AFWL-TR-70-16

21
CB

AFWL-TR-
70-16

A COMPUTATIONAL METHOD FOR EXACT, DIRECT, AND UNIFIED SOLUTIONS FOR AXISYMMETRIC FLOW OVER BLUNT BODIES OF ARBITRARY SHAPE (PROGRAM BLUNT)

Ronald H. Aungier
Capt USAF

TECHNICAL REPORT NO. AFWL-TR-70-16

13

July 1970

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base
New Mexico

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of AFWL (WLEE), Kirtland AFB, NM, 87117.

320

A COMPUTATIONAL METHOD FOR EXACT, DIRECT, AND UNIFIED SOLUTIONS FOR AXISYMMETRIC FLOW OVER BLUNT BODIES OF ARBITRARY SHAPE
(PROGRAM BLUNT)

Ronald H. Aungier
Capt USAF

TECHNICAL REPORT NO. AFWL-TR-70-16

FOREWORD

This report was performed under Program Element 62601F, Project 5791, Task 27.

Inclusive dates of research were May 1969 through December 1969. The report was submitted 11 May 1970 by the Air Force Weapons Laboratory Project Officer, Captain Ronald H. Aungier (WLEE).

Information in this report is embargoed under the U.S. Export Control Act of 1949, administered by the Department of Commerce. This report may be released by departments or agencies of the U.S. Government to departments or agencies of foreign governments with which the United States has defense treaty commitments, subject to approval of AFWL (WLEE).

This technical report has been reviewed and is approved.

Ronald H. Aungier

RONALD H. AUNGIER
Captain, USAF
Project Officer

Walter M. Hart, Jr.

WALTER M. HART, JR.
Lt Col, USAF
Chief, Fuzing Environment Branch

Carl F. Davis

CARL F. DAVIS
Colonel, USAF
Chief, Electronics Division

ABSTRACT

A time-dependent numerical method is presented that provides direct, exact, and unified solutions for axisymmetric flows about blunt nosed bodies of essentially arbitrary shape. The differencing scheme used ensures that the required stabilizing terms can be specified arbitrarily small and completely independent of the finite difference mesh sizes used. The method is shown to be more accurate than other reported time-dependent techniques. Computational procedures are introduced to enhance the computer efficiency obtainable with the time-dependent method. Extensive comparison with standard computational methods shows that the present method is comparable in both numerical accuracy and computer efficiency. A FORTRAN IV computer code and instructions for its use are provided.

(Distribution Limitation Statement No. 2)

AFWL-TR-70-16

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base
New Mexico

When U. S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is made available for study with the understanding that proprietary interests in and relating thereto will not be impaired. In case of apparent conflict or any other questions between the Government's rights and those of others, notify the Judge Advocate, Air Force Systems Command, Andrews Air Force Base, Washington, DC 20331.

DO NOT RETURN THIS COPY. RETAIN OR DESTROY.

CONTENTS

CONTENTS

CONTENTS

ILLUSTRATIONS

LIST OF SYMBOLS

A	weighting term defined by equation (45)
a	sound speed
B	weighting term defined by equation (46)
C_1	empirical constant used in equation (63)
C_2	empirical constant used in equation (63)
C_v	specific heat at constant volume
f	general function
G	metric of the coordinate system
g	general function
K	body curvature-also a constant used in the stability analysis
L	streamwise length of a body segment
M	Mach number
$\dot{m}$	mass flux across a wave
n	coordinate normal to the body
P	pressure
q	a general velocity component
r	body radius defined by equation (1)
S	entropy
s	coordinate parallel to the body
t	time
U	velocity component parallel to the shock wave
u	velocity component along s
V	velocity component normal to the shock wave
v	velocity component along n
W_s	the shock velocity
x	a general coordinate-also the coordinate normal to the shock wave
y	the coordinate parallel to the shock wave
z	distance along the symmetry axis, defined by equation (2)
α	a constant used in the stability analysis
β	a constant used in the stability analysis-also the shock angle
γ	the ratio of specific heats
δ	a constant defined by equation (88)

ζ	the ratio of the pressure in front of a wave to the pressure behind the wave
η	a nondimensional coordinate defined by equation (7)
θ	the body angle
κ	1 - Kn
μ	coefficients of the stabilizing terms
μ_0	an empirical constant used in equation (62)
ν	a constant used in the stability analysis
ξ	a coordinate defined by equation (6)
ρ	density
τ	transformed time variable defined by equation (8)
Φ	stabilizing term
ϕ	a constant used in the stability analysis

SUPERSCRIPTS

(ξ)	relative to the ξ coordinate
(η)	relative to the η coordinate
(x)	relative to the x coordinate

SUBSCRIPTS

m	maximum value of a quantity
s	value of a quantity at a point on the shock wave
t2	value behind a normal shock
o	value of a quantity when s = o
∞	free stream value of a quantity
2	behind a shock wave

SECTION I

INTRODUCTION

1. STATEMENT OF THE PROBLEM

A direct, exact, and unified numerical method for predicting the inviscid flow about high performance blunt bodies of essentially arbitrary shape is presented. This work was motivated by a requirement for extremely accurate definition of the edge conditions used in detailed boundary layer analyses. Accurate boundary layer analyses for slender high performance reentry vehicles must include consideration of entropy layers, bluntness induced pressure gradients, boundary layer displacement effects and vehicle shape deformation due to ablation mass loss. This requires an accurate inviscid flow solution capable of considering a general class of body shapes. Blunt body solutions of sufficient accuracy to be employed for these analyses are generally limited to body shapes defined by specific conic sections or power law profiles. Also, a description of the entire flow field usually requires the use of matched solutions from two or more numerical methods. A numerical method capable of considering the entire flow field is desirable, i.e., a unified solution procedure. An additional objective of this study was to develop a numerical method which can be easily generalized to three-dimensions to treat the angle-of-attack problem. An extensive development study considering three-dimensional problems is impractical due to the long computation times required to generate these solutions. Also, a numerical method for these problems should first be validated for axisymmetric flow, where standard solutions and experimental data are readily available. Consequently, it is believed that significant advances in developing techniques for three-dimensional problems must be based on axisymmetric flow studies.

2. THE SCOPE OF THIS STUDY

The time-dependent method suggested by the present author in reference 1 is generalized to realize the objectives of this study. Stability arguments are used to establish a differencing scheme that can be considered to be exact for practical purposes. A general body oriented coordinate system is used that is applicable to the desired class of body geometries. Computational procedures are introduced that greatly improve the computer efficiency of the time-dependent method. Extensive comparison with available exact solutions and experimental data is accomplished to demonstrate the accuracy of this method. A FORTRAN IV computer code and instructions for its use are included.

SECTION II

THE BLUNT BODY PROBLEM

1. DESCRIPTION OF THE PROBLEM

The blunt body problem is described by a set of nonlinear partial differential equations. These equations are elliptic in form in the subsonic region of the flow field and hyperbolic in the supersonic region. The mixed elliptic-hyperbolic form of these equations presents a formidable mathematical problem. In the supersonic region, the hyperbolic equations can be solved using the method of characteristics, once a starting line is established by a different numerical method. The method of characteristics has been generalized to three-dimensional flows, but with limited success. Solution of the elliptic equations in the subsonic region requires imposing boundary conditions at an unknown boundary, namely, the detached bow shock. The mixed elliptic-hyperbolic form of the governing equations has usually resulted in the use of two or more numerical techniques to describe the flow field. While a unified numerical solution is preferable to this matching procedure, a practical unified computational method for treating reentry vehicles of current interest has not been available. These other features can be identified as desirable characteristics for a blunt body solution. The solution should be direct, i.e., the body shape should be specified exactly. The governing equations should be treated in their entirety without simplifying assumptions, i.e., the numerical method should be exact. Finally, a practical computational method must allow efficient utilization of the computer to avoid excessive cost in generating solutions.

2. INVERSE TECHNIQUES

The inverse technique (Ref. 2 and 3) circumvents the problem of the unknown bow shock location by solving the flow field behind a specified bow shock. The body shape associated with this shock shape is computed during the solution. When solutions for specific body shapes are required, the shock shape is changed after each solution to achieve an approximation of the desired body shape through an iteration process. Clearly, the exact body shape cannot be achieved. Also, the body shape is extremely sensitive to the shock shape used. Small changes in shock shape can induce large changes in body shape or even preclude the existence of a solution. Consequently, this method is generally limited to body shapes for which the shock shape is well-known. Usually, only the subsonic portion of the flow field is treated due to the difficulty of obtaining convergence when the supersonic afterbody region is included.

3. THE METHOD OF INTEGRAL RELATIONS

The method of integral relations (Ref. 4) provides a direct solution to the blunt body problem. This method employs polynomial approximations to the flow field profile to reduce the governing equations to ordinary differential equations. The flow field is divided into strips to establish these ordinary differential equations. The number of equations that must be solved increases as more strips are used to improve the approximation. Further complication is introduced by the existence of movable saddle point singularities at the sonic line. The large number of equations that must be integrated through these singularities has generally limited the method to 1 or 2 strip approximations. An auxiliary assumption is required, typically that the maximum entropy streamline wets the body. This assumption is of questionable validity when three-dimensional effects are considered. Despite its limitations, the method of integral relations has provided some of the most accurate blunt body solutions reported to date.

4. THE TIME-DEPENDENT TECHNIQUE

The unsteady form of the governing equations are hyperbolic in form throughout the entire flow field. Consequently, the unsteady blunt body problem is a well-posed initial value problem. Several investigators (Refs. 1, 5-17) have employed the unsteady equations to advance the solution in time, asymptotically approaching the steady flow solution. Consequently, the unsteady equations become a means to an end. The additional complication of considering transient flows is justified by the relative simplicity of this problem as compared to the steady flow case. Additional terms must be introduced into the governing equations to stabilize an explicit finite difference scheme. If the influence of these terms on the solution is negligible, the method can be considered exact. The method is direct and, in principle, can provide a unified solution. In addition, it is easily generalized to treat complex body shapes, nonequilibrium flows and three-dimensional flow problems.

SECTION III

THE TIME-DEPENDENT METHOD

1. BACKGROUND

The time-dependent method was suggested by von Neumann and Richtmyer (Ref. 5) for treating flow problems involving shock waves. Their numerical method was motivated by the fact that shock waves tend to thicken in the presence of dissipative effects. Artificial dissipative terms are introduced into the governing equations to spread the shock to a finite thickness. Then, a differencing scheme can be established for the entire flow field which requires no special shock point computations. Lax and Wendroff (Refs. 6-8) developed more sophisticated differencing schemes using the conservation form of the governing equations. While these differencing schemes were motivated by mathematical rather than physical reasoning, the stabilizing terms used are, in effect, dissipative terms. Godunov, Zabrodin and Prokopov (Ref.9) applied the time-dependent method to the blunt body problem. Their method uses the one-dimensional Riemann problem to model the wave interaction between computational cells and at a discontinuous shock boundary. Burstein (Ref. 10) applied various differencing schemes to unsteady multidimensional flows. Bohachevsky, Rubin and Mates (Refs. 11, 12) used this technique to consider blunt body flows including bodies with corners, nonequilibrium flows and angles-of-attack. Moretti, Abbett and Bleich (Refs. 13, 14) introduced a one-dimensional unsteady characteristics technique to treat shock and body points for flows in two and three dimensions. This technique can be considered to be a generalization of Godunov's method where multidimensional effects are included through constant weighting terms. In reference 1, the present author suggested a differencing scheme which partially decouples the magnitude of the stabilizing terms from the finite difference mesh sizes used. This permits selecting the mesh sizes based on numerical accuracy requirements. Subsequently, the magnitude of the stabilizing terms can be established to properly model the inviscid flow problem.

2. ADVANTAGES

The governing equations for unsteady flow are hyperbolic in form. This results in a well-posed initial value problem. The unknown bow shock boundary is allowed to move in time until the proper steady-state profile is achieved. Since the procedure is applicable to the entire flow field, a unified solution procedure can be established. The method is direct and can be considered exact if the influence of the stabilizing terms is negligible. The method is easily programed for the computer, even when complex reentry phenomena are considered. While the time-dependent method is still in the early stages of development, no inherent limitations have been identified that would preclude its use for a practical computational method for generating exact, direct, and unified blunt body solutions.

3. COMPUTATIONAL ACCURACY

Time-dependent solutions reported in the literature often predict the correct surface pressure distribution. However, enthalpy and density distributions are usually in serious error. The errors are caused by the influence of the stabilizing terms and by the methods used to impose surface boundary conditions. It is believed that an exact solution can be achieved if a differencing scheme is developed that allows the magnitudes of the stabilizing terms to be specified arbitrarily small. In addition, a technique is required to achieve exact computations at points on the body surface. Since a solution to the blunt body problem requires the use of different mesh sizes throughout the flow field, the magnitudes of the stabilizing terms should be independent of the mesh sizes used. While a partial decoupling was achieved in reference 1, it is apparent that complete decoupling must be realized to obtain a practical computational method.

4. COMPUTER EFFICIENCY

Time-dependent methods reported in the literature have required extremely long computation times and extensive computer storage. As a consequence, no solutions for slender high performance vehicles of practical length have been reported. Even the relatively efficient numerical solution reported in reference 14 exhausted the storage of the IBM 7094 computer while treating the flow about an extremely short sphere-cone vehicle at angle of attack. The authors of reference 14 suggest that the consideration of more nodes on a larger computer would result in computation times too long for practical computation. The reduced computational field for axisymmetric problems will alleviate the computer storage problem. However, the computation time required to achieve a steady state increases with the body length. Consequently, an attempt to treat practical high performance vehicles with available time-dependent schemes must result in impractically long computation times. A significant reduction in the computer storage and computation times required by available time-dependent methods must be achieved to realize the full potential of this powerful technique.

SECTION IV

DESCRIPTION OF THE TECHNIQUE

1. THE COORDINATE SYSTEM

A body oriented coordinate system (Ref. 17) was selected for this numerical method. Body oriented coordinates allow a better finite difference nodal distribution than more conventional coordinate systems. As an example, the cylindrical coordinate system of reference 13 may be considered. The nodal spacing across the shock layer is reasonably good near the stagnation point of a blunt body, but the afterbody region for slender bodies must be treated with a very coarse mesh size in the axial direction. It is also apparent that a cylindrical afterbody cannot be treated with the transformed coordinate system of reference 13. The body oriented system is generally preferable for a numerical solution intended to treat a broad class of body shapes. The details of the coordinate system used in this study are illustrated in figure 1. The s coordinate is measured along the body surface from the stagnation point. The n coordinate is measured normal to the body surface with n = o on the body surface.

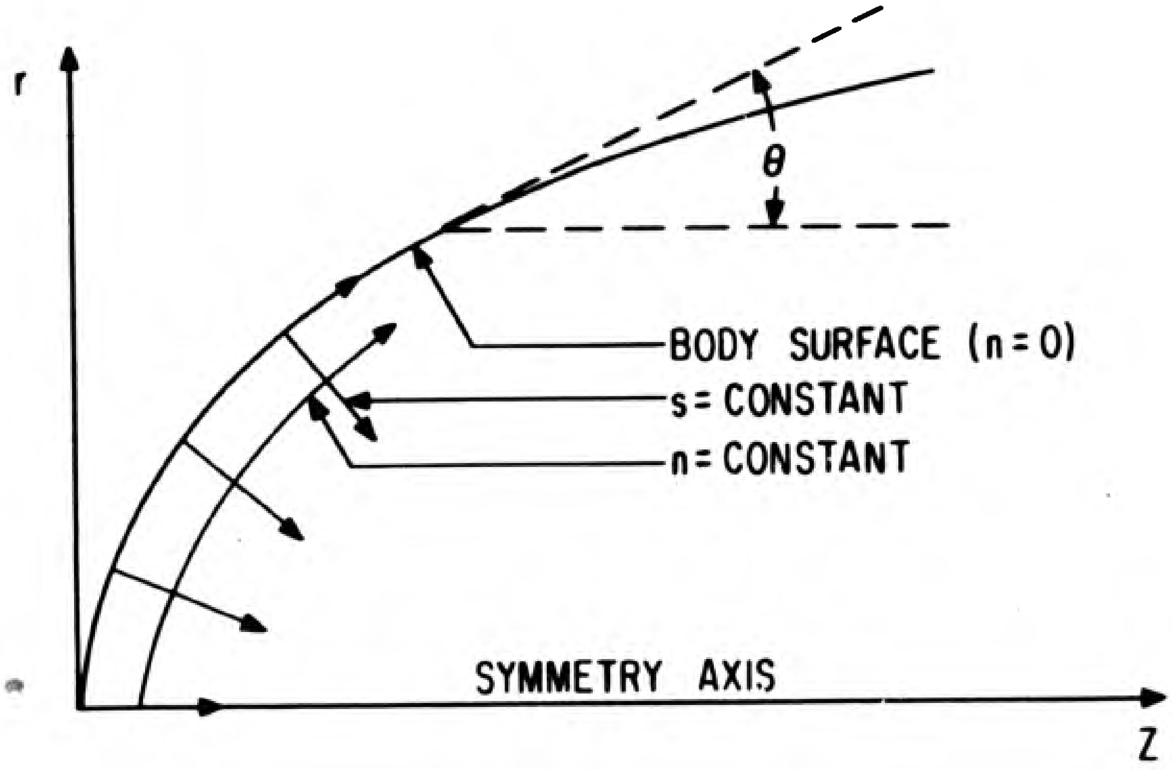


Figure 1. Body Oriented Coordinate System

It is easily shown that

$$r(s,n) = \int_{o}^{s} \sin\theta \, ds' - n \cos\theta \qquad (1)$$

$$z(s,n) = \int_{o}^{s} \cos\theta \, ds' + n \sin\theta \qquad (2)$$

The curvature of the body is given by

$$K\ (s) = -\frac{d\theta}{ds} \qquad (3)$$

Defining

$$\kappa = 1 - Kn \qquad (4)$$

the curvature of any line of constant n is given by K/κ. The metric tensor for this system can be expressed as

$$G = \begin{bmatrix} \kappa^2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & r^2 \end{bmatrix} \qquad (5)$$

These coordinates can be applied to smooth bodies, only, and are restricted to bodies whose curvature is continuous. It will be shown later that the latter restriction can be relaxed by properly staging the solution. Consequently, bodies with discontinuous curvature, such as sphere-cones, can be treated with the present technique. It is convenient to introduce the following transformations

$$\xi = s \tag{6}$$

$$\eta = n/n_s(s) \tag{7}$$

$$\tau = t \tag{8}$$

where $n_s(s)$ is the value of n at the shock and t is the time. Transformations of this type were applied in references 1 and 13 with considerable success. The principal merit is the simplification of the finite difference solution. It was shown in reference 1 that this type transformation can lead to serious errors unless applied with extreme care. The domain of solution is shown in figure 2.

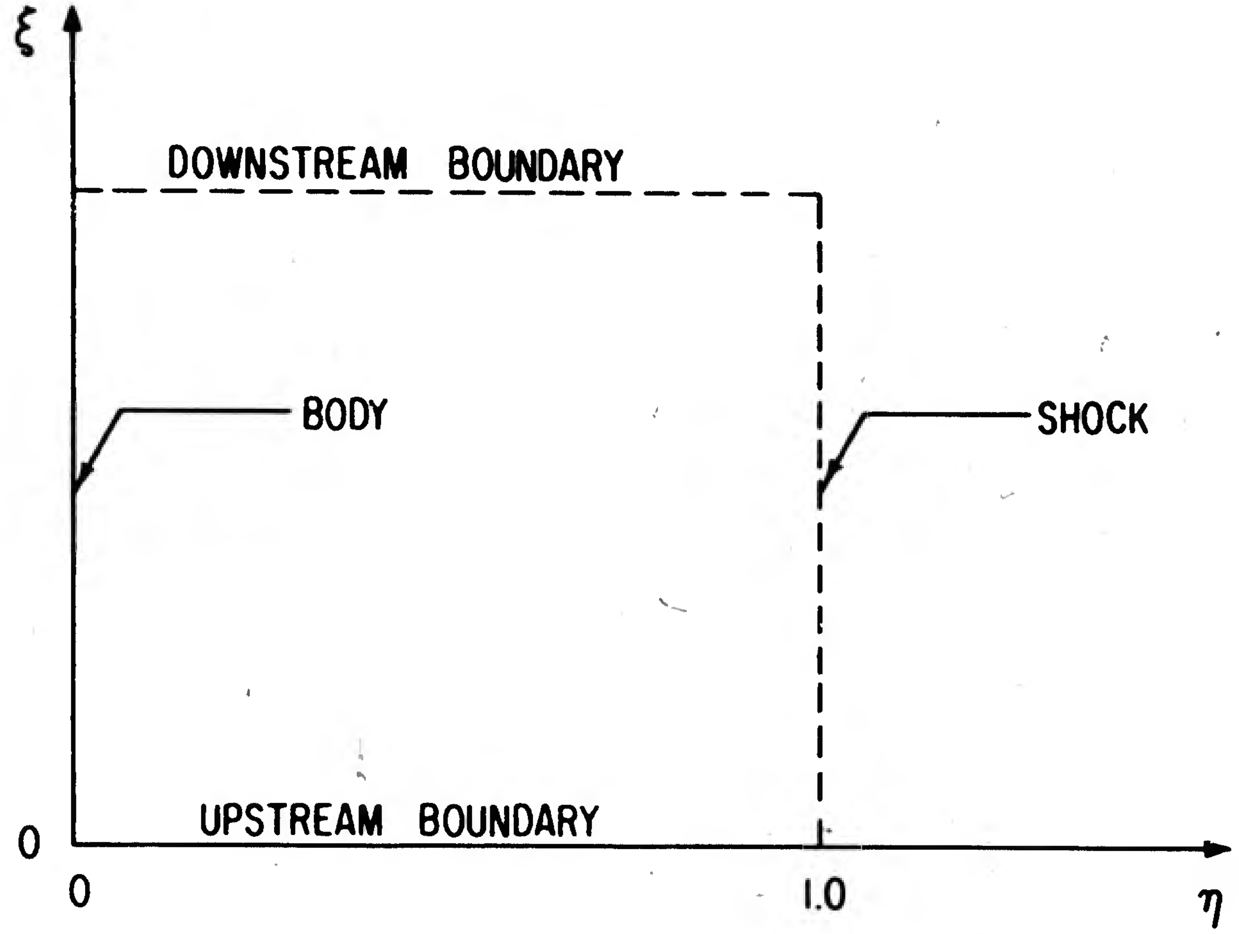


Figure 2. Transformed Coordinate System

It is generally possible to use a constant number of nodes in the η direction in the finite difference solution while maintaining a reasonably good node distribution. This greatly simplifies the numerical logic. Using equations (6) through (8) it is easily shown that

$$\frac{\partial}{\partial s} = \frac{\partial}{\partial \xi} - \frac{\eta}{n_s} \frac{\partial n_s}{\partial s} \frac{\partial}{\partial \eta} \tag{9}$$

$$\frac{\partial}{\partial n} = \frac{1}{n_s} \frac{\partial}{\partial \eta} \tag{10}$$

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial \tau} - \frac{\eta}{n_s} \frac{\partial n_s}{\partial t} \frac{\partial}{\partial \eta} \tag{11}$$

where $\frac{\partial n_s}{\partial t}$ is the shock velocity component in the n direction. Equations (9) through (11) allow solution of the governing equations for the (s,n) coordinate space in the (ξ,η) coordinate space.

2. THE GOVERNING EQUATIONS

The governing equations for adiabatic inviscid flow in the (s,n) coordinate space are

$$\kappa r\rho_t + (r\rho u)_s + (\kappa r\rho v)_n = 0 \tag{12}$$

$$\kappa u_t + u\, u_s + \kappa v u_n + \frac{1}{\rho} P_s = Kuv \tag{13}$$

$$\kappa v_t + uv_s + \kappa vv_n + \frac{\kappa}{\rho} P_n = -K_u^2 \tag{14}$$

$$\kappa S_t + uS_s + \kappa vS_n = 0 \tag{15}$$

where the subscripts imply partial differentiation, ρ is the density, P is the pressure, S is the entropy and u and v are the s and n velocity components, respectively. Equation (12) is singular at s = o and is replaced by

$$\kappa^2 \rho_t + 2\kappa(\rho u)_s + (\kappa^2 \rho v)_n = 0 \tag{16}$$

Equation (16) is obtained from equation (12) by evaluating the limit as s approaches zero using L'Hospital's rule.

For convenience, nondimensional variables will be employed. Defining

$$K' = K/K_o \tag{17}$$

$$\rho' = \rho/\rho_\infty \tag{18}$$

$$P' = P/P_\infty \qquad (19)$$

$$S' = (S - S_\infty)/C_V \qquad (20)$$

$$x' = K_o x \qquad (21)$$

$$q' = q/(P_\infty/\rho_\infty)^{1/2} \qquad (22)$$

$$t' = K_o(P_\infty/\rho_\infty)^{1/2} t \qquad (23)$$

where x refers to a general spatial coordinate and q refers to a general velocity component. It is easily shown that equations (12) through (16) remain valid for these nondimensional variables. The prime notation will be omitted in the remainder of this report, and all variables will be assumed to be nondimensional. To complete the set of governing equations, an appropriate state solution is needed. For a perfect gas, the appropriate form is

$$P = \rho^\gamma \text{ EXP } (S) \qquad (24)$$

3. FINITE DIFFERENCES FOR INTERIOR POINTS.

For interior points, i.e., points not on the shock or body surface, central difference approximations are employed for the spatial partial derivatives

$$f_\xi \leftarrow \left[f(\xi+\Delta\xi,\eta,\tau) - f(\xi-\Delta\ ,\eta,\tau)\right]/2\Delta\xi \qquad (25)$$

$$f_\eta \leftarrow \left[f(\xi,\eta+\Delta\eta,\tau) - f(\xi,\eta-\Delta\eta,\tau)\right]/2\Delta\eta \qquad (26)$$

and a forward difference approximation is employed for the partial derivatives in τ.

$$f_\tau \leftarrow \left[f(\xi,\eta,\tau+\Delta\tau) - f(\xi,\eta,\tau)\right]/\Delta\tau \qquad (27)$$

It is known that equations (25) through (27) result in an unconditionally unstable finite difference scheme when applied to equations (12) through (15). Von Neumann (Ref. 5) showed that stability could be achieved by including stabilizing terms in the governing equations. Denoting equations (12) through (15) by

$$g_t = f\ (s,\ n,\ \rho\ ,\ u,\ v,\ P) \qquad (28)$$

where g is ρ, u, v or S, the stabilizing terms are added in the form

$$g_t = f - \Phi \tag{29}$$

where

$$\Phi = \mu^{(\xi)} g_{\xi\xi} + \mu^{(\eta)} g_{\eta\eta} \tag{30}$$

In reference 1, $\mu^{(\xi)}$ and $\mu^{(\eta)}$ are equal and are given by

$$\mu^{(x)} = \frac{\nu (\Delta x)^2}{\Delta t} \tag{31}$$

where ν is a constant and x refers to ξ or η. Using equation (31), it was shown in reference 1 that the appropriate stability criterion is

$$\Delta t \lesssim \frac{\sqrt{\nu}\ \Delta x}{(q+a)} \tag{32}$$

where Δx is the smallest spatial mesh size, q is the corresponding velocity component and a is the sound speed. An outline of these calculations from reference 1 has been included in appendix I. Although this stability analysis is extremely simplified, equation (32) proved to be extremely accurate in predicting the time step allowable for stable computations (Ref. 1). The impressive success of equation (32) in the numerical method of reference 1 prompted an attempt to exploit the simple stability analysis further to obtain a more sophisticated numerical method. First, equation (32) is inverted to predict the minimum value of ν allowed for stability,

$$\nu \geq \left[(q+a)\ \Delta t/\Delta x\right]^2 \tag{33}$$

Then equation (31) and (33) are combined to yield

$$\mu^{(x)} = (q+a)^2\ \Delta t \tag{34}$$

Clearly, this decouples the magnitude of the stabilizing terms from the spatial mesh size. Equation (34) is now applied in the ξ and η directions to yield

$$\Phi = \Delta t\ \left[(u+a)^2\ g_{\xi\xi} + (v+a)^2\ g_{\eta\eta}\right] \tag{35}$$

which results in the magnitude of Φ being completely defined and minimized for any specified value of Δt. Equation (35) is applied at each node, independently in each coordinate direction, on each time iteration. As a result, the magnitudes of the stabilizing terms are minimized for all points and are proportional to Δt. The Courant-Friedricks-Lewy (CFL) stability criterion (Ref. 18) must be satisfied

$$\Delta t \leq \frac{\Delta x}{q+a} \tag{36}$$

Consequently, any value of Δt between zero and CFL limiting value can be used. The stabilizing term effects can be reduced by decreasing Δt, but the computation time required to reach a steady state will increase accordingly.

4. FINITE DIFFERENCES FOR BODY POINTS.

For nodes on the body surface, the velocity component normal to the body must vanish. A popular technique for satisfying this requirement is known as the reflection principle (Refs. 1,11,12). Basically, this involves assigning values for the flow parameters at a dummy set of nodes inside the body. The values are identical to those at nodes above the body surface except for the normal velocity component which is assigned the negative of its value above the surface. Then the differencing scheme used at interior points is applied at the body point. It is difficult to assess the accuracy of this scheme. A more appealing approach would be to develop a stable differencing scheme for the body points which does not rely on empirical schemes. This would allow an assessment of the error in terms of the finite difference errors of the various derivative approximations. It is shown in appendix I that a backward difference at the body surface should be stable when $\mu^{(\eta)}$ vanishes. This proved to be true in practice and is the basis for the body point solution presented here. Equation (25) is used to approximate the partial derivative in ξ. The partial derivatives in η were represented by forward differences (Note: this is equivalent to a backward difference in n).

$$f_\eta \rightarrow \left[f(\xi,\eta+\Delta\eta,r) - f(\xi,\eta,r)\right]/\Delta\eta \tag{37}$$

At body points, $\mu^{(\eta)}$ was allowed to vanish, i.e.,

$$\Phi = \Delta t(u+a)^2 g_{\xi\xi} \tag{38}$$

and the velocity component normal to the body was required to vanish on the body surface. With these exceptions, the differencing scheme for interior points was applied for the body points. Comparison between this technique and the reflected boundary condition logic is given in the presentation of results. It will become apparent at that time that this approach is considerably more accurate.

5. FINITE DIFFERENCES FOR THE STAGNATION STREAMLINE.

For axisymmetric flow, points on the stagnation streamline, like the body points, are characterized by the vanishing of one of the velocity components. The stability arguments of appendix I should also apply to these points. Consequently, the procedure for interior points are modified by allowing $\mu^{(\xi)}$ to vanish such that

$$\Phi = \Lambda t \ (v+a)^2 g_{\eta\eta} \tag{39}$$

The partial derivatives in ξ all vanish at s = 0 except for u_ξ. Since a forward, backward and central difference approximation for u_ξ are all identical at S = 0, it is of little concern as to which is applied. The velocity component, u, is required to vanish. The differencing scheme for interior points is modified to include equation (39) and applied at these points. Special attention is required at the stagnation point where the velocity vanishes. Equations (15) and (16) must be solved for entropy and density. The stability arguments of appendix I now indicate that both $\mu^{(\xi)}$ and $\mu^{(\eta)}$ can vanish, i.e., $\Phi = 0$. Then, equation (16) is solved using a forward difference approximation for the partial derivative in η. Equation (15) must be treated with care. The procedure used for equation (16) cannot be applied since equation (15) would then be degenerate. The proper value of entropy behind a steady normal shock could be imposed and held constant with advancing time. This would preclude the consideration of transient phenomena such as time varying free-stream conditions. An averaging procedure using entropy values at adjacent nodes was attempted, but resulted in longer computation times to achieve the steady-state solution. For steady-state solutions, the assignment of the value of entropy at the node upstream of the stagnation point at each time step proved to be a successful approach. However, for transient solutions, this proved to be unreliable, usually resulting in the generation of indefinite values in actual computation. A procedure that allows transient problems while providing rapid convergence to the steady-state solution is as follows:

$$S\ (o,o,\tau+\Delta\tau) = \mu\, S(o,o,\tau) + (1-\mu)\ S\ (o,\Delta\eta,\tau) \tag{40}$$

where

$$\mu = \left[\frac{S(o,o,\tau) - S(o,\Delta\eta,\tau)}{S(o,o,\tau) + S(o,\Delta\eta,\tau)}\right]^2 \tag{41}$$

When the values of entropy at $\eta = 0$ and $\eta = \Delta\eta$ are essentially equal, the stagnation point entropy is assigned a value approximately equal to its upstream value. When these values of entropy are quite different, the value is held approximately constant with a weak dependence on the upstream value. Since viscous effects predominate in this region in the physical problem, this procedure appears to be a reasonable way to obtain realistic transient solutions.

6. COMPUTATIONS FOR THE SHOCK POINTS.

The shock points were treated with a quasi-one-dimensional unsteady characteristics technique suggested by Moretti and Abbett (Ref. 13). This technique assumes that the variation in the flow variables normal to the shock are the relevant quantities for determining the shock behavior, while variations in the flow variables parallel to the shock can be assigned constant values for use as weighting functions. The method is quite similar to Godunov's method (Ref. 9) as applied in reference 16 except for the use of these constant weighting functions. The equations of motion are written for the shock fixed Cartesian coordinate system shown in figure 3, where β is the local angle between the shock and the free stream velocity vector. Denoting the x and y velocity components by V and U, respectively, the governing equations are

$$\rho_t + V\rho_x + \rho V_x = -A \tag{42}$$

$$V_t + VV_x + \frac{1}{\rho}P_x = -B \tag{43}$$

$$S_t = \text{constant} \tag{44}$$

where the y momentum equation has been omitted since the value of U is known from M_∞ and β. A and B are defined by

$$A = (\rho U)_y \tag{45}$$

$$B = UV_y \tag{46}$$

and will be treated as constants. In appendix II it is shown that the relevant characteristics can be defined by

$$\frac{dx}{dt} = V + a \tag{47}$$

$$\frac{dx}{dt} = V - a \tag{48}$$

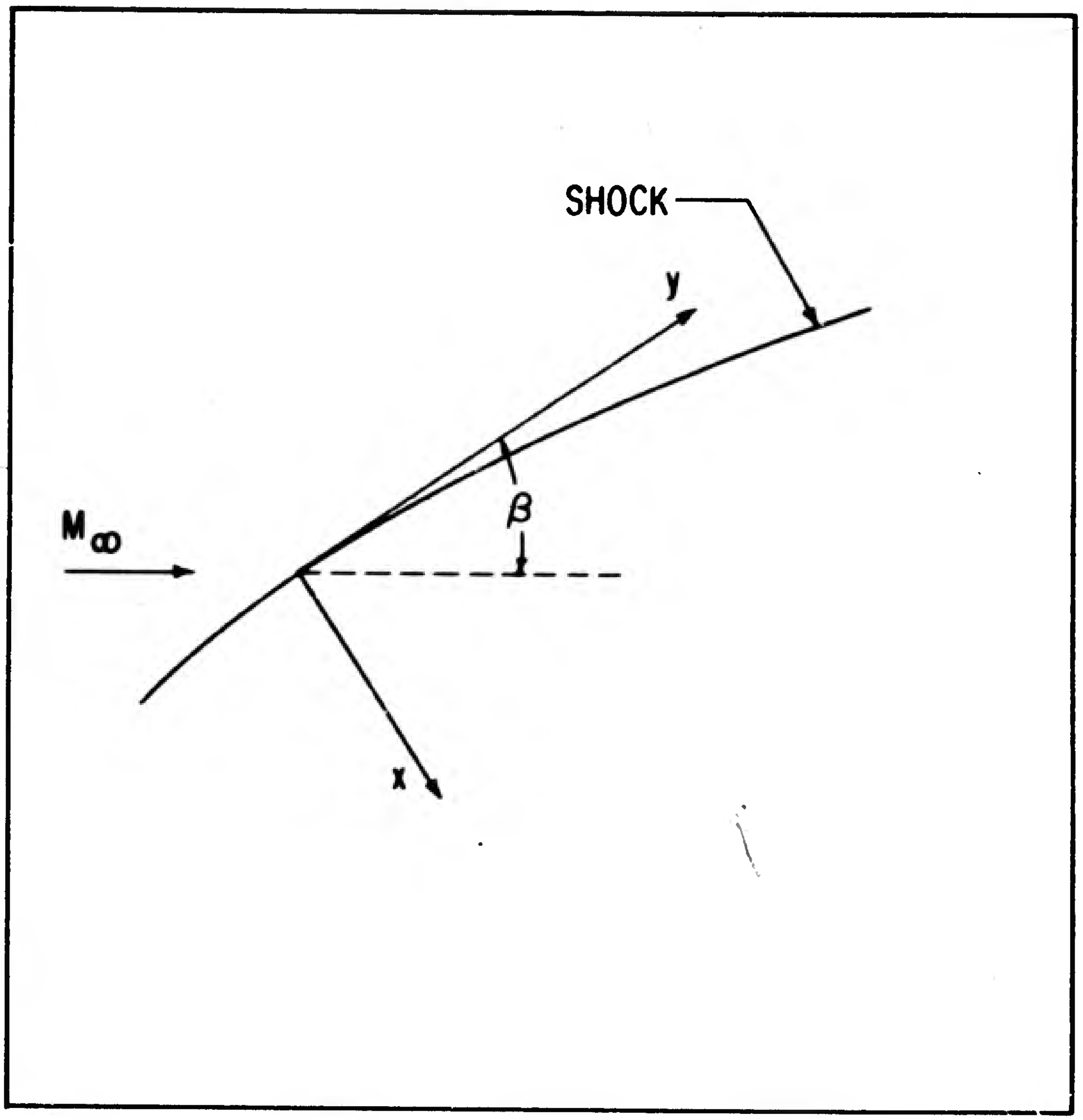


Figure 3. Shock Fixed Coordinates

and the compatability equation associated with equation (48) is

$$\frac{dP}{dt} = \left[\rho a \frac{dV}{dt} + (B-A \frac{a}{\rho})\right] \tag{49}$$

It is shown in appendix II that application of this method on the symmetry axis requires that A be multiplied by 2. The problem can be visualized as a one-dimensional Riemann problem with equation (49) replacing the usual Reimann invarients. The physical phenomena is illustrated in figure 4 where W_s refers to the shock velocity in the x direction. The problem is solved by an iteration procedure. The shock parameters are established from the Rankine-Hugonoit relations

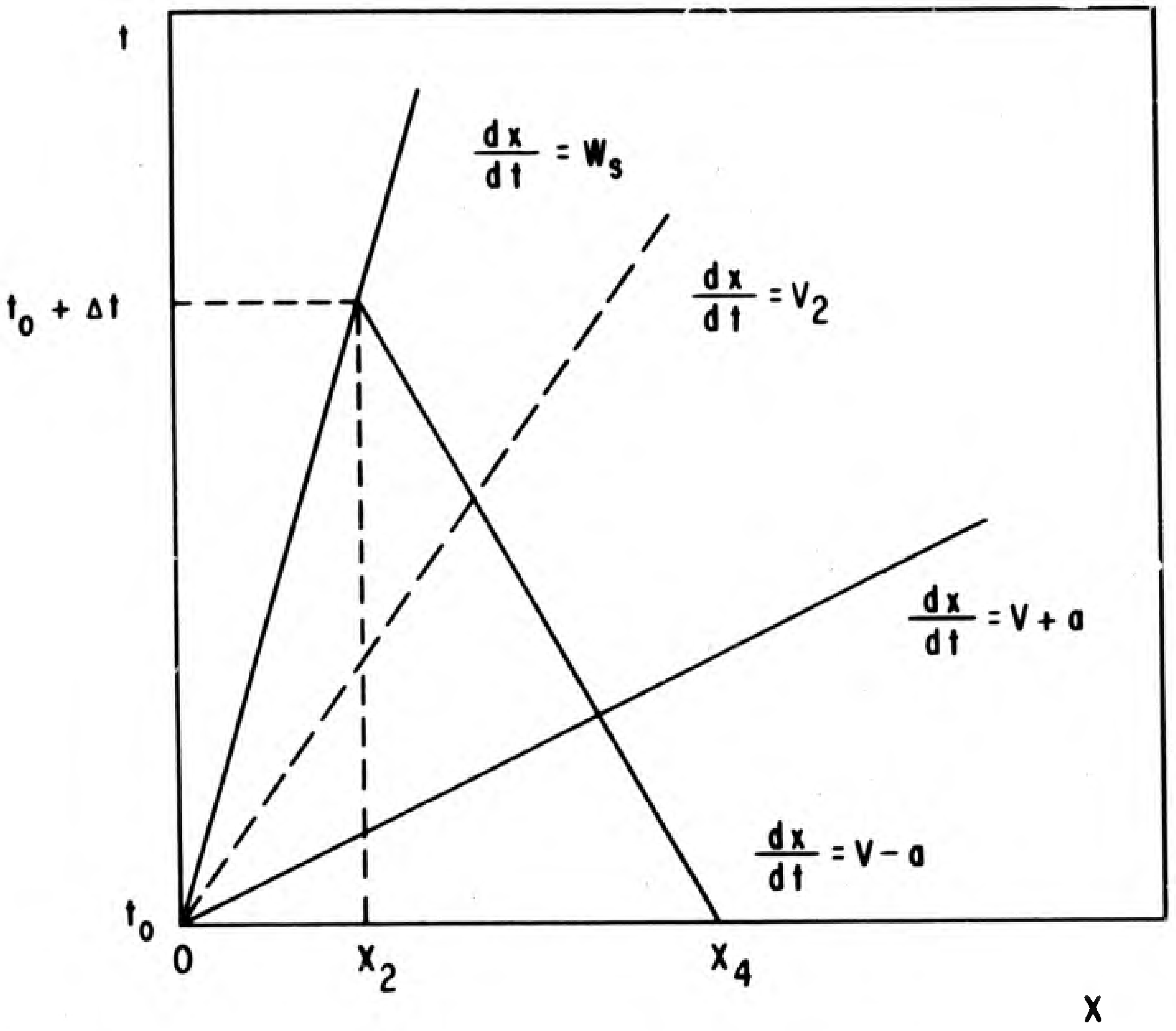


Figure 4. Shock Wave x-t Diagram

once satisfactory agreement between the Rankine-Hugonoit pressure prediction and the pressure prediction of equation (40) has been achieved. The Rankine-Hugonoit relations are applied in the form

$$\dot{m} = \left[\frac{\gamma+1}{2} P_s + \frac{\gamma-1}{2}\right]^{1/2} \tag{50}$$

$$\dot{m}V_s + P_s = \dot{m}V_\infty + 1 \tag{51}$$

$$W_s = V_\infty - \dot{m} \tag{52}$$

$$\rho_s = \frac{\dot{m}}{V_s - W_s} \tag{53}$$

where a perfect gas has been assumed and the subscript, s, refers to a quantity behind the shock. The velocity component in the y direction is unchanged across the shock

$$U_s = U_\infty \tag{54}$$

where the subscript ∞ refers to a free stream value. The entropy behind the shock is computed from the equation of state

$$S_s = \ln (P_s) - \gamma \ln(\rho_s) \tag{55}$$

The iteration procedure is diagramed in figure 5. Starting with the shock parameters at point 0 (Fig. 4) as an initial guess, point 2 is located by

$$X_2 = W_s \Delta t \tag{56}$$

and the flow variables at 2 are assumed identical to their values at 0, initally. Point 4 is located from equation (48) in the form

$$X_4 = X_2 + (V_2 - a_2)\, \Delta t \tag{57}$$

Then, the point 4 is redetermined from

$$X_4 = X_2 + \frac{1}{2}(V_4 - a_4 + V_2 - a_2) \tag{58}$$

and an iteration on equation (58) is performed until successive values of X_4 agree to an acceptable degree. Using the known solution at time t_o, the values of A and B are determined at 4, averaged with their values at point 0 and used to solve equation (49). The pressure at 2 is compared with the pressure predicted by equation (49). If they agree within an acceptable error, the shock parameters are computed from equations (50) through (55) and the iteration loop is terminated. If the agreement is not satisfactory, a new pressure at point 2 is computed as the average of the resident value and the value predicted by equation (49). Other shock parameters required are obtained from equation (50) through (53) and the entire iteration loop is repeated. The value of the shock angle, β, is held constant throughout the iteration procedure, equal to its value at t_o.

$$\beta = \theta - \tan^{-1}\left[\frac{1}{\kappa_s} \frac{\partial n_s}{\partial \xi}\right] \tag{59}$$

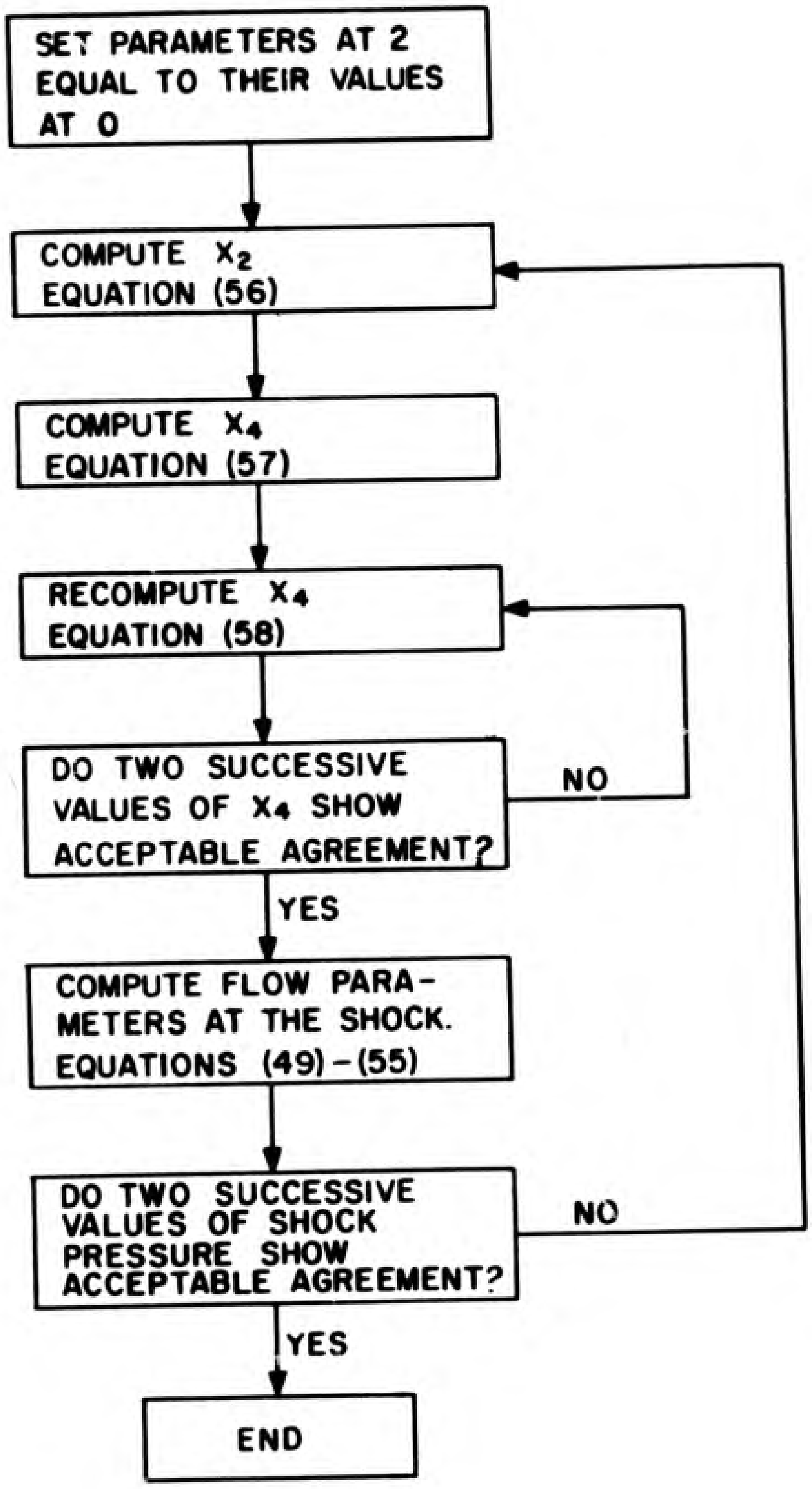

SET PARAMETERS AT 2 EQUAL TO THEIR VALUES AT 0
COMPUTE X_2 EQUATION (56)
COMPUTE X_4 EQUATION (57)
RECOMPUTE X_4 EQUATION (58)
DO TWO SUCCESSIVE VALUES OF X_4 SHOW ACCEPTABLE AGREEMENT?
NO
YES
COMPUTE FLOW PARAMETERS AT THE SHOCK. EQUATIONS (49)-(55)
DO TWO SUCCESSIVE VALUES OF SHOCK PRESSURE SHOW ACCEPTABLE AGREEMENT?
NO
YES
END

Figure 5. The Shock Point Iteration Scheme

The evaluation of the shock slope through the term containing the derivative of n_s with respect to ξ can be a source of serious error. When a central difference approximation is employed, the slope at any point is evaluated independently of the value of n_s at that point. If the initial guess for the shock shape is not carefully specified, a discontinuity in the predicted shock shape will often result. The present numerical scheme proved to be sufficiently reliable to avoid fatal errors in actual computer computations when this occured, but computation times were increased by factors of 20 to 30. The necessity of requiring extremely accurate initial guesses for shock shape would greatly complicate the program operation. The problem arises at regions where the shock curvature is large such as the supersonic region on a hemisphere. The actual difference approximation provides a reasonable value for the shock slope despite the fact that the shock is discontinuous at the point under consideration. The use of a backward difference approximation eliminated this problem, but proved to be inaccurate. To obtain sufficient accuracy while eliminating discontinuities in the shock, an average of two backward differences and one forward difference is employed.

$$\frac{dn_s}{d\xi} = \left[n_s(\xi+\Delta\xi) + n_s(\xi) - 2n_s(\xi-\Delta\xi)\right] /3\Delta\xi \qquad (60)$$

Satisfactory results are obtained at all points with this approximation while discontinuities in the shock are avoided. Introducing a dependence on the local value of n_s in the finite difference approximations acts as a damping influence on the computation. Moretti and Bleich (Ref. 19) experienced this problem for very blunt elliptical bodies. The shock shapes predicted in the reference are very similar to the type predicted with this method using the central difference approximation. It is probable that equation (60) could be used in their solution to avoid this difficulty.

7. POINTS ON THE UPSTREAM BOUNDARY.

Solution for nodes on the upstream boundary are treated by extending the solution linearly in ξ one extra mesh width. The interior point logic is then applied in its entirety. It is noted in appendix I that this is equivalent to employing backward differences for the partial derivatives in ξ and allowing $\mu^{(\xi)}$ to vanish. Since the linear extension logic was numerically faster and simpler, it has been adopted for this numerical method.

8. SELECTION OF THE TIME STEP.

The proper choice of a value of Δt for numerical computations requires that the quantity $\mu^{(x)}$ in equation (34) be small relative to the dynamic terms in the equations of motion, i.e., the effect of the stabilizing terms must be negligible. A simple estimate for the magnitude of the dynamic terms is given by the free-stream velocity. It is reasonable to require

$$\mu_o = \frac{\mu^{(x)}}{q_\infty} \tag{61}$$

where μ_o is a constant whose magnitude is small relative to unity. Using free-stream values for velocity and sound speed in equation (34) and combining that equation with equation (61)

$$\Delta t = \frac{\mu_o q_\infty}{(q_\infty + a_\infty)^2} \tag{62}$$

Equation (62) has been used to predict Δt for a range of free-stream Mach numbers from 1.5 to 500. The percent error resulting from the stabilizing terms was essentially equal in all cases, indicating that equation (62) is a proper expression for computing Δt. The computer code makes an additional check to ensure that the Courant-Friedricks-Lewy stability criterion, equation (36), is satisfied. A typical value for the constant, μ_o, is 0.04.

9. SEGMENTED SOLUTIONS

A computational procedure has been developed that avoids the problems of long computation times and excessive computer storage requirements. When the steady-state flow over typical high performance blunt bodies with relatively long afterbody regions is considered, e.g., sphere-cones with axial lengths in excess of 50 nose radii; the procedure used will be referred to as the segmented approach. The flow field is computed in a series of discrete segments marching back along the body. The steady-state solution is obtained for each segment before proceeding to the next. The solution at the end of each segment is used as a constant starting line for the next segment. The computations on each segment are terminated at the downstream boundary with the linear extension procedure described previously. The lengths of the segments is arbitrary except for the first, which must include the subsonic region and terminate in supersonic flow. The nodal spacing along the body and the number of nodes across the shock layer can be changed for each segment before starting the solution at that segment. Typical slender blunt bodies may require 5 nodes across the shock in the stagnation region while as many as 40 may be required far back on the afterbody. The number of nodes considered along the body can vary. Generally, far downstream on the afterbody a coarse nodal spacing using one computational node in the streamwise direction is the optimum choice for minimum computational time. An empirical expression to predict the physical time required for essentially steady-state solutions, τ_f, is

$$\tau_f = (C_1 + C_2 L) / (u_o + a_o) \tag{63}$$

where C_1 and C_2 are constants, L is the streamwise length of the segment and u_o and a_o are the velocity and sound speed at the body on the upstream boundary of the segment. This expression has been used for segments of widely varying values of L and has always predicted proper values for τ_f. Equation (63) permits the segmented approach to be used in the solution without complicated input instructions. In practice, computations proceed until τ exceeds τ_f. Then the shock velocity at all points is checked to insure that all values are less than 10^{-3}. If any shock velocity is greater than that value, further iterations are performed. As a result, the input instructions to the computer code are minimal. Since the exact number of iterations to be performed for a given problem is not known in advance, an accurate estimate of computer time is difficult. To avoid wasting large amounts of computer time due to a time limit termination, flow data for each segment is stored in sequential files on a magnetic tape. A simple restart option allows renewing computations at the first incomplete segment with little waste of computer time. A solution for a sphere-cone vehicle with a 10° cone half-angle and axial length of 50 nose radii was performed in under 6 minutes on the AFWL CDC 6600 computer using the segmented approach. To resolve the afterbody flow, 17 nodes across the shock layer were required. A mesh size of 0.15 was required to resolve the flow in the stagnation region. Without the segmented approach these two quantities would have had to remain constant throughout the flow field. Using the maximum value of the denominator in equation (63) to ensure the estimate of computation time would be a minimum, equation (63) and the known computational efficiency of the computer code resulted in a predicted computational time of 72 hours for the same computer code, without the segmented approach, on the same computer. The estimate is probably less than the time which would actually be required due to the use of the maximum value of the denominator in equation (63). It is easy to understand why time-dependent solutions for this type of vehicle have not been reported previously.

SECTION V

PRESENTATION OF RESULTS

1. INVESTIGATION OF THE STABILIZING TERM EFFECTS

A value of the parameter μ_0 must be selected empirically. A comparison of the predicted stagnation point pressure, P_0, with the known stagnation pressure behind a normal shock, P_{t2}, was accomplished for this purpose. The error in P_0 and the number of iterations required for an approximate steady state are shown in figure 6 for a range of values of μ_0. The case considered was Mach 4 flow over a sphere with 5 nodal points across the shock layer. The errors shown in figure 6 include both stabilizing term effects and finite difference errors. Since the finite difference errors should be approximately constant for all values of μ_0, the relative values of the errors indicate the effects of the stabilizing terms. For values of $\mu_0 > 0.05$, the Courant-Friedricks-Lewy (CFL) stability limit was exceeded, causing the computer code to ignore the specified values of μ_0. Based on the results presented in figure 6, a value of $\mu_0 = 0.04$ was selected and used for all other cases presented in this report. The error in stagnation point pressure is 0.53 percent for this value of μ_0. This should be adequate for most applications. Since the computation time required varies approximately as the inverse of μ_0, requiring greater accuracy will greatly increase the computation time while accepting larger errors will not greatly reduce it. The importance of the stabilizing term effects is further demonstrated in figures 7 and 8, where the entropy and total enthalpy distributions on the body predicted by the present technique and the method of reference 1 are compared. The method of reference 1 was selected for this comparison since it was shown (Ref.1) that accurate pressure distributions can be predicted with that method. It is seen that the present method predicts constant entropy on the body while the method of reference 1 shows considerable variation in that quantity. The error in total enthalpy on the body is less than 0.5 percent for the present method, while reference 1 predicts errors greater than 8 percent for this quantity. Clearly, the validation of a time-dependent technique cannot be accomplished by examining pressure profiles only.

2. UNSTEADY FLOW

By assuming an initial shock shape extremely close to the body surface, an approximate description of the flow over a body started impulsively from rest can

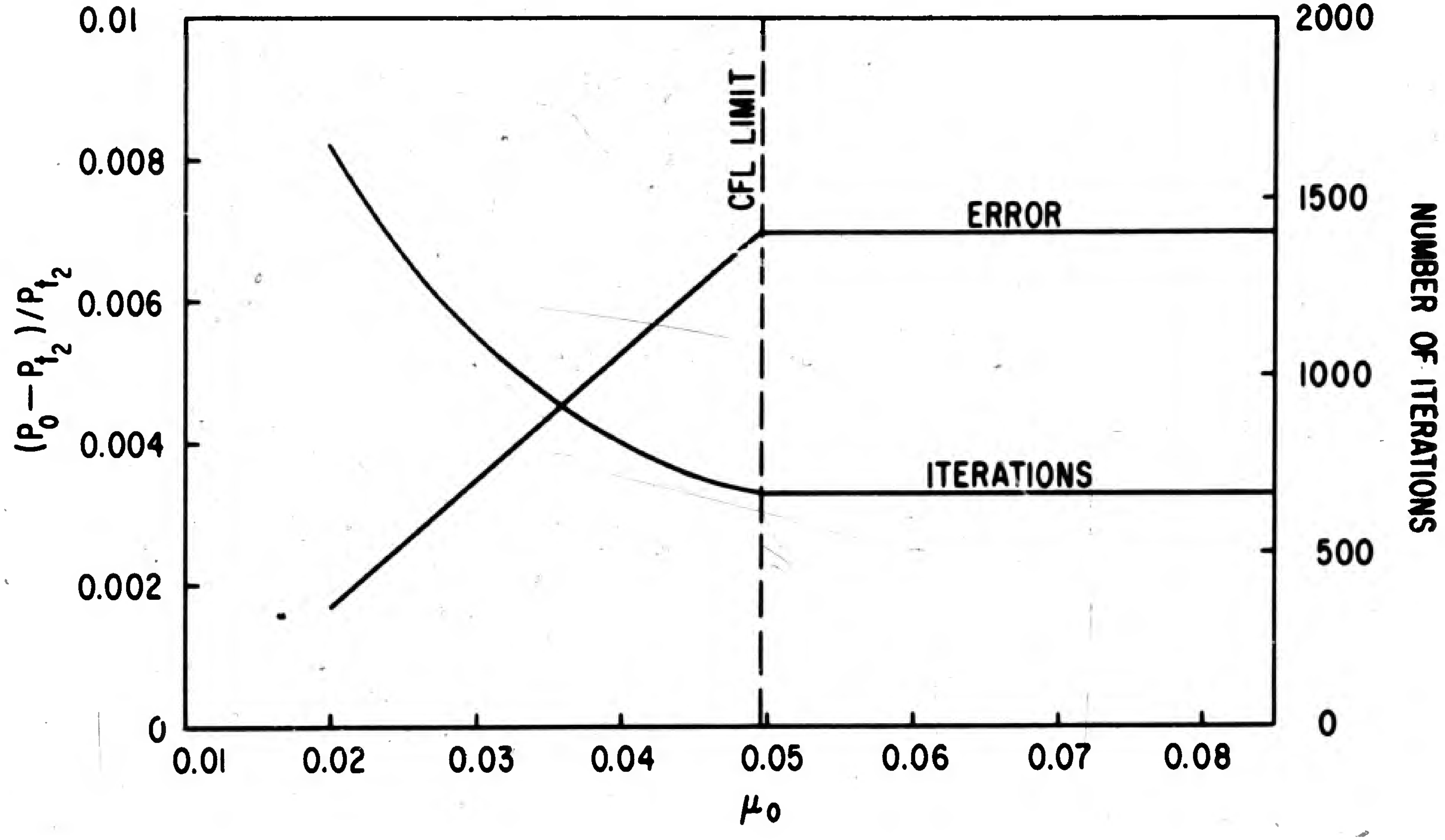


Figure 6. Investigation Of The Stabilizing Term Effects

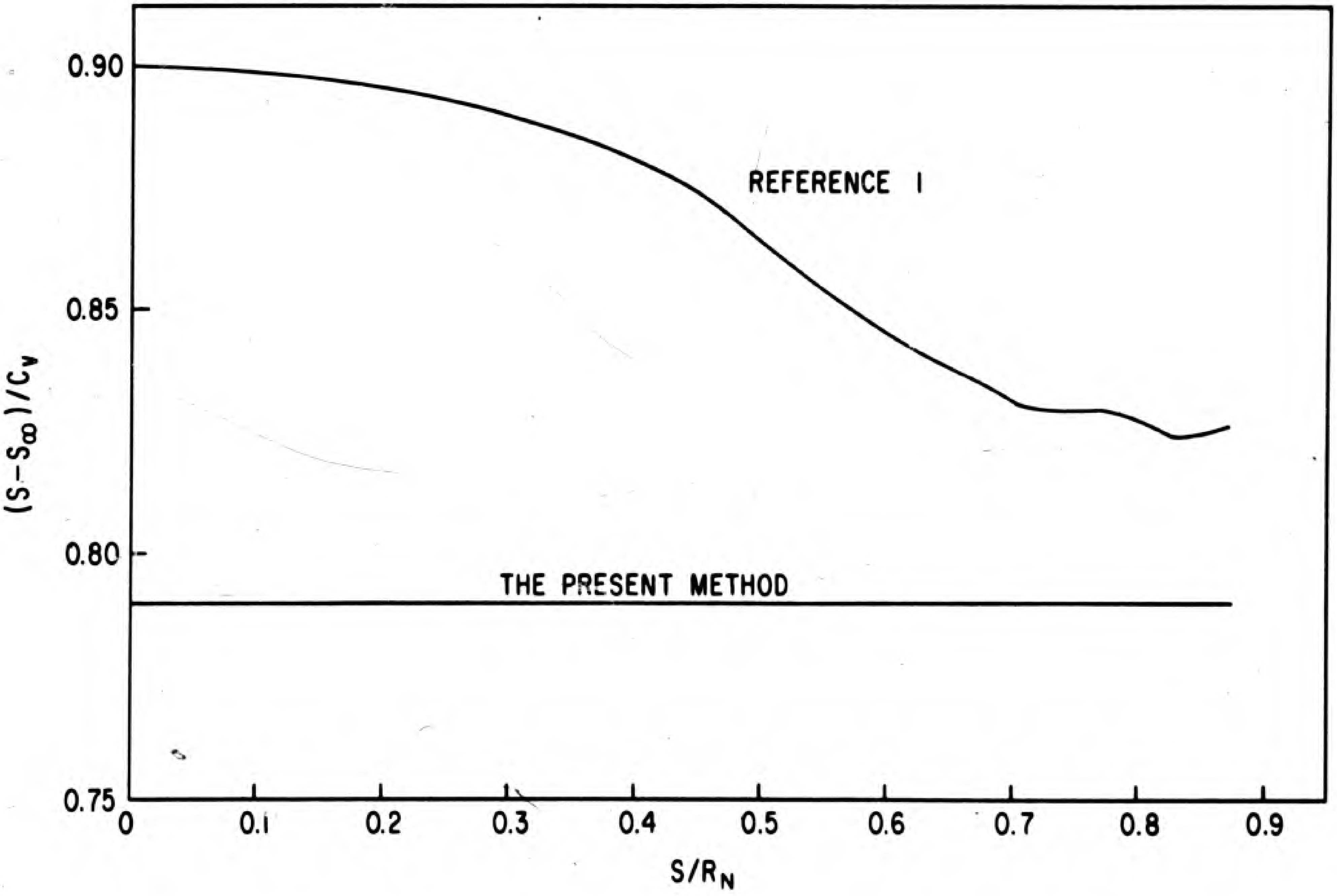


Figure 7. Surface Entropy Profiles For A Sphere In Mach 4 Flow

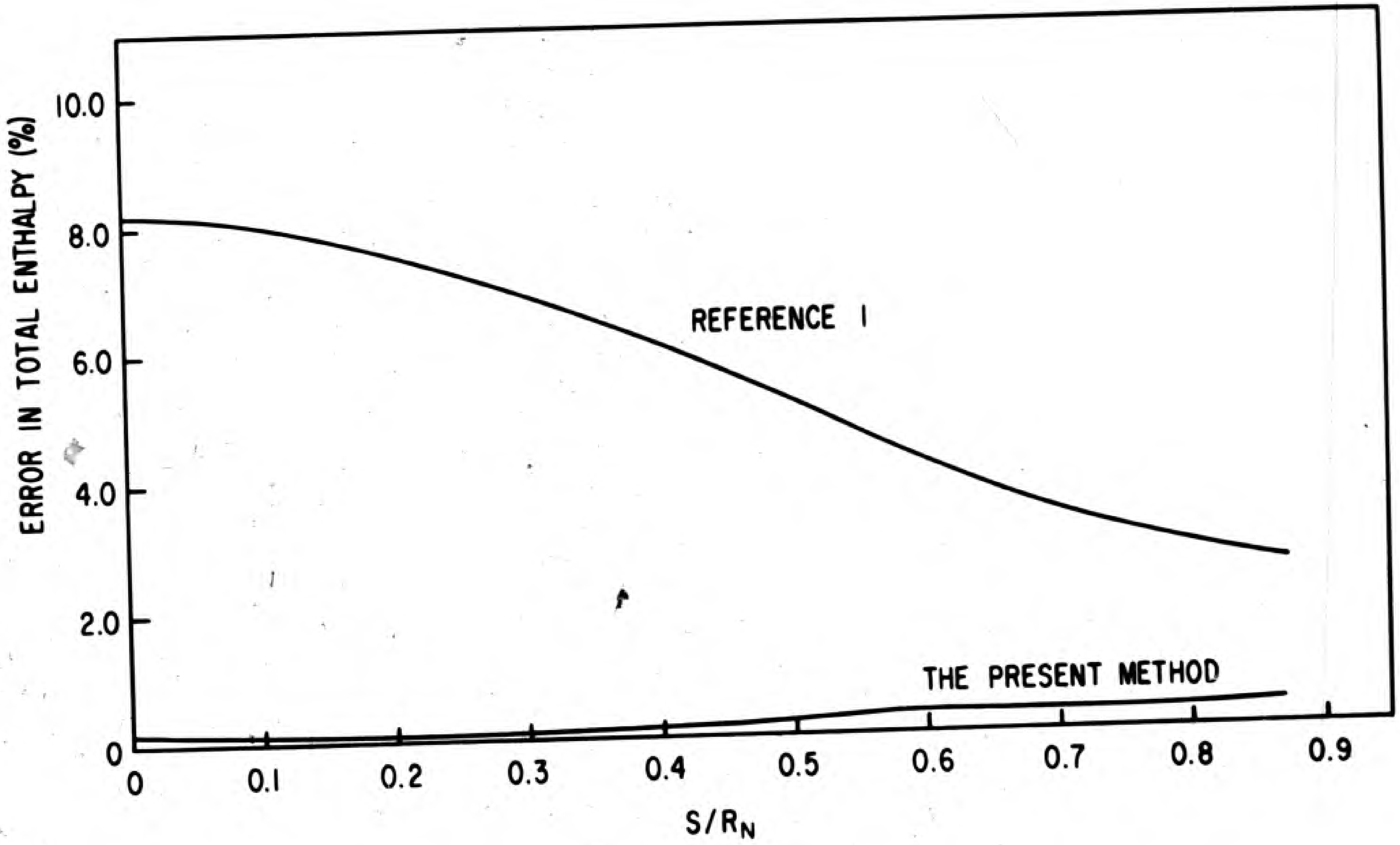


Figure 8. Surface Total Enthalpy Profiles For A Sphere In Mach 4 Flow

be achieved. Figure 9 illustrates the results of this procedure for a sphere in a Mach 30 flow. It is seen that the shock propagates out from the body and stabilizes to its steady-state profile after about 500 iterations. The stabilization of the shock shape has been used as a criterion for specifying an approximate steady state. This case demonstrated that this criterion is not sufficient since the flow field parameters had not stabilized after 500 iterations. Figure 10 illustrates the shock velocity at various times for this case. A reliable specification for achieving steady state flow was found to be the requirement that all shock velocities be less than 10^{-3}.

3. BLUNT BODY FLOW AT VARIOUS MACH NUMBERS

Many available solution procedures are extremely reliable for a limited Mach number range while experiencing difficulty at low and high Mach numbers.

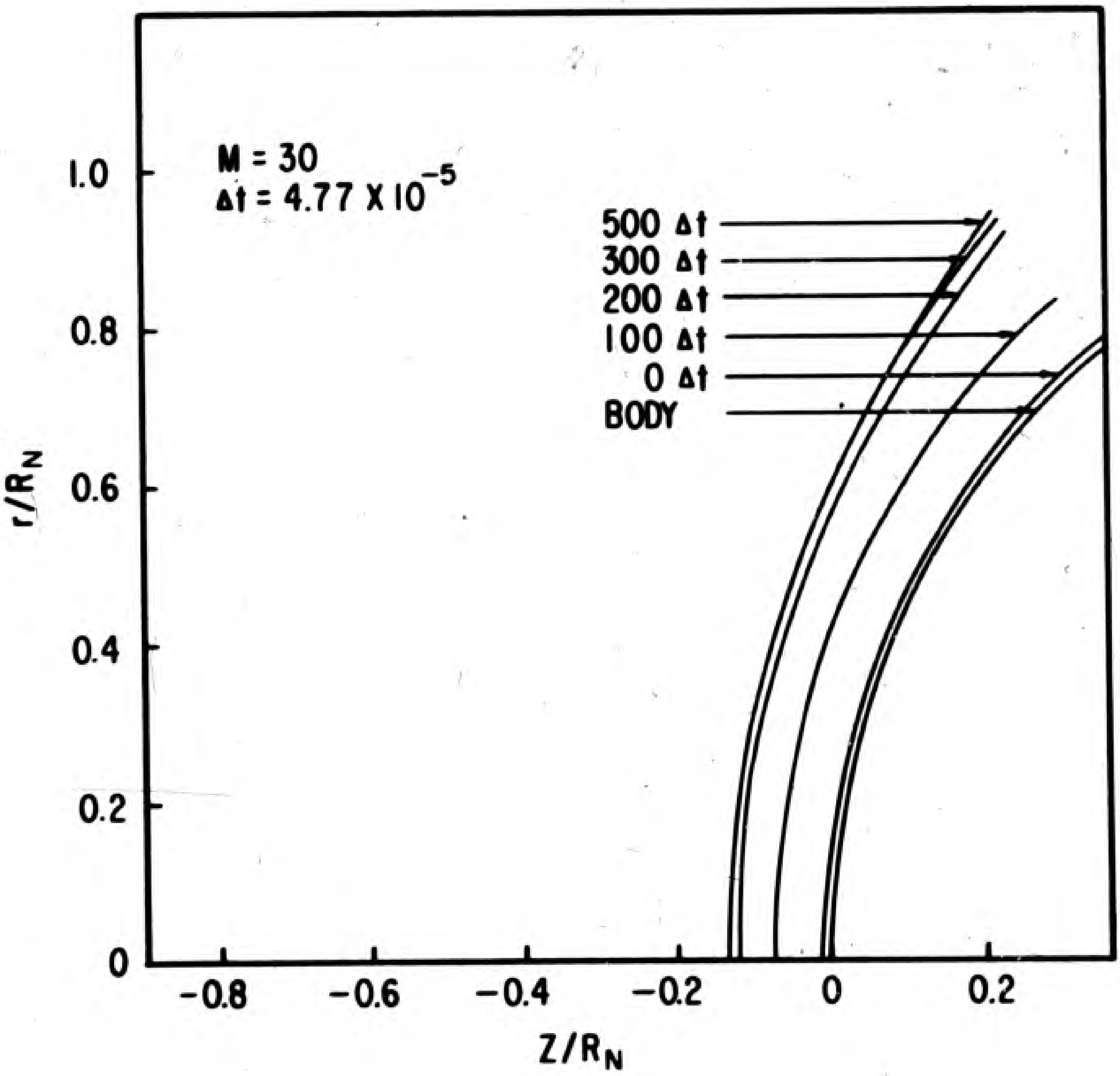


Figure 9. Shock Shapes At Various Times For Unsteady Flow Over A Sphere

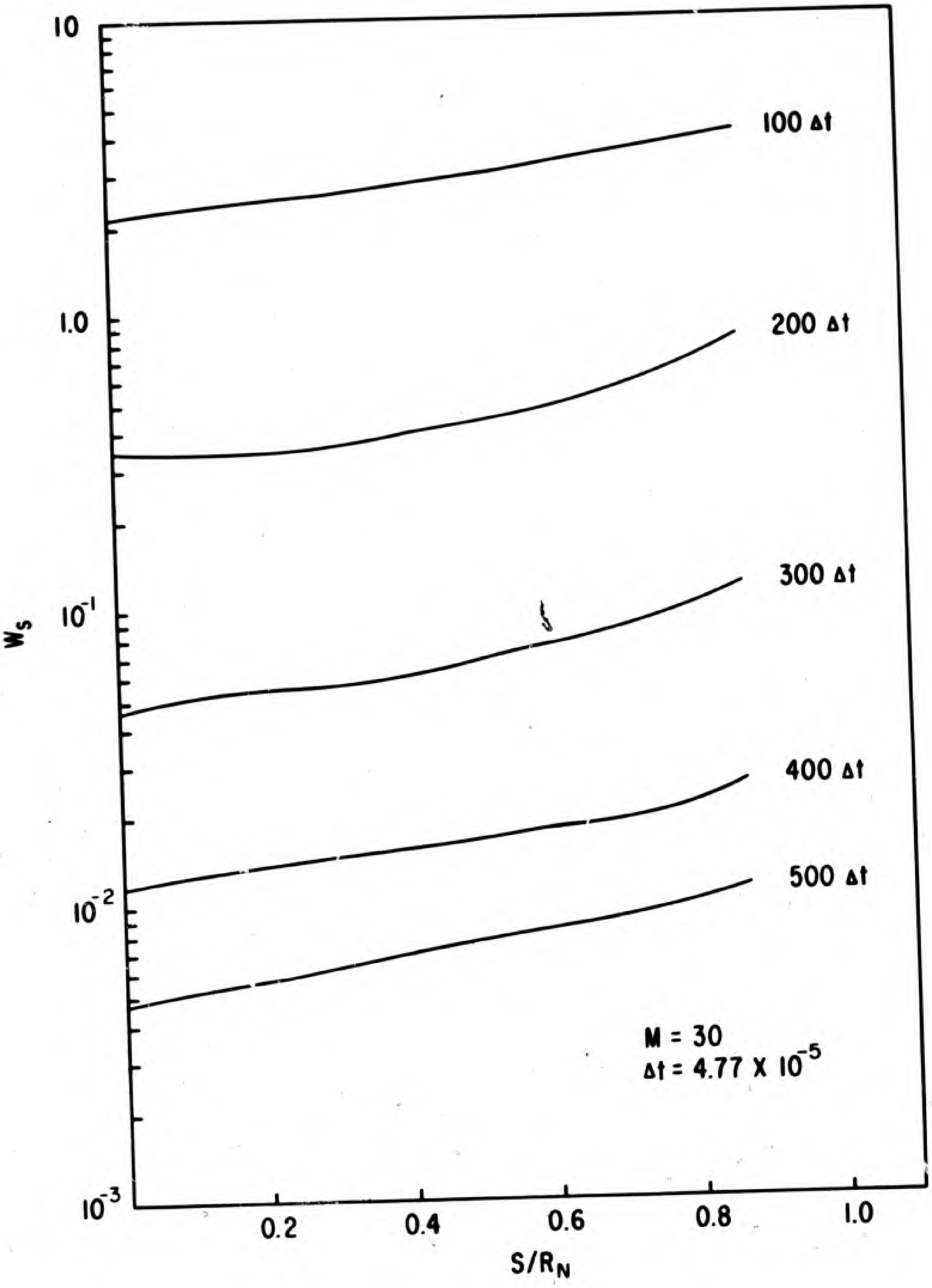


Figure 10. Shock Velocity At Various Times For Unsteady Flow Over A Sphere

Figure 11 illustrates shock shapes predicted for flow over a hemisphere at Mach numbers ranging from 1.5 to 500. No difficulties were encountered for any of these cases. Shock shapes predicted by the method of reference 13 are also shown. Since the shock point calculations for the present method are based on the method of reference 13, it is useful to ensure that both give the same results.

4. VALIDATION OF THE METHOD FOR BLUNT BODY FLOWS

To establish the accuracy of the present method for blunt body flows, the 3 strip integral relations solution of Belotserkovskii (data taken from reference 16) was used as a standard. Figure 12 illustrates the surface pressure distribution for a sphere in a Mach 4 flow. It is seen that the present method and Belotserkovskii's solution are in excellent agreement. Also shown are several time-dependent solutions reported in the literature. The solution by Moretti and Abbett (Ref. 13) is in excellent agreement with the present method in the subsonic region but predicts higher surface pressures in the supersonic region than either the present method or the integral relations solution. Solutions by the Godunov method reported by Godunov, Zabrodin and Prokopov (Ref. 9) and Masson (Ref. 16) are also shown. The solution by Masson is seen to agree very well with Belotserkovskii's results and the present method. It is concluded that the present method and Masson's method provide the best predictions of the time-dependent methods presented. Figure 13 illustrates the surface density distribution predicted by the present method, Belotserkovskii and Masson. Agreement between the present method and Belotserkovskii's solution is excellent. It is seen that Masson's solution does not predict the correct density distribution. It is concluded that the accuracy of the present method is superior to other reported time-dependent methods and equivalent to the method of integral relations.

5. AFTERBODY FLOWS

To validate the present method for afterbody flows, comparison with the NASA inverse scheme-method of characteristics program (Ref. 3) was accomplished. Since the surface entropy has been shown to be constant and correct, comparison of the predicted surface pressure distributions is sufficient to establish validity. Figures 14, 15 and 16 show the comparison of predicted surface pressures for sphere-cone vehicles with various cone half-angles and Mach numbers of 4, 6 and 10. Agreement between these methods is seen to be excellent.

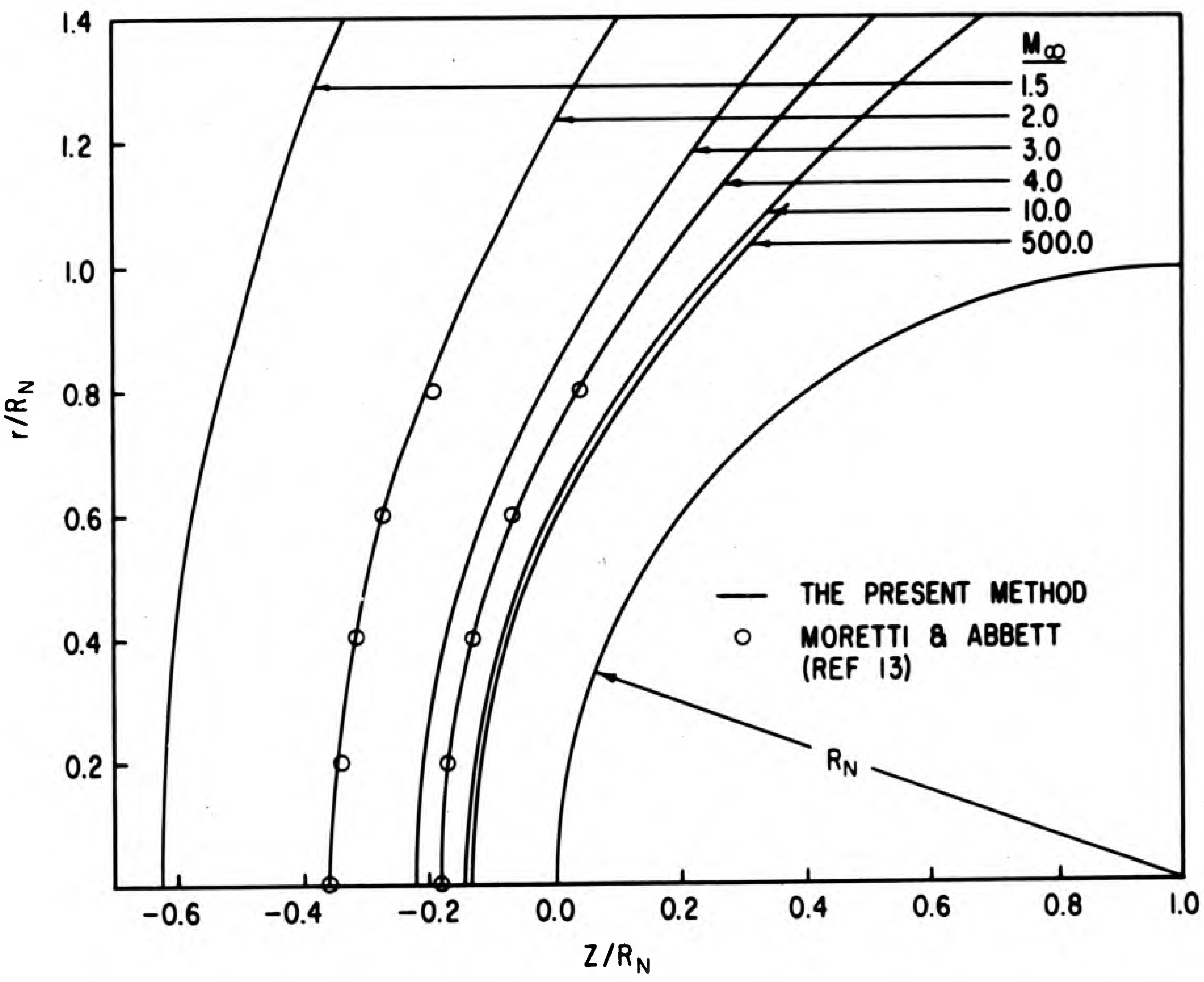


Figure 11. Shock Shapes For Flow Over A Hemisphere At Several Mach Numbers

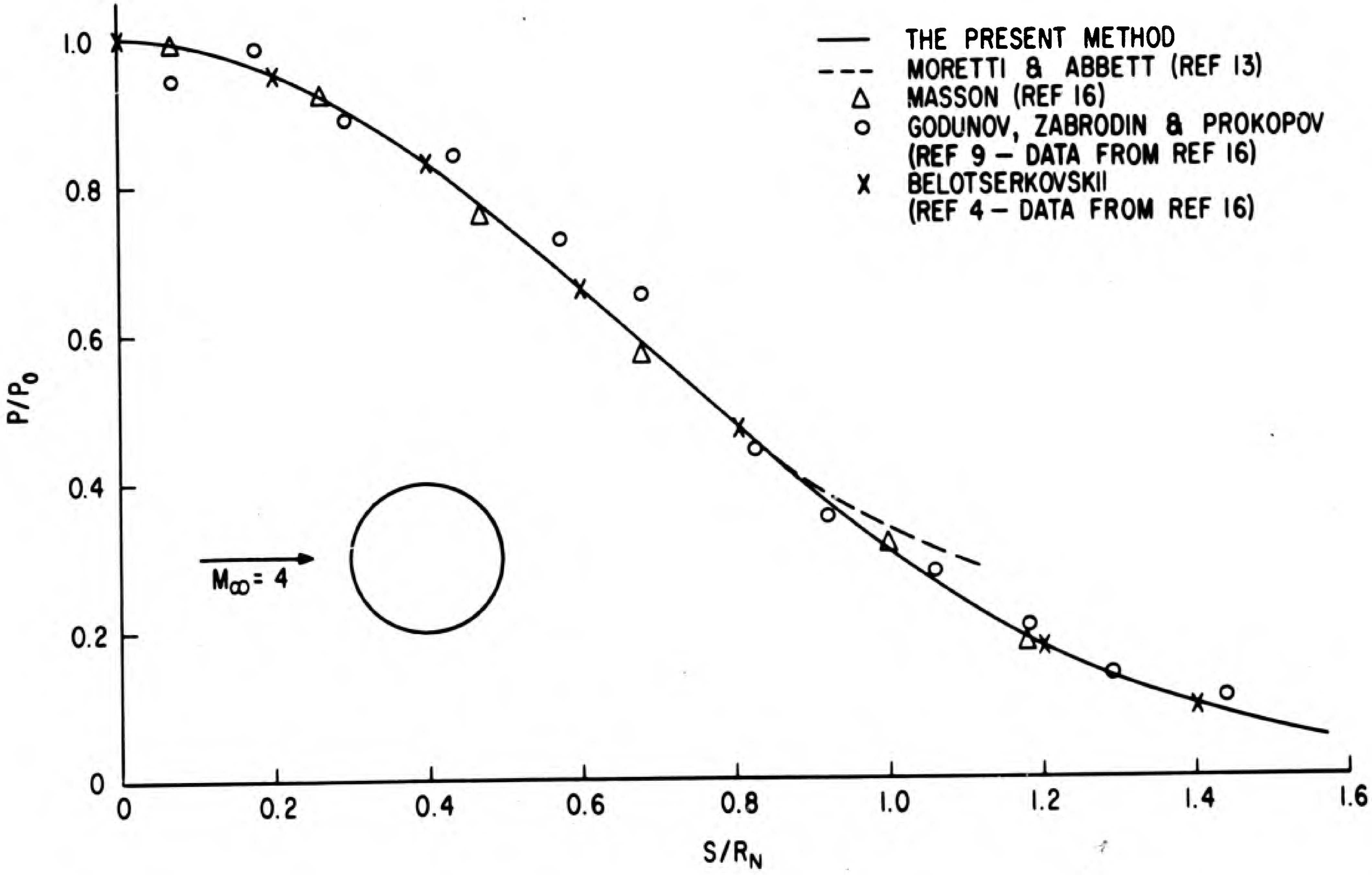


Figure 12. Comparison Of Time-Dependent Methods With An Integral Relations Solution

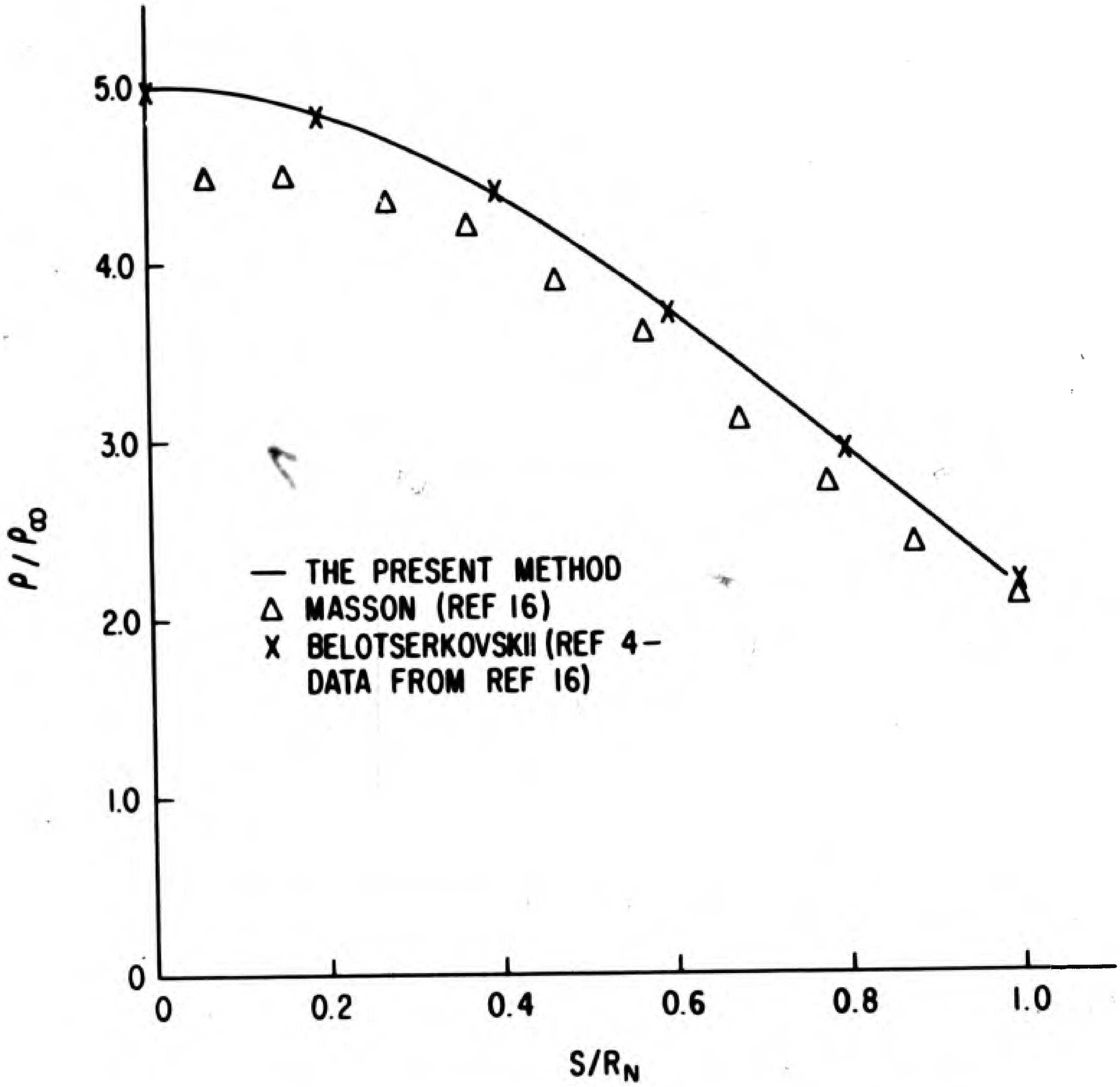


Figure 13. Surface Density Distributions For A Sphere In Mach 4 Flow

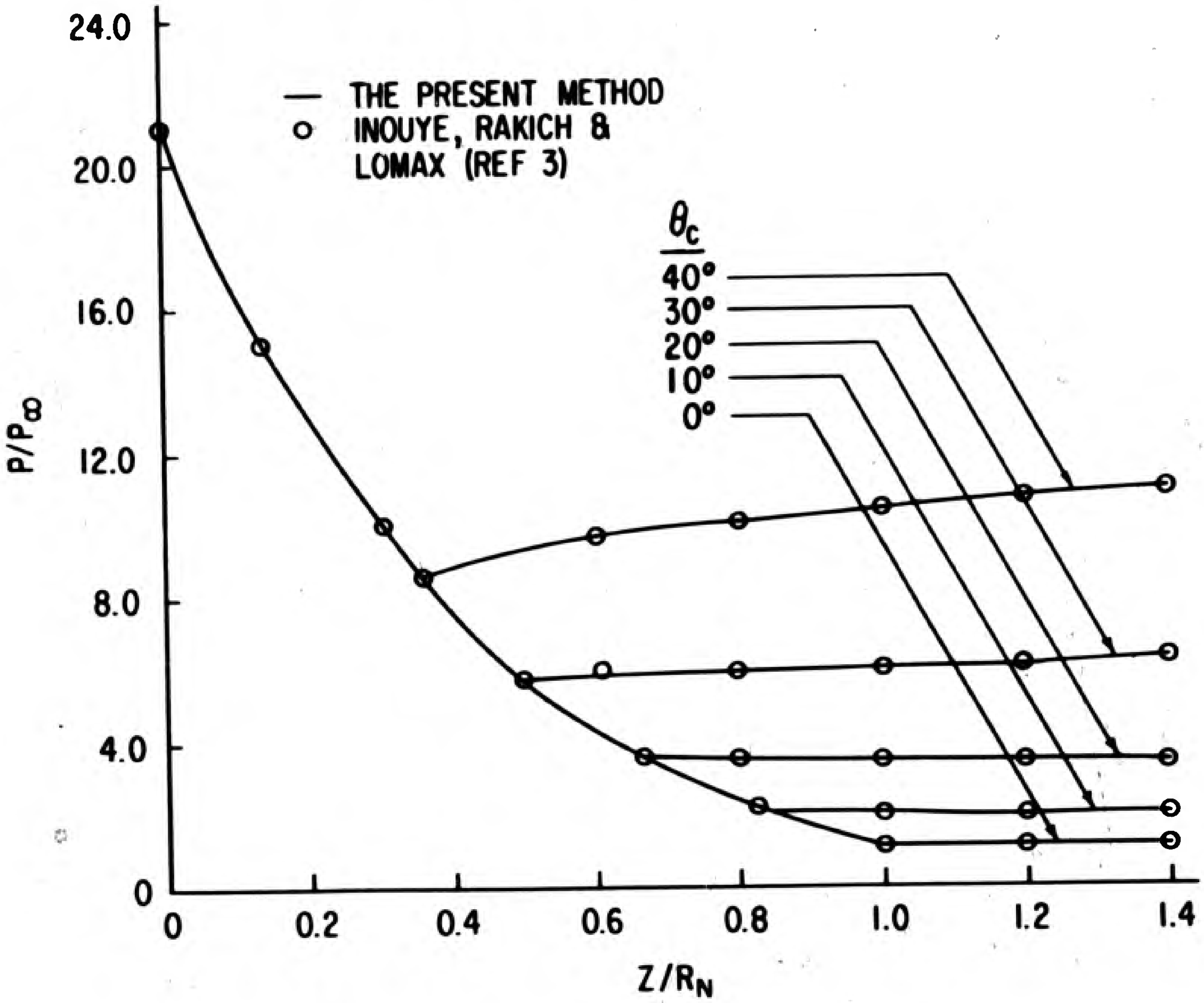


Figure 14. Surface Pressure Distributions For Sphere-Cones In Mach 4 Flow

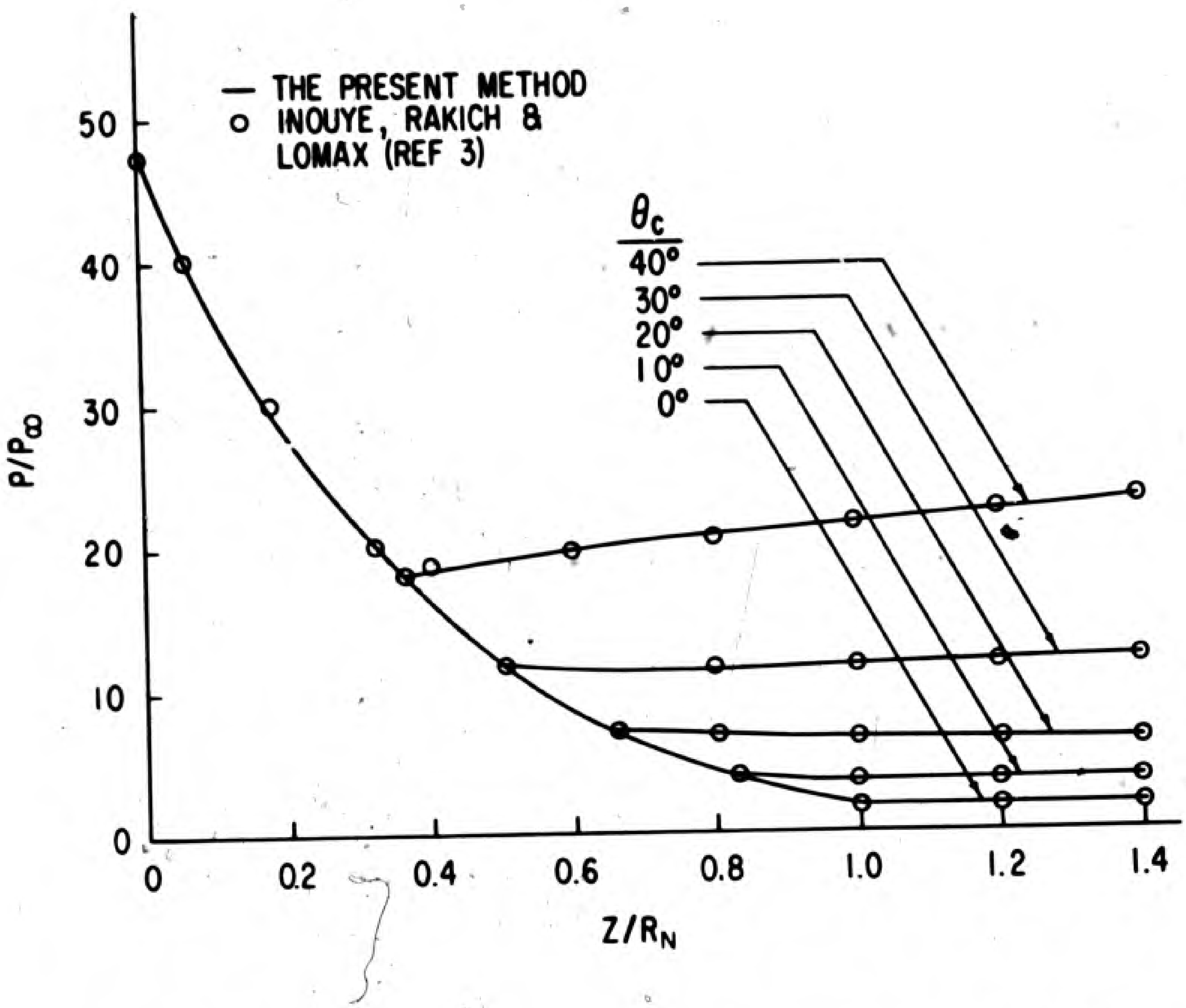


Figure 15. Surface Pressure Distribution For Sphere-Cones In Mach 6 Flow

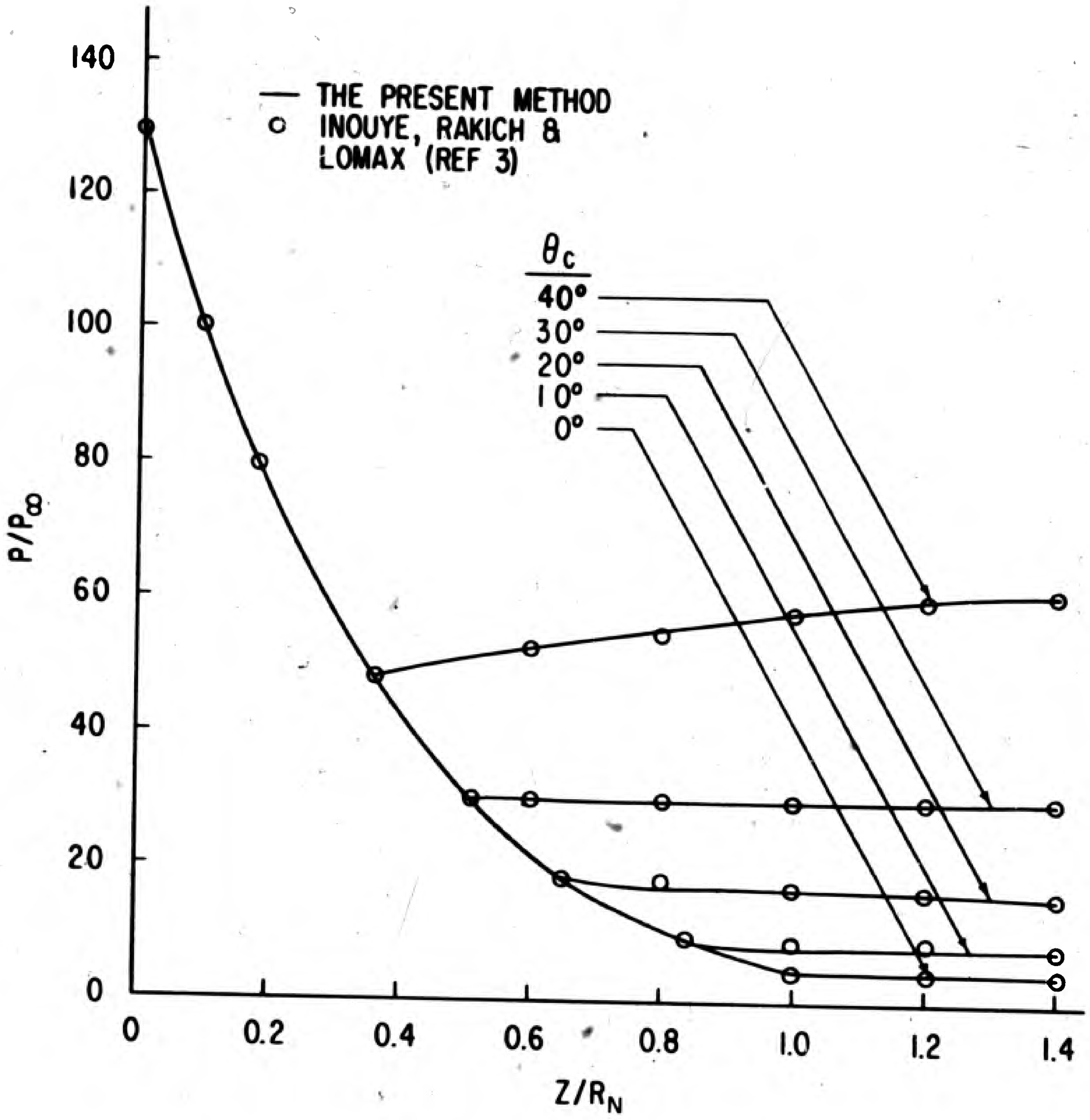


Figure 16. Surface Pressure Distribution For Sphere-Cones In Mach 10 Flow

6. ARBITRARY BODY SHAPES

The previous solutions for sphere-cone vehicles employed internal relations in the computer code to define the body geometry. To validate the numerical computations used to define an arbitrary body shape from input data, ellipsoid-cylinder vehicles were considered for comparison with reference 3. Figure 17 shows a comparison of surface pressures between these methods. It is concluded that the arbitrary body shape logic provides accuracy equivalent to reference 3 for those cases where comparison is possible.

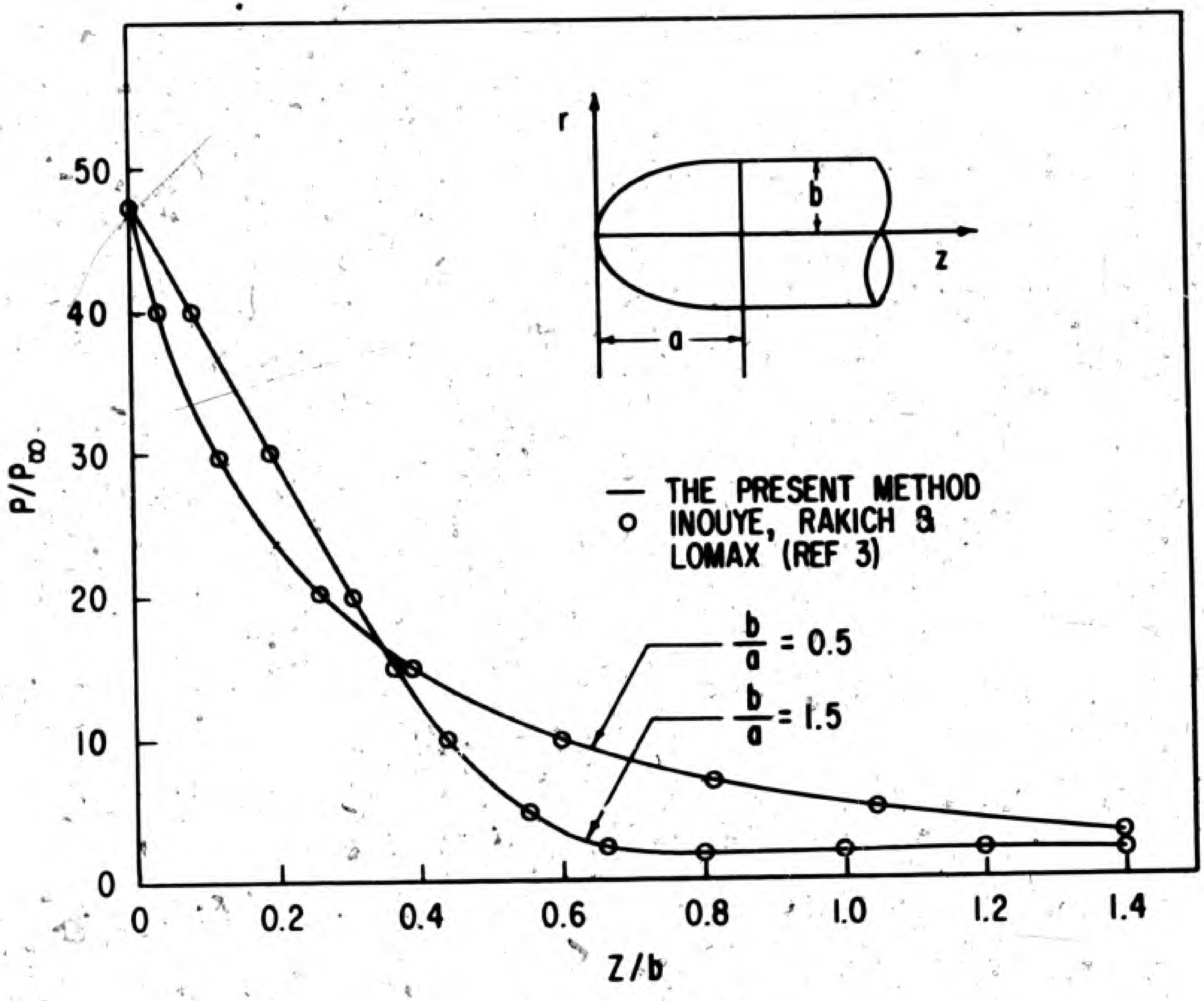


Figure 17. Surface Pressure Distributions For Ellipsoid-Cylinders In Mach 6 Flow

7. APPLICATION TO TYPICAL REENTRY VEHICLES

Since the accuracy of the present method has been established for the relatively simple problems normally used to demonstrate the capability of a computational method, it remains only to attempt a practical reentry problem. The AFWL version of the NASA computer code of reference 3 and the present method were used to generate a solution for a sphere-cone vehicle with a 10° cone half-angle in a flow with a Mach number of 20. The solution was carried downstream 50 nose radii measured along the symmetry axis. The surface pressure distribution comparison is shown in figure 18. Some disagreement is seen in the strong expansion region. Additional reduction of the stabilizing terms and a solution not using the segmented approach were attempted but had no significant effect on the predictions by the present method. It was concluded that this minor disagreement could not readily be explained. A comparison of the surface Mach number predictions is shown in figure 19. Except for the local disagreement in the strong expansion region, the solutions appear to be equivalent. To further investigate the local disagreement, a comparison of surface pressure distribution predicted by the NASA program and the present method are compared to experimental data reported in reference 19. Figure 20 shows that the present method agrees better with the experimental data than the NASA solution. This closer agreement is undoubtedly fortuitous, but does indicate that the local disagreement between the two computational methods is insignificant when compared to experimental data. The solution shown in Figures 17 and 18 represents the only known solution for a practical high performance reentry vehicle generated with a time-dependent method. The solution was generated in less than 6 minutes on the CDC 6600 computer and required only 40,000 octal locations for storage. As previously noted, the computation time required to treat this problem without the segmented approach is estimated to be in excess of 70 hours on the same computer.

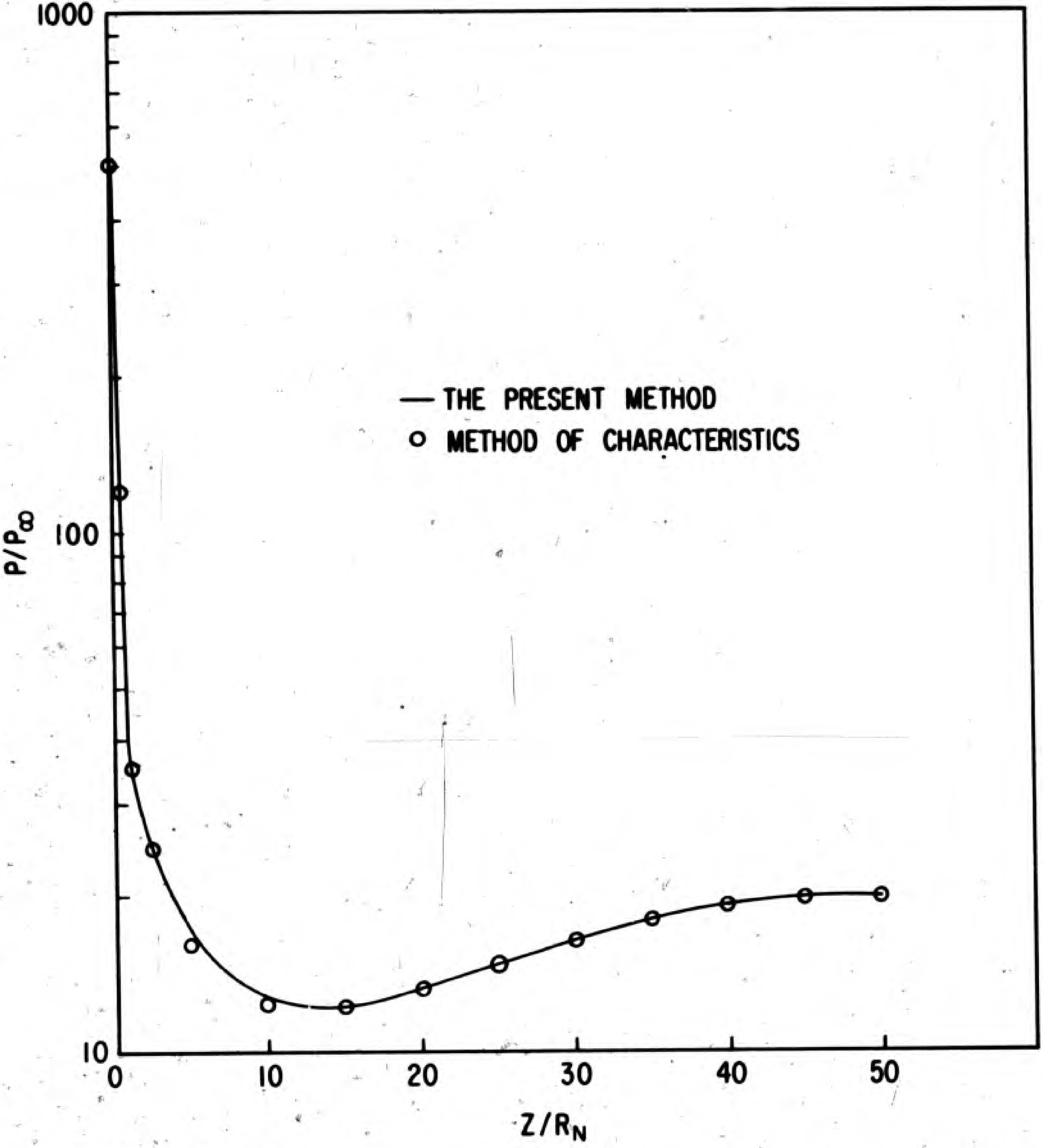


Figure 18. Surface Pressure Distribution For A 10° Sphere-Cone In Mach 20 Flow

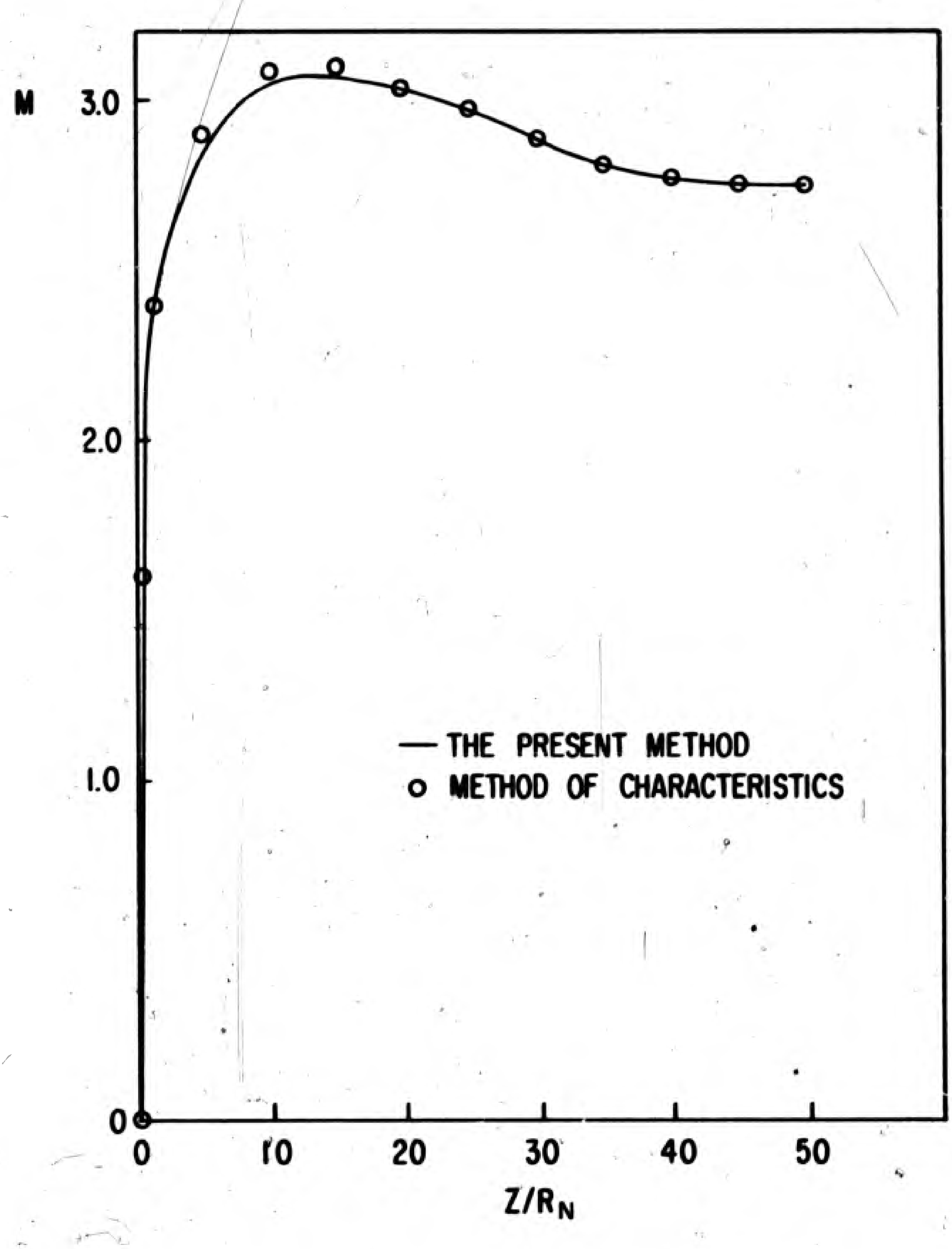


Figure 19. Surface Mach Number Distribution For A 10° Sphere-Cone In Mach 20 Flow

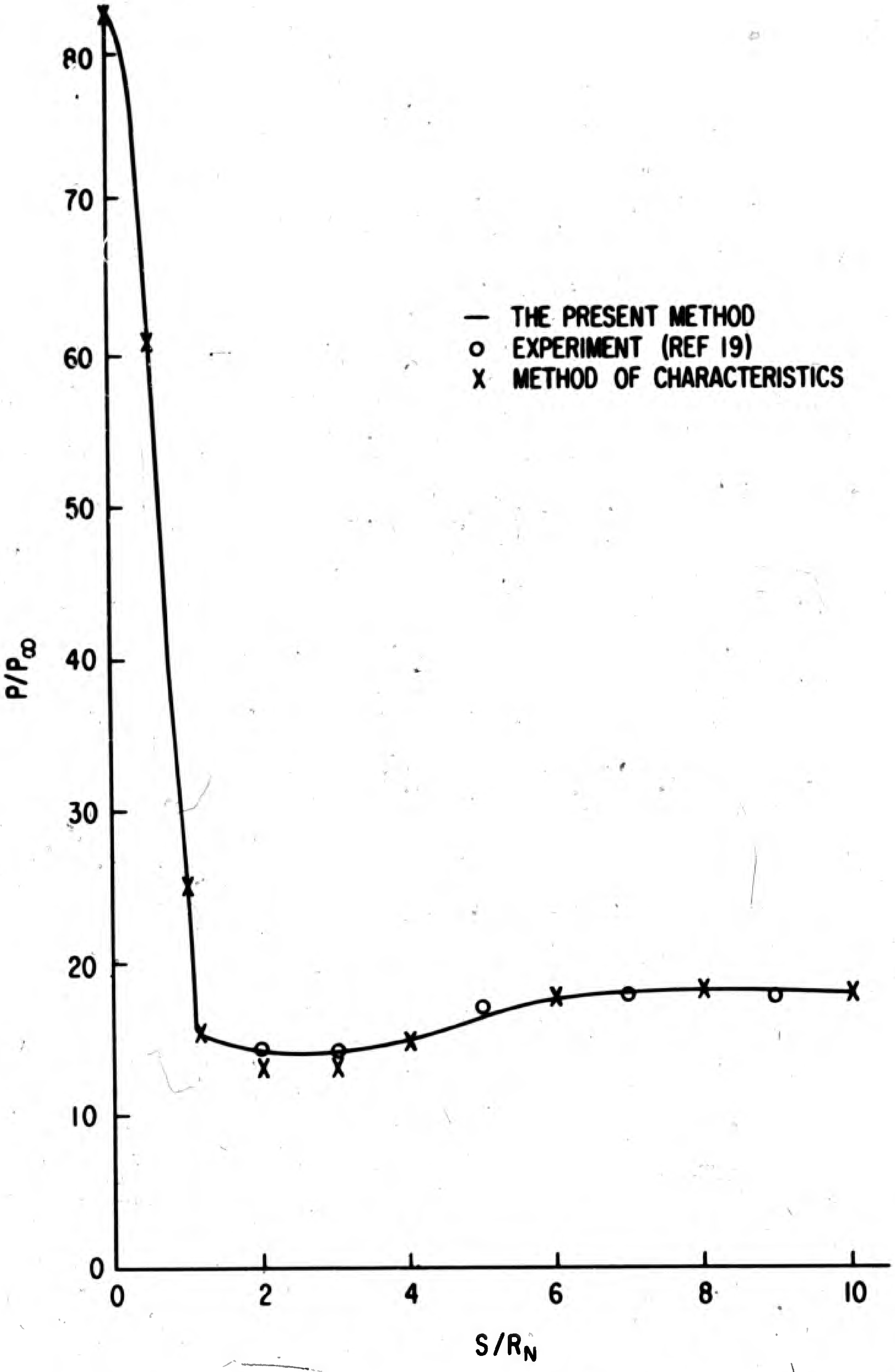


Figure 20. Surface Pressure Distribution For A 25° Sphere-Cone In Mach 8 Flow

SECTION VI

THE FORTRAN IV COMPUTER CODE

1. MACHINE ROUTINES

The computer code employs 3 machine routines available on the CDC 6600 computer. When a different computer is used, these routines may not be available. Substitute routines should be provided, as required, for the following routines:

INT (X) - a FUNCTION routine which converts the floating point variable, X, to integer form.

FLOAT (I) - a FUNCTION routine which converts the integer variable, I, to floating point form.

SKFILE (FILE, N) - a SUBROUTINE which skips N files on a magnetic tape unit specified by the Hollerith argument FILE. To simplify the replacing of this routine, calls to it are contained only in the dummy routine, SUBROUTINE SKIP.

2. DIMENSIONED VARIABLES

The variable dimensions currently contained in the computer code allow up to 15 nodes along the body and 40 nodes across the shock layer. To change these dimensions, new values for IMAX and NMAX should first be supplied in a DATA statement in the main program, where

IMAX - specifies the maximum number of nodes along the body permitted by the dimension sizes.

NMAX - specifies the maximum number of nodes across the shock layer permitted by the dimension sizes.

Then, the following COMMON statements should be modified consistent with the values of IMAX and NMAX:

```
COMMON/1/ V(IMAX,NMAX,4), D(NMAX,4), P(3,NMAX)
COMMON/2/ RB(IMAX), ZB(IMAX), TH(IMAX), CTH(IMAX), K(IMAX),
   NS(IMAX), WS(IMAX)
COMMON/11/ MAXIT, ISTOP, ITD, TIME, SI(IMAX), NST(IMAX)
COMMON/SPLINE/ C(4,IMAX)
```

SUBROUTINES INTERP and INTEG contain a dimensional variable which should be dimensioned X(IMAX).

3. THE MAIN PROGRAM

A. PURPOSE

The main program is designated as PROGRAM BLUNT. It's major functions are:

a) Initialize constants and some of the case data variables.

b) Define values for the control variables to select the proper options.

c) Accomplish checks to determine if any errors were sensed while reading the input data. When errors are sensed, an error message is printed, the case terminated, and the next case started.

d) Performs all flow field computations except for points on the shock and on the body surface.

e) Schedules the flow field data output as requested by the user.

f) Schedules the storing of the flow field data on magnetic tape.

B. VARIABLE LIST

AMACH1 - M_∞

AMUO - μ_o, equation (62)

AVN - parameter used to introduce the stabilizing terms relative to the n coordinate.

AVS - parameter used to introduce the stabilizing terms relative to the s coordinate.

A1 - A, equation (49), evaluated at the shock.

BNOSE - nose bluntness parameter

B1 - B, equation (49), evaluated at the shock.

CDEL - cos $(\beta-\theta)$

CN - $(\kappa^2\rho V)_n$

CS - $2(\rho u)_s$

CTH - array of cos θ values

C1 - $u/(2\Delta\xi)$

C2 - $-\eta\frac{\partial n_s}{\partial s}\frac{u}{\chi(v-W_s\eta)}$

C3 - $\chi(v-W_s\eta)/(2n_s\Delta\eta)$

C30 - $\frac{1}{2n_s\Delta\eta}$

D - array used for temporary storage of flow field parameters at $r + \Delta r$.

DBET - $\beta - \theta$

DELS - length of a body segment measured along the surface.

DET - $\Delta\eta$

DE2 - $2\Delta\eta$

DKAP - $\Delta\kappa = K\Delta\eta$

DN - Δn

DR - $\Delta r = \Delta n \cos\theta$

DRSM - r at $\xi - \Delta\xi$

DRSP - r at $\xi + \Delta\xi$

DT - $\Delta\tau$

DTO - 0.9 / $(q_\infty + a_\infty)$. Used for CFL stability check.

DT1 - the value of $\Delta\tau$ evaluated from equation (62).

DUM - dummy variable used for intermediate computations.

DXI - $\Delta\xi$

DX2 - $2\Delta\xi$

DX3 - $3\Delta\xi$

ERR - array specifying acceptable errors in convergence on n and P for shock iteration procedure.

ETA - η

GAMA - $(\gamma+1)/2$

GAMB - $(\gamma-1)/2$

GAM1 - γ

HT1 - total enthalpy.

I - index specifying body station in DO loops.

IGEOM - control variable. Specifies method for selecting the number of streamwise nodes to be used.

IM - I - 1

IMAX - maximum number of streamwise nodes allowed by the dimension sizes.

IMIN - value of I at which computations should start on the current segment.

INTAP - the magnetic tape unit number to be used for flow-field input (when required).

IP - I + 1

IPR - array specifying iteration numbers or times when flow field output is to be provided.

IPRINT - iteration number when next flow field output is to be provided.

ISEG - segment number.

ISTOP - error indicator.

IT - index on major iteration DO loop.

ITD - total number of iterations performed for the current segment.

JPR - array used for temporary storage of IPR.

K - array containing values of K.

KAP - κ

KBAR - $(K(\xi+\Delta\xi) + 2K(\xi) + K(\xi-\Delta\xi))/4$

KGEOM - control variable defining the option to be used for geometry calculations.

KSEG - index used to suppress the check on NSEG (see NSEG).

KZERO - body curvature at S = 0. Used to nondimesionalize all length parameters.

L - specifies first index in the P array corresponding to I.

LM - specifies first index in the P array corresponding to IM.

LP - specifies first index in the P array corresponding to IP.

LPLUS - array used to specify LP. LP = LPLUS (L).

M - dummy index.

MAXI - total number of streamwise nodes employed to treat current segment.

MAXIT - total number of iterations to be performed.

MAXN - the total number of nodes across the shock layer.

MI - MAXI - 1

MN - MAXN - 1

M1 - used for temporary storage of M_∞.

N - index specifying nodes across shock layer in DO loops.

NCASE - case number (see section VII).

NFILE - number of the first segment to be treated (see section VII).

NGEOM - control variable specifying method for selecting MAXN.

NM - N - 1

NMAX - total number of nodes across the shock layer allowed by dimension sizes.

NMIN - MAXN - 2

NOPT - control variable specifying method to be used in selecting MAXIT.

NP - N + 1

NPR - index used to select proper value of IPRINT from the IPR array.

NS - array containing values of n_s.

NSEG - total number of segments to be treated.

NTAP - control variable specifying the method to be used for assigning initial conditions.

OTAP - magnetic tape unit number on which flow field data should be stored.

P - array containing values of P at the 3 body stations required for local computations.

PI - π

PI2 - $\pi/2$

PN - $\frac{\partial P}{\partial n}$

PS - $\frac{\partial P}{\partial s}$

PT2 - total pressure behind a normal shock.

P1 - P_∞

P90 - empirical constant used to assign initial conditions.

Q1 - q_∞

R - r

RB - array containing values of r at the body stations.

RGAS - gas constant.

RHO1 - ρ_∞

RN - scale factor used for dimensional output (see section VII).

RSM - r at $(\xi-\Delta\xi)$

RSP - r at $(\xi+\Delta\xi)$

SDEL - sin $(\beta-\theta)$

TCHECK - estimated value of τ for steady state, equation (63).

TCRIT - array containing constants used in equation (63).

TDEL - tan $(\beta-\theta)$

TH - array containing values of θ at all body stations.

THETC - cone half-angle.

TIME - τ

TPR - array containing values of τ when output should be provided.

T1 - T_∞

V - array containing values of ρ, u, v and S at all nodes.

VA - array containing the values of averages used to introduce the stabilizing terms.

VIS - 1.0 - AVS - AVN. Used to compute averages stored in VA.

VK - Kv

VMIN - array used to store streamwise derivatives required for shock computations.

VN - array containing values of derivatives with respect to n.

VS - array containing values of streamwise derivatives.

VS1 - a_∞

VT - array containing values of derivatives with respect to r.

WS - array containing values of W_s for all streamwise stations.

WSH - component of $W_s(\xi)$ normal to the body.

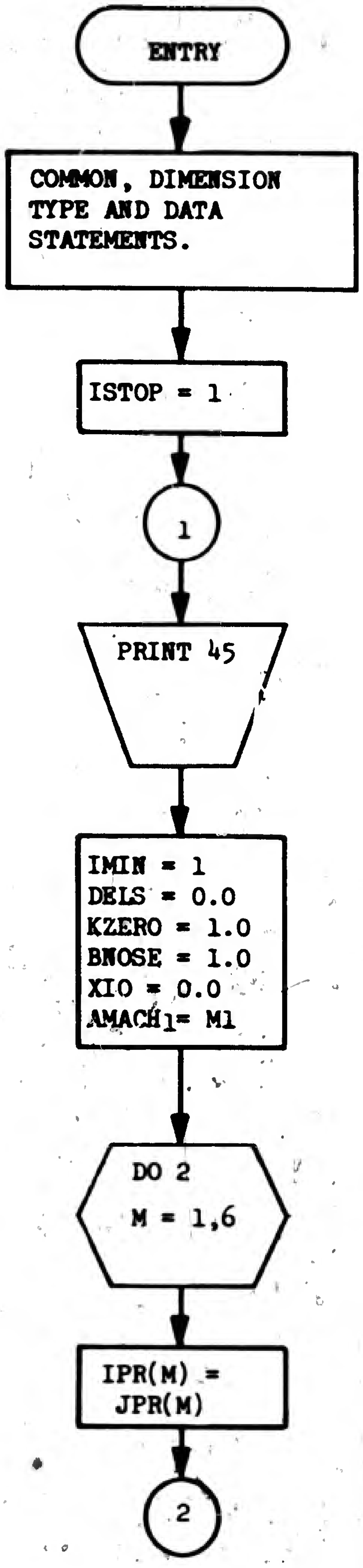
XIO - value of ξ at start of segment.

ZB - array containing values of z at all body stations.

ZRN - axial length (Z/R_N) for sphere-cones.

ZZ - dummy variable used for intermediate computations.

C. FLOW CHART



ENTER MAIN PROGRAM.

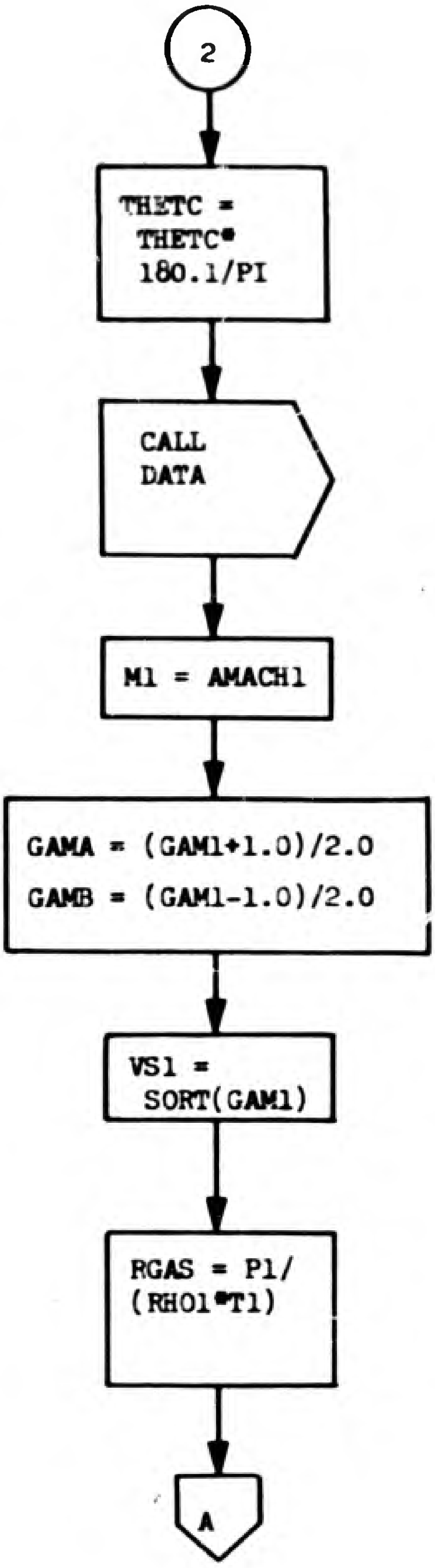
SET ERROR INDICATOR.

START POINT FOR ALL CASES.

PRINT TITLE.

INITIALIZE BODY GEOMETRY CONSTANTS AND RECOVER THE MACH NUMBER READ ON THE PREVIOUS CASE.

RECOVER PRINTOUT ARRAY READ ON PREVIOUS CASE.



CHANGE UNITS ON θ_c
TO DEGREES.

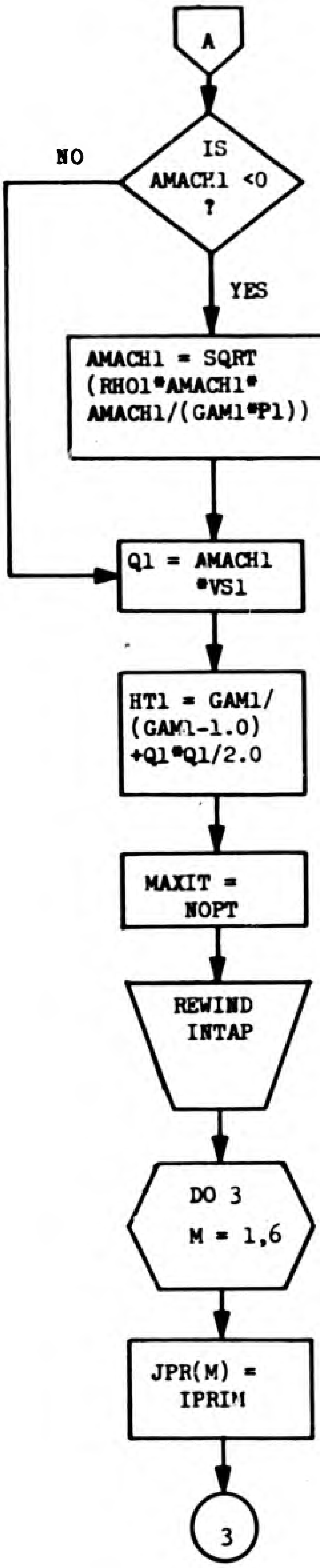
LOAD CASE DATA.

SAVE MACH NUMBER.

STORE $\frac{\gamma+1}{2}$ AND $\frac{\gamma-1}{2}$
FOR SHOCK COMPUTATIONS.

COMPUTE NONDIMENSIONAL
FREE STREAM SOUND SPEED.

COMPUTE GAS CONSTANT
FOR USE IN DATA PRINTOUT.



WAS A FREE STREAM VELOCITY READ RATHER THAN A MACH NUMBER ?

COMPUTE FREE STREAM MACH NUMBER.

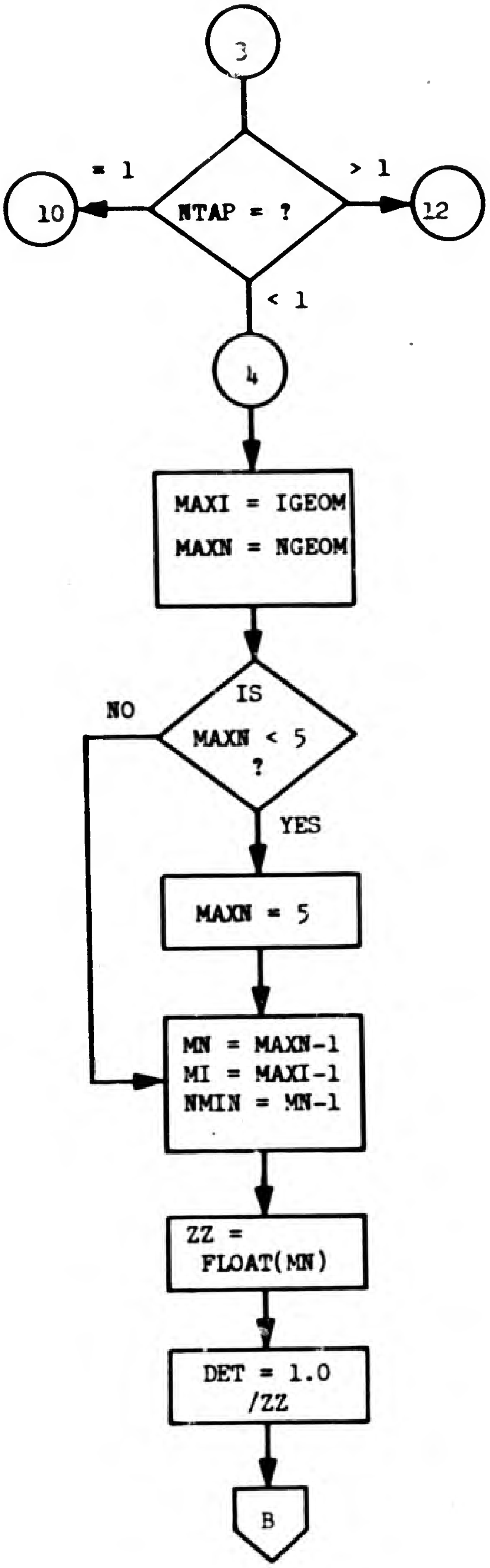
COMPUTE FREE STREAM VELOCITY.

COMPUTE TOTAL ENTHALPY

STORE VALUE OF MAXIT READ IN CASE DATA.

REWIND INPUT MAGNETIC TAPE UNIT.

STORE PRINTOUT ARRAY.



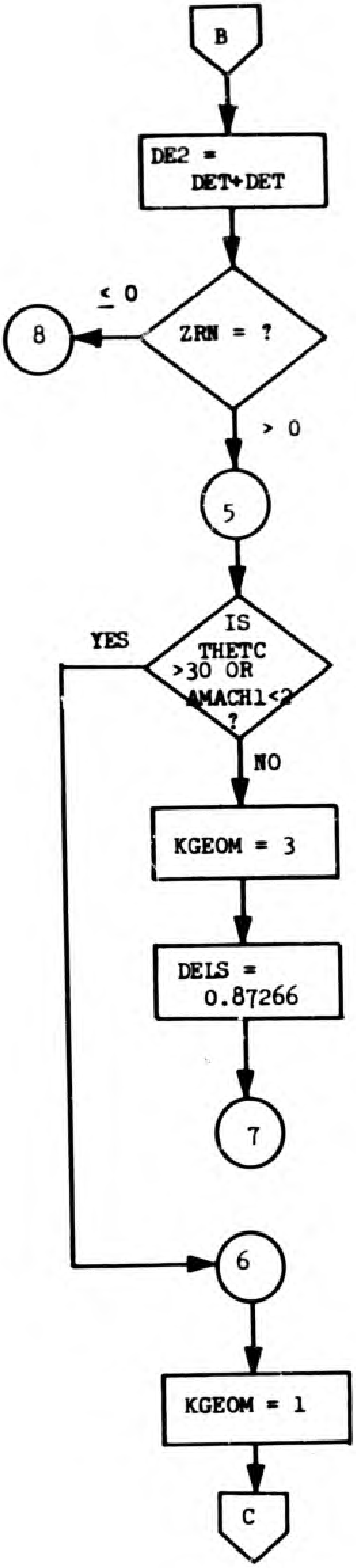
NEW CASE, RESTART OR
ADDITIONAL ITERATIONS ?

LOAD THE SPECIFIED
NUMBER OF NODES IN
THE ξ AND η DIRECTION.

MAXN < 5 ?

STORE MAXN-1, MAXI-1
AND MAXN-2.

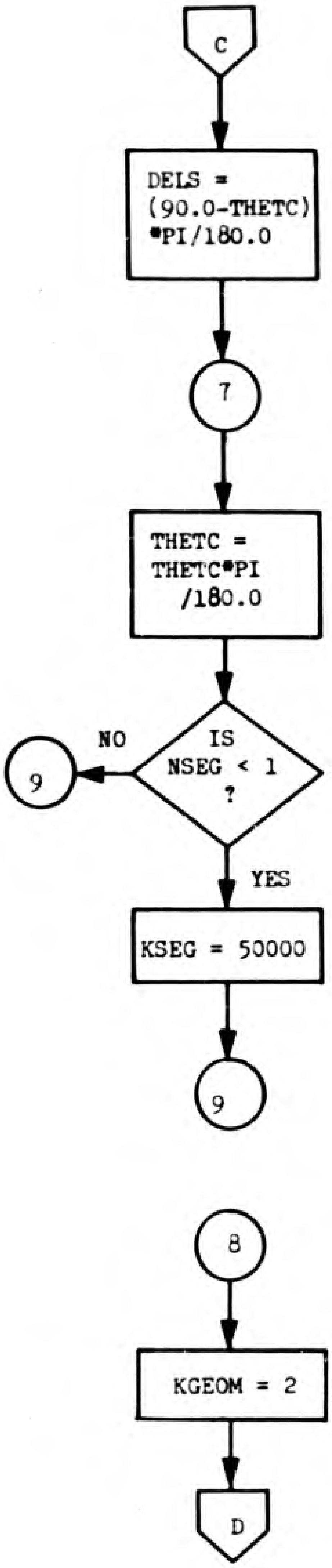
COMPUTE $\Delta\eta$.



SHOULD SPHERE NOSE BE TREATED WITH ONE OR TWO SEGMENTS ?

TWO SEGMENTS WILL BE USED.

ONE SEGMENT WILL BE USED.

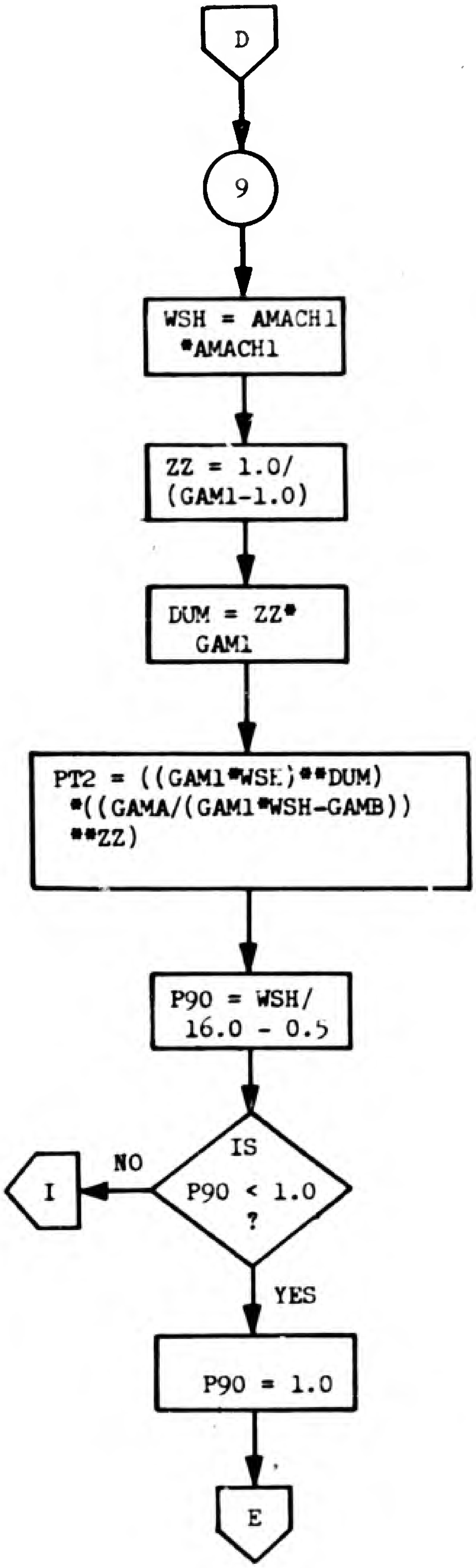


CONVERT θ_c TO RADIANS.

WAS THE NUMBER OF SEGMENTS TO BE TREATED SPECIFIED ?

NO. COMPUTATIONS WILL STOP WHEN THE TOTAL LENGTH, ZRN, HAS BEEN TREATED.

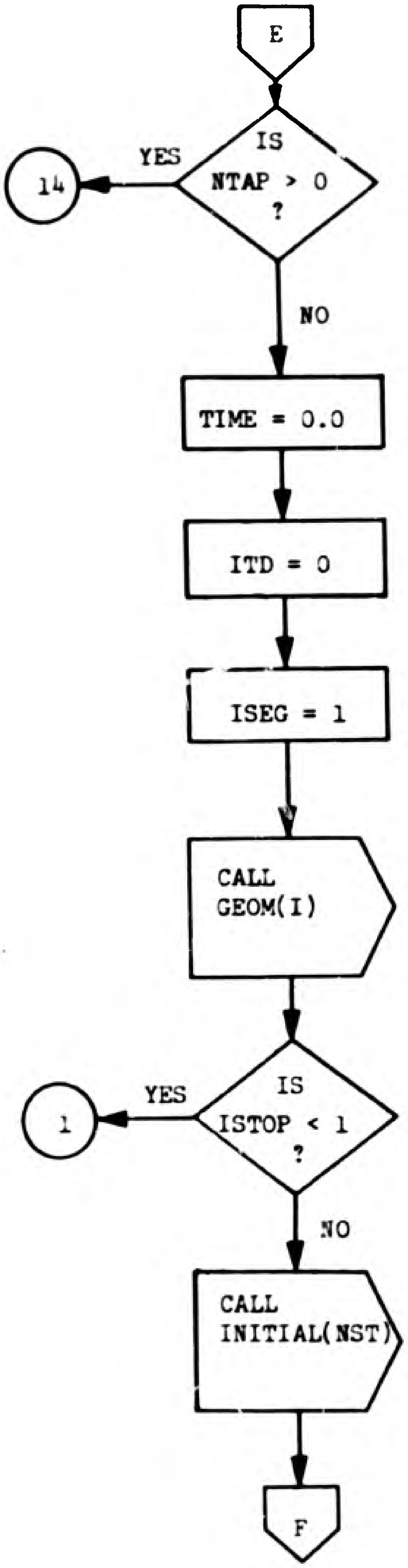
ARBITRARY BODY SHAPE OPTION.



COMPUTE CONSTANTS FOR STAGNATION PRESSURE CALCULATION.

COMPUTE STAGNATION PRESSURE BEHIND A NORMAL SHOCK. PERFECT GAS.

EMPIRICAL CONSTANT USED TO ESTIMATE INITIAL BODY PRESSURE.



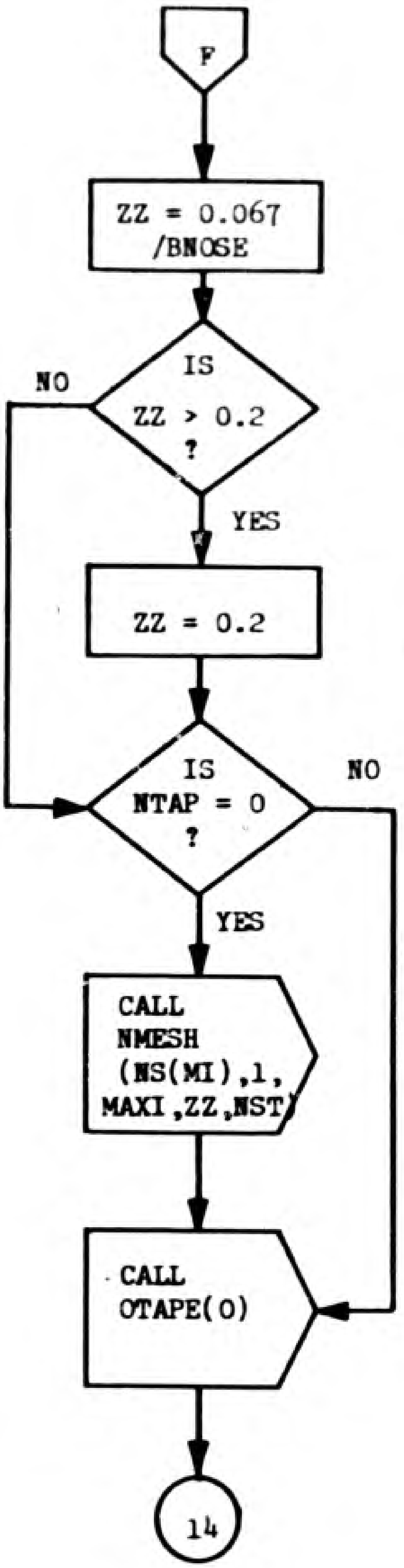
RESTART OR ADDITIONAL ITERATIONS ?

NO. NEW CASE. INITIALIZE CONSTANTS FOR FIRST SEGMENT.

SET UP BODY GEOMETRY FOR FIRST SEGMENT.

WAS DATA ERROR SENSED IN SUBROUTINE GEOM ?

INITIALIZE FLOW FIELD PARAMETERS FOR FIRST SEGMENT.



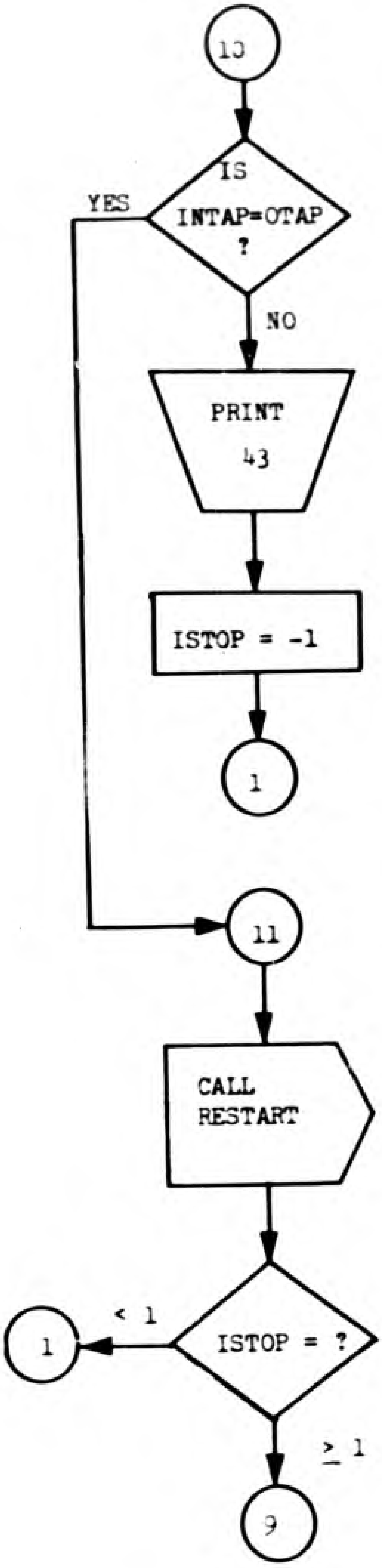
SET $\Delta\eta$ AT S = 0.

$\Delta\eta$ REQUIRED TO BE $\leq$ 0.2

NTAP <0, UNSTEADY FLOW CASE, DOES NOT EMPLOY AUTOMATIC NODAL SPACING OPTION.

SET NODAL SPACING ACROSS SHOCK LAYER.

STORE INPUT ARR·Y ON MAGNETIC TAPE.



RESTART OPTION.

THIS OPTION REQUIRES
INTAP = OTAP.

PRINT ERROR MESSAGE.

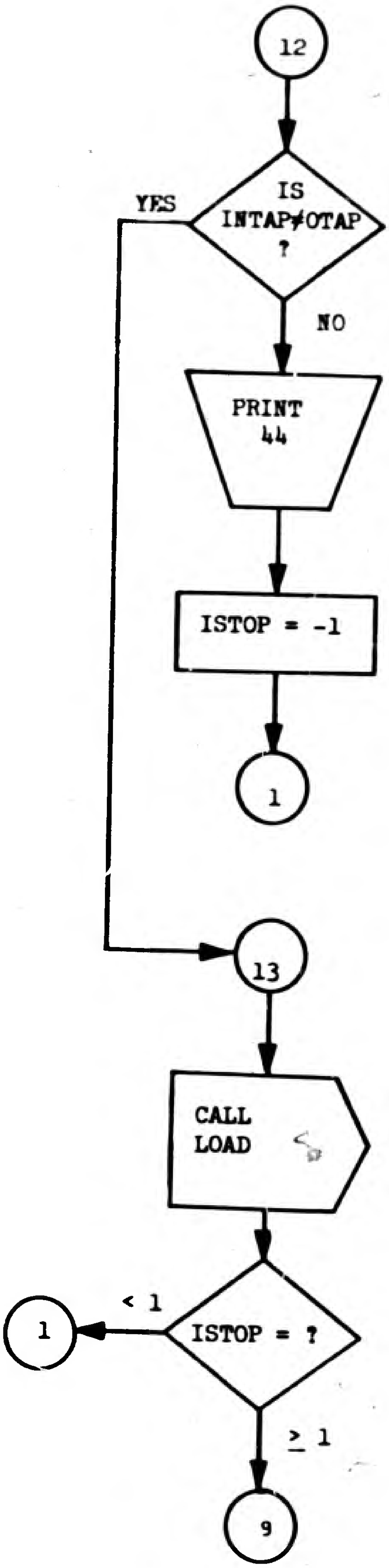
SET ERROR INDICATOR.

READ NEXT CASE.

LOAD DATA FOR RESTART.

WAS DATA ERROR SENSED
IN RESTART ?

GO TO 9 TO COMPUTE PT2.



ADDITIONAL ITERATIONS TO BE PERFORMED ON A CASE STORED ON MAGNETIC TAPE.

OPTION REQUIRES INTAP ≠ OTAP.

PRINT ERROR MESSAGE.

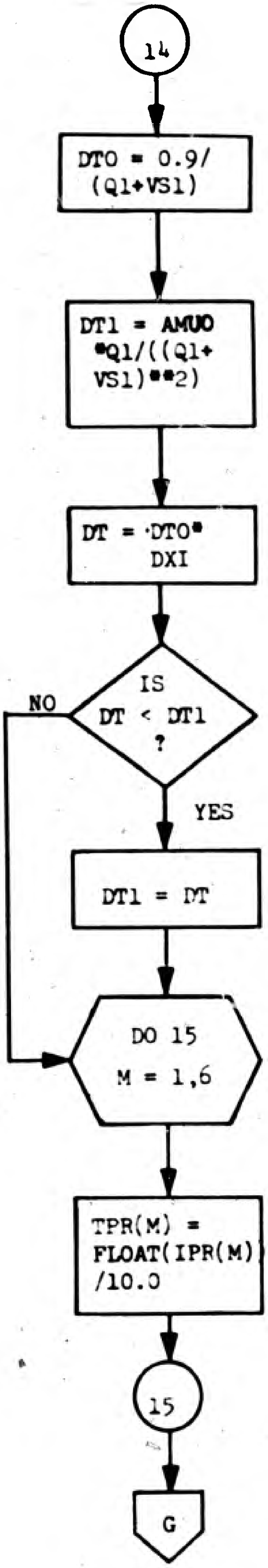
SET ERROR INDICATOR.

READ NEXT CASE.

LOAD DATA FROM MAGNETIC TAPE.

ERROR INDICATOR < 1 ? IN THIS CASE, IT SIGNIFIES THE END OF THE CASE.

GO TO 9 TO COMPUTE PT2.



COMPUTE CONSTANT, DTO, FOR USE IN CFL STABILITY CRITERION, EQUATION (36).

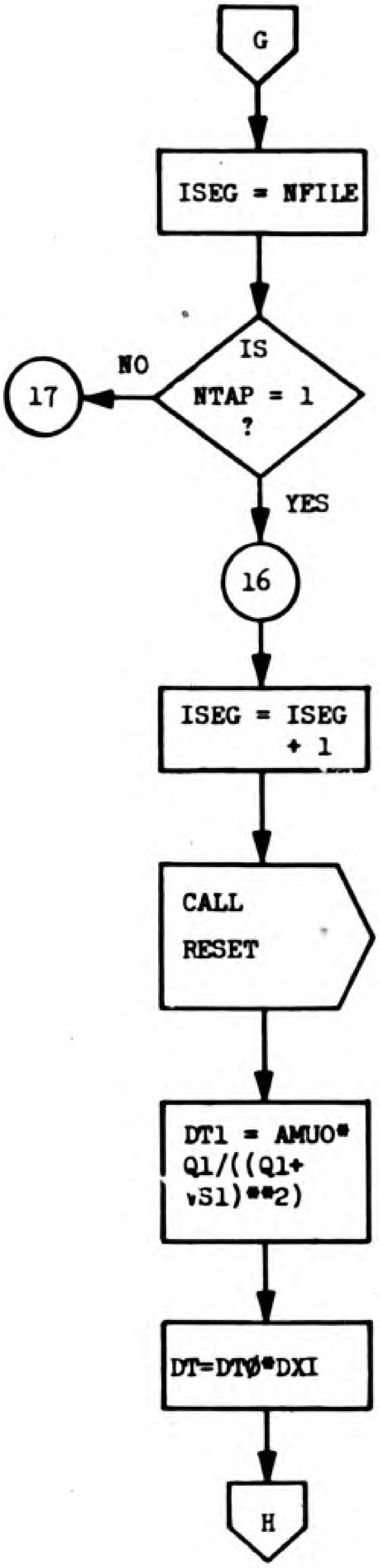
EQUATION (62).

COMPUTE CFL STABILITY LIMIT BASED ON $\Delta\xi$.

DOES DT1 EXCEED THIS LIMIT ?

YES. SET DT1 = DT.

SET TPR ARRAY FOR USE IN STEADY STATE FLOW OPTION.



INITIALIZE SEGMENT NUMBER.

RESTART OPTION ?

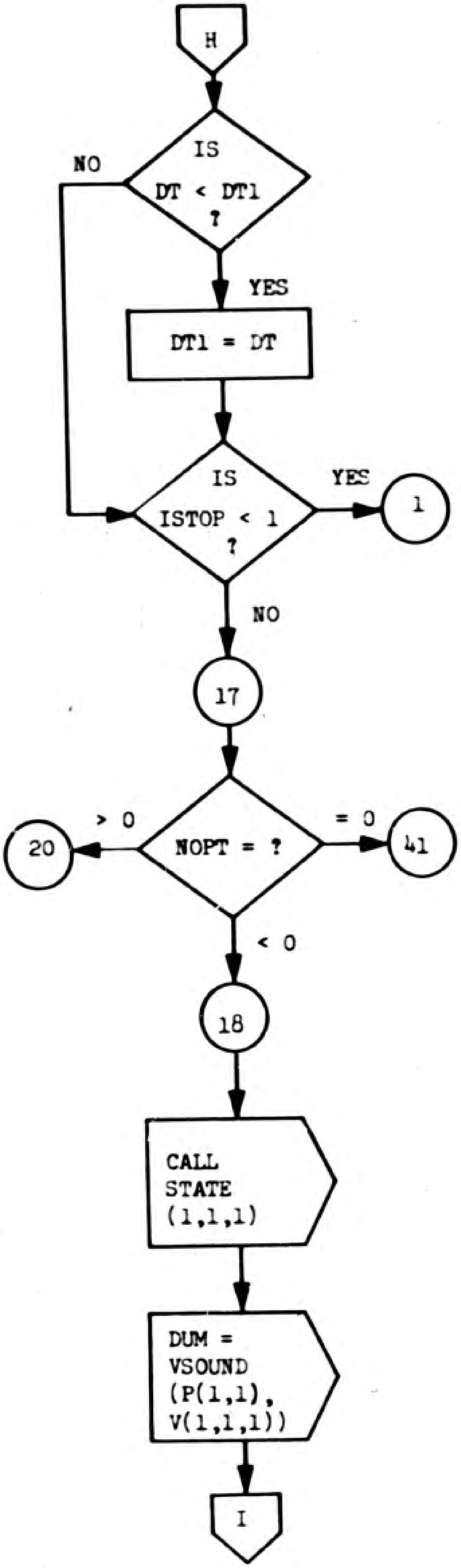
START OF SEGMENT COMPUTATIONS.

INCREMENT SEGMENT NUMBER.

SET UP INITIAL DATA FOR
PRESENT SEGMENT.

EQUATION (62).

COMPUTE CFL STABILITY LIMIT
BASED ON $\Delta\xi$.



DOES DT1 EXCEED THIS LIMIT ?

YES. SET DT1 = DT.

ERROR INDICATOR < 1 (THIS MEANS AN INPUT ERROR WAS SENSED OR THE CASE WAS COMPLETED).

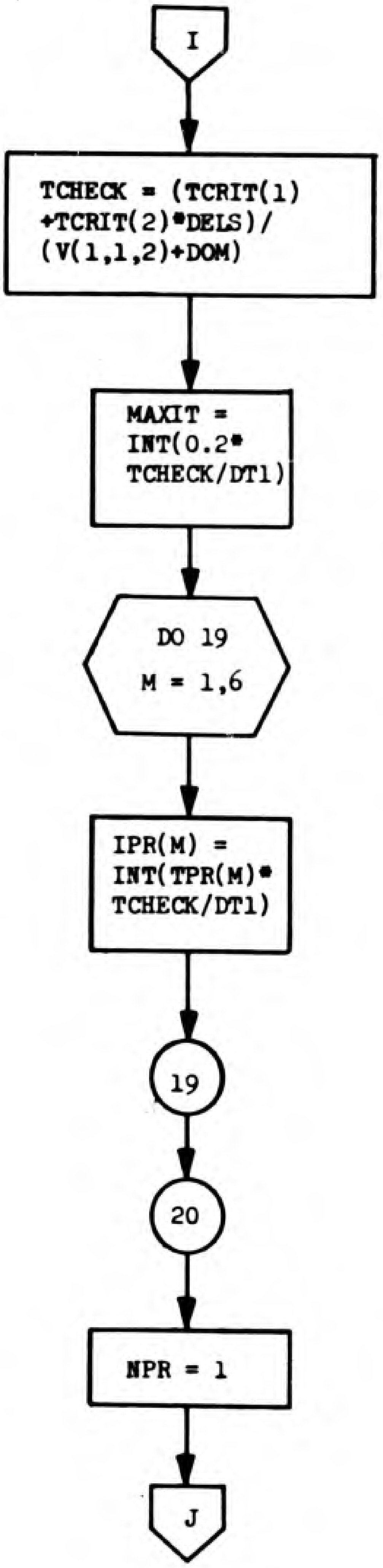
NOPT > 0 : NUMBER OF ITERATIONS SPECIFIED.

NOPT = 0 : NO ITERATIONS. JUST PRINT FLOW DATA.

NOPT < 0 : STEADY STATE FLOW SOLUTION.

COMPUTE BODY PRESSURE AT START OF SEGMENT.

COMPUTE BODY SOUND SPEED AT START OF SEGMENT.

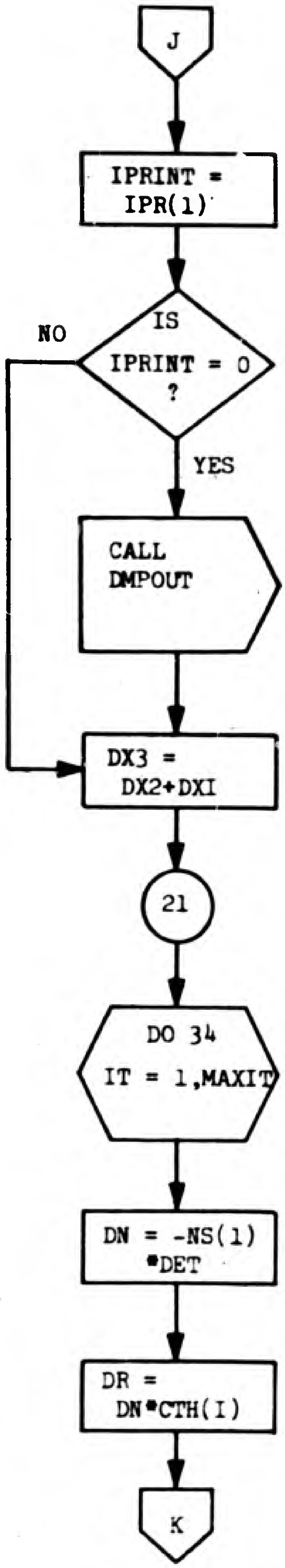

I
TCHECK = (TCRIT(1) +TCRIT(2)*DELS)/ (V(1,1,2)+DOM)
MAXIT = INT(0.2* TCHECK/DT1)
DO 19
M = 1,6
IPR(M) = INT(TPR(M)* TCHECK/DT1)
19
20
NPR = 1
J

ESTIMATE TIME REQUIRED FOR STEADY STATE SOLUTION, EQUATION (63).

SET MAXIT TO ALLOW SUFFICIENT ITERATIONS FOR A MEANINGFUL CFL STABILITY CALCULATION BASED ON $\Delta\eta$.

ESTIMATE ITERATION NUMBERS FOR WHICH PRINTOUT SHOULD BE PROVIDED.

INITIALIZE PRINTOUT INDEX.



SET CONTROL VARIABLE
FOR PRINTOUT.

ARE INITIAL CONDITIONS
TO BE PRINTED ?

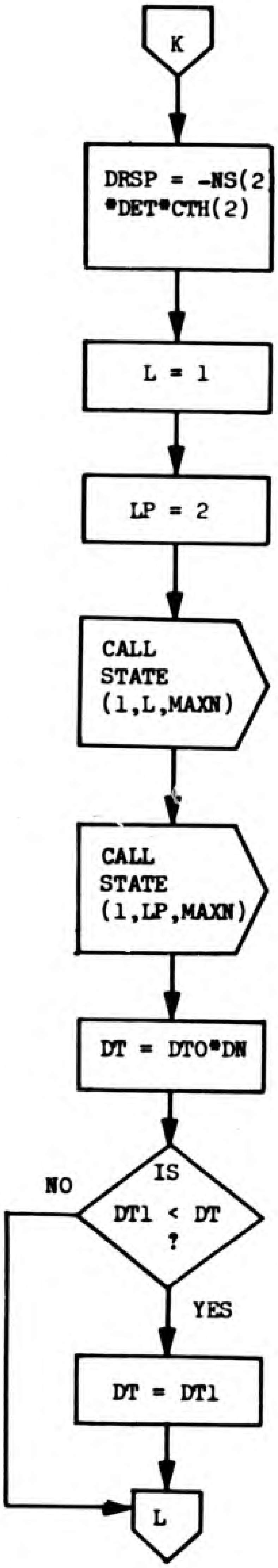
PRINT FLOW FIELD DATA.

STORE $3\Delta\xi$.

START OF ITERATION LOOP.

COMPUTE Δn AT START OF
SEGMENT; $\Delta n = -n_s \Delta\eta$.

COMPUTE Δr AT START OF
SEGMENT; $\Delta r = \Delta n \cos(\theta)$.



COMPUTE Δr AT SECOND NODE ALONG THE BODY FOR THIS SEGMENT.

INITIALIZE INDEX FOR PRESSURE AT ξ.

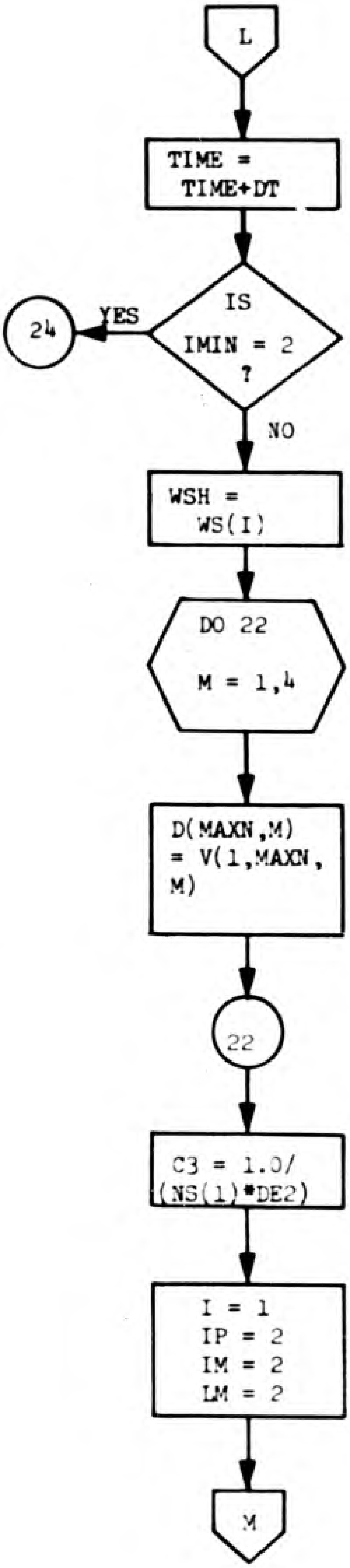
INITIALIZE INDEX FOR PRESSURE AT $\xi+\Delta\xi$.

COMPUTE PRESSURE ARRAY AT ξ.

COMPUTE PRESSURE ARRAY AT $\xi+\Delta\xi$.

COMPUTE CFL STABILITY LIMIT BASED ON Δn. EQUATION (36)

SHOULD Δt BE REDUCED BASED ON μ_o ?



INCREMENT TIME.

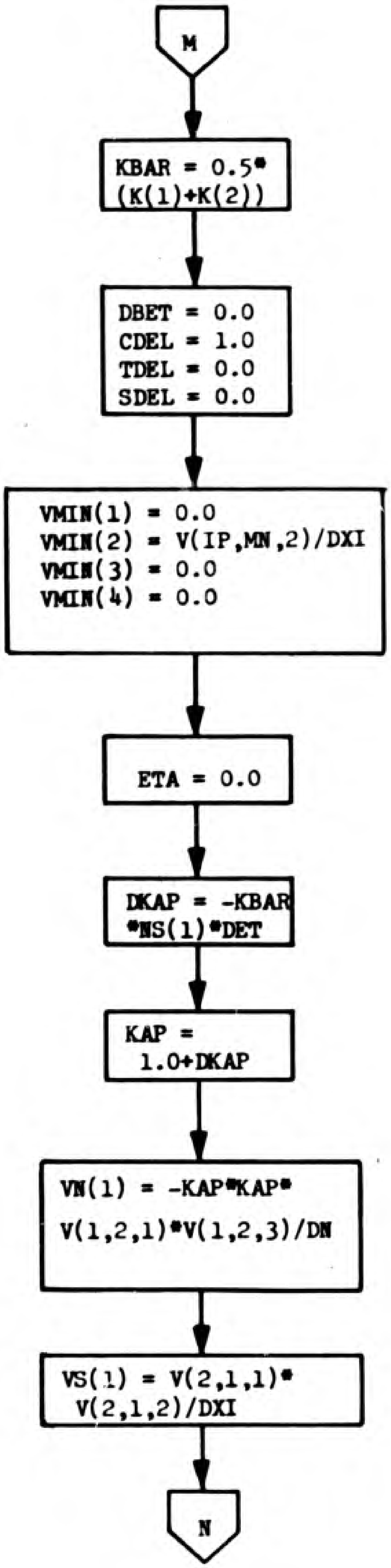
IMIN = 2 WHEN THE SEGMENT UNDER CONSIDERATION DOES NOT INCLUDE THE STAGNATION STREAMLINE.

COMPUTE SHOCK VELOCITY ALONG THE NORMAL COORDINATE.

STORE FLOW VARIABLES AT THE SHOCK.

COMPUTE CONSTANT FOR USE IN EVALUATING PARTIAL DERIVATIVES WITH RESPECT TO n.

SET NODAL INDICES FOR STAGNATION STREAMLINE COMPUTATIONS.



COMPUTE AVERAGE BODY CURVATURE FOR STAGNATION STREAMLINE DIFFERENCES.

SET β, COS (β), TAN (β), AND SIN (β) FOR NORMAL SHOCK COMPUTATIONS.

STORE $\frac{\partial \rho}{\partial \xi}$, $\frac{\partial u}{\partial \xi}$, $\frac{\partial v}{\partial \xi}$ AND $\frac{\partial s}{\partial \xi}$ AT THE FIRST NODE INSIDE THE SHOCK.

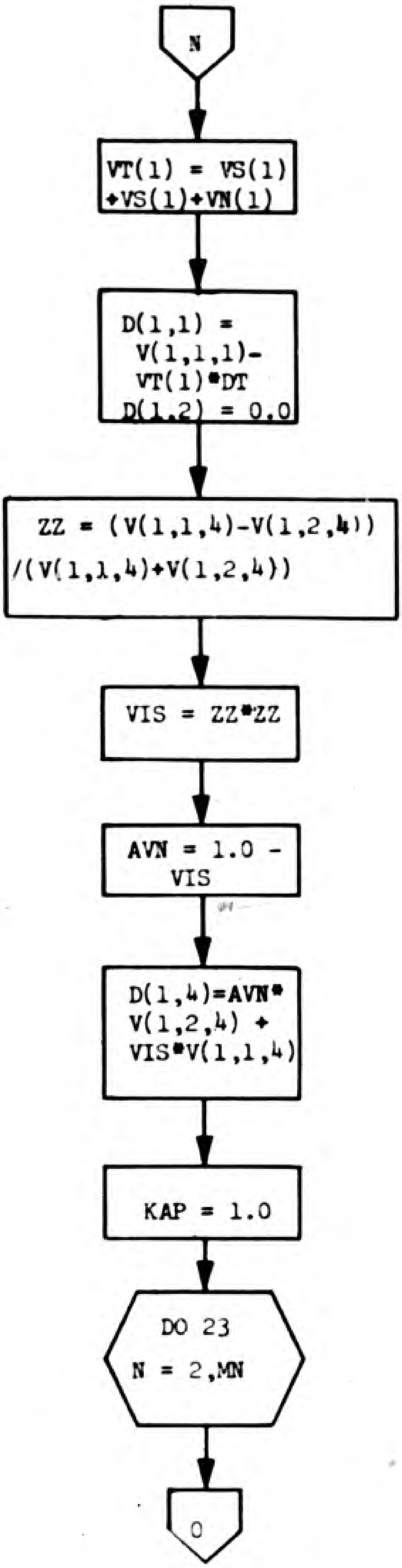
INITIALIZE η.

COMPUTE $\Delta\kappa = K\Delta n$.

COMPUTE κ AT $\eta = \Delta\eta$.

COMPUTE $\frac{\partial \kappa^2 \rho v}{\partial n}$ FOR USE IN EQUATION (16).

COMPUTE $\frac{\partial \rho u}{\partial s}$ FOR USE IN EQUATION (16).



COMPUTE $\frac{\partial \rho}{\partial t}$ AT THE STAGNATION POINT.

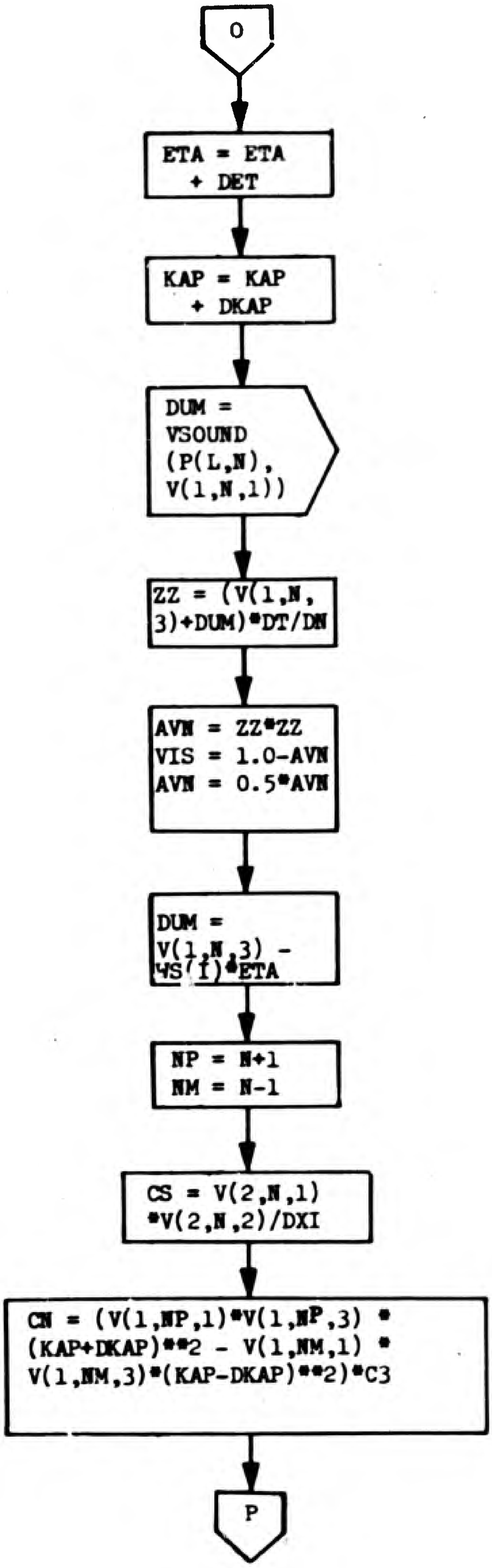
COMPUTE $\rho(\tau+\Delta\tau)$, EQUATION (16). SET u=o.

EQUATION (41).

COMPUTE $S(\tau+\Delta\tau)$, EQUATION (40).

INITIALIZE κ.

START LOOP ON N FOR POINTS ON THE STAGNATION STREAMLINE.


O
ETA = ETA + DET
KAP = KAP + DKAP
DUM = VSOUND (P(L,N), V(1,N,1))
ZZ = (V(1,N, 3)+DUM)*DT/DN
AVN = ZZ*ZZ
VIS = 1.0-AVN
AVN = 0.5*AVN
DUM = V(1,N,3) - WS(I)*ETA
NP = N+1
NM = N-1
CS = V(2,N,1) *V(2,N,2)/DXI
CN = (V(1,NP,1)*V(1,NP,3) * (KAP+DKAP)**2 - V(1,NM,1) * V(1,NM,3)*(KAP-DKAP)**2)*C3
P

INCREMENT η.

INCREMENT κ.

COMPUTE LOCAL SOUND SPEED.

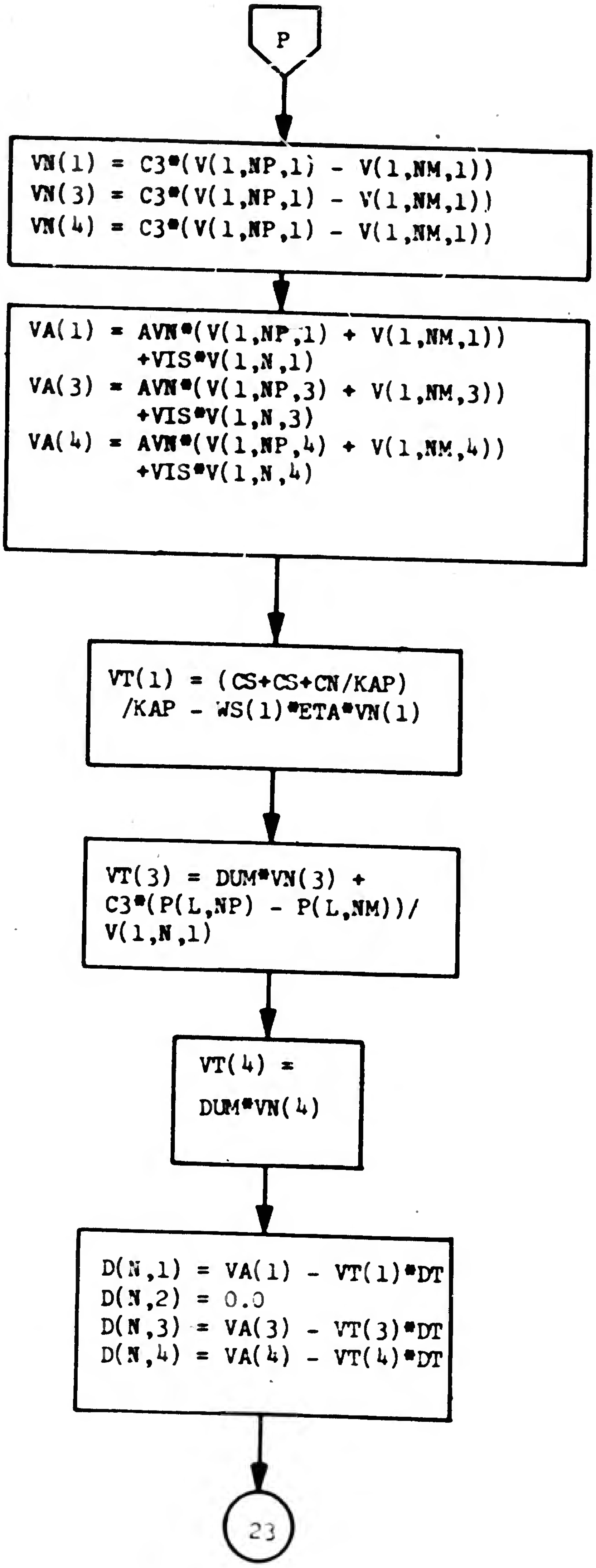
COMPUTE MINIMIZED STABILIZING TERMS.

STORE A USEFUL PARAMETER.

STORE INDICIES FOR THE NODES AT $\eta+\Delta\eta$ AND $\eta-\Delta\eta$.

COMPUTE $\frac{\partial \rho u}{\partial \xi}$

COMPUTE $\frac{\partial \kappa^2 \rho v}{\partial \eta}$



```
VN(1) = C3*(V(1,NP,1) - V(1,NM,1))
VN(3) = C3*(V(1,NP,1) - V(1,NM,1))
VN(4) = C3*(V(1,NP,1) - V(1,NM,1))
```

COMPUTE $\frac{\partial \rho}{\partial n}$, $\frac{\partial v}{\partial n}$, AND $\frac{\partial s}{\partial n}$.

```
VA(1) = AVN*(V(1,NP,1) + V(1,NM,1))
        +VIS*V(1,N,1)
VA(3) = AVN*(V(1,NP,3) + V(1,NM,3))
        +VIS*V(1,N,3)
VA(4) = AVN*(V(1,NP,4) + V(1,NM,4))
        +VIS*V(1,N,4)
```

COMPUTE AVERAGE VALUES OF ρ, V AND S TO INTRODUCE STABILIZING TERMS.

```
VT(1) = (CS+CS+CN/KAP)
 /KAP - WS(1)*ETA*VN(1)
```

COMPUTE $\frac{\partial \rho}{\partial \tau}$.

```
VT(3) = DUM*VN(3) +
C3*(P(L,NP) - P(L,NM))/
V(1,N,1)
```

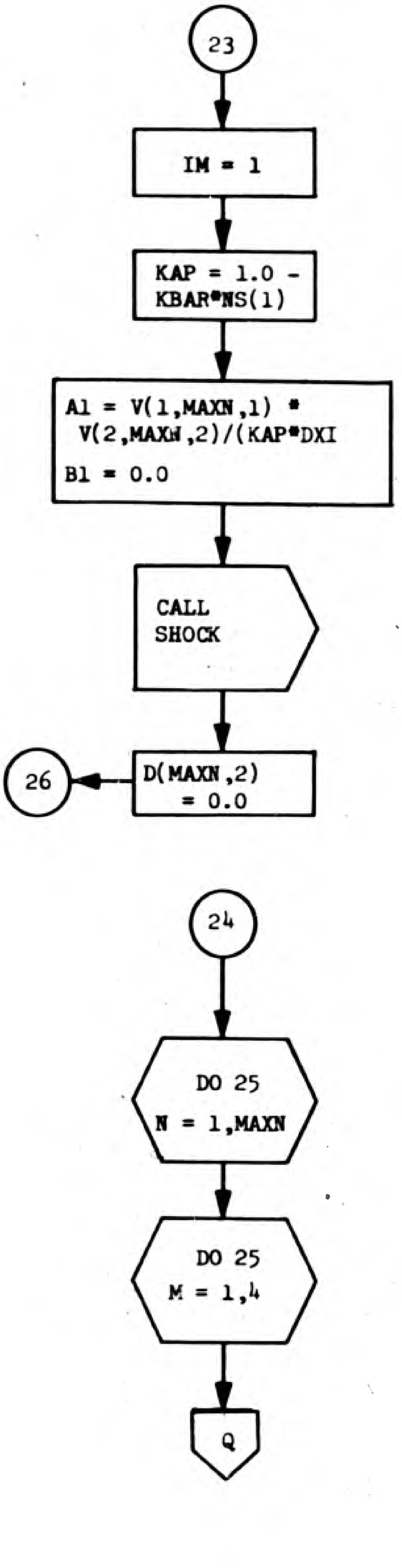
COMPUTE $\frac{\partial v}{\partial \tau}$.

```
VT(4) =
DUM*VN(4)
```

COMPUTE $\frac{\partial s}{\partial \tau}$.

```
D(N,1) = VA(1) - VT(1)*DT
D(N,2) = 0.0
D(N,3) = VA(3) - VT(3)*DT
D(N,4) = VA(4) - VT(4)*DT
```

SOLVE GOVERNING EQUATIONS.



SET IM TO INSURE PROPER FLOW VARIABLE STORAGE IN SUBROUTINE SHOCK.

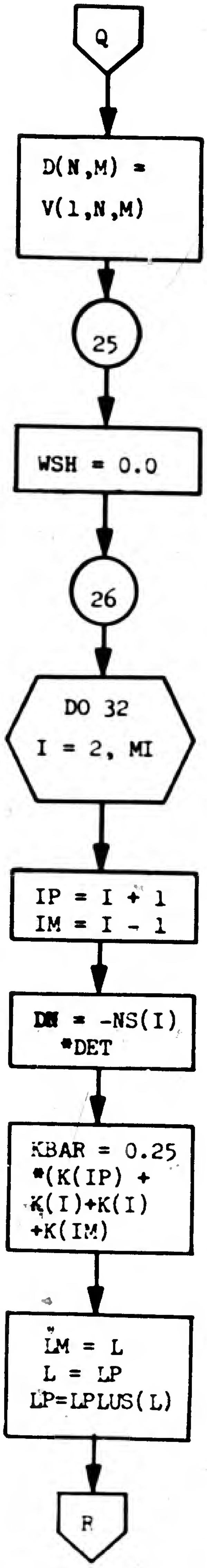
$\kappa = 1 - Kn_s$

COMPUTE CONSTANTS FOR SHOCK CALCULATIONS.

COMPUTE SHOCK PARAMETERS.

SET U = 0 AT THE SHOCK.

COMPUTATIONAL STARTING POINT WHEN IMIN = 2



STORE STARTING LINE PARAMETERS.

SET SHOCK VELOCITY COMPONENT NORMAL TO THE BODY TO ZERO FOR THE CONSTANT STARTING LINE.

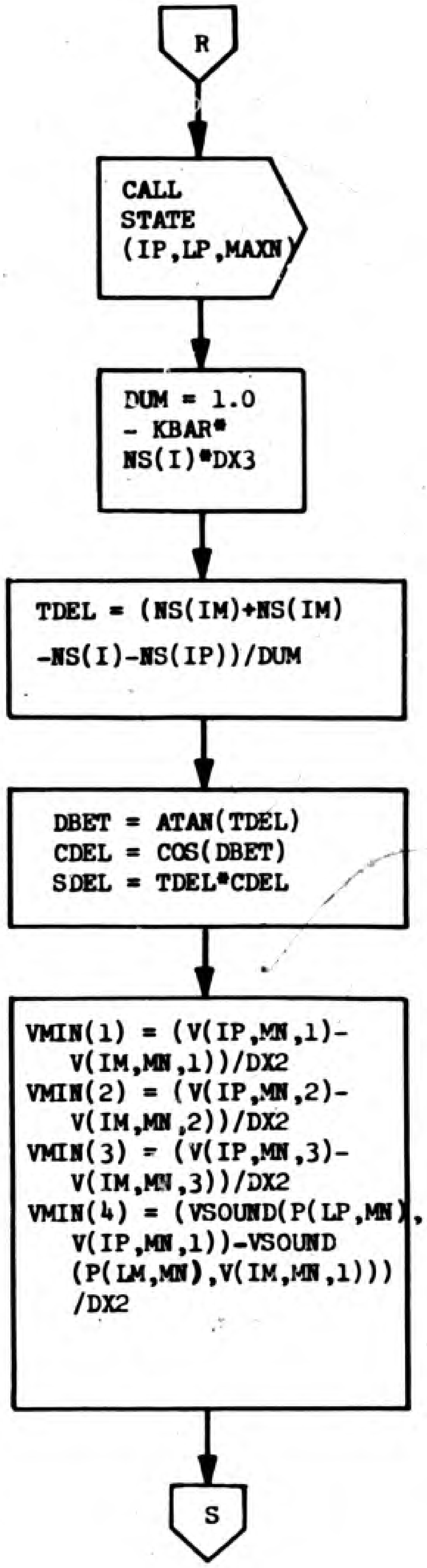
START OF LOOP TO TREAT POINTS NOT ON THE STAGNATION STREAMLINE.

COMPUTE INDICIES FOR NODES AT $\xi+\Delta\xi$ AND $\xi-\Delta\xi$.

COMPUTE Δn.

COMPUTE AVERAGE CURVATURE FOR FINITE DIFFERENCES.

SET INDICIES FOR THE PRESSURE ARRAY.



```
CALL
STATE
(IP,LP,MAXN)
```

COMPUTE PRESSURE FOR NODES AT $\xi+\Delta\xi$.

```
DUM = 1.0
- KBAR*
NS(I)*DX3
```

COMPUTE $3\kappa\Delta\xi$ AT THE SHOCK.

```
TDEL = (NS(IM)+NS(IM)
-NS(I)-NS(IP))/DUM
```

SHOCK SLOPE, EQUATION (60).

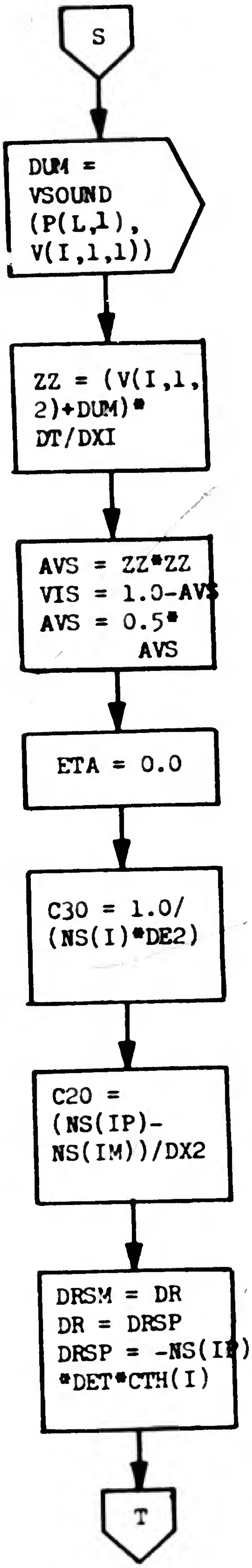
```
DBET = ATAN(TDEL)
CDEL = COS(DBET)
SDEL = TDEL*CDEL
```

COMPUTE CONSTANTS USED IN SHOCK TRANSFORMATIONS.

```
VMIN(1) = (V(IP,MN,1)-
   V(IM,MN,1))/DX2
VMIN(2) = (V(IP,MN,2)-
   V(IM,MN,2))/DX2
VMIN(3) = (V(IP,MN,3)-
   V(IM,MN,3))/DX2
VMIN(4) = (VSOUND(P(LP,MN),
   V(IP,MN,1))-VSOUND
   (P(LM,MN),V(IM,MN,1)))
   /DX2
```

COMPUTE $\frac{\partial \rho}{\partial \xi}$, $\frac{\partial u}{\partial \xi}$, $\frac{\partial v}{\partial \xi}$ AND $\frac{\partial a}{\partial \xi}$

AT THE FIRST NODE INSIDE THE SHOCK FOR USE IN SUBROUTINE SHOCK.



COMPUTE a AT BODY SURFACE.

EQUATION (33).

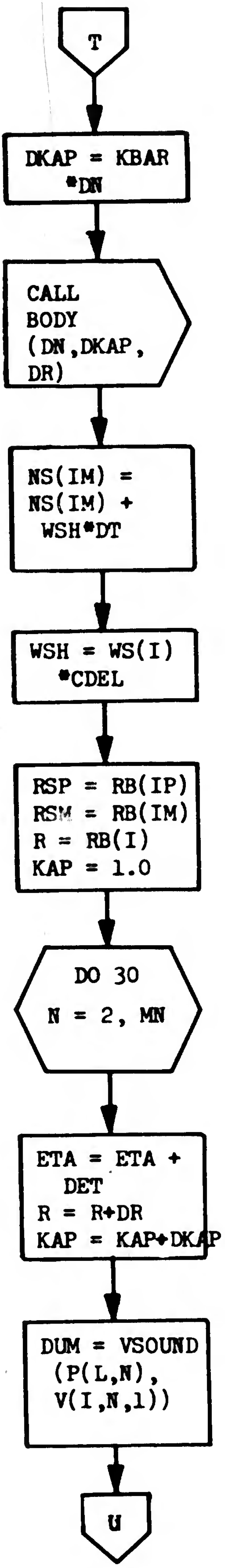
COMPUTE STABILIZING TERM COEFFICIENTS FOR BODY POINTS.

INITIALIZE n.

COMPUTE CONSTANT USED TO EVALUATE PARTIAL DERIVATIVES WITH RESPECT TO n.

COMPUTE CONSTANT USED TO EVALUATE PARTIAL DERIVATIVES WITH RESPECT TO S.

STORE Δr FOR ξ-Δξ, ξ AND ξ+Δξ.



COMPUTE $\Delta\kappa = K\Delta n$.

BODY POINT COMPUTATIONS.

COMPUTE NEW SHOCK POINT LOCATION FOR $\xi-\Delta\xi$.

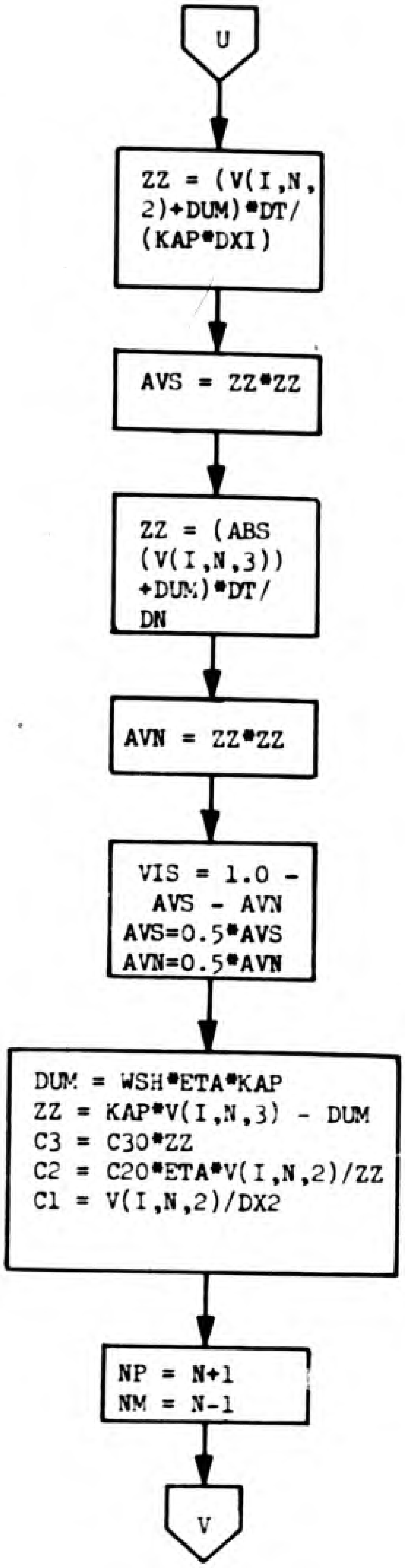
COMPUTE SHOCK VELOCITY COMPONENT NORMAL TO THE BODY.

INITIALIZE r AT $\xi+\Delta\xi$, $\xi-\Delta\xi$ AND ξ, AND κ.

START OF LOOP TO TREAT POINTS ALONG THE NORMAL COORDINATE AT ξ.

INCREMENT η, r AND κ.

COMPUTE SOUND SPEED.



```
ZZ = (V(I,N,
2)+DUM)*DT/
(KAP*DXI)

AVS = ZZ*ZZ

ZZ = (ABS
(V(I,N,3))
+DUM)*DT/
DN

AVN = ZZ*ZZ

VIS = 1.0 -
 AVS - AVN
AVS=0.5*AVS
AVN=0.5*AVN

DUM = WSH*ETA*KAP
ZZ = KAP*V(I,N,3) - DUM
C3 = C30*ZZ
C2 = C20*ETA*V(I,N,2)/ZZ
C1 = V(I,N,2)/DX2

NP = N+1
NM = N-1
```

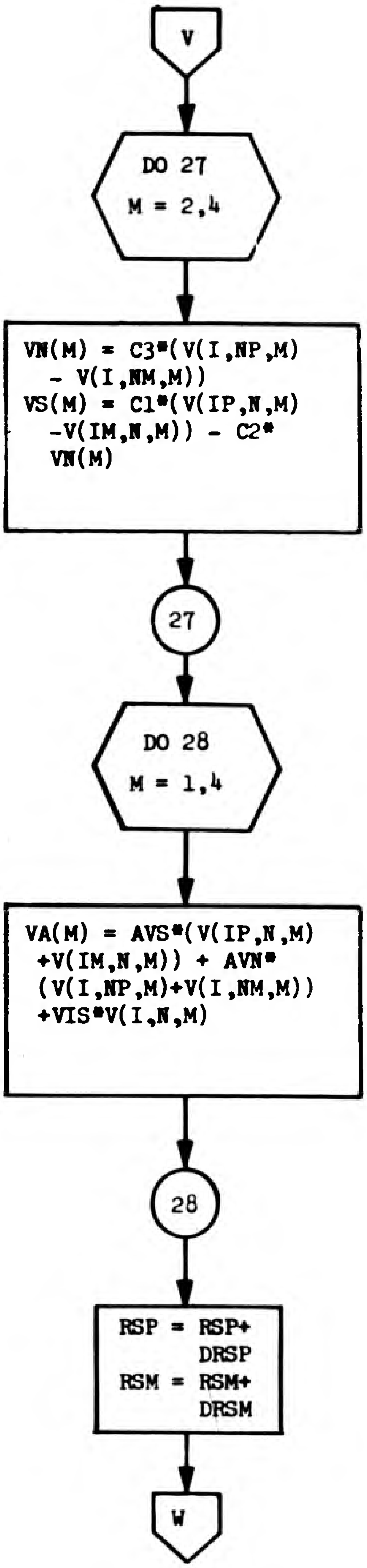
EQUATION (33) FOR ξ COORDINATE.

EQUATION (33) FOR η COORDINATE.

COEFFICIENTS FOR THE STABILIZING TERMS.

COMPUTE USEFUL PARAMETERS.

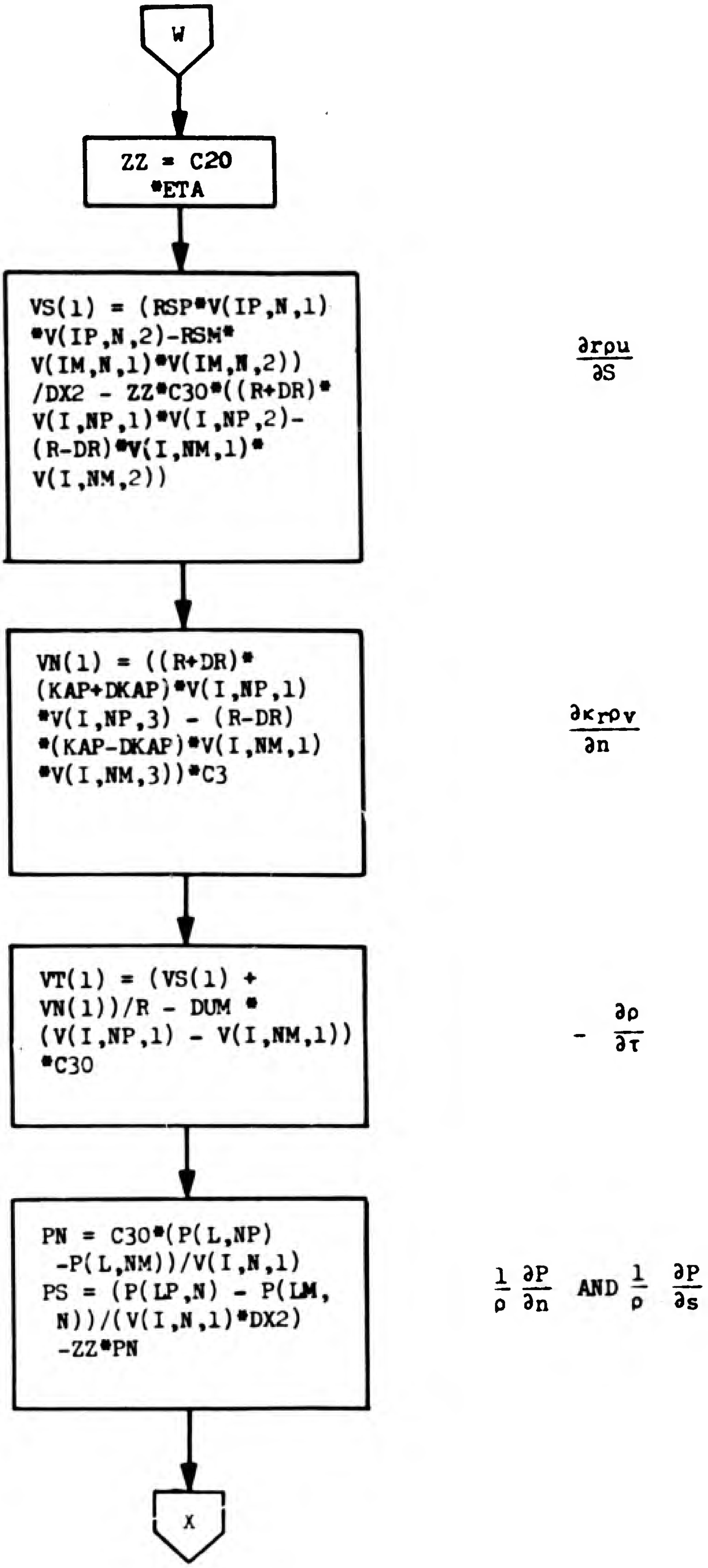
COMPUTE INDICIES OF NODES AT $n+\Delta\eta$ AND $n-\Delta\eta$.



COMPUTE PARTIAL DERIVATIVES FOR U, V AND S.

COMPUTE AVERAGES TO INTRODUCE STABILIZING TERMS FOR ρ, U, V AND S.

INCREMENT r AT ξ+Δξ AND ξ-Δξ.



```
ZZ = C20
*ETA
```

```
VS(1) = (RSP*V(IP,N,1)
*V(IP,N,2)-RSM*
V(IM,N,1)*V(IM,N,2))
/DX2 - ZZ*C30*((R+DR)*
V(I,NP,1)*V(I,NP,2)-
(R-DR)*V(I,NM,1)*
V(I,NM,2))
```

$\frac{\partial r\rho u}{\partial S}$

```
VN(1) = ((R+DR)*
(KAP+DKAP)*V(I,NP,1)
*V(I,NP,3) - (R-DR)
*(KAP-DKAP)*V(I,NM,1)
*V(I,NM,3))*C3
```

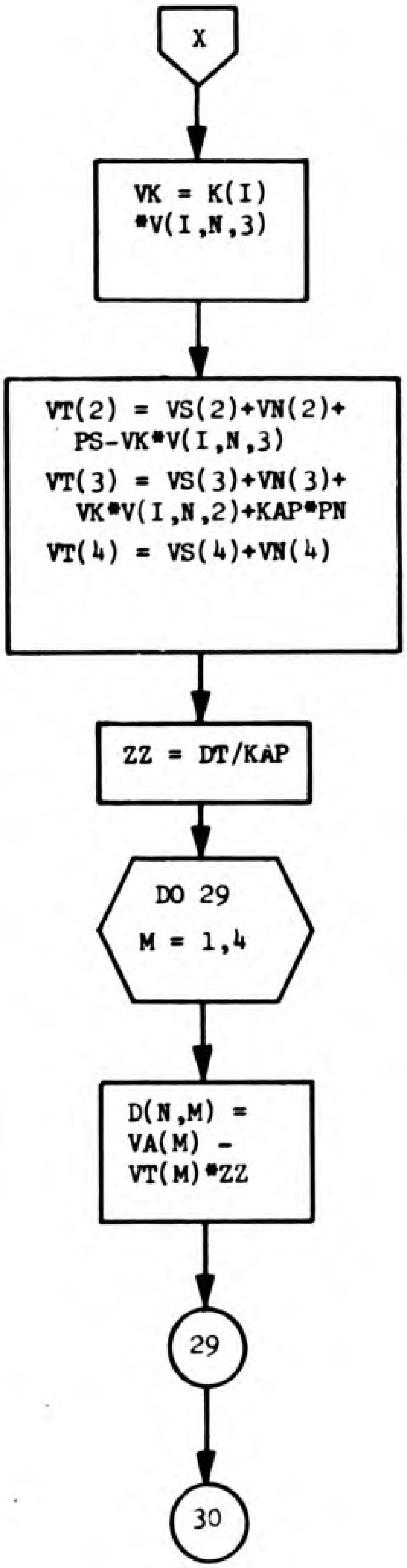
$\frac{\partial \kappa r\rho v}{\partial n}$

```
VT(1) = (VS(1) +
VN(1))/R - DUM *
(V(I,NP,1) - V(I,NM,1))
*C30
```

$- \frac{\partial \rho}{\partial \tau}$

```
PN = C30*(P(L,NP)
 -P(L,NM))/V(I,N,1)
PS = (P(LP,N) - P(LM,
 N))/(V(I,N,1)*DX2)
 -ZZ*PN
```

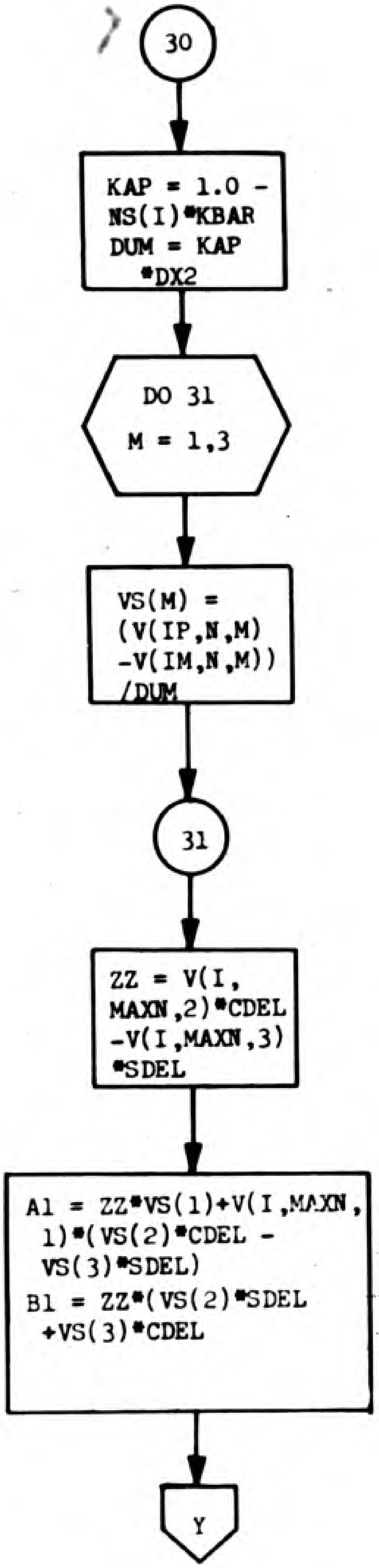
$\frac{1}{\rho}\frac{\partial P}{\partial n}$ AND $\frac{1}{\rho}\frac{\partial P}{\partial s}$



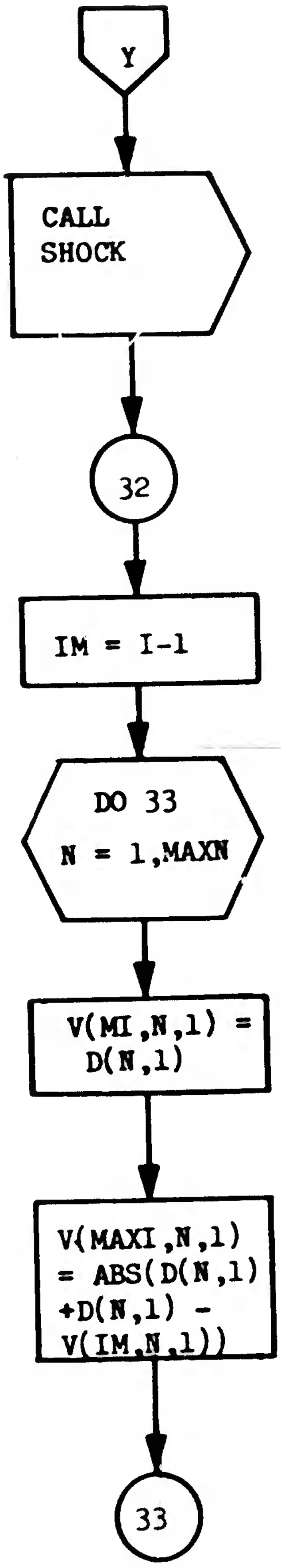
$- \frac{\partial u}{\partial t}, - \frac{\partial v}{\partial t}$ AND $- \frac{\partial S}{\partial t}$.

COMPUTE FLOW PARAMETERS
AT $\tau+\Delta\tau$.

END OF LOOP ON N.



COMPUTE CONSTANTS FOR SHOCK CALCULATIONS.

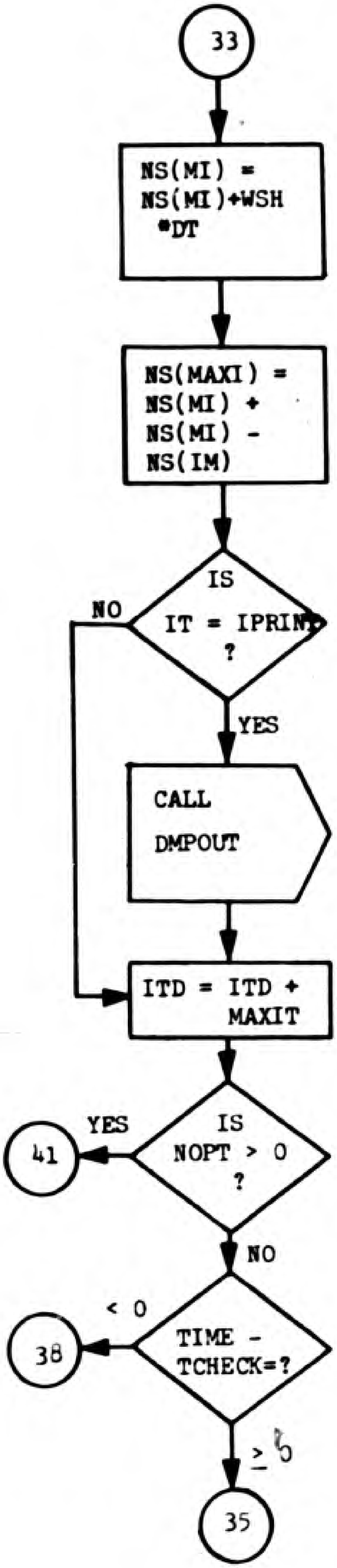


COMPUTE SHOCK PARAMETERS.

END OF LOOP ON I.

TRANSFER VALUES OF ρ FOR THE DOWNSTREAM SEGMENT BOUNDARY FROM TEMPORARY STORAGE.

LINEAR EXTENSION FOR ρ. ABSOLUTE VALUES USED TO INSURE INITIAL GUESS WILL NOT INTRODUCE NEGATIVE VALUES.



COMPUTE NEW SHOCK POINT LOCATION AT DOWNSTREAM SEGMENT BOUNDARY.

LINEAR EXTENSION OF SHOCK.

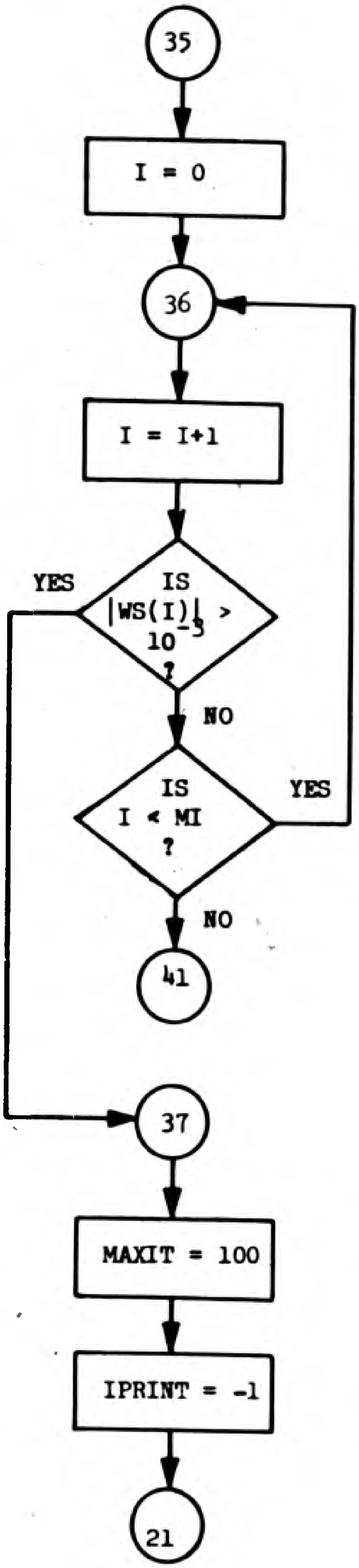
SHOULD PRINTOUT BE PROVIDED ?

PRINT FLOW FIELD VARIABLES.

COMPUTE TOTAL NUMBER OF ITERATIONS PERFORMED ON THIS SEGMENT.

WAS THE NUMBER OF ITERATIONS TO BE PERFORMED SPECIFIED ?

HAS THE ESTIMATED TIME FOR STEADY STATE BEEN EXCEEDED ?



INITIALIZE I.

INCREMENT I.

IS THE MAGNITUDE OF THE SHOCK VELOCITY GREATER THAN 10^{-3} ?

ARE THERE MORE POINTS IN THIS SEGMENT ?

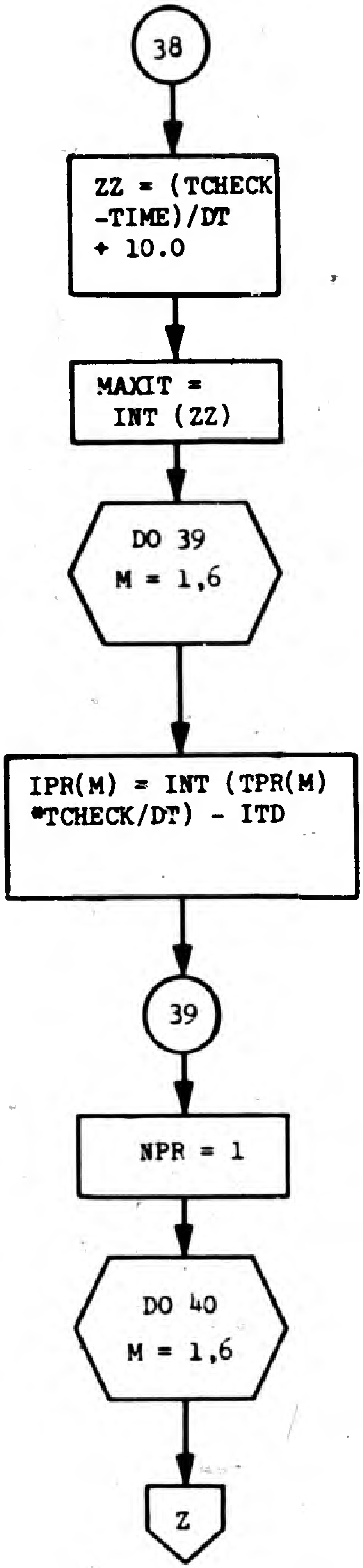
STEADY STATE ACHIEVED.

STEADY STATE NOT ACHIEVED.

SET MAXIT FOR 100 MORE ITERATIONS.

SUPPRESS EXTRA PRINTOUT.

START ITERATION LOOP.

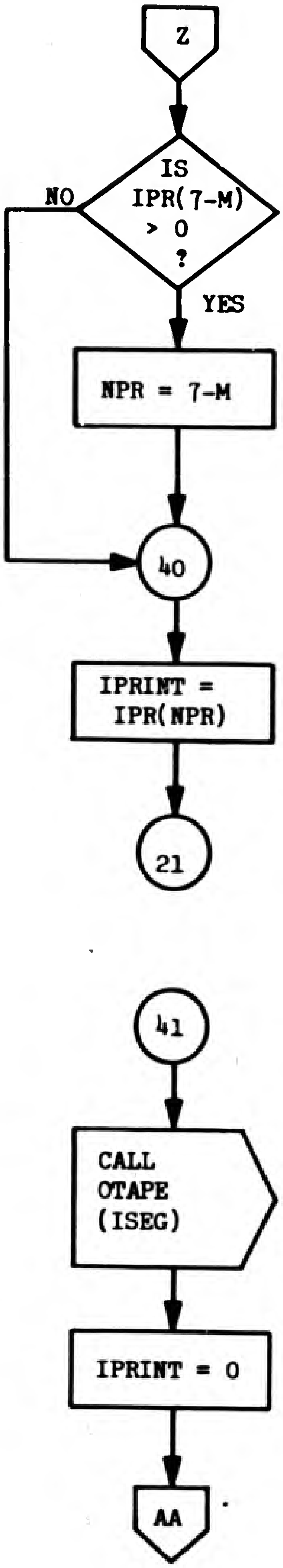


ESTIMATE THE NUMBER OF ADDITIONAL ITERATIONS REQUIRED FOR STEADY STATE.

CONVERT TO INTEGER FORM.

EVALUATE NEW PRINTOUT ARRAY.

INITIALIZE PRINTOUT CONTROL INDEX.



IS IPR(7-M)> 0 ?

RESET NPR.

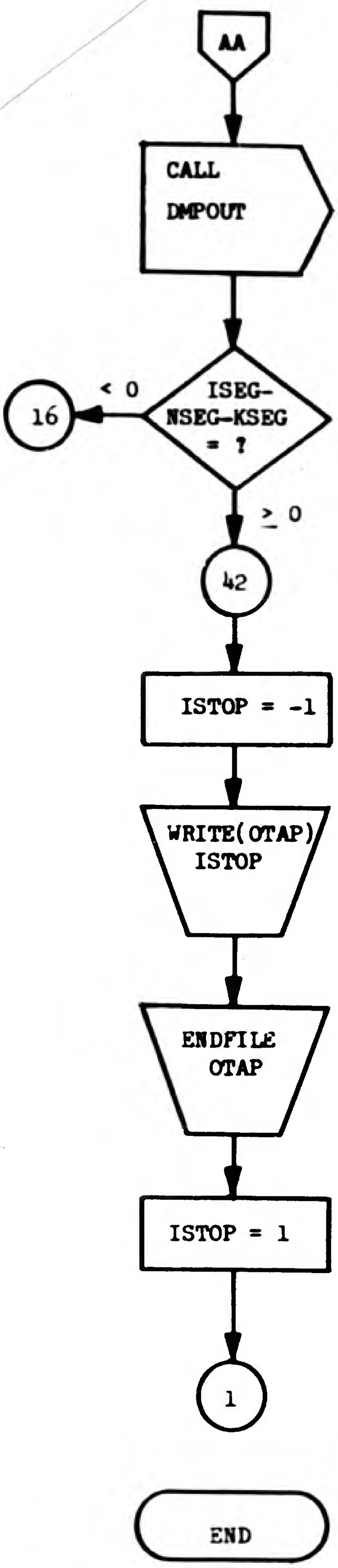
SET PRINT CONTROL VARIABLE.

START ITERATION LOOP.

SEGMENT COMPLETED.

WRITE FLOW FIELD DATA
ON MAGNETIC TAPE.

SET IPRINT FOR FINAL
PRINTOUT.



FINAL PRINTOUT FOR THIS SEGMENT.

HAVE ALL SEGMENTS BEEN TREATED ?

WRITE INDICATOR ON TAPE TO SIGNIFY END OF CASE.

ENDFILE TAPE.

SET ERROR INDICATOR.

READ NEXT CASE.

END.

D. LISTING

```
      PROGRAM BLUNT(INPUT,OUTPUT,TAPE1,TAPE2,TAPE5=INPUT)

       THIS PROGRAM SOLVES THE TIME-DEPENDENT INVISCID FLOW
      EQUATIONS FOR AXISYMMETRIC BLUNT BODIES OF ARBITRARY SHAPE
      FLYING AT SUPERSONIC SPEEDS. (AFWL-TR-70-15)

      COMMON /1/ V(15,40,4),D(40,4),F(3,40)
      COMMON /2/ RB(15),ZB(15),TH(15),CTH(15),K(15),NS(15),HS(15)
      COMMON /3/ MAXI,MI,IMIN,MAXN,MN,NMIN
      COMMON /4/ IPRINT,NPR,KGEOM,TPR(6),KSEG
      COMMON /5/ DXI,DET,DX2,DE2,DT
      COMMON /6/ Q1,VS1,HT1,A1,M1
      COMMON /7/ I,IP,IM,L,LP,LM
      COMMON /8/ VMIN(4),KBAR,Z20
      COMMON /9/ ISEG,XI0,DELS,KZERO,RNOSE
      COMMON /10/ VS(4),VN(4),VA(4),VT(4)
      COMMON /11/ MAXIT,ISTOP,ITO,TIME,SI(15),NST(15)
      COMMON /12/ DBET,CDEL,SDEL,TDEL
      COMMON /13/ AVS,AVN,VIS
      COMMON /INPUT/ GAM1,AMACH1,THETC,ZRN,TCRIT(2),P1,RHO1,RN,AMU0,T1,E
     *RR(2),IPR(6),INTAP,OTAP,NOPT,NCASE,NFILE,NSEG,IGEOM,NGEOM,NTAP
      COMMON /CONST/ PI,PI2,GAMA,GAMP,RGAS
      COMMON /MESH/ IMAX,NMAX
      COMMON /PRES/ PT2,P01
      DIMENSION LPLUS(3), JPR(6)
      INTEGER OTAP
      REAL NS,K,KAP,KBAR,KZERO,M1,NST
      DATA PI,PI2/3.14159265359,1.570796326795/
      DATA (LPLUS(M),M=1,3)/2,3,1/
      DATA (JPR(M),M=1,6)/-5,-5,-5,-5,-5,-5/
      DATA                AMU0/     0.04/,           TCRIT(1)/      2.0/,
     *             TCRIT(2)/     10.0/,               GAM1/      1.4/,
     *                 NSEG/        0/,                 T1/      1.0/,
     *                   P1/      1.0/,               RHO1/      1.0/,
     *                   RN/      1.0/,                ZRN/      0.0/,
     *               ERR(1)/  1.0E-05/,             ERR(2)/  1.0E-08/,
     *                INTAP/        2/,               OTAP/        1/,
     *                 NOPT/       -1/,              NFILE/        1/,
     *                 NTAP/        0/,              IGEOM/        0/,
     *                NGEOM/        0/,               IMAX/       15/,
     *                 NMAX/       40/,              NCASE/        1/
      ISTOP=5
      PRINT 45
      IMIN=1
      DELS=0.0
      KZERO=1.0
      RNOSE=1.0
      XI0=0.0
      KSEG=0
      AMACH1=M1
      DO 2 M=1,6
      IPR(M)=JPR(M)
```

NOT REPRODUCIBLE

```
      THETC=THETC*180.0/PI                                              PLUM  53
      CALL DATA (ISTOP)                                                 PLUM  54
      M1=AMACH1                                                         PLUM  55
      GAMA=(GAM1+1.0)/2.0                                               PLUM  56
      GAMB=(GAM1-1.0)/2.0                                               PLUM  57
      VS1=SQRT(GAM1)                                                    PLUM  58
      RGAS=P1/(RHO1*T1)                                                 PLUM  59
      IF (AMACH1.LT.2.0) AMACH1=SQRT(RHO1*AMACH1*AMACH1/(GAM1*P1))      PLUM  60
      U1=AMACH1*VS1                                                     PLUM  61
      HT1=GAM1/(GAM1-1.0)+U1*U1/2.0                                     PLUM  62
      MAXIT=NROT                                                        PLUM  63
      REWIND NTAP                                                       PLUM  64
      DO 3 M=1,5                                                        PLUM  65
    3 JPR(M)=IPR(M)                                                     PLUM  66
      IF (NTAP-1) 4,10,12                                               PLUM  67
    4 MAXI=IGEOM                                                        PLUM  68
      MAXN=NGEOM                                                        PLUM  69
      IF (MAXN.LT.1) MAXN=5                                             PLUM  70
      MN=MAXN-1                                                         PLUM  71
      MI=MAXI-1                                                         PLUM  72
      MMIN=MN-1                                                         PLUM  73
      ZZ=FLOAT(MN)                                                      PLUM  74
      DET=1.0/ZZ                                                        PLUM  75
      DE2=DET+DET                                                       PLUM  76
      IF (IPN) 8,8,5                                                    PLUM  77
      IF (THETC.GT.90.0.OR.AMACH1.LT.2.0) GO TO 6                       PLUM  78
      KGEOM=2                                                           PLUM  79
      DELS=0.87266                                                      PLUM  80
      GO TO 7                                                           PLUM  81
    6 DELS=(90.0-THETC)*PI/180.0                                        PLUM  82
      KGEOM=1                                                           PLUM  83
    7 THETC=THETC*PI/180.0                                              PLUM  84
      IF (NSEG.LT.1) NSEG=50000                                         PLUM  85
      GO TO 9                                                           PLUM  86
    8 KGEOM=3                                                           PLUM  87
    9 WSH=AMACH1*AMACH1                                                 PLUM  88
      ZZ=1.0/(GAM1-1.0)                                                 PLUM  89
      DUM=ZZ*GAM1                                                       PLUM  90
      PT2=((GAMA*WSH)**DUM)*((GAMA/(GAM1*WSH-GAMB))**ZZ)                PLUM  91
      DDD=NSH/16.0-0.5                                                  PLUM  92
      IF (DDD.LT.1.0) DDD=1.0                                           PLUM  93
      IF (NTAP.GT.0) GO TO 14                                           PLUM  94
      TIME=0.0                                                          PLUM  95
      IT=0                                                              PLUM  96
      ISEG=1                                                            PLUM  97
      CALL GEOM (I)                                                     PLUM  98
      IF (ISTOP.LT.1) GO TO 1                                           PLUM  99
      CALL INITIAL (N)                                                  PLUM 100
      ZZ=0.157/RNOSE                                                    PLUM 101
      IF (ZZ.GT.1.0) ZZ=1.0                                             PLUM 102
      IF (NTAP.EQ.0) CALL AMESH (NS(MI),1,MAXI,ZZ,N)                    PLUM 103
      CALL STAPE (0)                                                    PLUM 104
      GO TO 14                                                          PLUM 105
   10 IF (NTAP.EQ.MTAP) GO TO 11                                        PLUM 106
      PRINT 43                                                          PLUM 107
      ISTOP=-1
      GO TO 1
 11   CALL RESTART
      IF (ISTOP) 1,1,9
 12   IF (INITAP.NE.CTAP) GO TO 13
      PRINT 44
      ISTOP=-1
      GO TO 1
 13   CALL LOAD
      IF (ISTOP) 1,1,9
 14   DT1=0.8/(C1+VS1)
      DT1=AMU0*C1/((C1+VS1)**2)
      DT=DT1*DXT
      IF (DT.LT.DT1) DT1=DT
      DO 15 M=1,6
 15   TPR(M)=FLOAT(IPR(M))/10.0
      ISEG=NFILE
      IF (NTAP-1) 17,16,17
 16   ISEG=ISEG+1
      CALL RESET
      DT1=AMU0*C1/((C1+VS1)**2)
      DT=DT1*DXT
      IF (DT.LT.DT1) DT1=DT
      IF (ISTOP) 1,1,17
 17   IF (NOPT) 18,41,20
 18   CALL STATE (1,1,1)
      DUM=VSOUND(P(1,1),V(1,1,1))
      TCHECK=(TCRIT(1)+TCRIT(2)*DELS)/(V(1,1,2)+DUM)
      MAXIT=INT(0.2*TCHECK/DT1)
      DO 19 M=1,6
 19   IPR(M)=INT(TPR(M)*TCHECK/DT1)
 20   NPR=1
      IPRINT=IPR(1)
      IF (IPRINT.EQ.0) CALL DMPOUT
      DX2=DX2+DXI
 21   DO 24 IT=1,MAXIT
      DN=-NS(1)*DET
      DP=DN*CTH(1)
      DPSP=-NS(2)*DET*CTH(2)
      L=1
      LP=2
      CALL STATE (1,L,MAXN)
      CALL STATE (2,LP,MAXN)
      DT=DT1*DN
      IF (DT1.LT.DT) DT=DT1
      TIME=TIME+DT
      IF (IMIN.EQ.2) GO TO 24
      WSH=NS(1)
      DO 22 M=1,4
 22   P(MAXN,M)=V(1,MAXN,M)
      DZ=1.0/(NS(1)*DE2)
      I=1
      IP=2
      IM=2
      LM=2
      KBAR=0.5*(K(I-1)+K(I))
      DPDT=0.0
      DPDL=1.0
      TPDL=0.0
      DPCL=0.0
      VMIN(1)=0.0
      VMIN(2)=V(I2,MN,2)/DXI
      VMIN(3)=0.0
      VMIN(4)=1.0
      ETA=0.0
      DKAP=-KBAR*WS(1)*DET
      KAP=1.0+DKAP
      VN(1)=-KAP*KAP*V(1,2,1)*V(1,2,3)/DN
      VS(1)=V(2,1,1)*V(2,1,2)/DXI
      VT(1)=VS(1)+VS(1)+VN(1)
      P(1,1)=V(1,1,1)-VT(1)*DT
      P(1,2)=0.0
      ZZ=(V(1,1,4)-V(1,2,4))/(V(1,1,4)+V(1,2,4))
      VIS=ZZ*ZZ
      AVM=1.0-VIS
      P(1,4)=AVM*V(1,2,4)+VIS*V(1,1,4)
      KAP=1.0
      DO 23 N=2,MN
      ETA=ETA+DET
      KAP=KAP+DKAP
      DUM=VSOUND(P(L,N),V(1,N,1))
      ZZ=(V(1,N,3)+DUM)*DT/DN
      AVM=ZZ*ZZ
      VIS=1.0-AVM
      AVM=0.5*AVM
      DUM=V(I,N,3)-WS(I)*ETA
      NP=N+1
      NM=N-1
      CS=V(2,N,1)*V(2,N,2)/DXI
      CN=(V(1,NP,1)*V(1,NP,3)*(KAP+DKAP)**2-V(1,NM,1)*V(1,NM,3)*(KAP-DKAP
     *)**2)*C2
      VM(1)=C3*(V(I,NP,1)-V(I,NM,1))
      VM(3)=C3*(V(I,NP,3)-V(I,NM,3))
      VM(4)=C3*(V(I,NP,4)-V(I,NM,4))
      VA(1)=AVM*(V(I,NP,1)+V(I,NM,1))+VIS*V(I,N,1)
      VA(3)=AVM*(V(I,NP,3)+V(I,NM,3))+VIS*V(I,N,3)
      VA(4)=AVM*(V(I,NP,4)+V(I,NM,4))+VIS*V(I,N,4)
      VT(1)=(CS+CS+CN/KAP)/KAP-WS(1)*ETA*VM(1)
      VT(3)=DUM*VM(3)+C3*(P(L,NP)-P(L,NM))/V(I,N,1)
      VT(4)=DUM*VM(4)
      P(N,1)=VA(1)-VT(1)*DT
      P(N,2)=0.0
      P(N,3)=VA(3)-VT(3)*DT
   23 P(N,4)=VA(4)-VT(4)*DT
      IM=1
      KAP=1.0-KBAR*WS(1)
      A1=V(1,MAXN,1)*V(2,MAXN,2)/(KAP*DXI)
      B1=0.0
      CALL SHOCK (2,3)
      P(MAXN,2)=0.0
      GO TO 26                                                          BLUN 218
24    DO 25 N=1,MAXN                                                    BLUN 219
      DO 25 M=1,4                                                       BLUN 220
25    D(M,N)=V(1,N,M)                                                   BLUN 221
      WTH=0.0                                                           BLUN 222
26    DO 32 I=2,MI                                                      BLUN 223
      IP=I+1                                                            BLUN 224
      IM=I-1                                                            BLUN 225
      DN=-NS(I)*DET                                                     BLUN 226
      KBAR=0.25*(K(IP)+K(I)+K(I)+K(IM))                                 BLUN 227
      LM=L                                                              BLUN 228
      L=LP                                                              BLUN 229
      LP=LPLUS(L)                                                       BLUN 230
      CALL STATE (IP,LP,MAXN)                                           BLUN 231
      DUM=(1.0-K313*NS(I))*DX3                                          BLUN 232
      TDEL=(NS(IM)+NS(IM)-NS(I)-NS(IP))/DUM                             BLUN 233
      DBET=ATAN(TDEL)                                                   BLUN 234
      CDEL=COS(DBET)                                                    BLUN 235
      SDEL=TDEL*CDEL                                                    BLUN 236
      VMTN(1)=(V(IP,MN,1)-V(IM,MN,1))/DX2                               BLUN 237
      VMTN(2)=(V(IP,MN,2)-V(IM,MN,2))/DX2                               BLUN 238
      VMTN(3)=(V(IP,MN,3)-V(IM,MN,3))/DX2                               BLUN 239
      VMTN(4)=(VSOUND(P(LP,MN),V(IP,MN,1))-VSOUND(P(LM,MN),V(IM,MN,1)))/BLUN 240
     *DX2                                                               BLUN 241
      DUM=VSOUND(P(L,1),V(I,1,1))                                       BLUN 242
      ZZ=(V(I,1,2)+DUM)*DT/DXI                                          BLUN 243
      AVS=ZZ*ZZ                                                         BLUN 244
      VIS=1.0-AVS                                                       BLUN 245
      AVS=1.5*AVS                                                       BLUN 246
      ETA=0.0                                                           BLUN 247
      CTC=1.0/(NS(I)*DE2)                                               BLUN 248
      D2D=(NS(IP)-NS(IM))/DX2                                           BLUN 249
      D2SM=D2                                                           BLUN 250
      DP=D2SP                                                           BLUN 251
      D2SP=-NS(IP)*DET*CTH(IP)                                          BLUN 252
      DKAP=KBAR*DN                                                      BLUN 253
      CALL BODY (DN,DKAP,DP)                                            BLUN 254
      NS(I1)=NS(IM)+WSH*DT                                              BLUN 255
      WSH=NS(I)*CDEL                                                    BLUN 256
      PSP=PB(IP)                                                        BLUN 257
      PSM=PB(IM)                                                        BLUN 258
      P=PB(I)                                                           BLUN 259
      KAP=1.0                                                           BLUN 260
      DO 31 N=2,MN                                                      BLUN 261
      ETA=ETA+DET                                                       BLUN 262
      P=P+DP                                                            BLUN 263
      KAP=KAP+DKAP                                                      BLUN 264
      DUM=VSOUND(P(L,N),V(I,N,1))                                       BLUN 265
      ZZ=(V(I,N,2)+DUM)*DT/(KAP*DXI)                                    BLUN 266
      AVS=ZZ*ZZ                                                         BLUN 267
      ZZ=(ABS(V(I,N,3))+DUM)*DT/DN                                      BLUN 268
      AVN=ZZ*ZZ                                                         BLUN 269
      VIS=1.0-AVS-AVN                                                   BLUN 270
      AVS=1.5*AVS                                                       BLUN 271
      AVN=1.5*AVN                                                       BLUN 272
      DUM=DCH*ETA*KAP                                                   PLUN 273
      ZZ=KAP*V(I,N,3)-DUM                                               PLUN 274
      C3=C31*ZZ                                                         PLUN 275
      C1=C31*ETA*V(I,N,2)/ZZ                                            PLUN 276
      C2=V(I,N,2)/DX2                                                   PLUN 277
      NP=N+1                                                            PLUN 278
      NM=N-1                                                            PLUN 279
      DO 27 M=2,4                                                       PLUN 280
      VN(M)=C3*(V(I,NP,M)-V(I,NM,M))                                    PLUN 281
   27 VS(M)=C1*(V(IP,N,M)-V(IM,N,M))-C2*VN(M)                           PLUN 282
      DO 28 M=1,4                                                       PLUN 283
   28 V1(M)=AVS*(V(IP,N,M)+V(IM,N,M))+AVN*(V(I,NP,M)+V(I,NM,M))+VIS*V(I,PLUN 284
     *N,M)                                                              PLUN 285
      PSP=PSP+DPSP                                                      PLUN 286
      PSM=PSM+DPSM                                                      PLUN 287
      ZZ=C31*ETA                                                        PLUN 288
      VS(1)=(PSP*V(IP,N,1)*V(IP,N,2)-PSM*V(IM,N,1)*V(IM,N,2))/DX2-ZZ*C31PLUN 289
     **((P+DP)*V(I,NP,1)*V(I,NP,2)-(P-DP)*V(I,NM,1)*V(I,NM,2))          PLUN 290
      VN(1)=((P+DP)*(KAP+DKAP)*V(I,NP,1)*V(I,NP,3)-(P-DP)*(KAP-DKAP)*V(IPLUN 291
     *,NM,1)*V(I,NM,3))*C33                                             PLUN 292
      VT(1)=(VS(1)+VN(1))/R-DUM*(V(I,NP,1)-V(I,NM,1))*C30               PLUN 293
      PN=C30*(P(L,NP)-P(L,NM))/V(I,N,1)                                 PLUN 294
      PS=(P(LP,N)-P(LM,N))/(V(I,N,1)*DX2)-ZZ*PN                         PLUN 295
      VK=K(I)*V(I,N,2)                                                  PLUN 296
      VT(2)=VS(2)+VN(2)+PS-VK*V(I,N,3)                                  PLUN 297
      VT(3)=VS(3)+VN(3)+VK*V(I,N,2)+KAP*PN                              PLUN 298
      VT(4)=VS(4)+VN(4)                                                 PLUN 299
      ZZ=DT/KAP                                                         PLUN 300
      DO 29 M=1,4                                                       PLUN 301
      V(IM,N,M)=D(N,M)                                                  PLUN 302
   29 D(N,M)=V1(M)-VT(M)*ZZ                                             PLUN 303
   30 CONTINUE                                                          PLUN 304
      KAP=1.0-NS(I)*KPAP                                                PLUN 305
      DUM=KAP*DX2                                                       PLUN 306
      DO 31 M=1,3                                                       PLUN 307
   31 VS(M)=(V(IP,MAXN,M)-V(IM,MAXN,M))/DUM                             PLUN 308
      ZZ=V(I,MAXN,2)*COEL-V(I,MAXN,3)*SOEL                              PLUN 309
      S1=ZZ*VS(1)+V(I,MAXN,1)*(VS(2)*COEL-VS(3)*SOEL)                   PLUN 310
      R1=ZZ*(VS(2)*SOEL+VS(3)*COEL)                                     PLUN 311
   32 CALL SHOCK (1,0).                                                 PLUN 312
      IM=MI-1                                                           PLUN 313
      DO 33 N=1,MAXN                                                    PLUN 314
      V(MI,N,1)=D(N,1)                                                  PLUN 315
      V(MAXI,N,1)=ABS(D(N,1)+D(N,1)-V(IM,N,1))                          PLUN 316
      DO 33 M=2,4                                                       PLUN 317
      V(MI,N,M)=D(N,M)                                                  PLUN 318
   33 V(MAXI,N,M)=D(N,M)+D(N,M)-V(IM,N,M)                               PLUN 319
      NS(MI)=NS(MT)+WSH*DT                                              PLUN 320
      NS(MAXI)=NS(MI)+NS(MI)-NS(IM)                                     PLUN 321
   34 IF (IT.EQ.IPRINT) CALL DMPOUT                                     PLUN 322
      ITP=ITP+MAXIT                                                     PLUN 323
      IF (MOPT.GT.0) GO TO 41                                           PLUN 324
      IF (TIME-TCHECK) 33,35,35                                         PLUN 325
   35 I=2                                                               PLUN 326
   36 I=I+1                                                             PLUN 327
```

```
      IF ([illegible](I)*[illegible](I).GT.1.E-[illegible]) GO TO 37
      IF ([illegible]) GO TO 35
      [illegible]
    7 [illegible]
      I[illegible]INT=1
      [illegible]
      [illegible] ([illegible]-ITM[illegible])/DT+1.
      [illegible]T(77)
      [illegible]
   38 ITP(K)=INT(TP(K)*T[illegible]/DT)-IT[illegible]
      [illegible]
      DO 40 J=1,6
   40 IF (ITP(7-J).GT.0) IPP=7-J
      [illegible]INT=ITP(IPP)
      GO TO 21
   41 CALL [illegible]TAPE (I?=6)
      ICTINT=0
      CALL OUTPUT
      IF (ISEG-NSEG-KSEG) 16,42,42
   42 ISTOP=-1
      WRITE (OTAPE) ISTOP
      END FILE OTAPE
      ISTOP=1
      GO TO 1
C
   43 FORMAT (////40X,13HERROR IN INPUT DATA/40X,34HRESTART OPTION REQUI
     *RES INTAPE=OTAPE/40X,15HCASE TERMINATED)
   44 FORMAT (//////40X,13HERROR IN INPUT DATA/40X,29HNTAP=2 REQUIRES IN
     *TAP.NE.OTAP/40X,15HCASE TERMINATED)
   45 FORMAT(1H1//////25X,59HAFWL TIME-DEPENDENT FLOW FIELD PROGRAM FOR
     *AXISYMMETRIC BLUNT BODIES/25X,27HAFWL-TR-70-16, R.H. BUNGIER)
      END
```

4. SUBROUTINE BODY

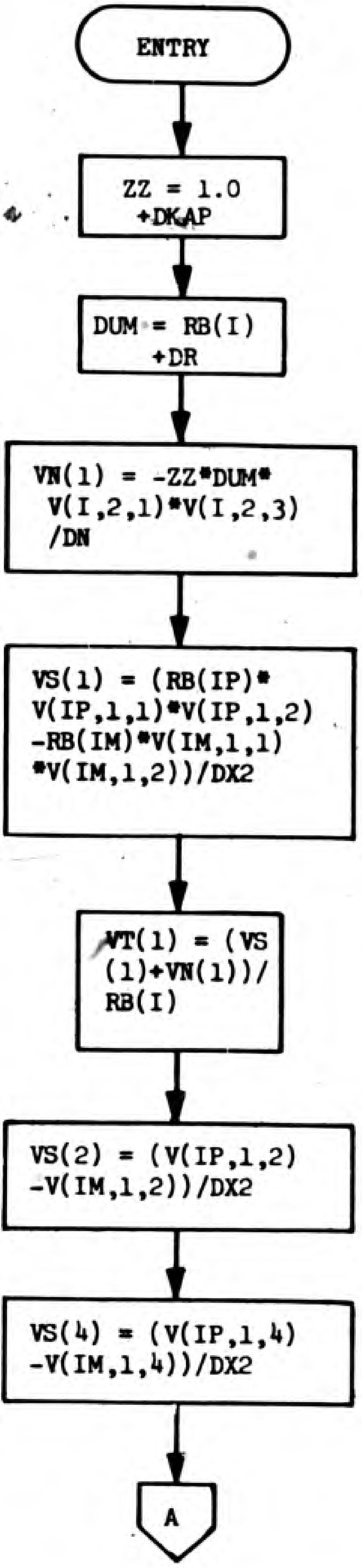
A. PURPOSE

This subroutine solves the governing equations at nodes on the body surface, except for the stagnation point. The special body point logic developed in section IV is employed.

B. VARIABLE LIST

All variables used in this subroutine have been defined in the description of the main program.

C. FLOW CHART


ENTRY
ZZ = 1.0 +DKAP
DUM = RB(I) +DR
VN(1) = -ZZ*DUM* V(I,2,1)*V(I,2,3) /DN
VS(1) = (RB(IP)* V(IP,1,1)*V(IP,1,2) -RB(IM)*V(IM,1,1) *V(IM,1,2))/DX2
VT(1) = (VS (1)+VN(1))/ RB(I)
VS(2) = (V(IP,1,2) -V(IM,1,2))/DX2
VS(4) = (V(IP,1,4) -V(IM,1,4))/DX2
A

ENTER SUBROUTINE BODY.

COMPUTE κ AT $\eta = \Delta\eta$.

COMPUTE r AT $\eta = \Delta\eta$.

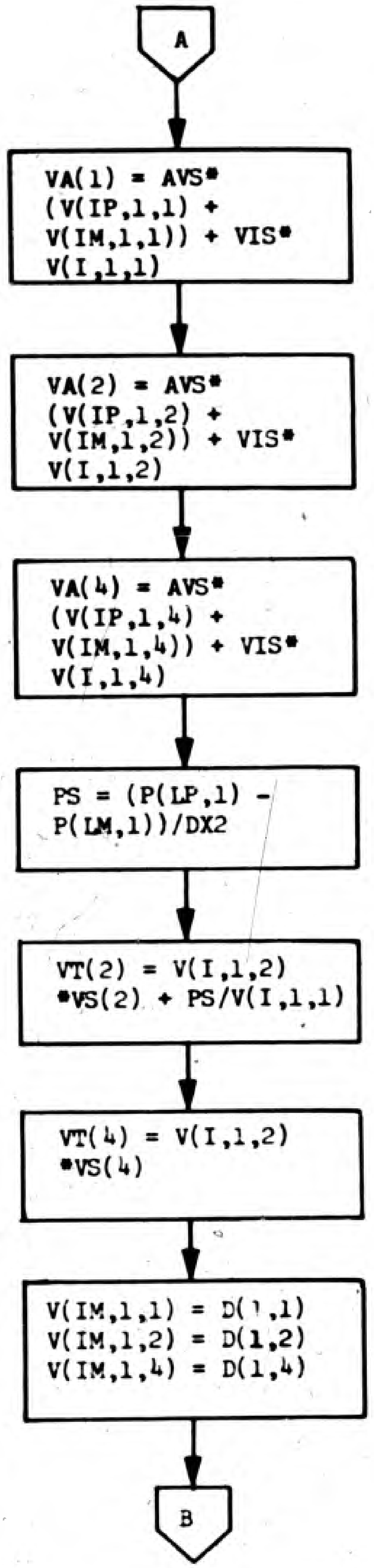
COMPUTE $(\kappa r \rho V)_n$.

COMPUTE $(r\rho u)_s$.

COMPUTE ρ_τ.

COMPUTE u_s.

COMPUTE S_s.



COMPUTE AVERAGE VALUE OF ρ TO INTRODUCE THE STABILIZING TERM.

COMPUTE AVERAGE VALUE OF u TO INTRODUCE THE STABILIZING TERM.

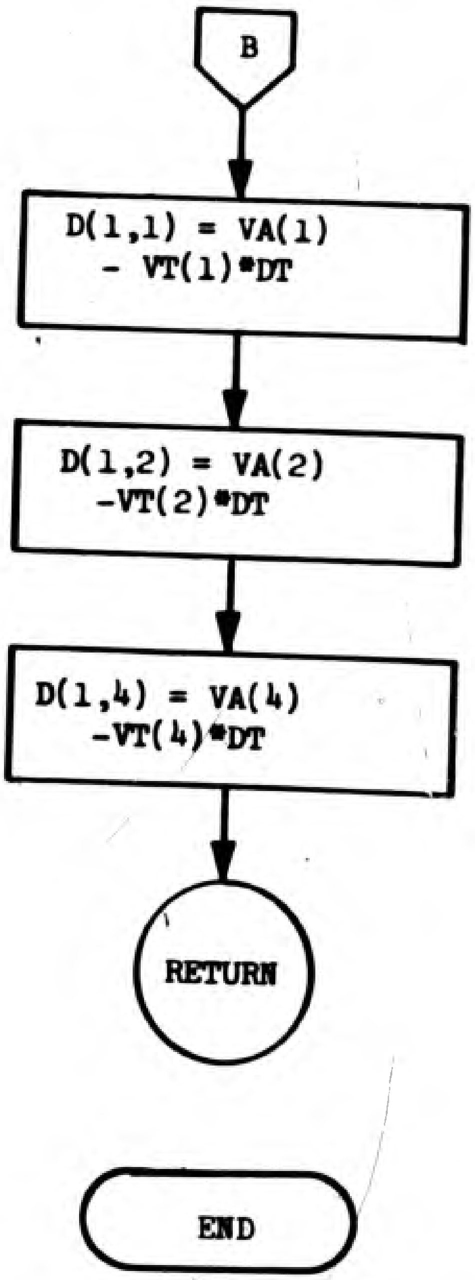
COMPUTE AVERAGE VALUE OF S TO INTRODUCE THE STABILIZING TERM.

COMPUTE P_s.

COMPUTE U_τ.

COMPUTE S_τ.

TRANSFER ρ, U AND S AT S-ΔS AND $\tau+\Delta\tau$ FROM TEMPORARY TO PERMANENT STORAGE.



SOLVE FOR ρ AT τ+Δτ.

SOLVE FOR u AT τ+Δτ.

SOLVE FOR S AT τ+Δτ.

RETURN.

END STATEMENT.

D. LISTING

```
      SUBROUTINE BODY (DN,DKAP,DR)
C
C      THIS ROUTINE TREATS THE NODES ON THE BODY SURFACE EXCLUSIVE
C     OF THE STAGNATION POINT.
C
      COMMON /1/ V(15,41,4),D(41,4),P(3,41)
      COMMON /2/ RB(15),ZB(15),TH(15),CTH(15),K(15),NS(15),WS(15)
      COMMON /5/ DXI,DET,DX2,DE2,DT
      COMMON /7/ I,IP,IM,L,LP,LM
      COMMON /10/ VS(4),VN(4),VA(4),VT(4)
      COMMON /13/ AVS,AVN,VIS
      REAL NS,K,KAP
      ZZ=1.0+DKAP
      DUM=RB(I)+DR
      VN(1)=-ZZ*DUM*V(I,2,1)*V(I,2,2)/DN
      VS(1)=(RB(IP)*V(IP,1,1)*V(IP,1,2)-RB(IM)*V(IM,1,1)*V(IM,1,2))/DX2
      VT(1)=(VS(1)+VN(1))/RB(I)
      VS(2)=(V(IP,1,2)-V(IM,1,2))/DX2
      VS(4)=(V(IP,1,4)-V(IM,1,4))/DX2
      VA(1)=AVS*(V(IP,1,1)+V(IM,1,1))+VIS*V(I,1,1)
      VA(2)=AVS*(V(IP,1,2)+V(IM,1,2))+VIS*V(I,1,2)
      VA(4)=AVS*(V(IP,1,4)+V(IM,1,4))+VIS*V(I,1,4)
      PS=(P(LP,1)-P(LM,1))/DX2
      VT(2)=V(I,1,2)*VS(2)+PS/V(I,1,1)
      VT(4)=V(I,1,2)*VS(4)
      V(IM,1,1)=D(1,1)
      V(IM,1,2)=D(1,2)
      V(IM,1,4)=D(1,4)
      D(1,1)=VA(1)-VT(1)*DT
      D(1,2)=VA(2)-VT(2)*DT
      D(1,4)=VA(4)-VT(4)*DT
      RETURN
      END
```

5. SUBROUTINE COEFF

A. PURPOSE

This subroutine generates coefficients for a cubic fit of the data supplied under the arbitrary body shape option. Conventional curve fit techniques such as least squares or spline fit techniques were found to be inadequate for this application. The present curve fit procedure requires that the user supply a fairly complete description of the body geometry at all data points supplied (r, z, θ, and K). It is recognized that approximate methods may be required to generate this data for many body shapes of interest. However, this data is often known explicitly, but is not easily specified at equal intervals along the body, e.g., an ellipse. It was decided that an accurate technique should be incorporated into the program, leaving the choice of more approximate schemes to the user when that becomes necessary. The cubic fit is accomplished using arrays which specify the values of x, y and $\frac{dy}{dx}$ at a number of points. The relevant constraint relations for a cubic fit can be expressed as

$$y_j = \sum_{i=1}^{4} C_{ij} X_j^{(i-1)} \qquad (64)$$

$$\left(\frac{dy}{dx}\right)_j = \sum_{i=2}^{4} (i-2)\, C_{ij} X_j^{(i-2)} \qquad (65)$$

At end points of the array of X_y values, the values of y_j and $(\frac{dy}{dx})_j$ at 2 adjacent points are used to generate the coefficient c_{ij}. Defining the matrixes

$$\left[B_{ij}\right] = \begin{bmatrix} 1 & X_1 & X_1^2 & X_1^3 \\ 1 & X_2 & X_2^2 & X_2^3 \\ 0 & 1 & 2X_1 & 3X_1^2 \\ 0 & 1 & 2X_2 & 3X_2^2 \end{bmatrix} \qquad (66)$$

$$\left[Y_j\right] = \begin{bmatrix} y_1 \\ y_2 \\ (\frac{dy}{dx})_1 \\ (\frac{dy}{dx})_2 \end{bmatrix} \qquad (67)$$

a matrix, $\left[A\right]$, is generated from $\left[B\right]$ by replacing the i^{th} column with $\left[Y_j\right]$ and the determinants $|A|$ and $|B|$ are used to generate C_{ij}

$$C_{ij} = |A_{ij}| / |B_{ij}| \qquad (68)$$

For interior points, the values of y at 2 adjacent points are used with the local value and slope to define $\left[B_{ij}\right]$ by

$$\left[B_{ij}\right] = \begin{bmatrix} 1 & X_1 & X_1^2 & X_1^3 \\ 1 & X_2 & X_2^2 & X_2^3 \\ 1 & X_3 & X_3^2 & X_3^3 \\ 0 & 1 & 2X_2 & 3X_1^2 \end{bmatrix} \qquad (69)$$

where the point under consideration is presumed to be X_2. Using

$$\left[Y_j\right] = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ (\frac{dy}{dx})_2 \end{bmatrix} \qquad (70)$$

Equation (68) supplies the desired cubic coefficients for the interior points.

B. VARIABLE LIST

A - $[A_{ij}]$

B - $[B_{ij}]$

C - $[C_{ij}]$

DET - $|B|$

DETC - $|A|$

DYDX - array containing values of $\frac{dy}{dx}$.

I - index for DO loops.

J - index for DO loops.

K - index for DO loops.

MN - index of last interior point.

N - index specifying the location of the point under consideration in the input arrays.

NEX - integer exponent used in loading arrays.

NL - index of first point to be considered.

NM - N - 1

NN - index of first interior point.

NP - N + 1

NPTS - number of points to be considered.

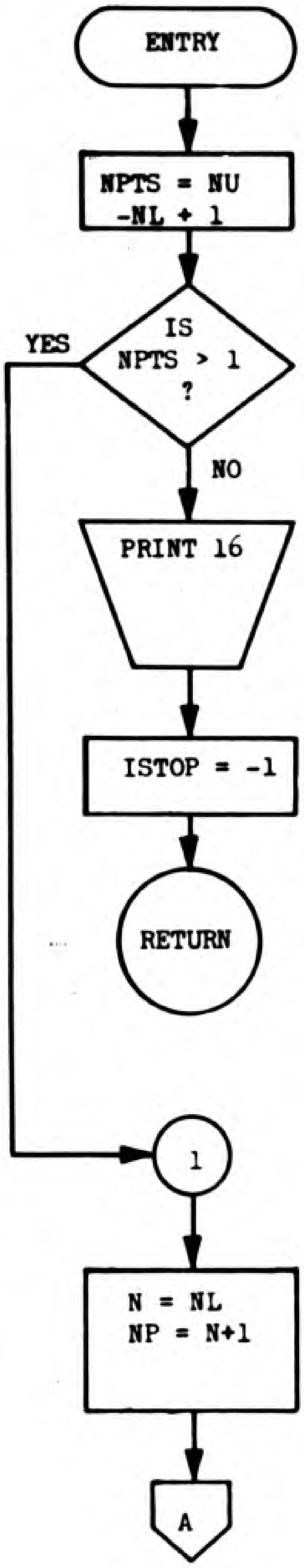
NU - index of the last point.

X - array containing values of X.

Y - array containing values of Y.

YC - $[Y_j]$

C. FLOW CHART



ENTER SUBROUTINE COEFF.

COMPUTE TOTAL NUMBER OF POINTS TO BE CONSIDERED.

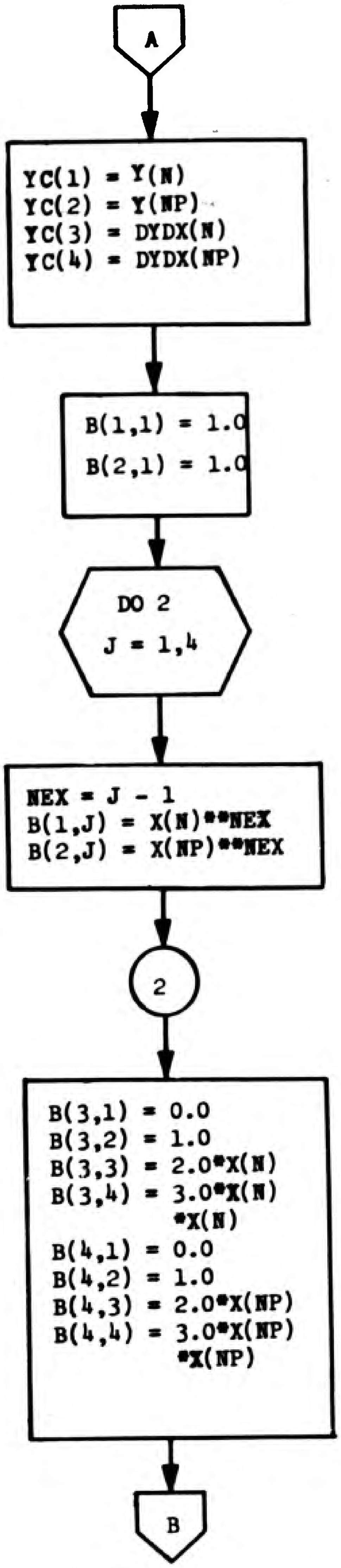
WERE AT LEAST 2 POINTS PROVIDED ?

PRINT ERROR MESSAGE.

SET ERROR INDICATOR.

RETURN.

SET INDICES FOR FIRST POINT.



```
YC(1) = Y(N)
YC(2) = Y(NP)
YC(3) = DYDX(N)
YC(4) = DYDX(NP)
```

LOAD THE YC ARRAY FOR THE FIRST POINT.

```
B(1,1) = 1.0
B(2,1) = 1.0
```

SET b_{11} AND b_{21}.

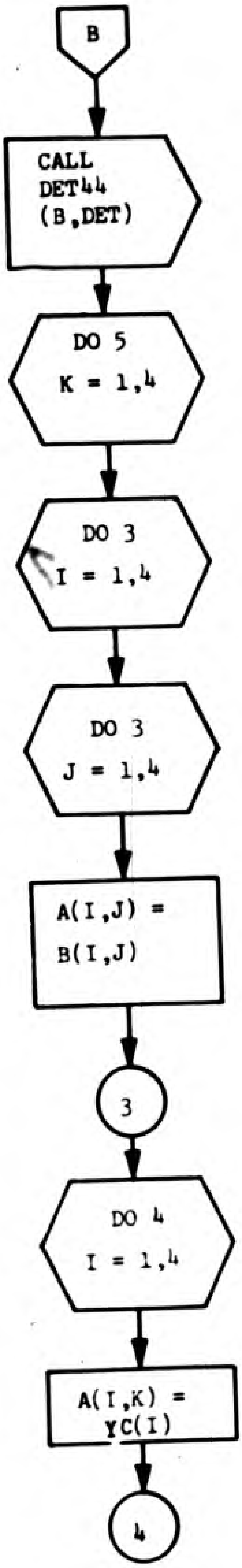
```
DO 2
J = 1,4
```

```
NEX = J - 1
B(1,J) = X(N)**NEX
B(2,J) = X(NP)**NEX
```

LOAD COLUMNS 2 TO 3 OF ROWS 1 AND 2 OF THE B ARRAY FOR THE FIRST POINT.

```
B(3,1) = 0.0
B(3,2) = 1.0
B(3,3) = 2.0*X(N)
B(3,4) = 3.0*X(N)
         *X(N)
B(4,1) = 0.0
B(4,2) = 1.0
B(4,3) = 2.0*X(NP)
B(4,4) = 3.0*X(NP)
         *X(NP)
```

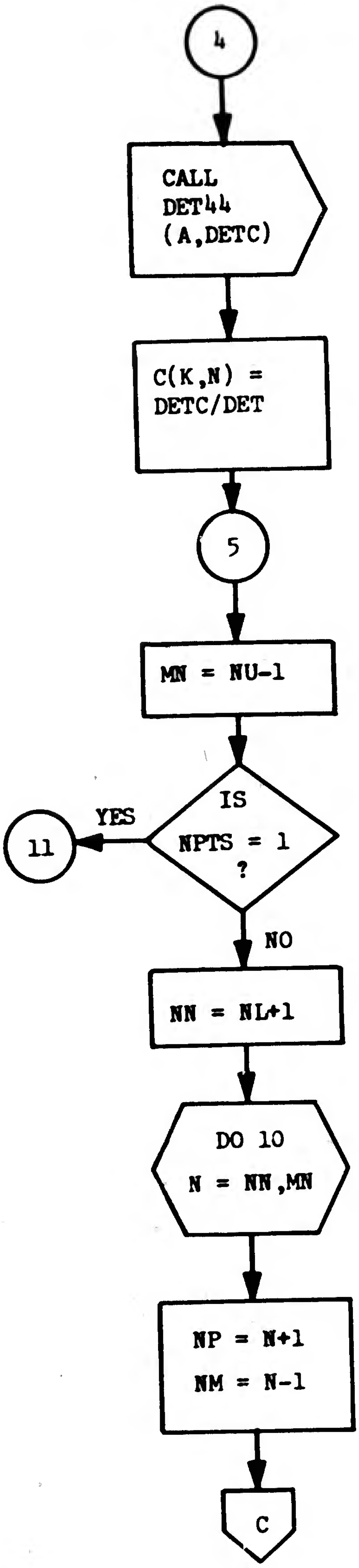
LOAD REMAINING TERMS IN THE B ARRAY FOR THE FIRST POINT.



COMPUTE DETERMINANT OF B.

LOAD A ARRAY FOR THE FIRST POINT.

LOAD YC ARRAY INTO THE A ARRAY FOR COMPUTATION OF C_k

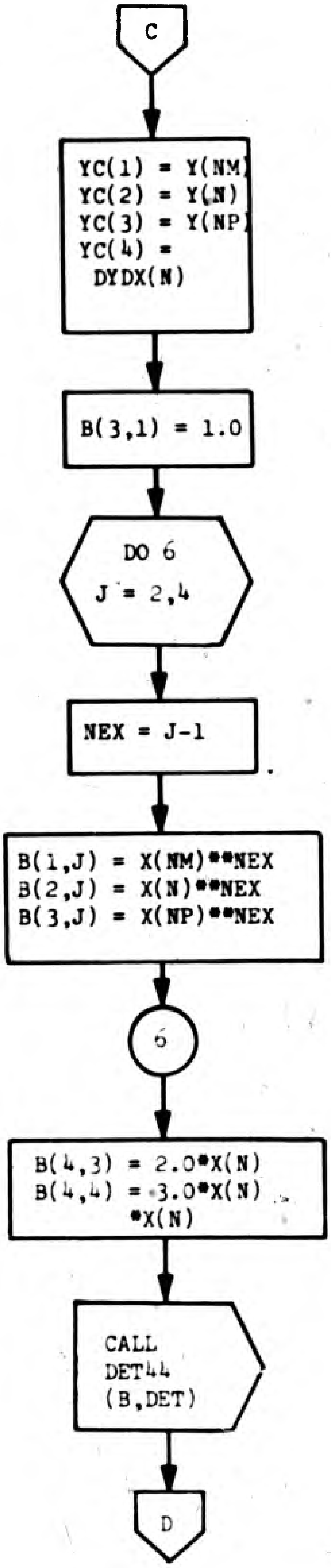


COMPUTE DETERMINANT OF A.

COMPUTE C_k's FOR THE FIRST POINT.

WERE ONLY 2 POINTS PROVIDED ?

START LOOP FOR INTERIOR POINTS.



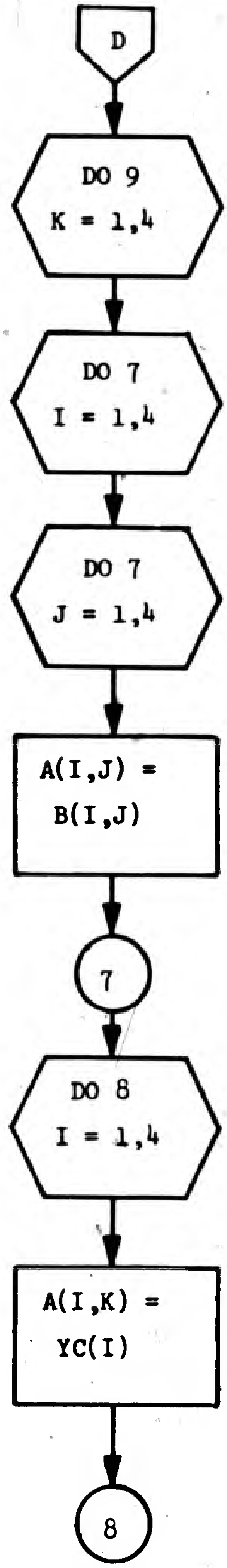
LOAD YC ARRAY.

COMPUTE EXPONENT.

LOAD COLUMNS 2, 3 AND 4
OF ROWS 1, 2 AND 3.

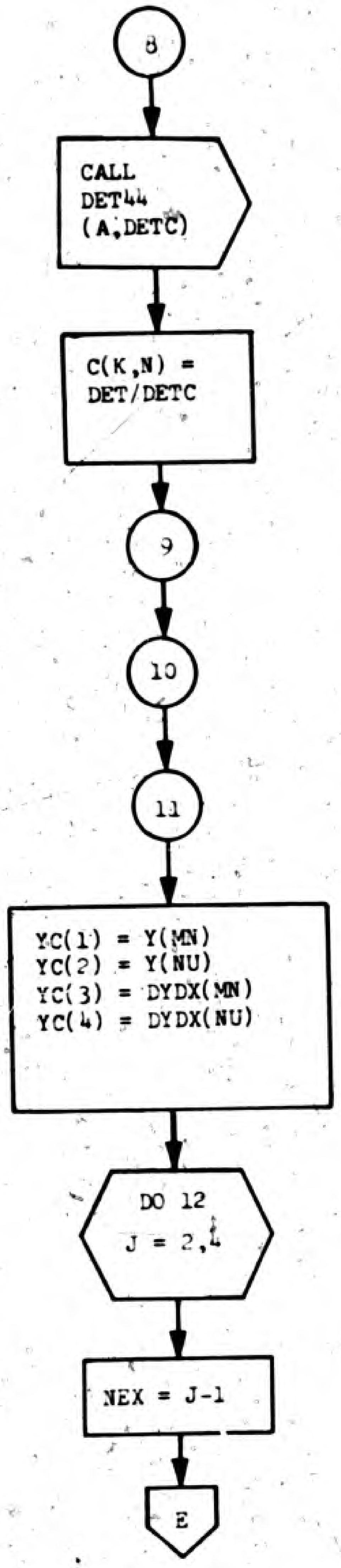
LOAD REMAINING TERMS.

COMPUTE DETERMINATION OF B.



LOAD A ARRAY FROM B ARRAY.

LOAD YC VALUES INTO THE KTH COLUMN OF THE A ARRAY.



COMPUTE DETERMINANT OF A.

COMPUTE C_k's.

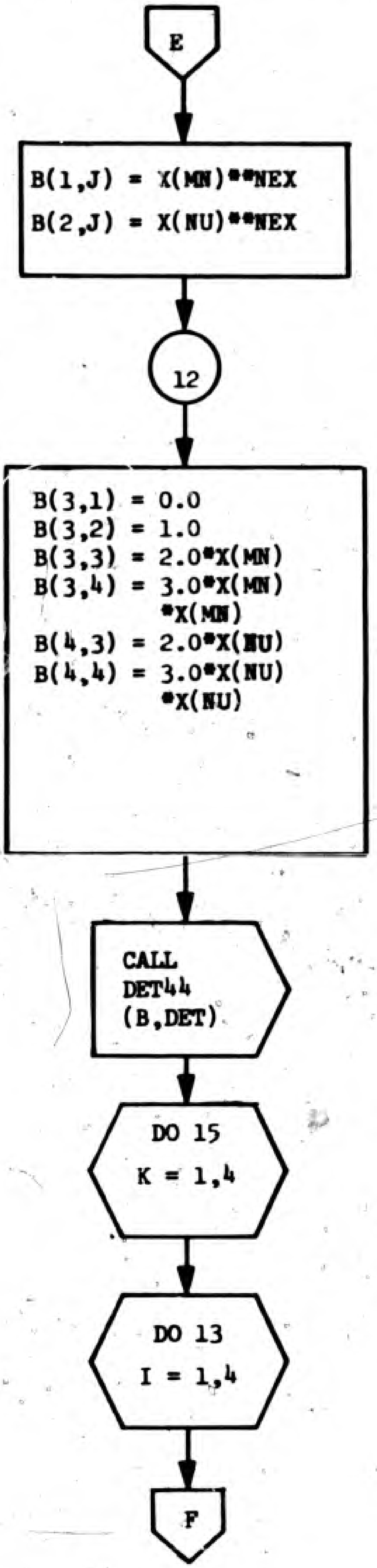
END OF DO LOOP ON K.

END OF DO LOOP OVER ALL INTERIOR POINTS.

START COMPUTATIONS FOR THE LAST POINT.

LOAD YC FOR LAST POINT.

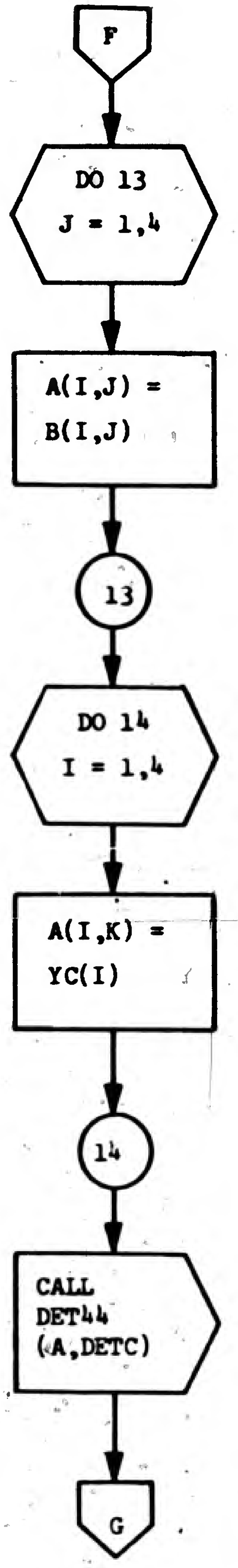
COMPUTE EXPONENT FOR THE J^{TH} COLUMN.



LOAD COLUMNS 2 TO 4 OF ROWS 1 AND 2 OF THE B ARRAY.

LOAD REMAINING TERMS INTO THE B ARRAY.

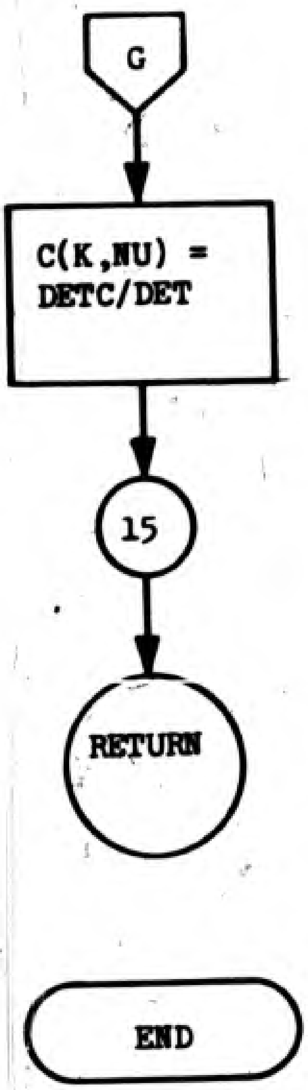
COMPUTE DETERMINANT OF B.



LOAD A ARRAY FROM THE B ARRAY.

LOAD YC VALUES INTO KTH COLUMN OF A ARRAY.

COMPUTE DETERMINANT OF A.



COMPUTE C_k FOR THE LAST POINT.

RETURN.

END

C. LISTING

```
      SUBROUTINE COEFF (NL,NU,X,Y,DYDX)
C
C      THIS ROUTINE COMPUTES THE COEFFICIENTS FOR A CUBIC FIT WHERE
C     THE VALUE OF THE FUNCTION AND ITS SLOPE AT THE POINT UNDER
C     CONSIDERATION AND ITS VALUES AT THE TWO ADJACENT POINTS ARE USED.
C
      COMMON /11/ IDUMMY,ISTOP
      COMMON /SPLINE/ C(4,15)
      DIMENSION X(15), Y(15), DYDX(15), YC(4), B(4,4), A(4,4)
      NPTS=NU-NL+1
      IF (NPTS.GT.1) GO TO 1
      PRINT 16
      ISTOP=-1
      RETURN
1     N=NL
      NP=N+1
      YC(1)=Y(N)
      YC(2)=Y(NP)
      YC(3)=DYDX(N)
      YC(4)=DYDX(NP)
      B(1,1)=1.0
      B(2,1)=1.0
      DO 2 J=2,4
      NEX=J-1
      B(1,J)=X(N)**NEX
2     B(2,J)=X(NP)**NEX
      B(3,1)=0.0
      B(3,2)=1.0
      B(3,3)=2.0*X(N)
      B(3,4)=3.0*X(N)*X(N)
      B(4,1)=0.0
      B(4,2)=1.0
      B(4,3)=2.0*X(NP)
      B(4,4)=3.0*X(NP)*X(NP)
      CALL DET44 (B,DET)
      DO 5 K=1,4
      DO 3 I=1,4
      DO 3 J=1,4
3     A(I,J)=B(I,J)
      DO 4 I=1,4
4     A(I,K)=YC(I)
      CALL DET44 (A,DETC)
5     C(K,N)=DETC/DET
      MN=NU-1
      IF (NPTS.EQ.2) GO TO 11
      NN=NL+1
      DO 11 N=NN,MN
      NP=N+1
      NM=N-1
      YC(1)=Y(NM)
      YC(2)=Y(N)
      YC(3)=Y(NP)
```

NOT REPRODUCIBLE

```
      YC(4)=DYDX(N)
      B(3,1)=1.0
      DO 6 J=2,4
      NEX=J-1
      B(1,J)=X(NM)**NEX
      B(2,J)=X(N)**NEX
    6 B(3,J)=X(NP)**NEX
      B(4,3)=2.0*X(N)
      B(4,4)=3.0*X(N)*X(N)
      CALL DET44 (B,DET)
      DO 9 K=1,4
      DO 7 I=1,4
      DO 7 J=1,4
    7 A(I,J)=B(I,J)
      DO 8 I=1,4
    8 A(I,K)=YC(I)
      CALL DET44 (A,DETC)
    9 C(K,N)=DETC/DET
   10 CONTINUE
   11 YC(1)=Y(MN)
      YC(2)=Y(NU)
      YC(3)=DYDX(MN)
      YC(4)=DYDX(NU)
      DO 12 J=2,4
      NEX=J-1
      B(1,J)=X(MN)**NEX
   12 B(2,J)=X(NU)**NEX
      B(3,1)=0.0
      B(3,2)=1.0
      B(3,3)=2.0*X(MN)
      B(3,4)=3.0*X(MN)*X(MN)
      B(4,3)=2.0*X(NU)
      B(4,4)=3.0*X(NU)*X(NU)
      CALL DET44 (B,DET)
      DO 15 K=1,4
      DO 13 I=1,4
      DO 13 J=1,4
   13 A(I,J)=B(I,J)
      DO 14 I=1,4
   14 A(I,K)=YC(I)
      CALL DET44 (A,DETC)
   15 C(K,NU)=DETC/DET
      RETURN
C
   16 FORMAT (//30X,25HERROR IN SUBROUTINE CCEFF/10X,33HNUMBER OF POINTS
     * IS LESS THAN TWO/10X,15HCASE TERMINATED)
      END
```

6. SUBROUTINE DATA

A. PURPOSE

This subroutine loads all of the input data except for the arbitrary body shape data. It sorts the input data by matching alphanumeric specifications on the data cards with known variable names and with special alphanumeric flags signaling the start and end of a data packet. Consequently, the number of cards provided and the order of these cards are arbitrary. Since the computer code does not alter the values of the input variables, only variables whose resident values are to be changed need be input by the user. This simplifies the input data required for multicase runs. The principal reason for including a sorting routine of this type was to greatly simplify the input data required without imposing unnecessary restrictions on the numerical method. In addition to the empirical constants required for the numerical method, several option control variables had to be included. For example, to apply this method as a practical computational technique, a restart option had to be included since the exact number of iterations that will be required cannot be specified in advance. Also, if a steady state is not realized for a problem, an option to recover the data and perform additional iterations should be available. Other input controls considered desirable are specifications for additional flow field outputs, dimensional output, and nodal geometry. A a result, 23 input variables were required for a completely general computational method. Most of these variables have been supplied initial resident values in the main program which will be adequate for the majority of the problems encountered. Consequently, while the capability of inputting values for all 23 variables is available, the input required is usually quite brief, typically 3 to 5 cards. It is easily seen that without a routine of this type, a general input capability would be very complicated.

B. VARIABLE LIST

ENFILE - alphanumeric word LAST in A10 format. Used to check for the end of the data packet.

I - index for an implied DO loop in a print statement.

IERR - error indicator. Called ISTOP in the main program.

IV - array containing all integer variables in COMMON block /INPUT/.

K - array used for temporary storage of the values of input integer variables.

M - index used for various applications.

T - array containing the alphanumeric names of all allowed input variables.

TITLE - alphanumeric word TITLE in A10 format. Used to identify the first card in a data packet.

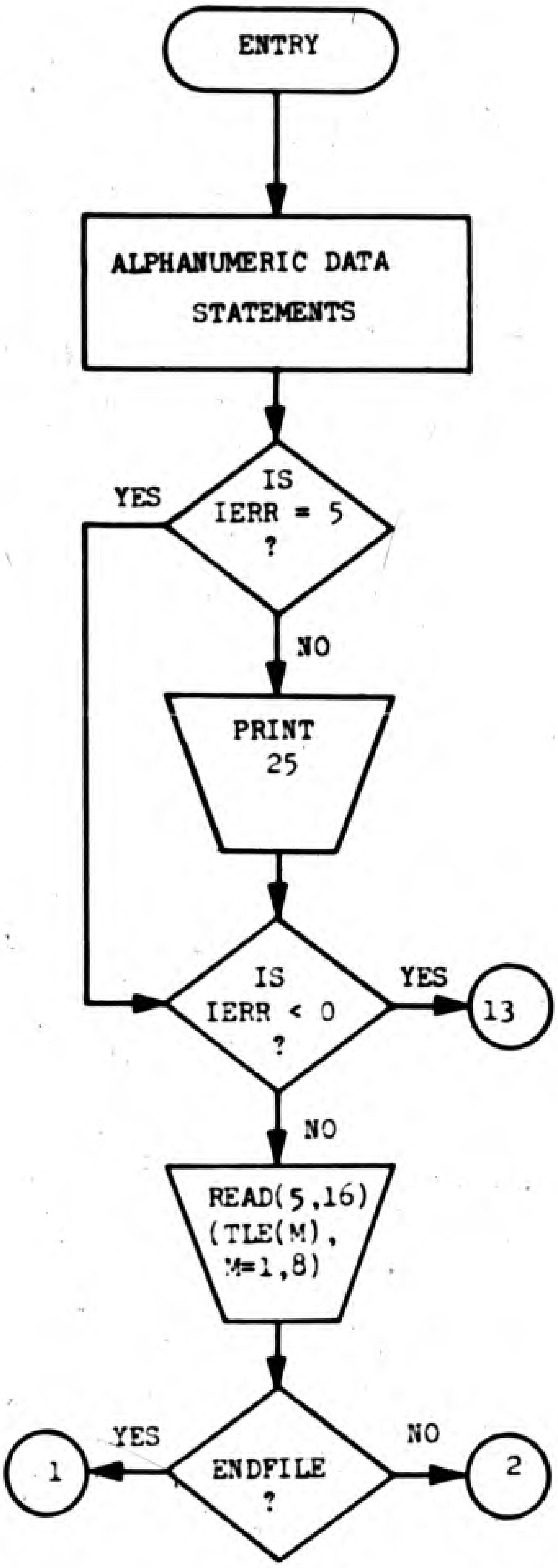
TLE - array used to read and print an alphanumeric title.

V - array containing all floating point variables in COMMON block /INPUT/.

VAR - variable used to read the input variable name from data cards.

ZZ - variable used for temporary storage of the values of input floating point variables.

C. FLOW CHART



ENTER SUBROUTINE DATA.

SET ALPHANUMERIC TITLES
OF THE CASE DATA VARIABLES.

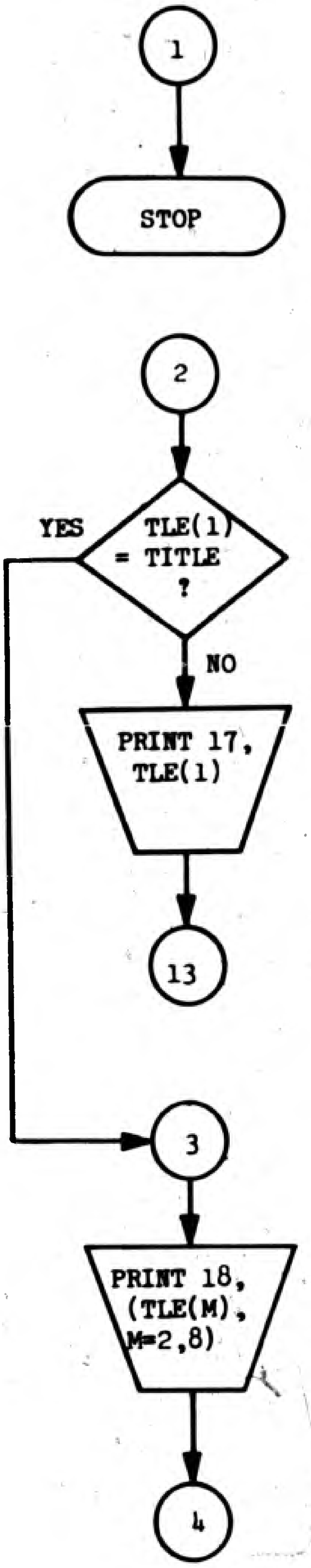
FIRST CASE ?

START A NEW PAGE.

ERROR SENSED IN
PREVIOUS CASE ?

READ TITLE CARD.

ENDFILE ENCOUNTERED ?

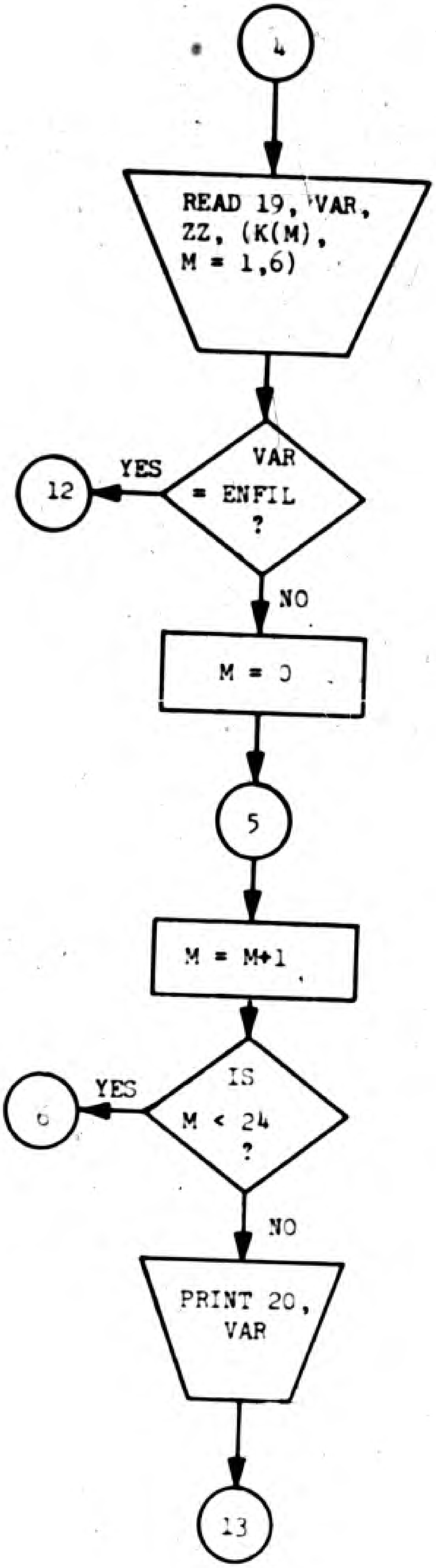


TERMINATE JOB.

WAS TITLE INDICATOR CORRECT ?

PRINT ERROR MESSAGE.

PRINT TITLE.



READ CASE DATA CARD.

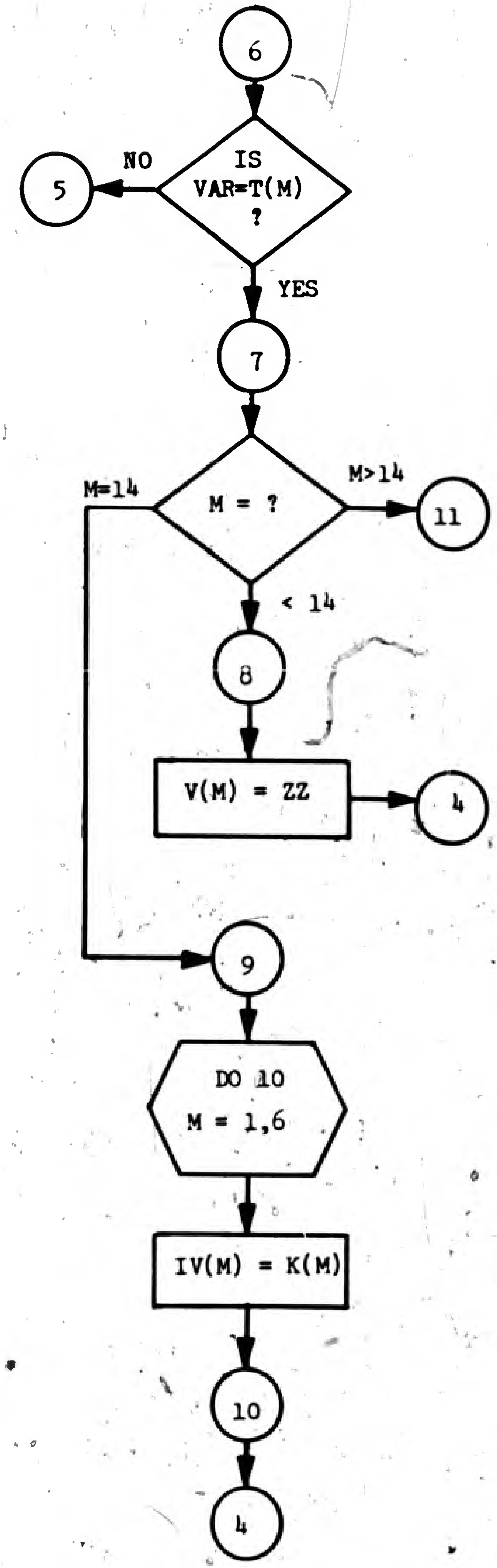
LAST CARD IN THE CASE ?

INITIALIZE INDEX, M, TO SEARCH TABLE OF VARIABLE NAMES FOR THE PRESENT VARIABLE.

INCREMENT M.

IS M WITHIN THE ALLOWABLE NUMBER OF CASE VARIABLES?

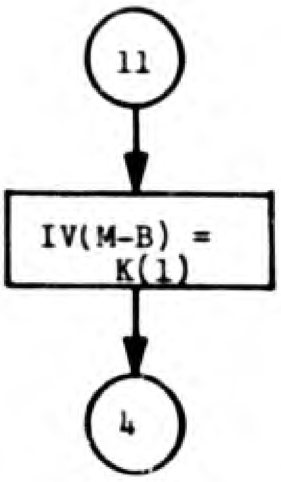
VARIABLE NAME COULD NOT BE MATCHED WITH AN INPUT VARIABLE. PRINT ERROR MESSAGE.



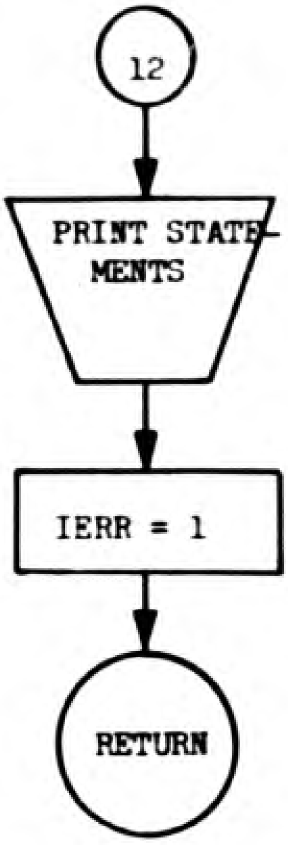
DOES VARIABLE NAME READ MATCH THE NAME STORED IN T(M) ?

FLOATING POINT VARIABLE. STORE IN PROPER COMMON LOCATION.

IPR(6) ARRAY WAS READ. STORE IN PROPER COMMON LOCATION.



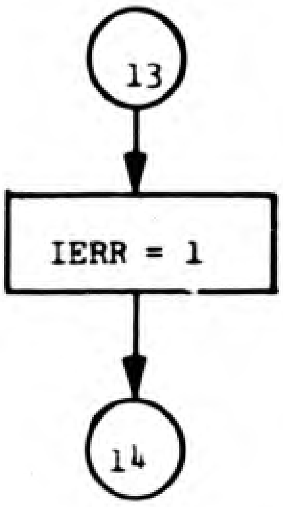
INTEGER VARIABLE. STORE IN PROPER COMMON LOCATION.



PRINT INPUT DATA.

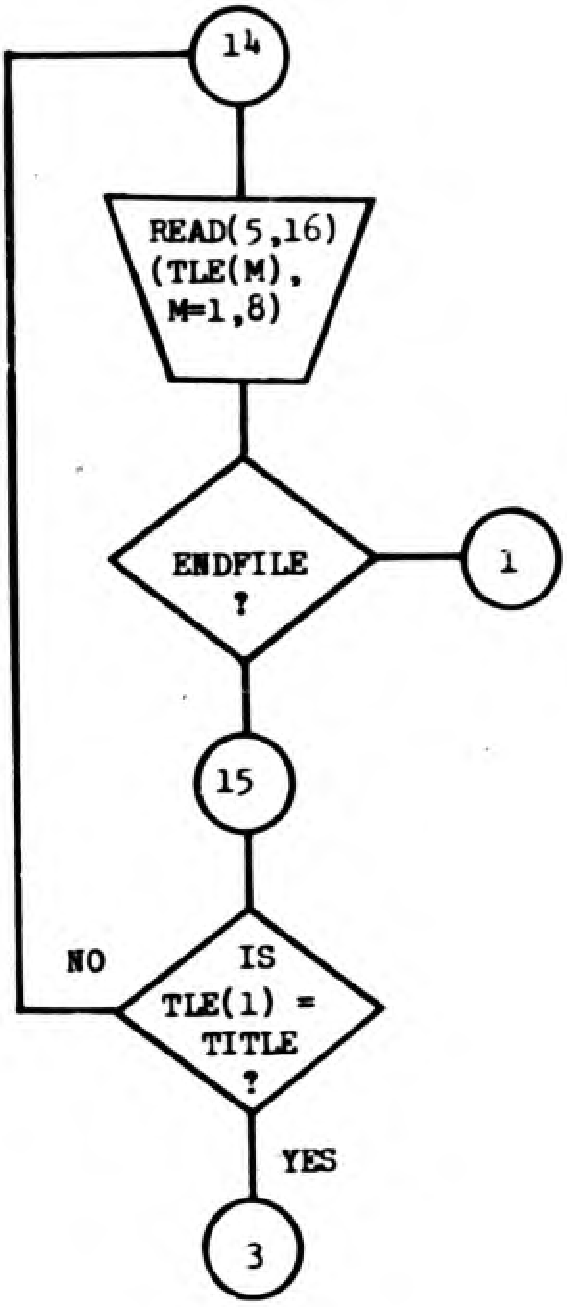
SET ERROR INDICATOR.

RETURN.



START PROCEDURES TO BYPASS ERRONEOUS DATA.

SET ERROR INDICATOR.



READ DATA CARD.

WAS AN ENDFILE ENCOUNTERED ?

WAS A TITLE CARD READ ?

B. LISTING

```
      SUBROUTINE DATA (IERR)
C
C     THIS ROUTINE LOADS THE CASE DATA.
C
      COMMON /IN/ TLE(8),ZZ,VAR
      COMMON /INPUT/ V(13),IV(15)
      DIMENSION T(23), K(6)
      DATA (T(M),M=1,23)/
     *10HGAMMA     ,10HM         ,10HTHETA     ,10HZ/RN      ,
     *10HTCRIT1    ,10HTCRIT2    ,10HP1        ,10HRHO1      ,
     *10HR1        ,10HMU        ,10HT1        ,10HERRORN    ,
     *10HERRORR    ,10HIPR       ,10HINTAP     ,10HOTAP      ,
     *10HMAXIT     ,10HNCASE     ,10HNFILE     ,10HNSEG      ,
     *10HMAXI      ,10HMAXN      ,10HNTAP      /
      DATA TITLE,ENFIL/10HTITLE       ,10HLAST       /
      IF (IERR.LT.0) GO TO 13
      IF (IERR.NE.0) PRINT 25
      READ (5,16) (TLE(M),M=1,8)
      IF (ENDFILE 5) 1,2
1     STOP
2     IF (TLE(1).EQ.TITLE) GO TO 3
      PRINT 17, TLE(1)
      GO TO 13
3     PRINT 18, (TLE(M),M=2,8)
4     READ 19, VAR,ZZ,(K(M),M=1,6)
      IF (VAR.EQ.ENFIL) GO TO 12
      M=0
5     M=M+1
      IF (M.LT.24) GO TO 6
      PRINT 20, VAR
      GO TO 13
6     IF (VAR-T(M)) 5,7,5
7     IF (M-14) 8,9,11
8     V(M)=ZZ
      GO TO 4
9     DO 10 M=1,6
10    IV(M)=K(M)
      GO TO 4
11    IV(M-8)=K(1)
      GO TO 4
12    PRINT 21
      PRINT 22, ((T(I+8),IV(I)),I=7,15)
      PRINT 23, ((M,IV(M)),M=1,6)
      PRINT 24, ((T(M),V(M)),M=1,13)
      IERR=0
      RETURN
13    IERR=1
14    READ (5,16) (TLE(M),M=1,8)
      IF (ENDFILE 5) 1,15
15    IF (TLE(1).EQ.TITLE) GO TO 3
      GO TO 14
C
```

```
16    FORMAT (8A10)
17    FORMAT (1H1,30X,19HERROR IN INPUT DATA/30X,21HTITLE CARD NOT SEN*E
     *)/30X,4HREAD,1X,3H10H,A10,1X,25HRATHER THAN 10HTITLE      /30X,15HC
     *ASE TERMINATED)
18    FORMAT(//25X,13HCASE TITLE - ,7A10)
19    FORMAT (A10,F10.4,6I10)
20    FORMAT (/30X,19HERROR IN INPUT DATA/30X,42HAN INCORRECT VARIABLE N
     *AME HAS ENCOUNTERED/30X,12HTHE NAME WAS,1X,A10/30X,15HCASE TERMINA
     *TED)
21    FORMAT (//25X,80H******************************************* INPUT DATA ***
     *********************************)
22    FORMAT (/15X,3(10X,A10,I10))
23    FORMAT (/15X,3(10X,4HTPR(,I1,1H),4X,I10))
24    FORMAT (/15X,3(10X,A10,F10.3))
25    FORMAT (1H1//////)
      END
```

7. DETERMINANT SUBROUTINES

A. PURPOSE

SUBROUTINE DET44 computes the determinant of a 4 X 4 array. It accompliplishes a standard expansion by minors using the explicit determinant solution for a 3 X 3 array contained in SUBROUTINE DET33.

B. VARIABLE LIST

A - 4 X 4 array for which the determinant is to be evaluated.

B - 3 X 3 array containing various minors of the A array.

DET - value of the determinant evaluated in both routines.

DET1 - determinant of the first minor.

DET2 - determinant of the second minor.

DET3 - determinant of the third minor.

DET4 - determinant of the fourth minor.

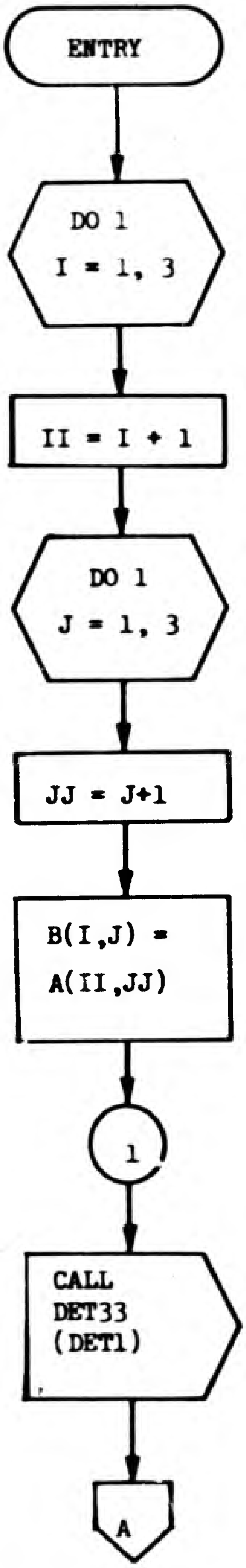
I - index used to load arrays.

II - index used to load arrays.

J - index used to load arrays.

JJ - index used to load arrays.

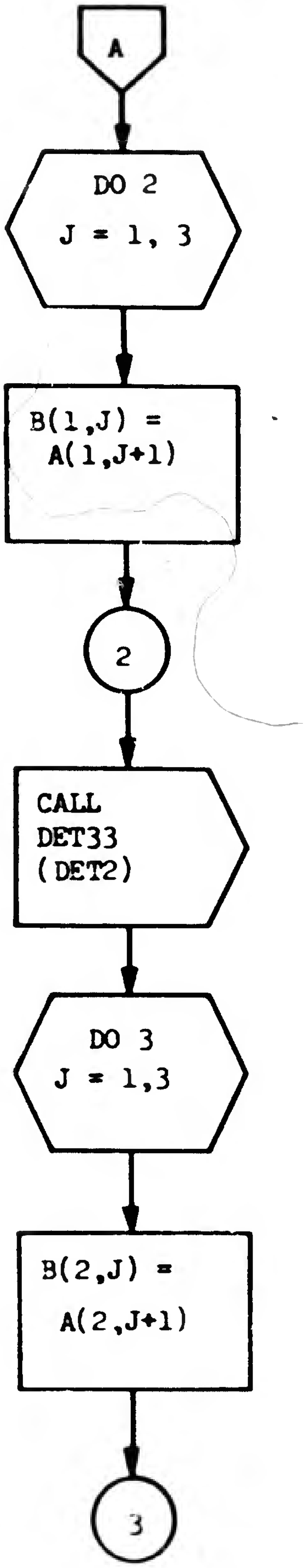
C. FLOW CHART



ENTER SUBROUTINE DET44.

LOAD 3X3 ARRAY FOR THE FIRST MINOR.

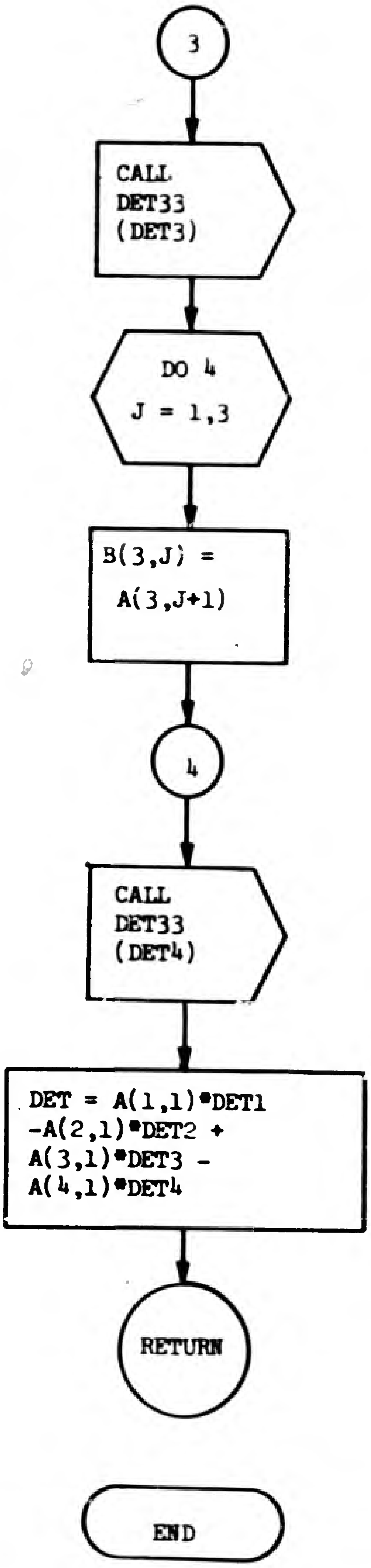
EVALUATE FIRST MINOR.



LOAD 3X3 ARRAY FOR
SECOND MINOR.

EVALUATE SECOND MINOR.

LOAD 3X3 ARRAY FOR
THIRD MINOR.



EVALUATE THIRD MINOR.

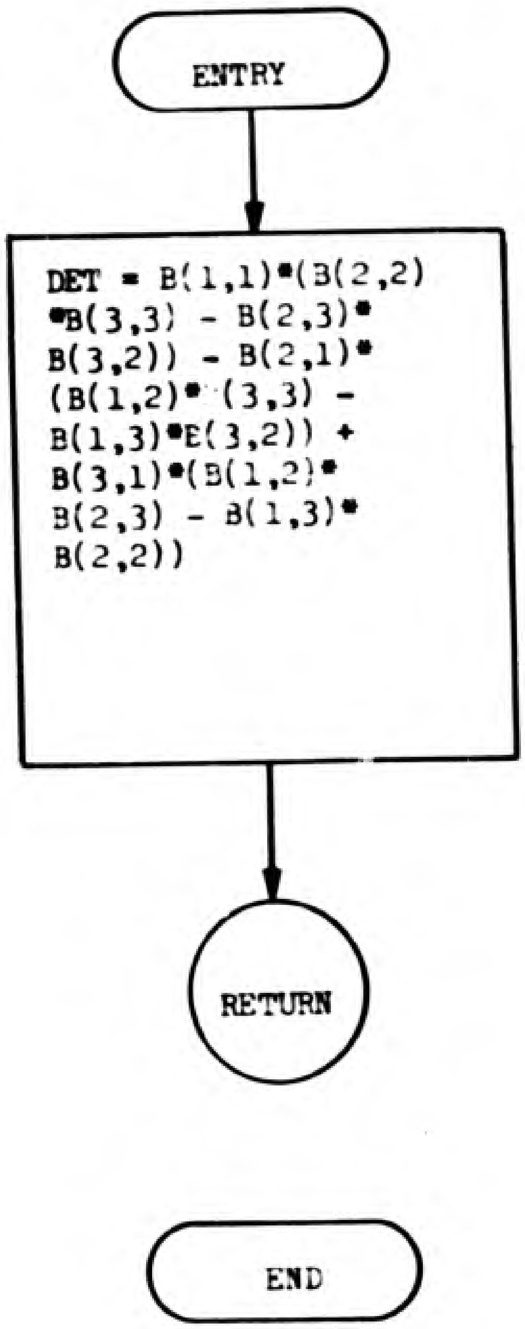
LOAD 3X3 ARRAY FOR THE FOURTH MINOR.

EVALUATE FOURTH MINOR.

EVALUATE DETERMINANT.

RETURN.

END.



ENTER SUBROUTINE DET33.

EXPLICIT SOLUTION FOR
A 3X3 DETERMINANT.

RETURN.

END.

D. LISTING

```
      SUBROUTINE DET44 (A,DET)                                          DET44   1
C                                                                       DET44   2
C      THIS ROUTINE SOLVES A 4X4 DETERMINANT.                           DET44   3
C                                                                       DET44   4
      COMMON /DETERM/ B(3,3)                                            DET44   5
      DIMENSION A(4,4)                                                  DET44   6
      DO 1 I=1,3                                                        DET44   7
      II=I+1                                                            DET44   8
      DO 1 J=1,3                                                        DET44   9
      JJ=J+1                                                            DET44  10
1     B(I,J)=A(II,JJ)                                                   DET44  11
      CALL DET33 (DET1)                                                 DET44  12
      DO 2 J=1,3                                                        DET44  13
2     B(1,J)=A(1,J+1)                                                   DET44  14
      CALL DET33 (DET2)                                                 DET44  15
      DO 3 J=1,3                                                        DET44  16
3     B(2,J)=A(2,J+1)                                                   DET44  17
      CALL DET33 (DET3)                                                 DET44  18
      DO 4 J=1,3                                                        DET44  19
4     B(3,J)=A(3,J+1)                                                   DET44  20
      CALL DET33 (DET4)                                                 DET44  21
      DET=A(1,1)*DET1-A(2,1)*DET2+A(3,1)*DET3-A(4,1)*DET4               DET44  22
      RETURN                                                            DET44  23
      END                                                               DET44  24


      SUBROUTINE DET33 (DET)                                            DET33   1
C                                                                       DET33   2
C      THIS ROUTINE SOLVES A 3X3 DETERMINANT.                           DET33   3
C                                                                       DET33   4
      COMMON /DETERM/ B(3,3)                                            DET33   5
      DET=B(1,1)*(B(2,2)*B(3,3)-B(2,3)*B(3,2))-B(2,1)*(B(1,2)*B(3,3)-B(1DET33   6
     *,3)*B(3,2))+B(3,1)*(B(1,2)*B(2,3)-B(1,3)*B(2,2))                  DET33   7
      RETURN                                                            DET33   8
      END                                                               DET33   9
```

8. SUBROUTINE DMPOUT

A. PURPOSE

This subroutine prints the flow-field data for each segment. Printed outputs are provided as requested by the user through the input variable IPR, and after the last iteration for each segment. The output is printed in dimensional form (except for entropy) based on the values of P1, T1, RHO1 and RN supplied in the input data. When all of these variables are unity, nondimensional output is provided. The normal velocity component, the coordinate A and the shock velocity are defined positive outward from the body in the printed output. The printed output is set up to include as many streamwise stations per page as possible, based on a maximum of 60 lines per page.

B. VARIABLE LIST

All variables previously defined in the description of the main program have been omitted from this list.

DAT - array containing the values of various body parameters for each station.

DS - ΔS

IND - control variable specifying when a value of DAT should be included on a given printed line.

IPAGE - number of streamwise stations which have been printed on the present page.

M - index used in print statements.

NPAGE - number of streamwise stations which can be printed on each page.

NST - array used to store the pressure coefficient at each body station for later output.

PREFIX - array containing the alphanumeric names of body parameters.

Q - $[P_\infty \rho_\infty]^{1/2}$

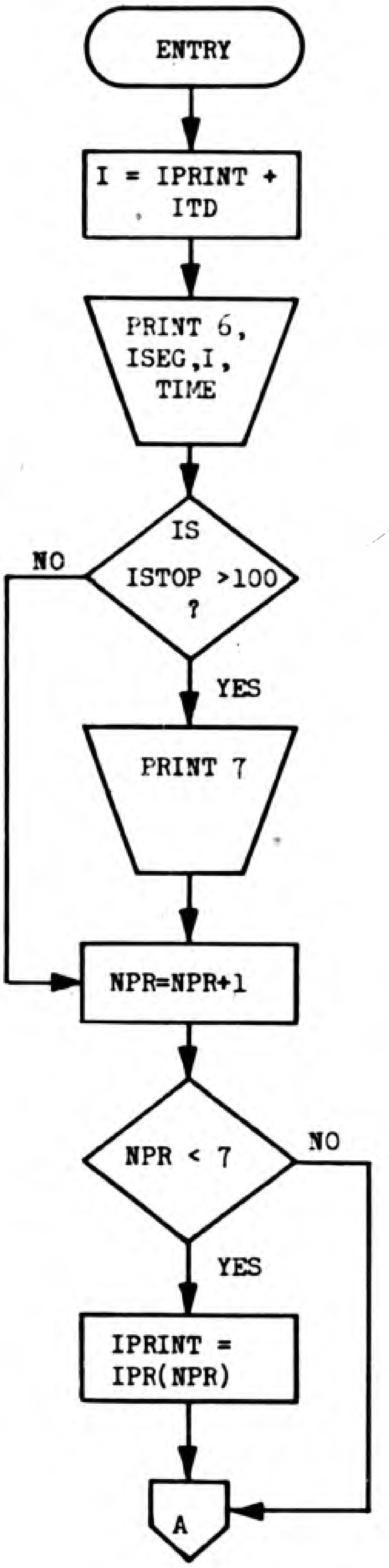
QSQ - $u^2 + v^2$

SI - array used to store the values of the ratio of local pressure to total pressure on the body for later output.

VS - array used to store dimensional values of output values as they are evaluated for later output.

ZZ - dummy variable used for intermediate computations.

C. FLOW CHART



ENTER SUBROUTINE DMPOUT.

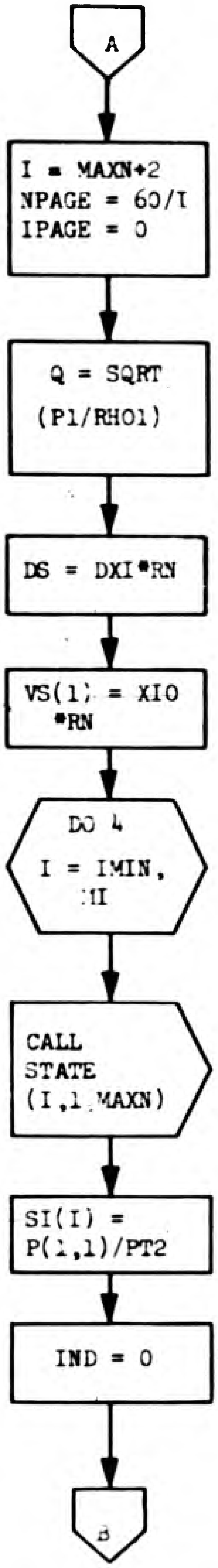
COMPUTE TOTAL NUMBER OF ITERATIONS PERFORMED ON THIS SEGMENT.

WAS THE MAXIMUM NUMBER OF NODES ACROSS THE SHOCK LAYER EXCEEDED IN SUBROUTINE GEOM ?

INCREMENT NPR.

ARE ANY VALUES OF IPR UNUSED ?

LOAD ITERATION NUMBER FOR NEXT PRINT.



COMPUTE THE NUMBER OF STATIONS WHICH CAN BE PRINTED ON EACH PAGE. INITIALIZE IPAGE.

COMPUTE SCALE FACTOR FOR DIMENSIONAL VELOCITIES.

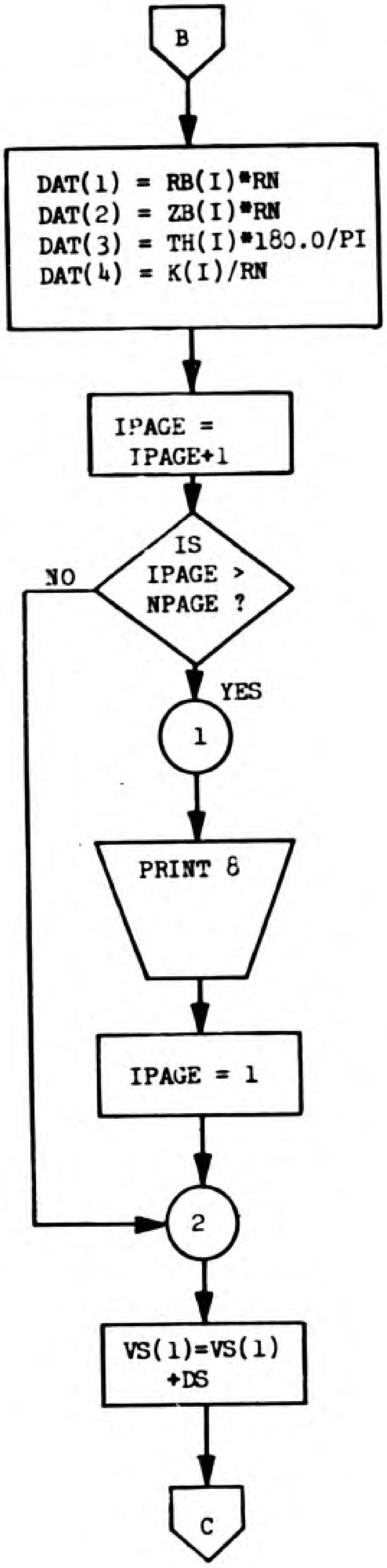
COMPUTE DIMENSIONAL ΔS.

INITIALIZE S.

COMPUTE PRESSURES.

STORE P/P_o.

INITIALIZE IND.



COMPUTE LOCAL BODY PARAMETERS.

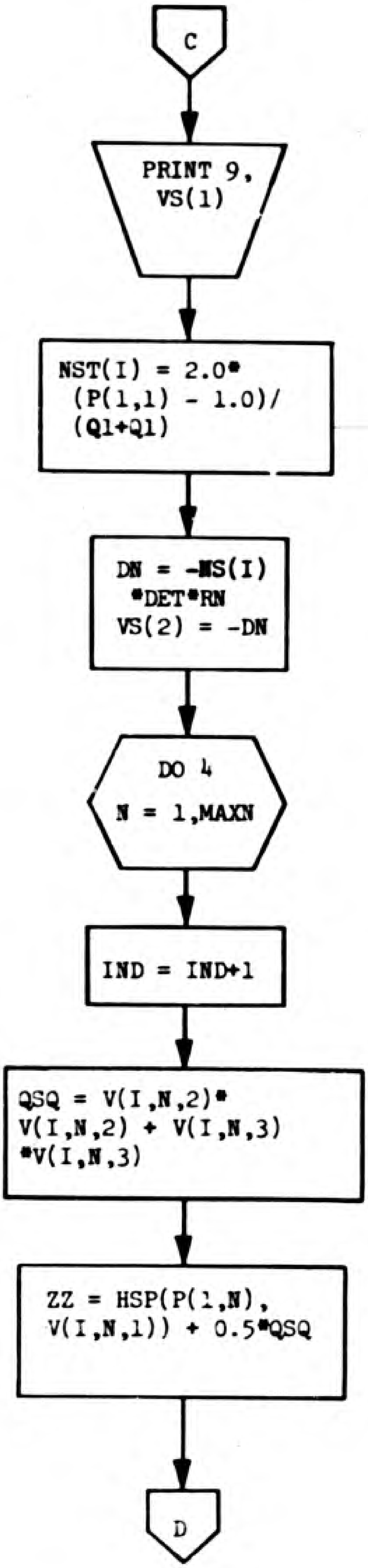
INCREMENT IPAGE.

START A NEW PAGE ?

NEW PAGE.

INITIALIZE IPAGE.

INCREMENT S.



PRINT S AND HEADING.

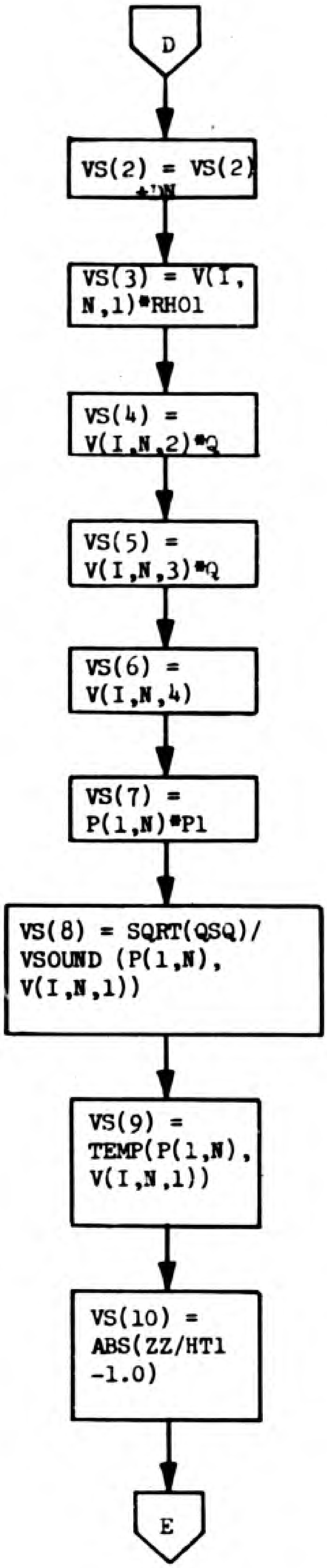
STORE PRESSURE COEFFICIENT AT THE BODY.

COMPUTE Δn AND INITIALIZE n.

INCREMENT IND.

COMPUTE MAGNITUDE OF LOCAL VELOCITY.

COMPUTE LOCAL TOTAL ENTHALPY.



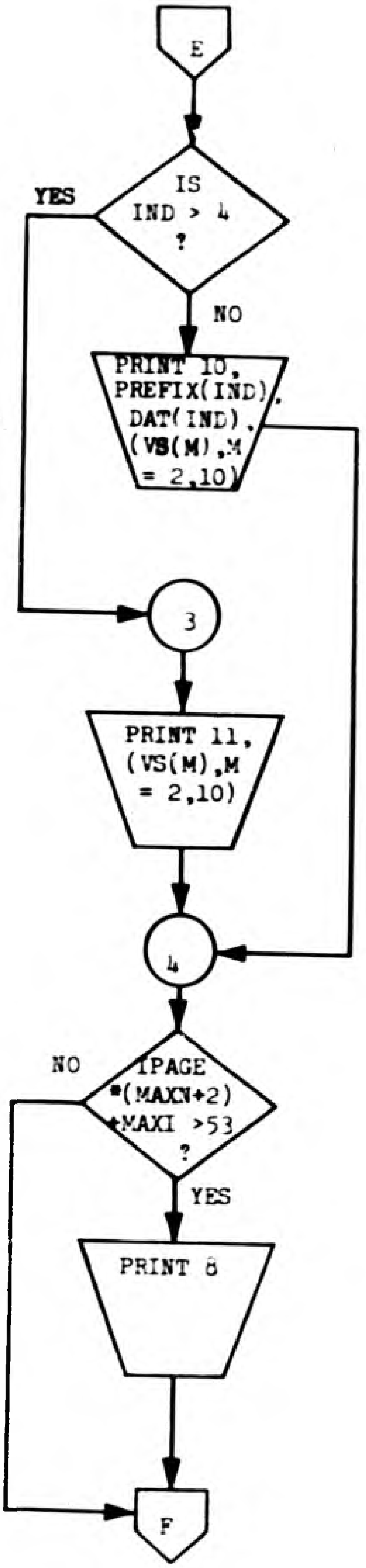
INCREMENT n.

COMPUTE ρ.

COMPUTE U.

COMPUTE V.

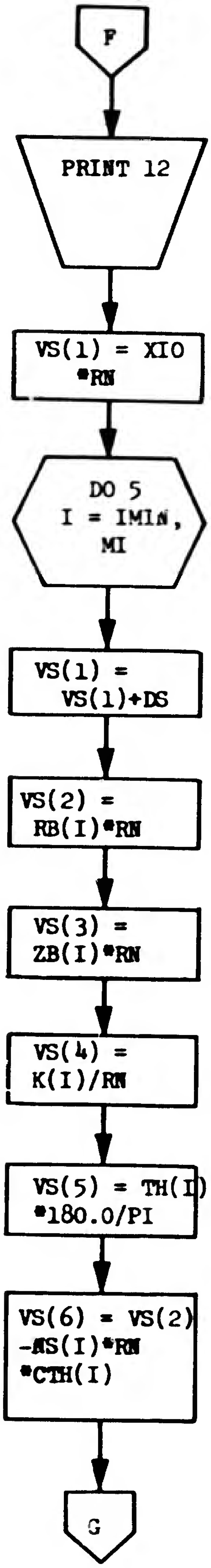
STORE S.



PRINT BODY PARAMETERS
AND FLOW FIELD DATA.

PRINT FLOW FIELD DATA.

NEW PAGE FOR SHOCK AND
BODY PARAMETERS ?



PRINT HEADING FOR SHOCK AND BODY PARAMETERS.

INITIALIZE S.

INCREMENT S.

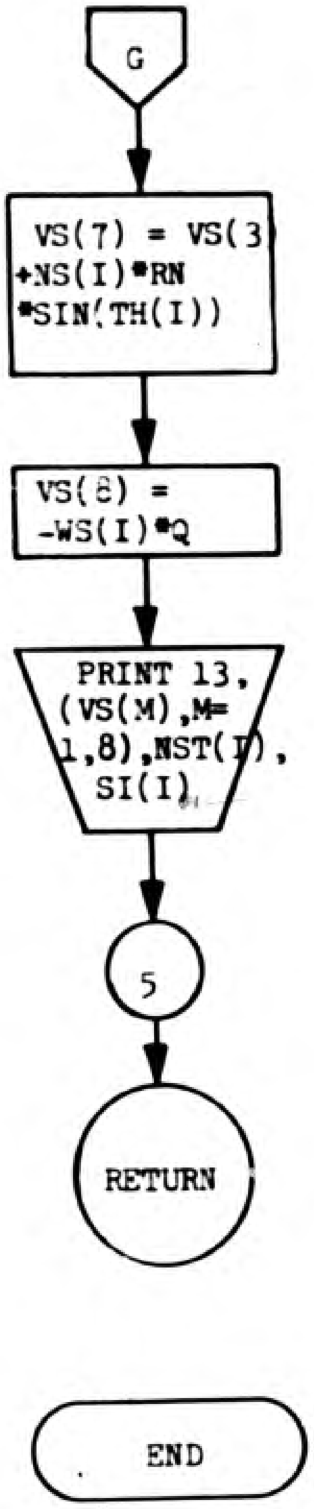
COMPUTE γ AT THE BODY.

COMPUTE Z AT THE BODY.

COMPUTE K AT THE BODY.

COMPUTE θ.

COMPUTE γ AT THE SHOCK.



COMPUTE Z AT THE SHOCK.

COMPUTE SHOCK VELOCITY.

PRINT SHOCK AND BODY PARAMETERS.

RETURN.

END OF SUBROUTINE DMPOUT.

D. LISTING

```
      SUBROUTINE DMPOUT
C
C      THIS ROUTINE PRINTS THE FLOW FIELD DATA.
C
      COMMON /1/ V(15,40,4),D(40,4),P(3,40)
      COMMON /2/ RB(15),ZB(15),TH(15),CTH(15),K(15),NS(15),WS(15)
      COMMON /3/ MAXI,MI,IMIN,MAXN,MN,NMIN
      COMMON /4/ IPRINT,NPR,KGEOM
      COMMON /5/ DXI,DET,DX2,DE2,DT
      COMMON /6/ Q1,VS1,HT1,A1,R1
      COMMON /9/ ISEG,XI0,DELS
      COMMON /10/ VS(10),DS,DN,C,CSC,ZZ
      COMMON /11/ MAXIT,ISTOP,ITD,TIME,SI(15),NST(15)
      COMMON /INPUT/ GAM1,AMACH1,THETC,ZRN,ICRIT(2),P1,RHO1,RN,AMUO,T1,F
     *PR(2),IPR(6),INTAP,OTAP,NCPT,NCASE,NFILE,NSEG,IGEOM,NGEOM,NTAP
      COMMON /CONST/ PI,PI2,GAMA,GAMB
      COMMON/PRES/PT2,P00
      DIMENSION PREFIX(4), DAT(4)
      DATA (PREFIX(M),M=1,4)/5H   RB=,6H    ZB=,5HTHETA=,6H     K=/
      REAL NS,K,NST
      I=IPRINT+ITD
      PRINT 6, ISEG,I,TIME
      IF (ISTOP.GT.100) PRINT 7
      NPR=NPR+1
      IF (NPR.LT.7) IPRINT=IPR(NPR)
      I=MAXN+2
      NPAGE=60/I
      IPAGE=0
      C=SQRT(P1/RHO1)
      DS=DXI*RN
      VS(1)=XI0*RN
      DO 4 I=IMIN,MI
      CALL STATE (I,1,MAXN)
      SI(I)=P(1,1)/PT2
      IND=0
      DAT(1)=RB(I)*RN
      DAT(2)=ZB(I)*RN
      DAT(3)=TH(I)*180.0/PI
      DAT(4)=K(I)/RN
      IPAGE=IPAGE+1
      IF (IPAGE-NPAGE) 2,2,1
    1 PRINT 8
      IPAGE=1
    2 VS(1)=VS(1)+DS
      PRINT 9, VS(1)
      NST(I)=2.0*(P(1,1)-1.0)/(Q1*Q1)
      DN=-NS(I)*DET*RN
      VS(2)=-DN
      DO 4 N=1,MAXN
      IND=IND+1
      QSQ=V(I,N,2)*V(I,N,2)+V(I,N,3)*V(I,N,3)
      ZZ=HSP(P(1,N),V(I,N,1))+0.5*QSQ
      VS(2)=VS(2)*RN
      VS(3)=V(I,N,1)*RHO1
      VS(4)=V(I,N,2)*Q
      VS(5)=-V(I,N,3)*Q
      VS(6)=V(I,N,4)
      VS(7)=P(I,N)*P1
      VS(8)=SQRT(QSQ)/VSOUND(P(I,N),V(I,N,1))
      VS(9)=TEMP(VS(7),VS(3))
      VS(10)=ABS(ZZ/HT1-1.0)
      IF (IND.GT.4) GO TO 3
      PRINT 10, PREFIX(IND),DAT(IND),(VS(M),M=2,10)
      GO TO 4
3     PRINT 11, (VS(M),M=2,10)
4     CONTINUE
      IF (IPAGE*(MAXN+2)+MAXT.GT.53) PRINT 8
      PRINT 12
      VS(1)=XI0*RN
      DO 5 I=IMIN,MI
      VS(1)=VS(1)+DS
      VS(2)=RR(I)*RN
      VS(3)=ZZ(I)*RN
      VS(4)=K(I)/RN
      VS(5)=TH(I)*180.0/PI
      VS(6)=VS(2)-NS(I)*RN*CTH(I)
      VS(7)=VS(3)+NS(I)*RN*SIN(TH(I))
      VS(8)=-WS(I)*Q
5     PRINT 13, (VS(M),M=1,8),NST(I),ST(I)
      RETURN
C
6     FORMAT (1H1,20X,14HSEGMENT NUMBER,I3,10X,16HITERATION NUMBER,I5,10
     *X,5HTIME=,E12.4)
7     FORMAT (30X,32HEXCEEDED MAXIMUM NUMBER OF NODES IN NORMAL DIRECTIO
     *N/30X,23HMAXIMUM NUMBER WAS USED)
8     FORMAT (1H1)
9     FORMAT (/4X,2HS=,F8.2,7X,1HN,11X,3HRHO,9X,1HU,11X,1HV,8X,3H(S-S1)/
     *3V,6X,1HP,11X,1HM,11X,1HT,8X,5HDH/H1)
10    FORMAT (1X,A6,F7.2,1X,9E12.4,F8.5)
11    FORMAT (10X,9E12.4,F8.5)
12    FORMAT (///47X,25HSHOCK AND BODY PARAMETERS//6X,1HS,11X,2HRB,10X,
     *2HZB,10X,1HK,9X,5HTHETA,7X,2HRS,10X,2HZS,10X,2HWS,10X,2HCP,9X,
     *4HD/D0)
13    FORMAT (10F12.4)
      END
```

NOT REPRODUCIBLE

9. SUBROUTINE GEOM

A. PURPOSE

This subroutine computes the body geometry parameters and the initial guess for the shock shape. An option for generating these parameters for a sphere-cone vehicle is available. Currently, other body shapes must be specified with data cards. These geometry data cards are read in this routine. A computed GO TO statement selects the proper option based on the value of a variable KGEOM. Explicit computations for body shapes other than sphere-cones could be incorporated by using additional values of this variable (currently, values of 1 to 3 are used). When the body geometry is read from data cards, several options are available. The data can be read in at specified equal intervals along the body or at a series of representative points. In the latter case, curve fit routines accomplish the necessary integrations and interpolations to specify the data at equal intervals along the body. The initial shock shape can be input or computed using empirical relations. The number of nodes to be used along the body and across the shock layer can be specified for each segment. These specifications will be ignored if these quantities were defined in the input read in SUBROUTINE DAιA. The body curvature can be changed between segments at the common segment boundary. If no specification for the number of nodes to be used along the body is provided, the number of data cards provided is used as the value for this quantity. An error check is accomplished to ensure that the number of points provided for a segment has not exceeded the dimension sizes. The segment number for which a current set of data cards has been provided is checked with the number of the segment under consideration. Data are skipped until the correct segment number is located. This allows the complete geometry data packet to be included in a restart. even though the first few segments will not be required. If a segment number greater than the value of the segment under consideration is used, it is considered as a data error. All data errors result in an error message and termination of the current case. An subsequent case will then be treated.

B. VARIABLE LIST

All variables previously defined in the description of the main program have been omitted from this list.

A - array used for body shape computations under the arbitrary body shape option.

ANS - variable used for temporary storage of the value of n_s at the end of a segment previously completed.

BLANK - alphanumeric blank under A10 format. Used to check for discontinuities in body curvature.

C - array used to store cubic fit coefficients for each data point under the arbitrary body shape option.

DNS - change in n_s between body stations.

DR - change in r at the body between body stations for a sphere-cone.

DUM - dummy variable used for intermediate computations.

DZ - change in Z between body stations for a sphere-cone vehicle.

IC - body station index at which integration with respect to r should be replaced with integration with respect to Z.

IP - dummy index used for various applications.

IST - index of the downstream boundary body station for the previous segment.

KSH - control variable specifying whether the initial shock shape has been specified on the data cards.

M - dummy index used for various applications.

NPTS - number of data cards to be read for the current segment.

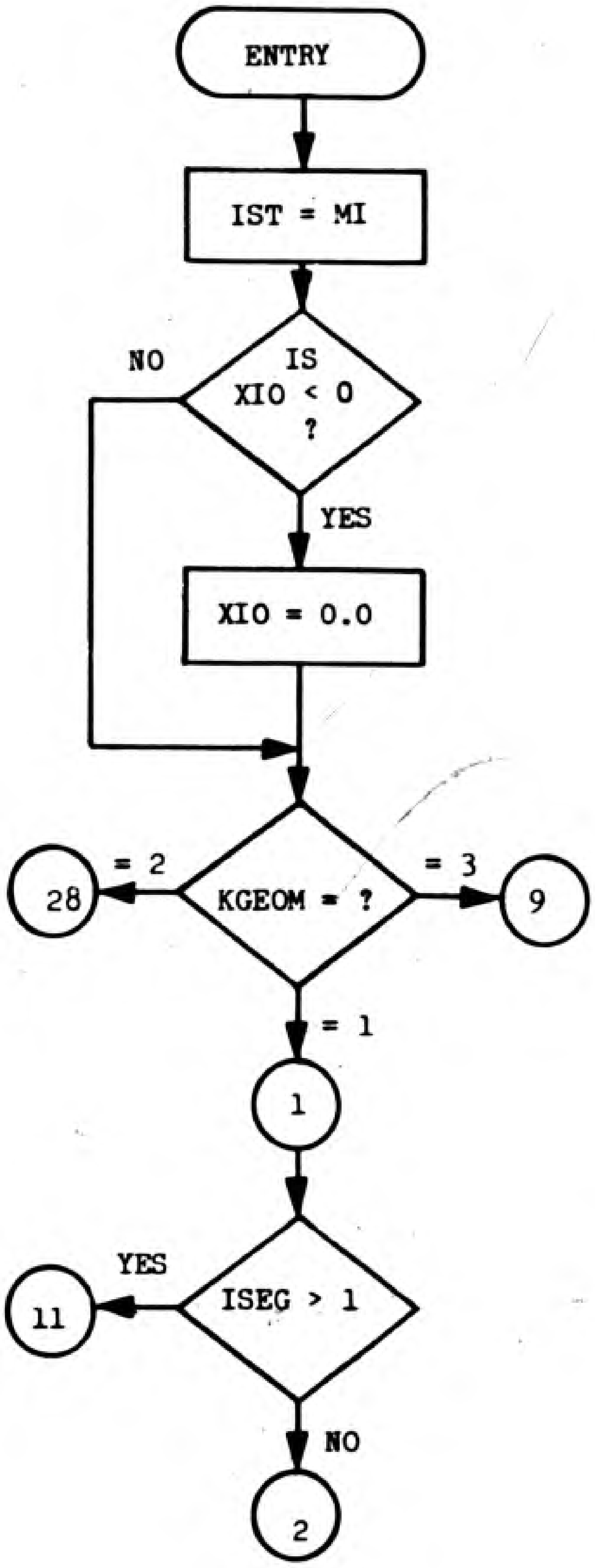
NST - array used for temporary storage of an initial shock shape specification as provided on the data cards.

SI - array used for body shape computations under the arbitrary body shape option.

SUM - result of a numerical integration.

ZZ - dummy variable used for intermeaiate computations.

C. FLOW CHART



ENTER SUBROUTINE GEOM.

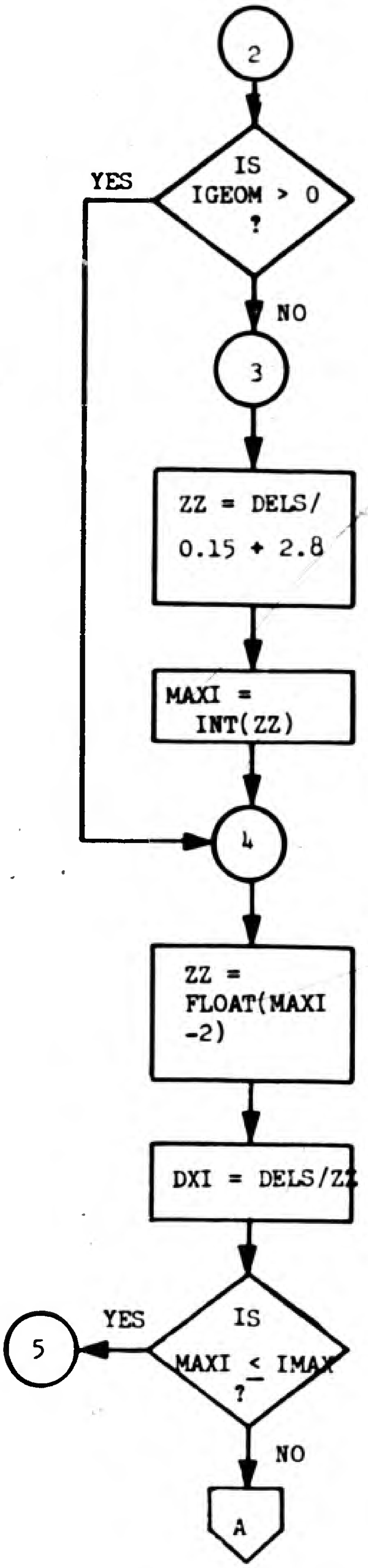
SAVE INDEX OF THE DOWNSTREAM BOUNDARY OF THE PREVIOUS SEGMENT.

IS THE VALUE OF ξ AT UPSTREAM BOUNDARY OF THE LAST SEGMENT NEGATIVE ?

WHAT GEOMETRY OPTION ?

SPHERE-CONE; SPHERE TREATED WITH A SINGLE SEGMENT.

HAS THE FIRST SEGMENT BEEN COMPLETED ?

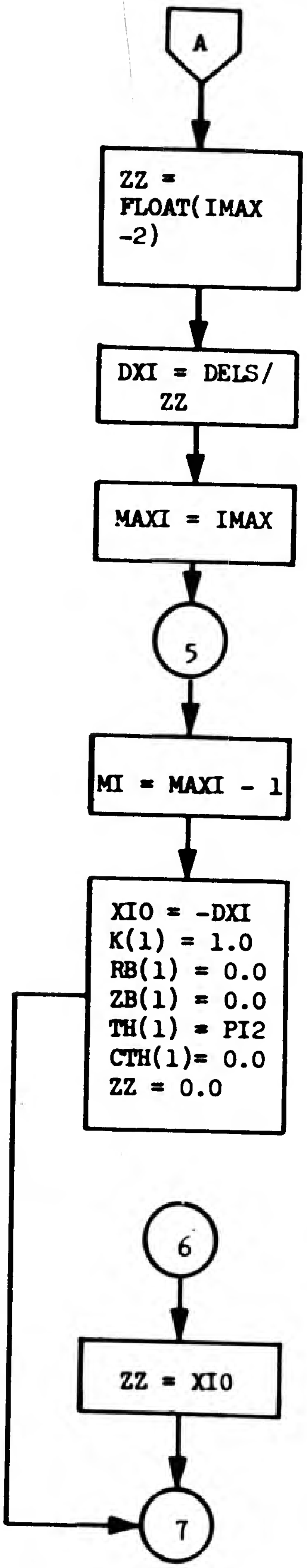


WAS THE NUMBER OF NODES ALONG THE BODY PER SEGMENT SPECIFIED IN THE CASE DATA ?

COMPUTE MAXI.

COMPUTE Δξ.

ARE DIMENSIONED VARIABLES LARGE ENOUGH FOR THE VALUE OF MAXI ?



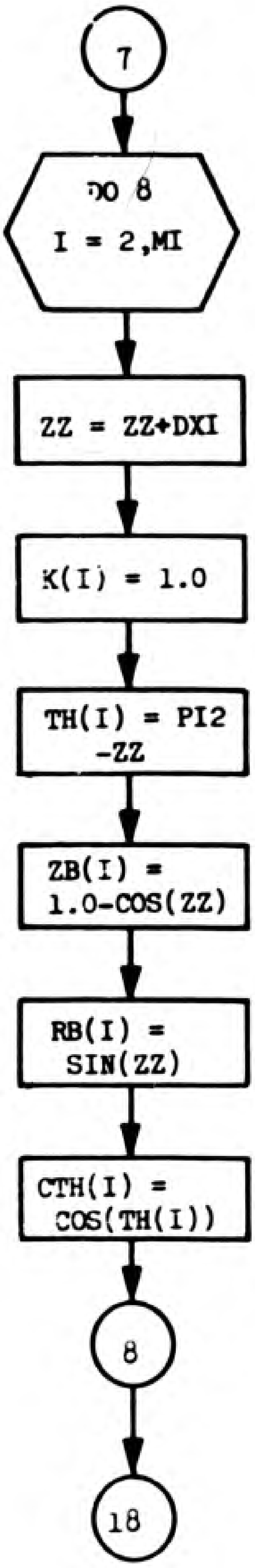
COMPUTE MINIMUM ALLOWABLE VALUE OF $\Delta\xi$.

SET MAXI TO MAXIMUM VALUE.

COMPUTE INDEX OF DOWNSTREAM BOUNDARY NODE.

INITIALIZE BODY PARAMETERS FOR STAGNATION POINT.

INITIALIZE ξ FOR SEGMENTS OTHER THAN THE FIRST.



START DO LOOP TO COMPUTE BODY GEOMETRY FOR A SPHERE.

INCREMENT ξ.

CURVATURE.

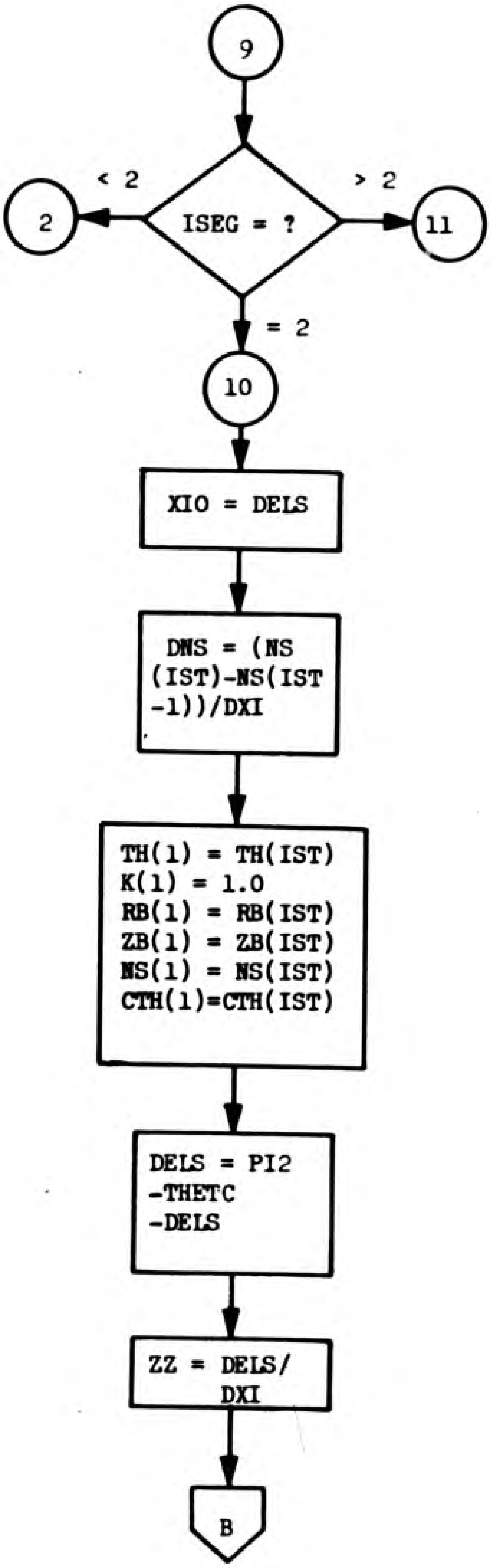
θ.

Z_b.

r_b.

$\cos(\theta)$.

END OF LOOP.



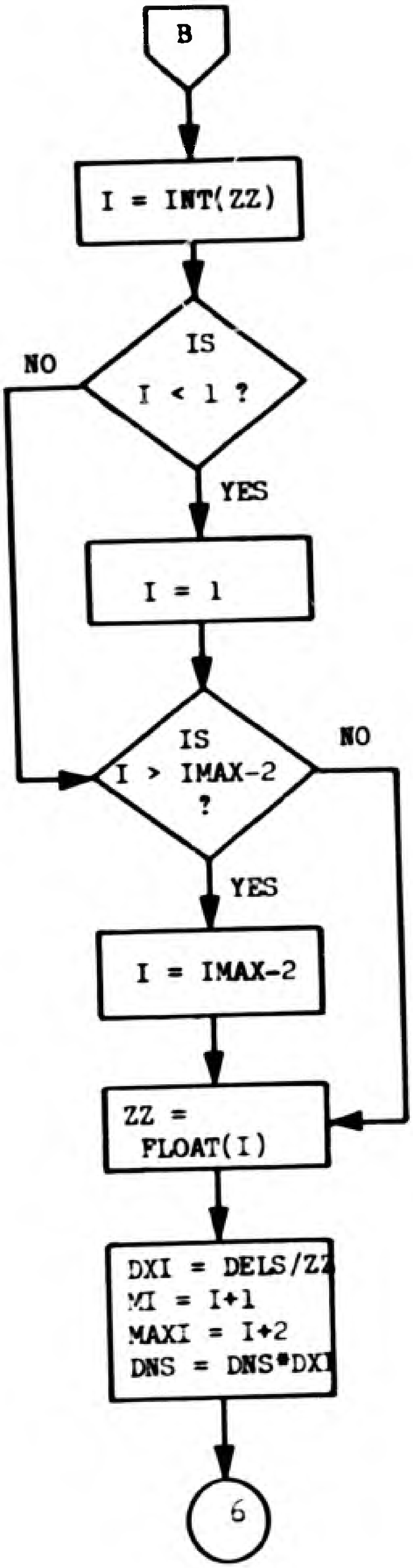
SPHERE-CONE VEHICLE WITH SPHERE TREATED WITH 2 SEGMENTS.

INITIALIZE XIO FOR SEGMENT 2.

SAVE SHOCK SLOPE AS COMPUTED AT THE DOWNSTREAM BOUNDARY OF SEGMENT 1.

INITIALIZE BODY GEOMETRY PARAMETERS FOR UPSTREAM BOUNDARY OF SEGMENT 2.

COMPUTE SEGMENT STREAMWISE LENGTH.

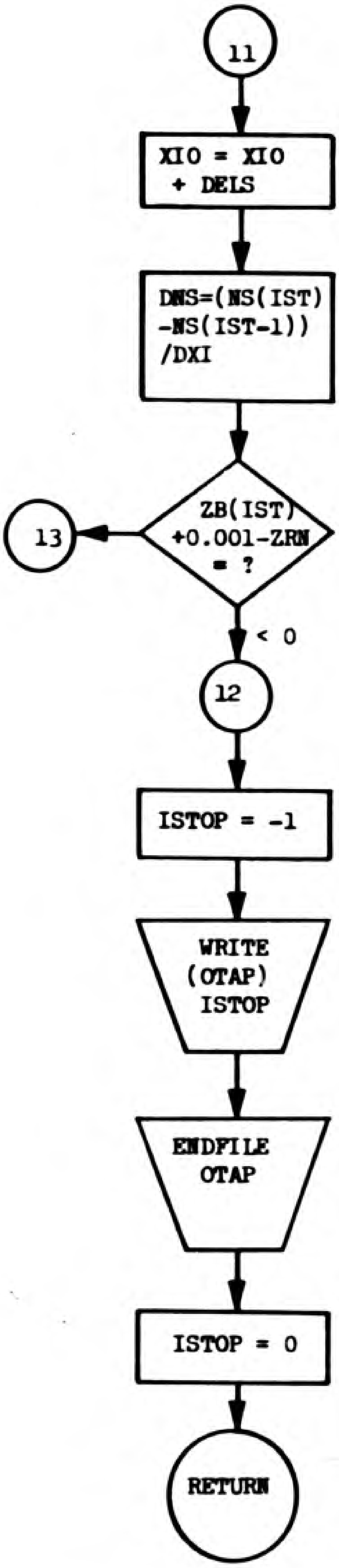


COMPUTE THE NUMBER OF NODES REQUIRED TO GIVE THE SAME VALUE OF Δξ AS USED FOR SEGMENT 1.

EXCEEDED MAXIMUM NUMBER OF NODES ALLOWED ?

USE MAXIMUM NUMBER.

COMPUTE Δξ, MI, MAXI AND THE CHANGE IN NS PER EACH Δξ BASED ON LINEAR EXTRAPOLATION.



CONICAL AFTERBODY.

INCREMENT XIO.

COMPUTE SHOCK SLOPE FOR UPSTREAM STARTING LINE.

HAS THE INPUT AXIAL LENGTH BEEN EXCEEDED ON PREVIOUS SEGMENT ?

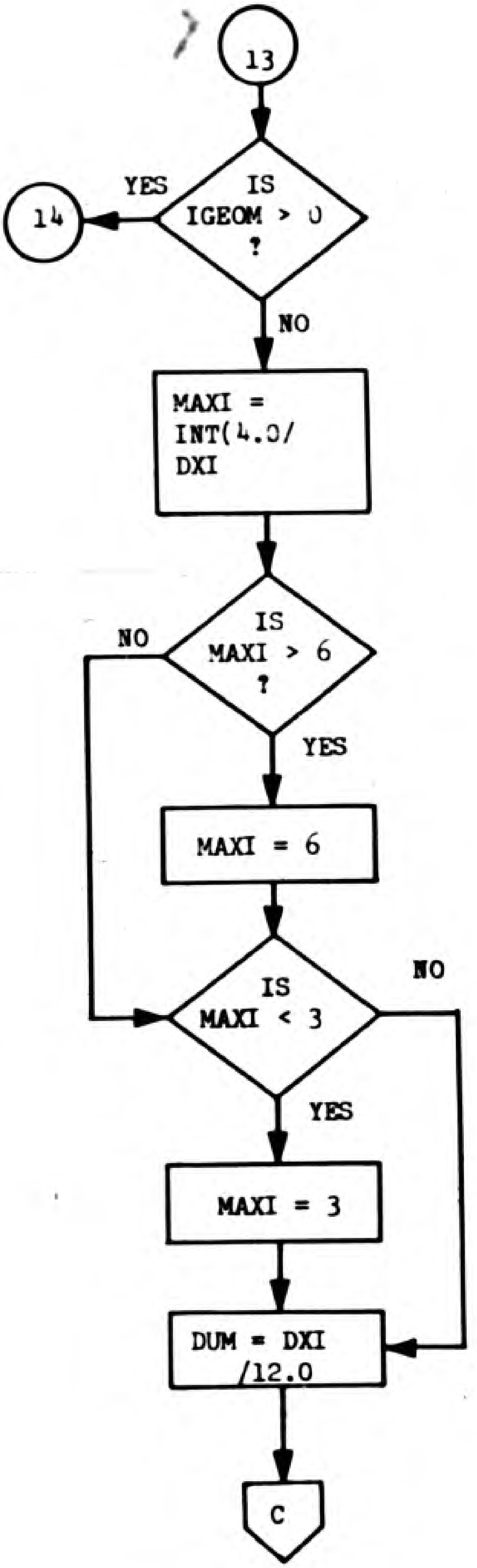
CASE COMPLETED.

WRITE INDICATOR ON OUTPUT TAPE UNIT TO SIGNIFY END OF CASE.

ENDFILE OUTPUT TAPE UNIT.

SET ERROR INDICATOR TO TERMINATE CASE (NO ERROR).

RETURN.



CASE INCOMPLETE.

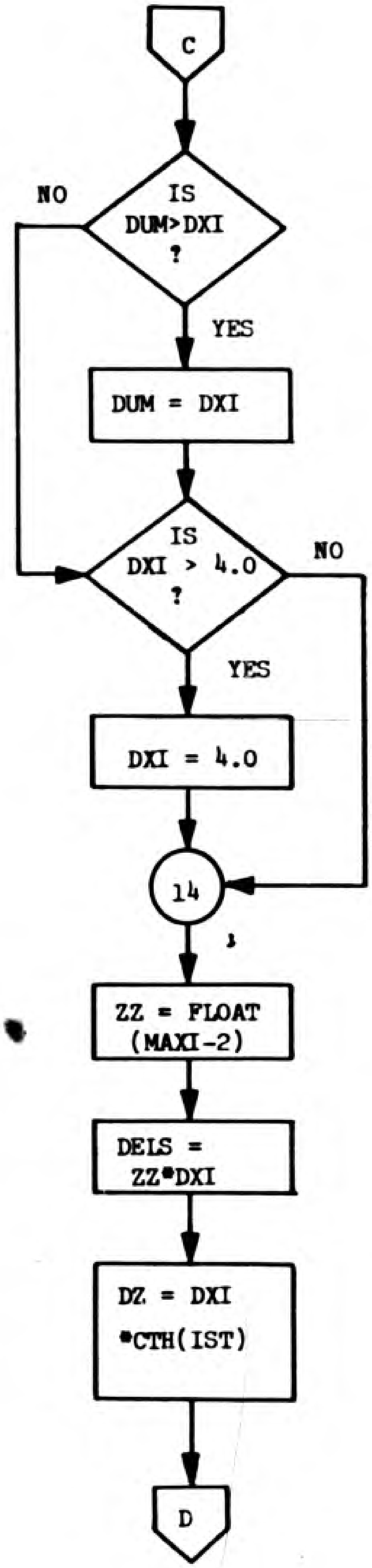
WAS MAXI SPECIFIED IN THE CASE DATA ?

ESTIMATE THE OPTIMUM NUMBER OF NODES FOR THIS SEGMENT.

USE MAXIMUM OF 6 NODES.

MUST HAVE AT LEAST 3 NODES.

ESTIMATE OPTIMUM VALUE OF Δξ.

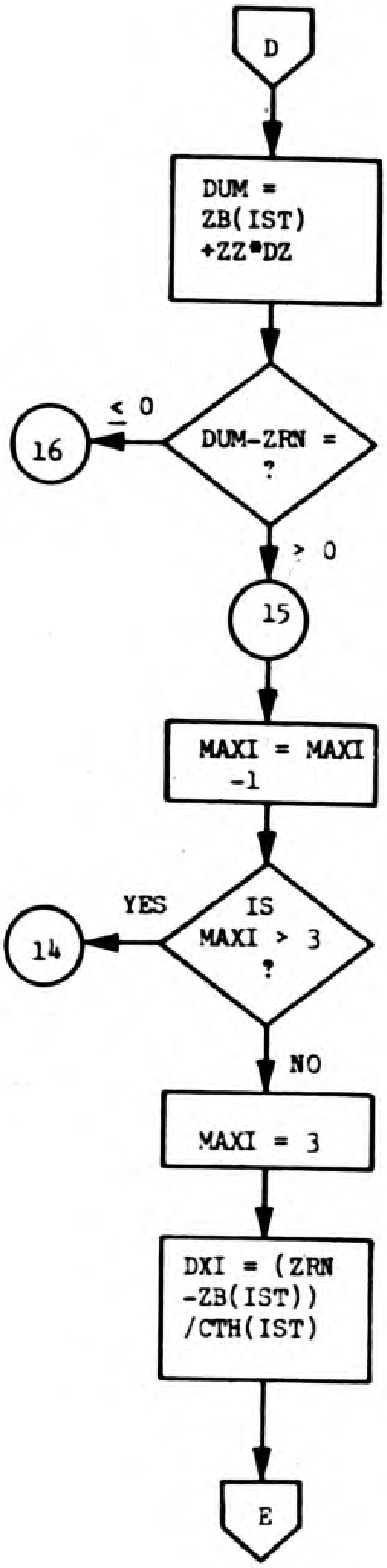


Δξ ≥ PREVIOUS VALUE

Δξ ≤ 4.0.

COMPUTE SEGMENT LENGTH.

ΔZ = Δξ cos θ.



COMPUTE Z/RN AT THE END OF THE CURRENT SEGMENT.

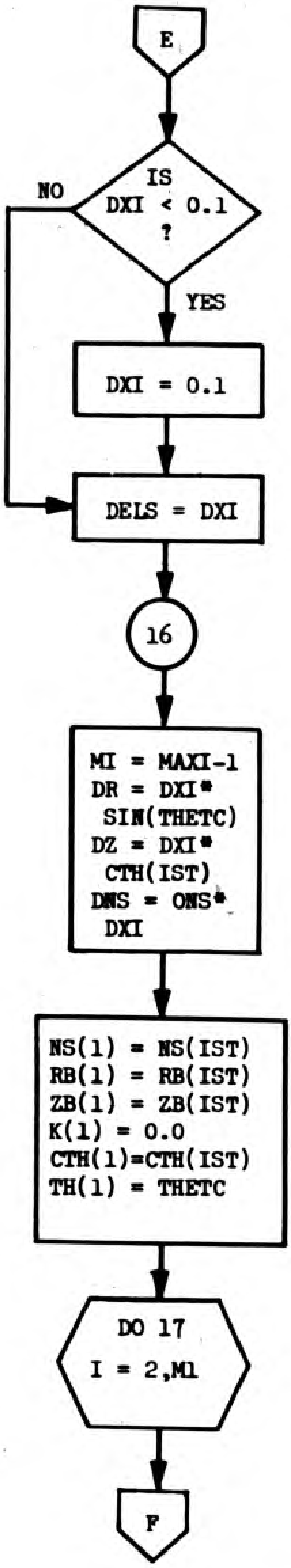
EXCEEDED SPECIFIED VEHICLE LENGTH ?

REDUCE MAXI.

ALLOWABLE VALUE ?

USE 3 NODES TO FINISH CASE.

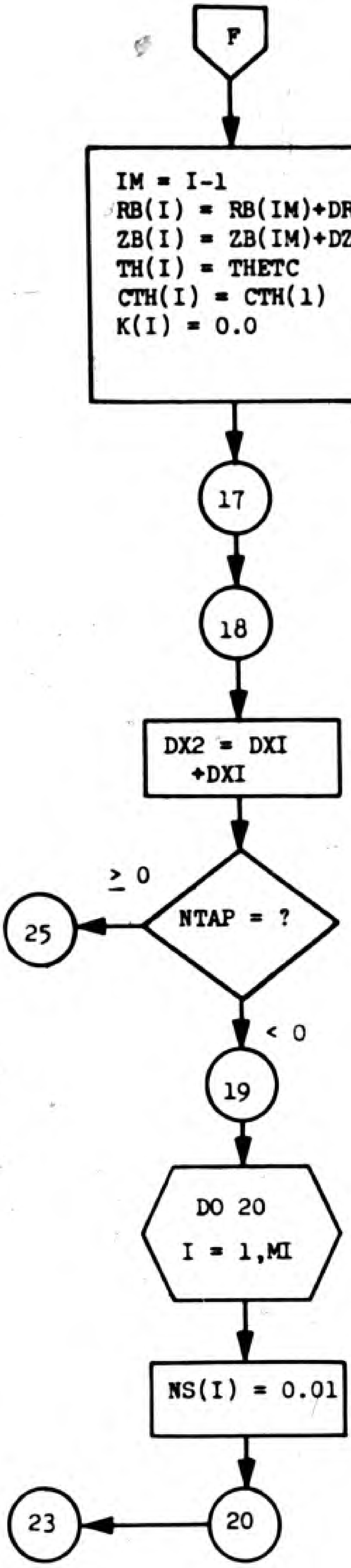
COMPUTE Δξ REQUIRED TO COMPLETE CASE.



REQUIRE $\Delta\xi \geq 0.1$.

COMPUTE INDEX OF NODE AT DOWNSTREAM BOUNDARY; Δr, ΔZ AND Δn, BETWEEN NODES.

ASSIGN BODY GEOMETRY PARAMETERS AT UPSTREAM STARTING LINE.



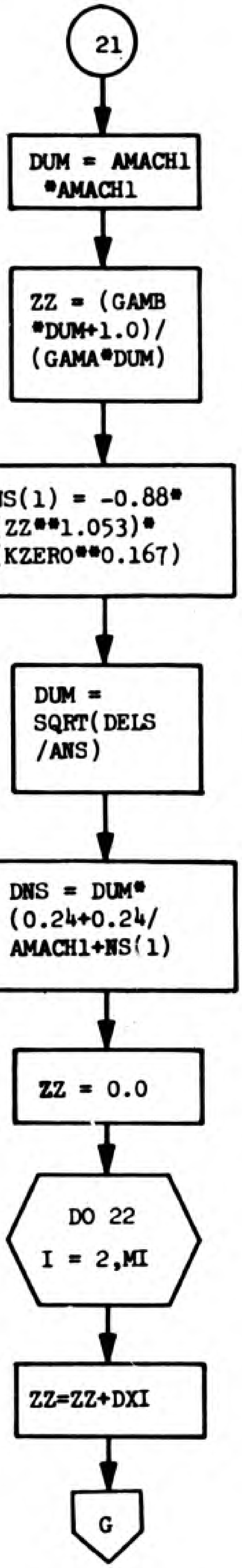
COMPUTE BODY GEOMETRY PARAMETERS AT REMAINING POINTS.

START SHOCK GEOMETRY COMPUTATIONS (ALL OPTIONS).

COMPUTE $2\Delta\xi$.

UNSTEADY FLOW OPTION.

SHOCK SPECIFIED AS A CONSTANT DISTANCE FROM THE BODY OF 0.01.


21
DUM = AMACH1 *AMACH1
ZZ = (GAMB *DUM+1.0)/ (GAMA*DUM)
NS(1) = -0.88* (ZZ**1.053)* (KZERO**0.167)
DUM = SQRT(DELS /ANS)
DNS = DUM* (0.24+0.24/ AMACH1+NS(1)
ZZ = 0.0
DO 22 I = 2,MI
ZZ=ZZ+DXI
G

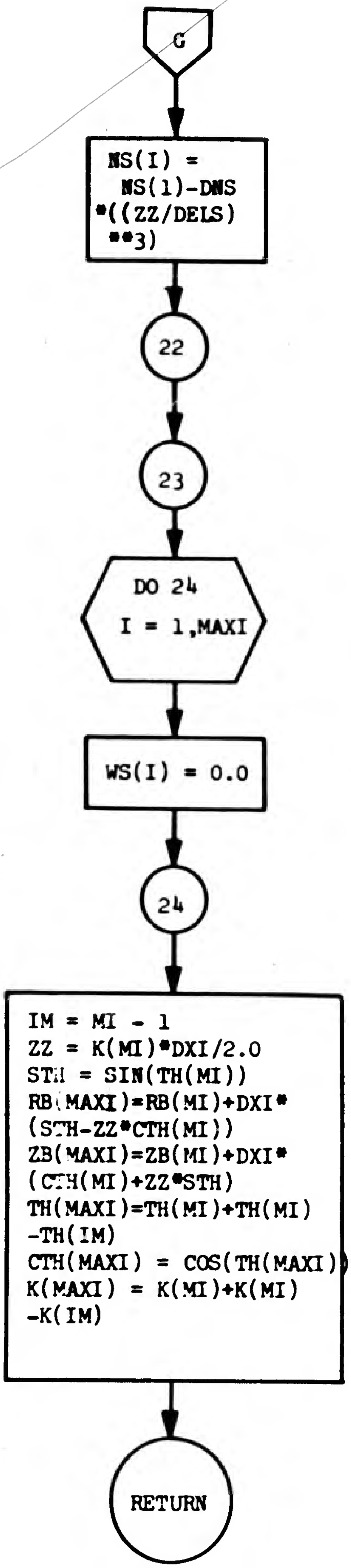
ESTIMATE SHOCK SHAPE FOR FIRST SEGMENT, STEADY STATE OPTION.

ESTIMATE SHOCK STANDOFF DISTANCE.

ESTIMATE Δn_s BETWEEN NODES.

INITIALIZE ξ.

INCREMENT ξ.



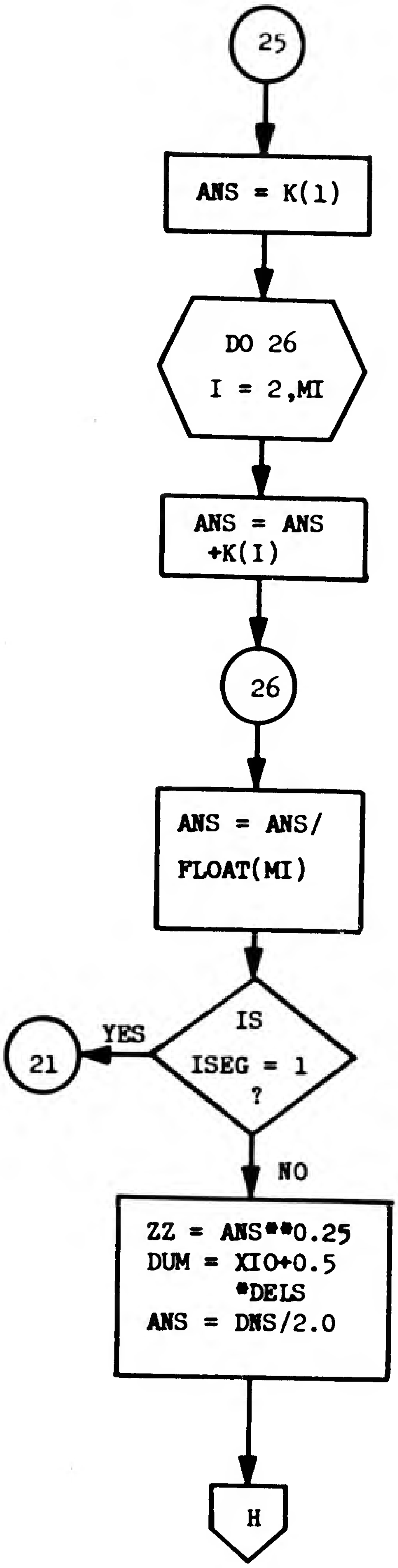
ESTIMATE SHOCK SHAPE
FOR SEGMENT 1.

FINAL SERIES OF
COMPUTATIONS FOR ALL OPTIONS.

SET SHOCK VELOCITIES
TO ZERO.

EXTEND BODY GEOMETRY PARAMETERS
TO ADDITIONAL DOWNSTREAM NODE.

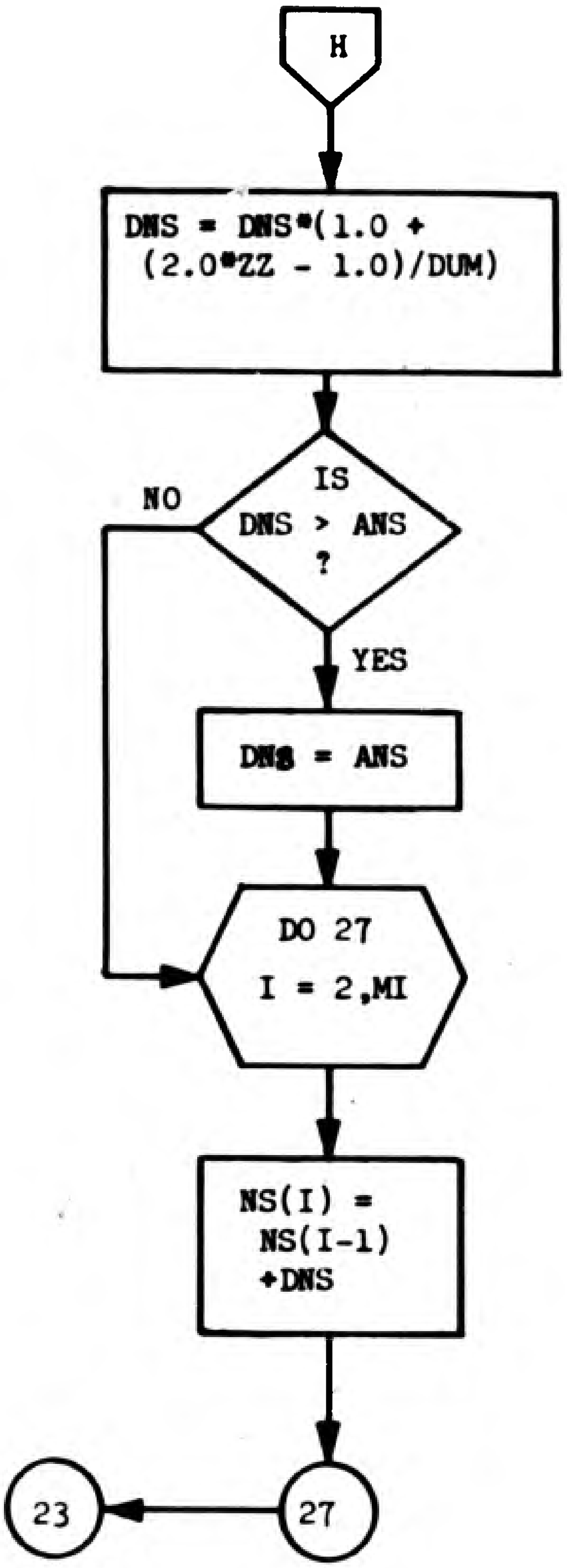
RETURN.



ENTRY POINT FOR SHOCK COMPUTATIONS WHEN STEADY STATE INITIAL SHOCK SHAPE IS TO BE ESTIMATED.

COMPUTE AVERAGE BODY CURVATURE OF THE SEGMENT.

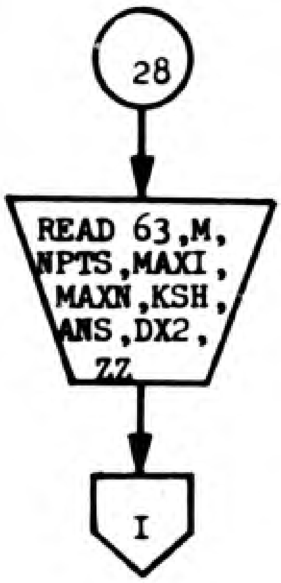
FIRST SEGMENT ?



ESTIMATE Δn_s BETWEEN NODES.

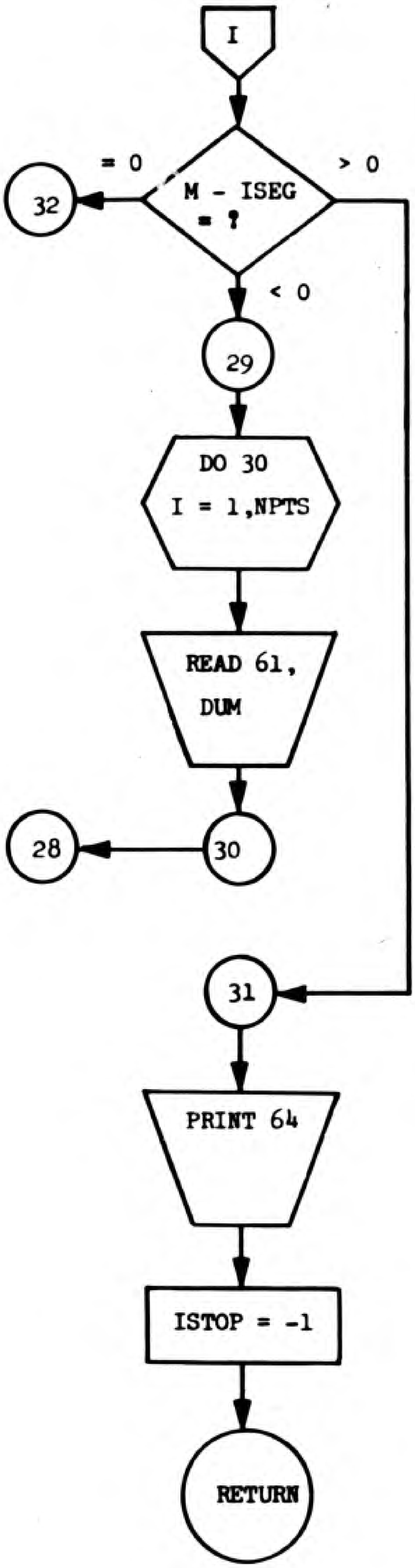
IS ESTIMATED VALUE SMALLER IN MAGNITUDE THAN HALF OF THE VALUE PREDICTED BY THE SHOCK SLOPE ?

ESTIMATE SHOCK SHAPE FOR OTHER POINTS.



ARBITRARY BODY SHAPE OPTION.

READ FIRST DATA CARD FOR THIS SEGMENT.



CORRECT SEGMENT NUMBER ?

SEGMENT NUMBER READ, M,
WAS LESS THAN THE CURRENT
SEGMENT NUMBER.

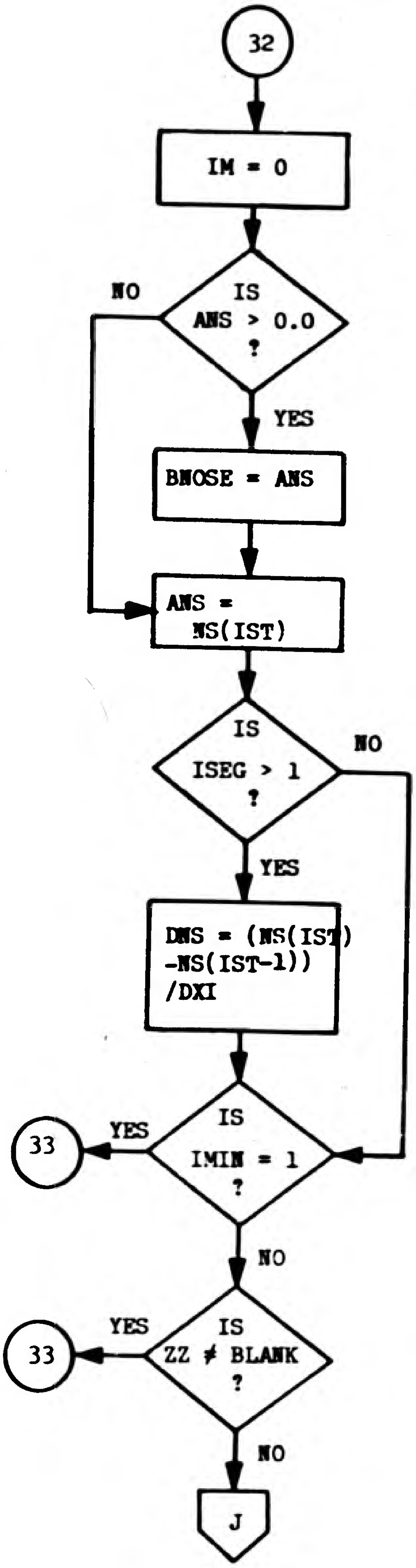
CLEAR UNNEEDED DATA CARDS
AND RETURN TO 28 TO READ
NEXT SET OF DATA CARDS.

FATAL ERROR. SEGMENT NUMBER
READ WAS GREATER THAN THE
CURRENT SEGMENT NUMBER.

PRINT ERROR MESSAGE.

SET ERROR INDICATOR.

RETURN.



CORRECT SEGMENT NUMBER READ.

SET IM TO READ GEOMETRY DATA FOR I = 1, MI.

WAS A NOSE BLUNTNESS PARAMETER PROVIDED ?

LOAD NOSE BLUNTNESS PARAMETER.

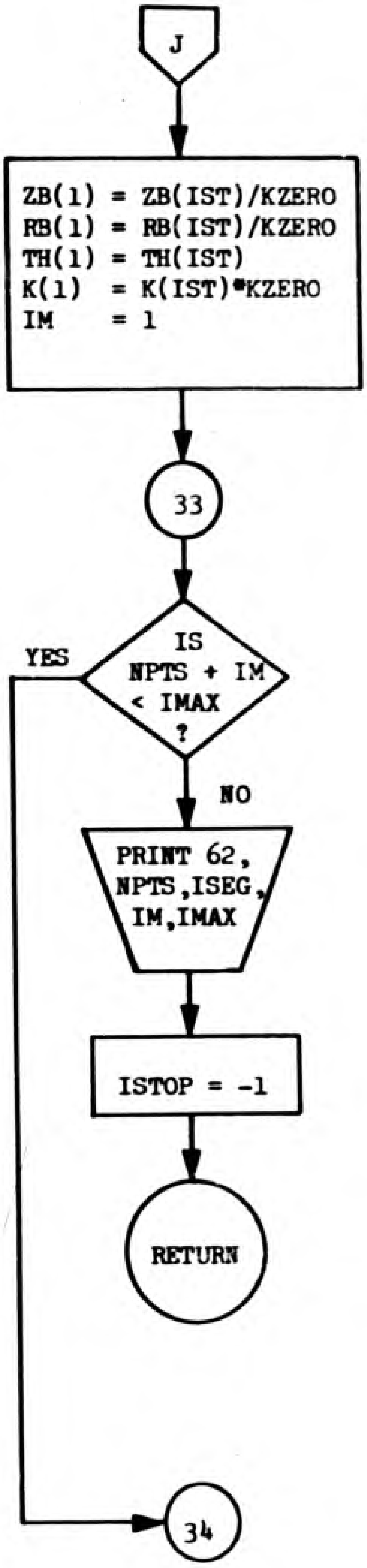
SAVE SHOCK LOCATION FOR UPSTREAM STARTING LINE.

SEGMENT NUMBER ≠ 1.

SAVE SHOCK SLOPE AT DOWNSTREAM BOUNDARY OF PREVIOUS SEGMENT.

NEED DATA FOR I = 1 ?

CHANGE CURVATURE AT I = 1 ?



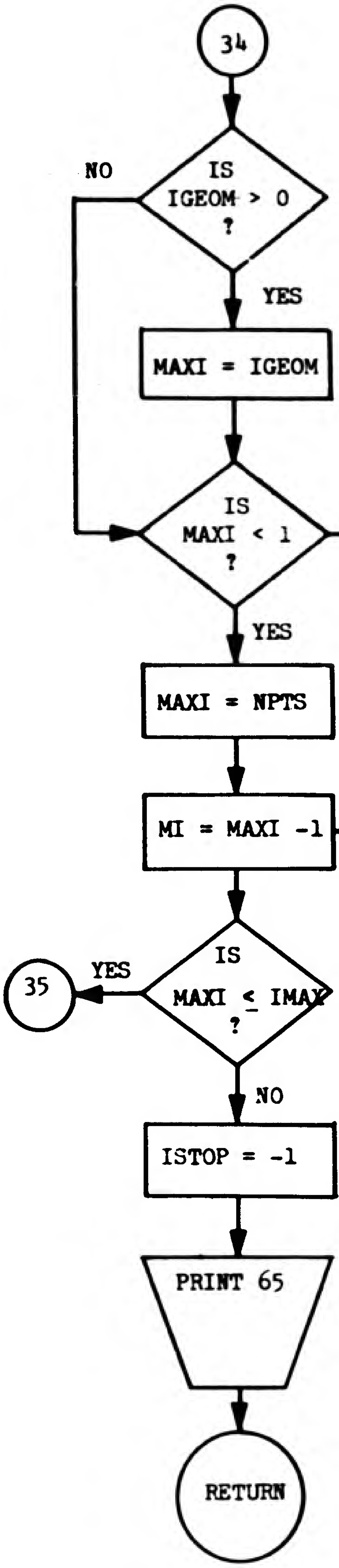
ASSIGN BODY GEOMETRY PARAMETERS FROM DOWNSTREAM BOUNDARY OF THE PREVIOUS SEGMENT TO THE UPSTREAM BOUNDARY OF THE CURRENT SEGMENT, SET IM TO READ DATA FOR I = 2, MI.

ARE DIMENSIONED VARIABLES LARGE ENOUGH TO HOLD ALL THE POINTS REQUIRED ?

PRINT ERROR MESSAGE.

SET ERROR INDICATOR.

RETURN.



WAS MAXI SPECIFIED IN CASE DATA ?

USE VALUE PROVIDED IN CASE DATA.

IS MAXI UNSPECIFIED ?

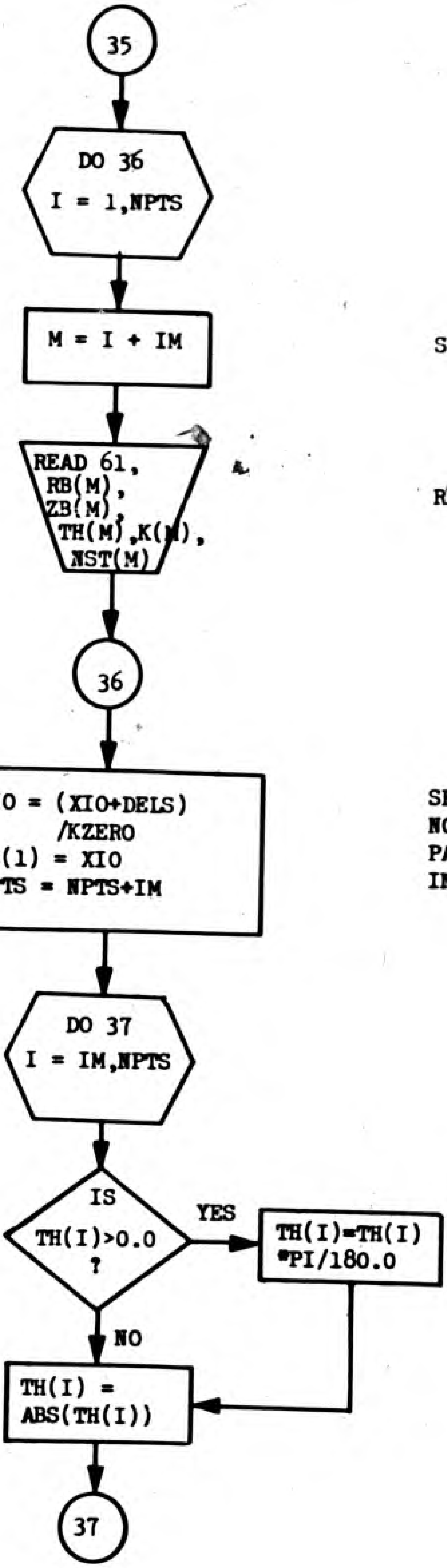
SET MAXI = NPTS.

DIMENSIONS LARGE ENOUGH FOR THIS VALUE OF IMAX ?

SET ERROR INDICATOR.

PRINT ERROR MESSAGE.

RETURN.

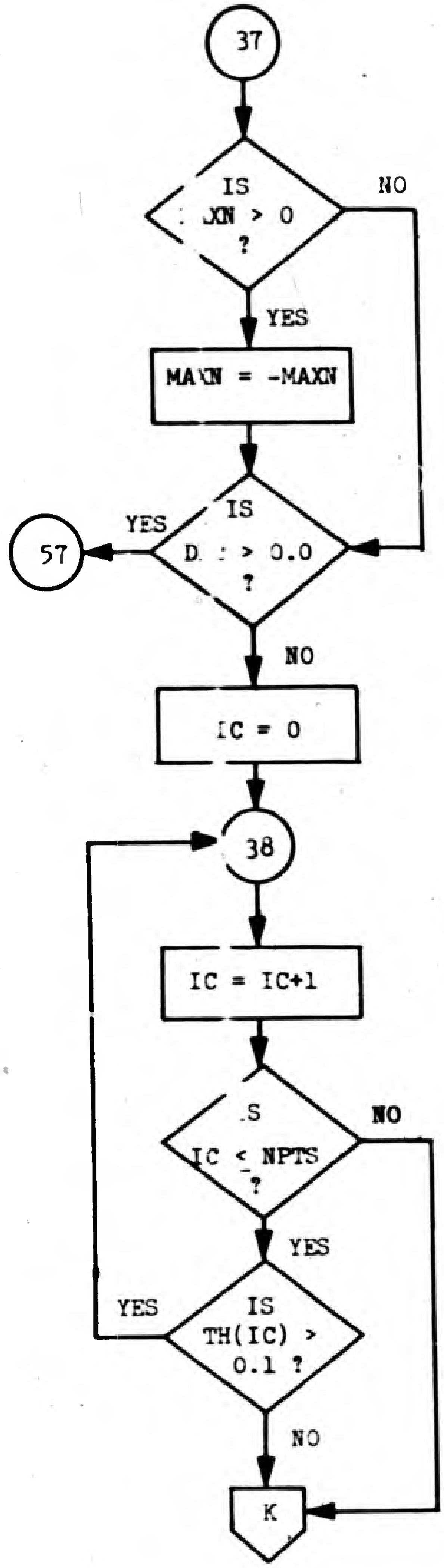


SET INDEX.

READ BODY GEOMETRY DATA.

SET CONSTANTS TO COMPUTE NONDIMENSIONAL GEOMETRY PARAMETERS AT CONSTANT INTERVALS OF ξ.

θ IN DEGREES ?



WAS MAXN SPECIFIED ON
FIRST DATA CARD ?

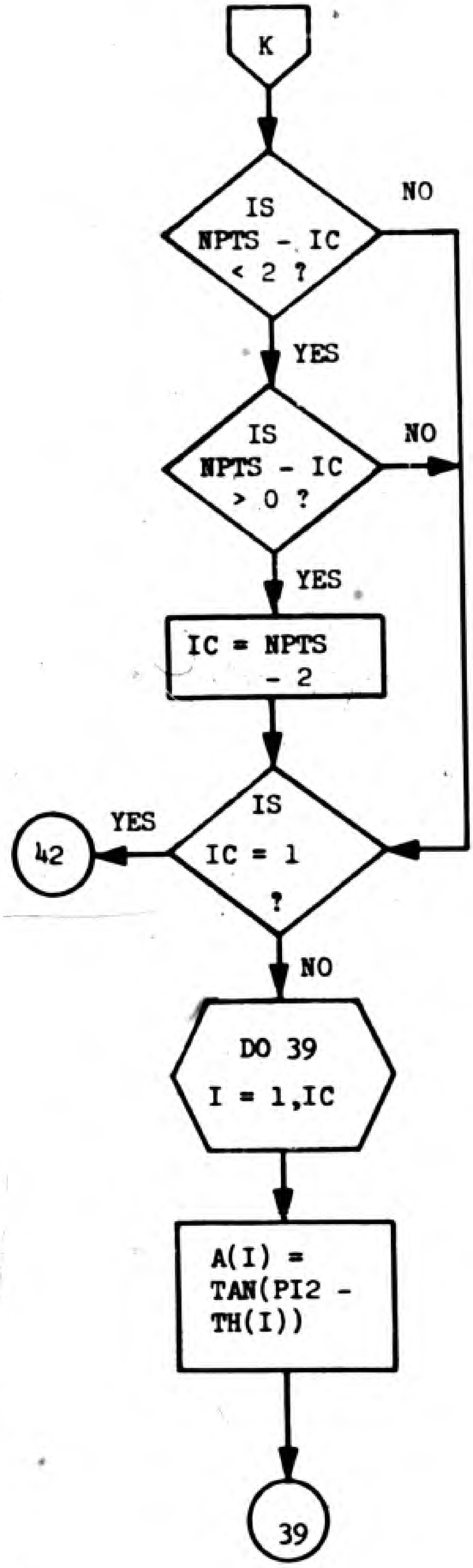
SET MAXN NEGATIVE TO
INDICATE THIS VALUE
SHOULD BE USED.

HAS DATA BEEN PROVIDED
AT EQUAL INTERVALS OF
ξ ?

SET IC TO AVOID
SINGULAR DERIVATIVES.

IS IC WITHIN THE
SPECIFIED NUMBER OF POINTS ?

IS $\theta > 0.1$?



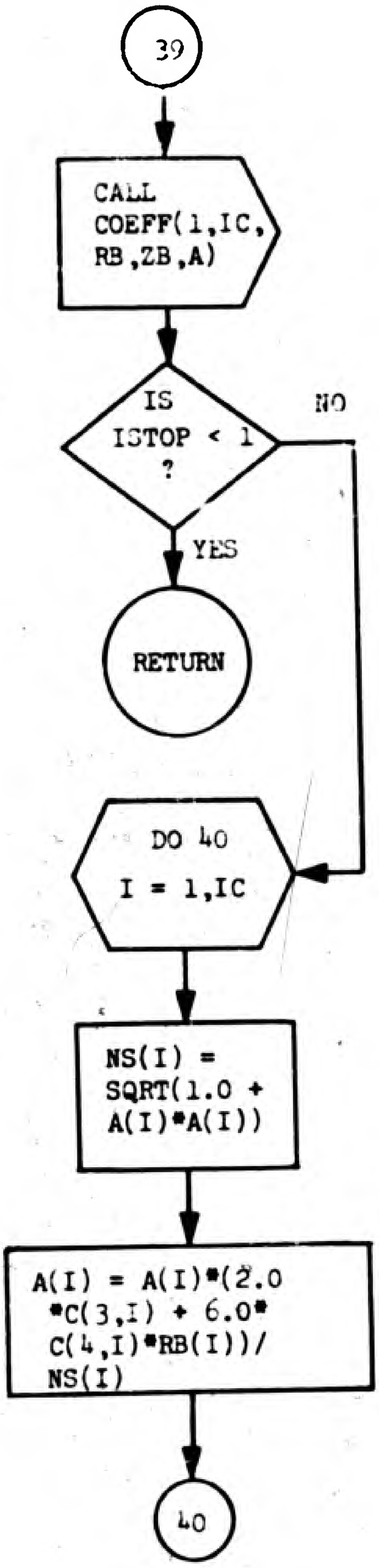
HAVE LESS THAN 2 POINTS BEEN ALLOCATED FOR INTEGRATION WITH RESPECT TO Z ?

HAVE ANY POINTS BEEN ALLOCATED ?

REQUIRE AT LEAST 2 IF ANY WERE USED.

ARE ALL INTEGRATIONS TO BE PERFORMED WITH RESPECT TO Z ?

LOAD tan $(\frac{\pi}{2} - \theta)$


39
CALL COEFF(1,IC, RB,ZB,A)
IS ISTOP < 1 ?
NO
YES
RETURN
DO 40 I = 1,IC
NS(I) = SQRT(1.0 + A(I)*A(I))
A(I) = A(I)*(2.0 *C(3,I) + 6.0* C(4,I)*RB(I))/ NS(I)
40

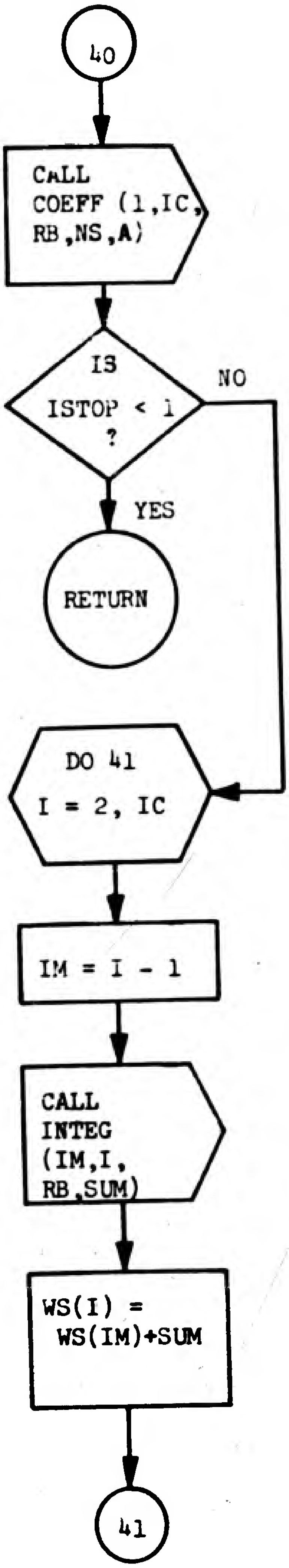
COMPUTE CUBIC COEFFICIENTS FOR INTEGRATION WITH RESPECT TO r.

ERROR IN SUBROUTINE COEFF ?

RETURN.

COMPUTE $\sqrt{1 + (\frac{dz}{dr})^2}$

COMPUTE $\frac{d}{dr} \sqrt{1 + (\frac{dz}{dr})^2}$

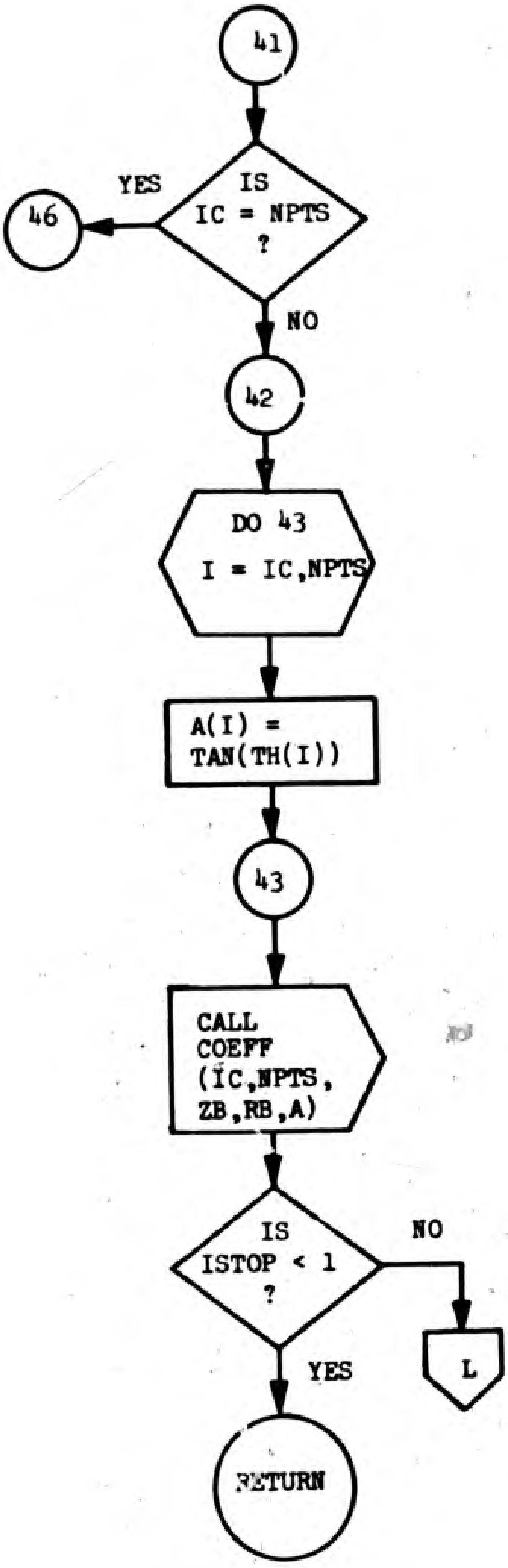


COMPUTE CUBIC COEFFICIENTS
FOR INTEGRATION WITH
RESPECT TO r.

ERROR IN SUBROUTINE COEFF ?

INTEGRATE FOR ξ AT I.

STORE ξ.



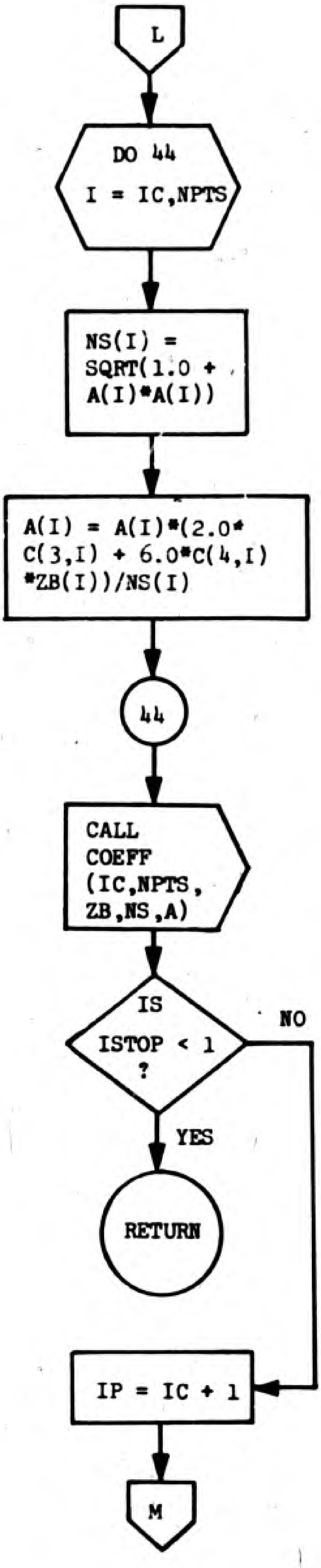
ARE ALL INTEGRATIONS TO BE PERFORMED WITH RESPECT TO r ?

COMPUTE $\frac{dr}{dz}$

COMPUTE COEFFICIENTS FOR A CUBIC FIT OF Z VS r.

ERROR IN SUBROUTINE COEFF ?

RETURN.



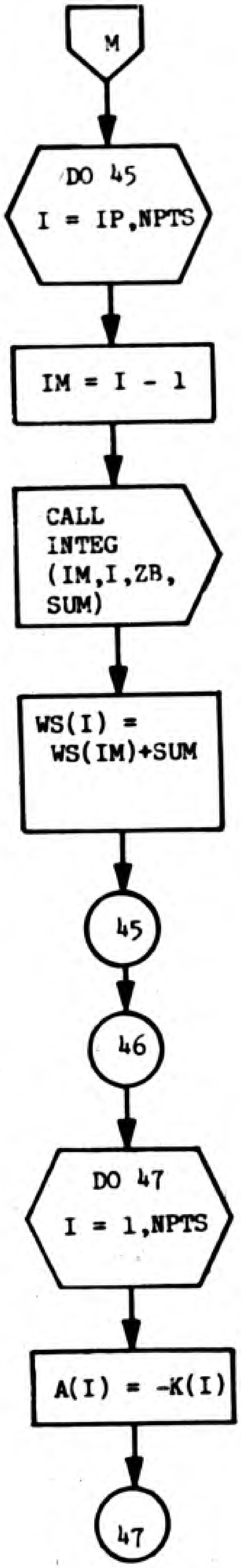
COMPUTE $\sqrt{1 + (\frac{dr}{dz})^2}$

COMPUTE $\frac{d}{dz} \sqrt{1 + (\frac{dr}{dz})^2}$

COMPUTE CUBIC COEFFICIENTS FOR INTEGRATION WITH RESPECT TO Z.

ERROR IN SUBROUTINE COEFF ?

RETURN.

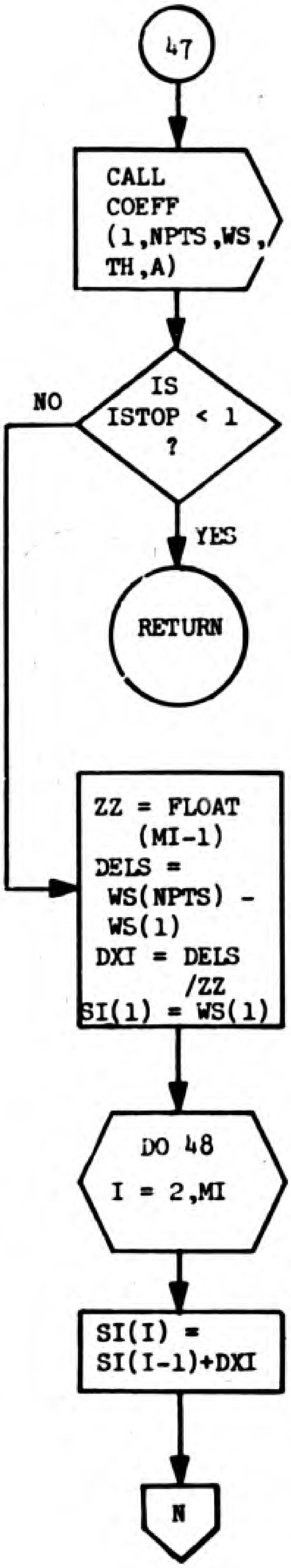


INTEGRATE FOR ξ AT I.

STORE ξ.

START INTERPOLATION.

LOAD $\frac{d\theta}{d\xi}$.

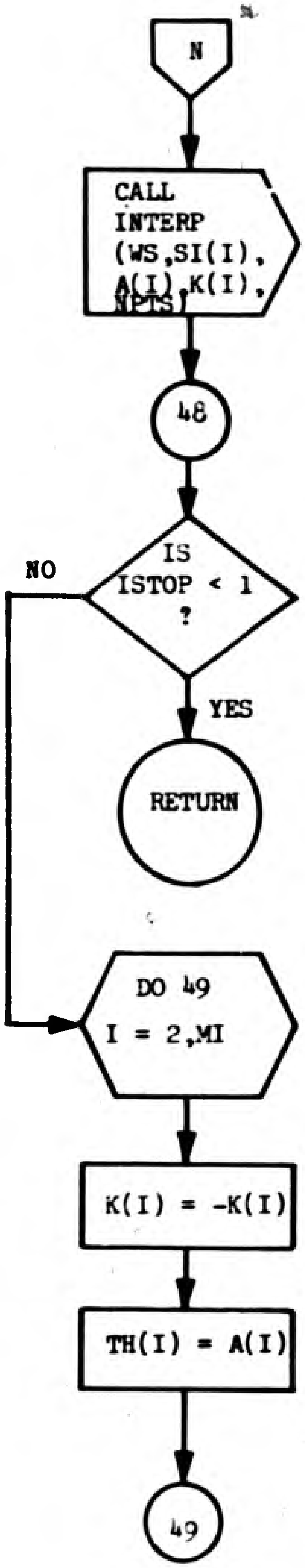


COMPUTE CUBIC COEFFICIENTS
FOR θ VS ξ.

ERROR IN SUBROUTINE COEFF ?

RETURN.

COMPUTE DELS AND Δξ.



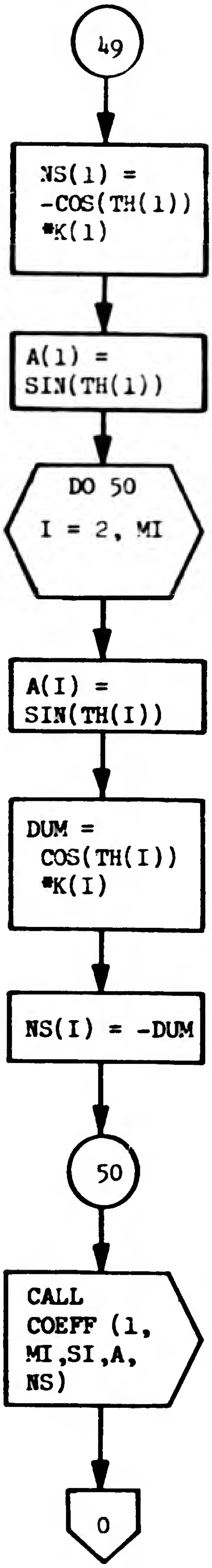
INTERPOLATE FOR θ AT ξ.

ERROR I; SUBROUTINE INTERP ?

RETURN.

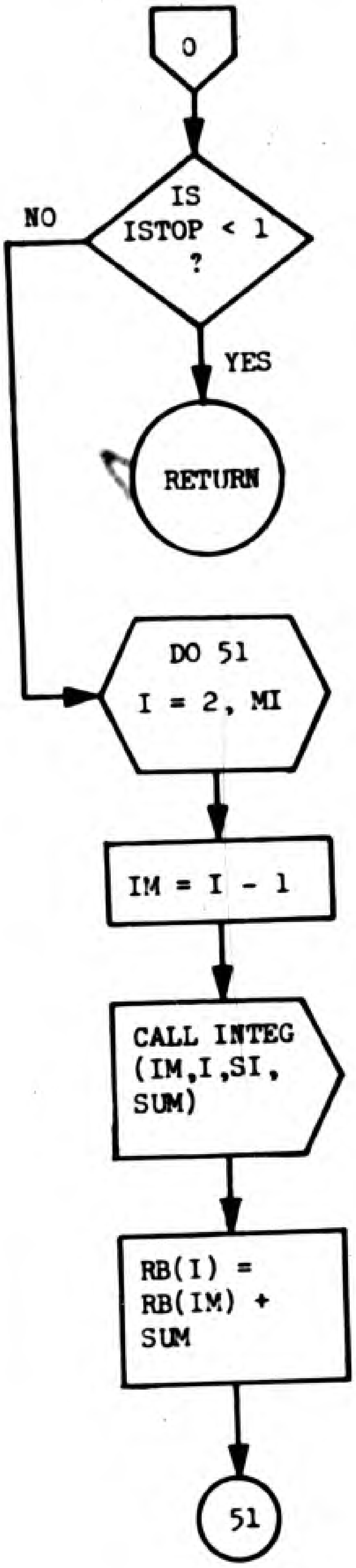
LOAD K AT ξ.

LOAD θ AT ξ.



LOOP TO LOAD PARAMETERS IN EQUATION (1) TO INTEGRATE FOR RB(I).

COMPUTE CUBIC COEFFICIENTS TO INTEGRATE FOR RB(I).

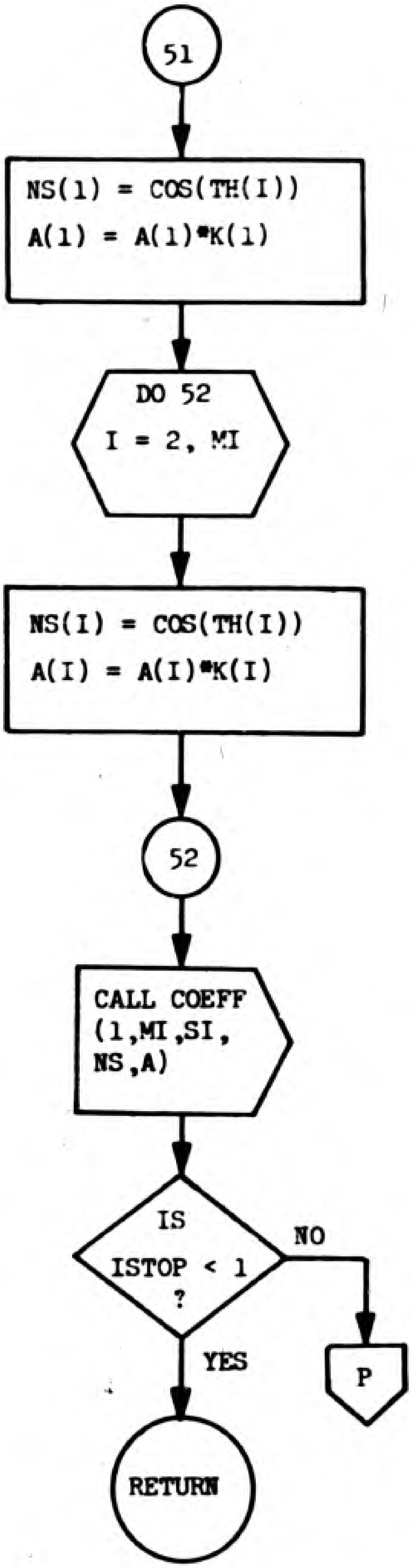


ERROR IN SUBROUTINE COEFF ?

RETURN.

INTEGRATE EQUATION (1).

STORE RB(I).

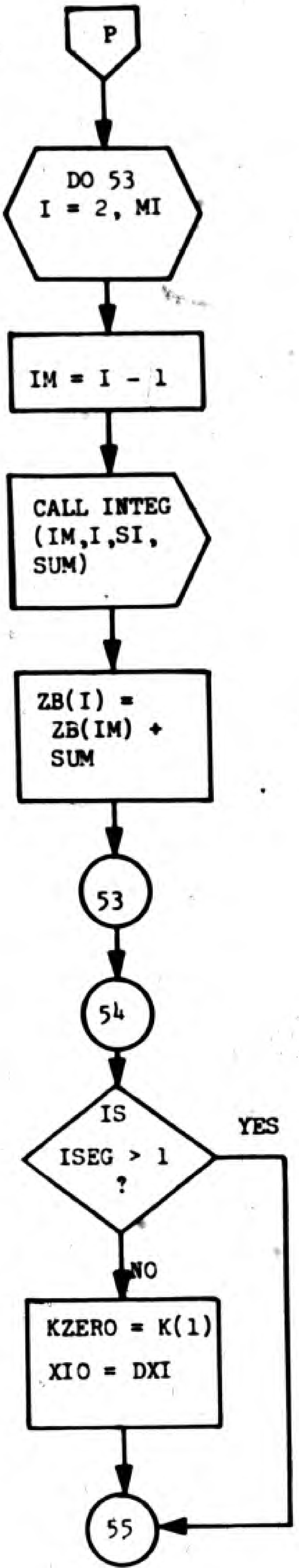


LOOP TO LOAD PARAMETERS FOR SOLVING EQUATION (2) FOR ZB(I).

COMPUTE CUBIC COEFFICIENTS FOR SOLVING EQUATION (2).

ERROR IN SUBROUTINE COEFF ?

RETURN.



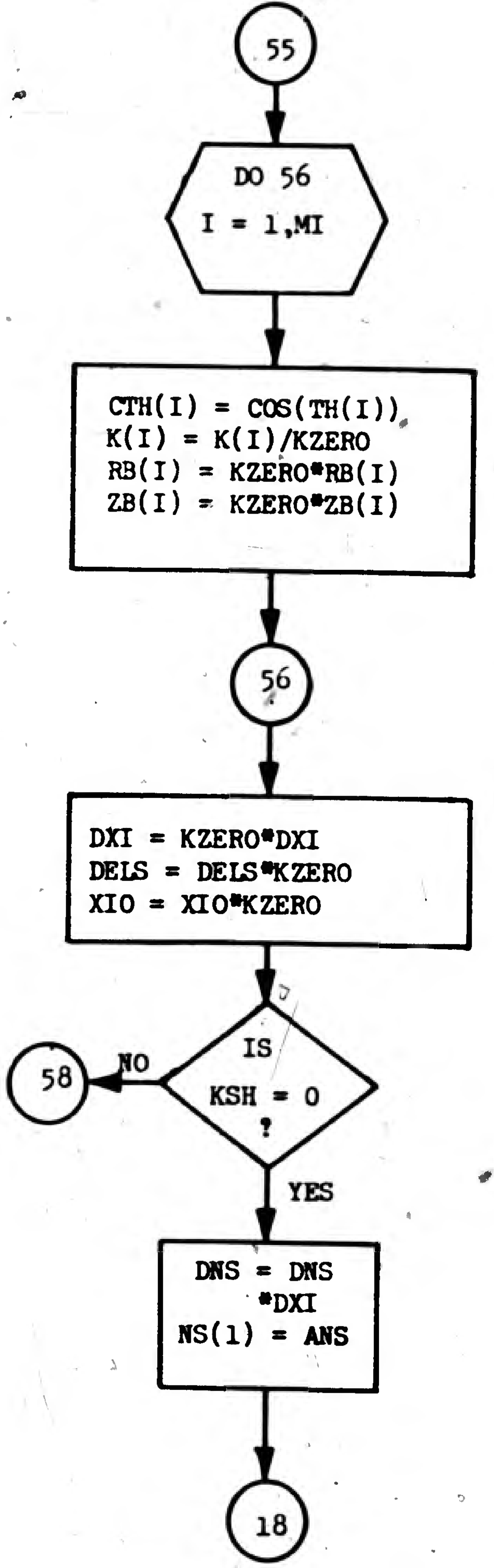
SOLVE EQUATION (2).

STORE ZB(I).

ENTRY POINT TO NON-DIMENSIONALIZE THE BODY GEOMETRY.

SEGMENT NUMBER > 1 ?

SAVE K(1) AND SET XIO FOR FIRST SEGMENT.

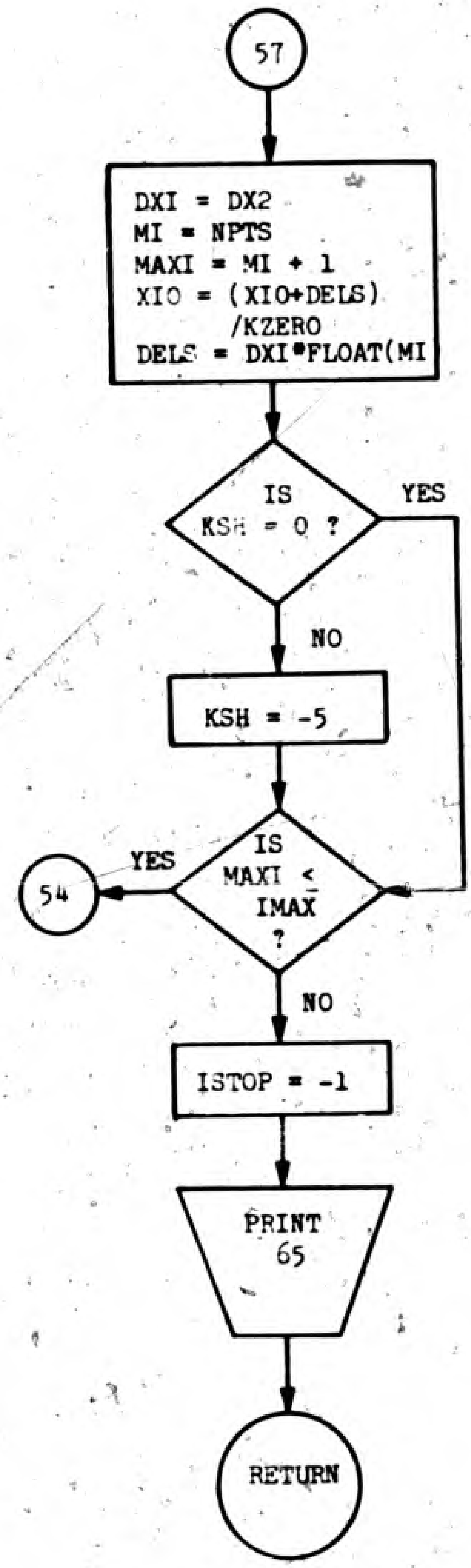


COMPUTE COS (Θ) AND NON-DIMENSIONALIZE K, RB AND ZB.

NONDIMENSIONALIZE DXI, DELS AND XIO.

WAS THE SHOCK SHAPE READ IN ?

STORE SHOCK SLOPE AND NS AT THE CONSTANT STARTING LINE.



BODY GEOMETRY DATA WAS INPUT
IN EQUAL INCREMENTS OF S.

SET UP VALUES OF BODY GEOMETRY
PARAMETERS.

IS THE SHOCK SHAPE TO BE
ESTIMATED ?

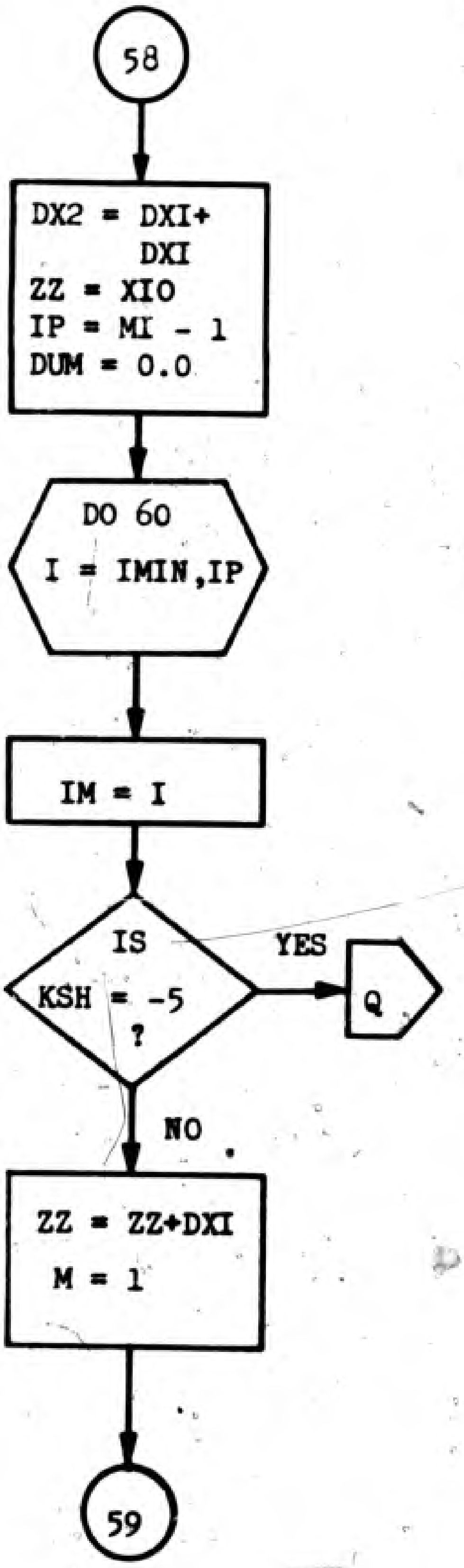
NO. PROVIDED IN INPUT DATA.
SET INDICATOR.

ARE DIMENSION SIZES LARGE ENOUGH
FOR THE NUMBER OF POINTS SUPPLIED ?

NO. SET ERROR FLAG.

PRINT ERROR MESSAGE.

RETURN.

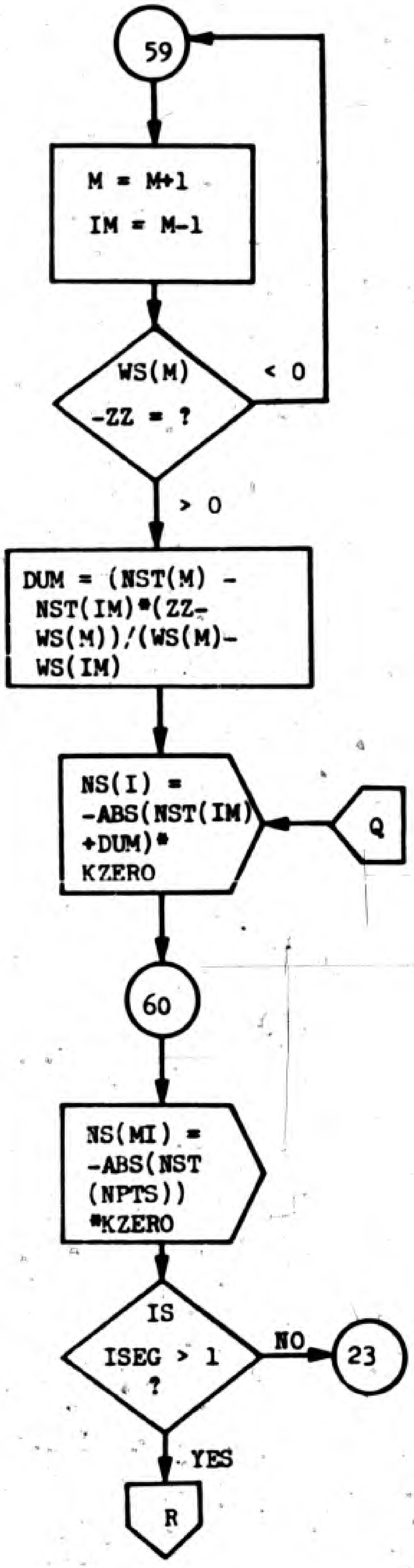


SHOCK SHAPE HAS BEEN INPUT.

SET GEOMETRY PARAMETERS.

START LOOP TO SET n_s.

WAS DATA INPUT IN EQUAL INCREMENTS OF S ?



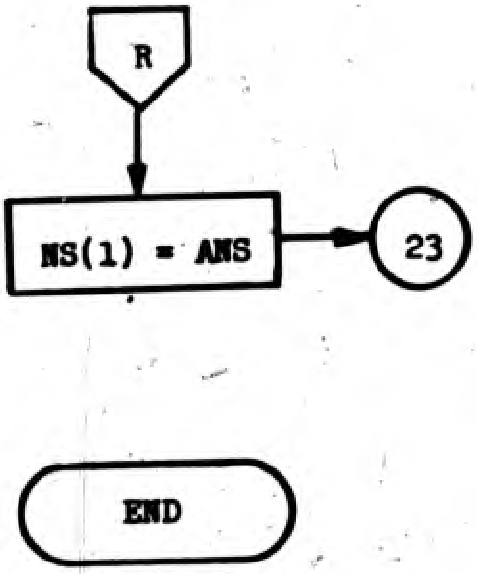
SEARCH FOR CORRECT INPUT SHOCK VALUE.

INTERPOLATE FOR POINT OF INTEREST.

SET n_s.

SET DOWNSTREAM VALUE OF n_s.

SEGMENT NUMBER > 1 ?



RECOVER UPSTREAM VALUE OF n_s.

D. LISTING

```
      SUBROUTINE GEOM (IST)
C
C      THIS ROUTINE PERFORMS THE BODY AND SHOCK GEOMETRY CALCULATIONS
C     FOR EACH BODY SEGMENT.
C
      COMMON /2/ RB(15),ZB(15),TH(15),CTH(15),K(15),NS(15),WS(15)
      COMMON /3/ MAXI,MI,IMIN,MAXN,MN,NMIN
      COMMON /4/ IPRINT,NPR,KGEOM
      COMMON /5/ DXI,DET,DX2,DE2,DT
      COMMON /6/ C1,VS1,HT1,A1,B1
      COMMON /9/ ISEG,XI0,DELS,KZERO,BNOSE
      COMMON /11/ MAXIT,ISTOP,ITD,TIME,SI(15),NST(15)
      COMMON /INPUT/ GAM1,AMACH1,THETC,ZRN,TCRIT(2),P1,RHO1,RN,AMU0,T1,E
     *RR(2),IPR(6),INTAP,OTAP,NOPT,KCASE,NFILE,NSEG,IGEOM,NGEOM,NTAP
      COMMON /CONST/ PI,PI2,GAMA,GAMB
      COMMON /MESH/ IMAX,NMAX
      COMMON /SPLINE/ C(4,15)
      DIMENSION A(1)
      DATA BLANK/10H          /
      INTEGER OTAP
      REAL NS,K,KZERO,NST
      EQUIVALENCE (A,CTH)
      IST=MI
      IF (XI0.LT.0.0) XI0=0.0
      GO TO (1,28,9), KGEOM
1     IF (ISEG.GT.1) GO TO 11
2     IF (IGEOM) 3,3,4
3     ZZ=DELS/0.15+2.0
      MAXI=INT(ZZ)
4     ZZ=FLOAT(MAXI-2)
      DXI=DELS/ZZ
      IF (MAXI.LE.IMAX) GO TO 5
      ZZ=FLOAT(IMAX-2)
      DXI=DELS/ZZ
      MAXI=IMAX
5     MI=MAXI-1
      XI0=-DXI
      K(1)=1.0
      RB(1)=0.0
      ZB(1)=0.0
      TH(1)=PI2
      CTH(1)=0.0
      ZZ=0.0
      GO TO 7
6     ZZ=XI0
7     DO 8 I=2,MI
      ZZ=ZZ+DXI
      K(I)=1.0
      TH(I)=PI2-ZZ
      ZB(I)=1.0-COS(ZZ)
      RB(I)=SIN(ZZ)
8     CTH(I)=COS(TH(I))
```

```
      GO TO 13
9     IF (IBEG-2) 10,11,11
10    XTO=DELS
      DNS=(NS(IST)-NS(IST-1))/DXI
      TH(1)=TH(IST)
      K(1)=1.0
      RB(1)=RB(IST)
      ZB(1)=ZB(IST)
      NS(1)=NS(IST)
      CTH(1)=CTH(IST)
      DELS=PI2-THETC-DELS
      ZZ=DELS/DXI
      I=INT(ZZ)
      IF (I.LT.1) I=1
      IF (IGEOM.GT.0) I=IGEOM-2
      IF (I.GT.IMAX-2) I=IMAX-2
      DXI=DELS/FLOAT(I)
      MI=I+1
      MAXI=I+2
      DNS=DNS*DXI
      GO TO 6
11    XTO=XTO+DELS
      DNS=(NS(IST)-NS(IST-1))/DXI
      IF (ZB(IST)+0.001-ZBN) 13,13,12
12    ISTOP=-1
      WRITE (ITAP) ISTOP
      END FILE ITAP
      ISTOP=0
      RETURN
13    IF (IGEOM.GT.0) GO TO 14
      MAXI=INT(4.0/DXI)
      IF (MAXI.GT.6) MAXI=6
      IF (MAXI.LT.3) MAXI=3
      DUM=XI0/12.0
      IF (DUM.GT.DXI) DXI=DUM
      IF (DXI.GT.4.0) DXI=4.0
14    ZZ=FLOAT(MAXI-2)
      DELS=ZZ*DXI
      DZ=DXI*CTH(IST)
      DUM=ZB(IST)+ZZ*DZ
      IF (DUM-ZBN) 16,16,15
15    MAXI=MAXI-1
      IF (MAXI.GT.3) GO TO 14
      MAXI=3
      DXI=(ZBN-ZB(IST))/CTH(IST)
      IF(DXI.LT.0.1) DXI=0.1
      DELS=DXI
16    MI=MAXI-1
      DR=DXI*SIN(THETC)
      DZ=DXI*CTH(IST)
      DNS=DNS*DXI
      NS(1)=NS(IST)
      RB(1)=RB(IST)
      ZB(1)=ZB(IST)
      K(1)=1.0
      CTH(1)=CTH(IST)                                                   GEOM 108
      TH(1)=THETC                                                       GEOM 109
      DO 12 I=2,MI                                                      GEOM 110
      IM=I-1                                                            GEOM 111
      RR(I)=RR(IM)+DR                                                   GEOM 112
      ZR(I)=ZR(IM)+DZ                                                   GEOM 113
      TH(I)=THETC                                                       GEOM 114
      CTH(I)=CTH(1)                                                     GEOM 115
 12   K(I)=0.0                                                          GEOM 116
 11   DX2=DXT+DXI                                                       GEOM 117
      IF (NTAP) 19,25,25                                                GEOM 118
 19   DO 20 I=1,MI                                                      GEOM 119
 20   NS(I)=-0.01                                                       GEOM 120
      GO TO 23                                                          GEOM 121
 24   DUM=AMACH1*AMACH1                                                 GEOM 122
      ZZ=(GAMD*DUM+1.0)/(GAMA*DUM)                                      GEOM 123
      NS(1)=-0.88*(ZZ**1.957)*(KZERO**0.157)                            GEOM 124
      DUM=SQRT(DELS/ANS)                                                GEOM 125
      DNS=DUM*(0.24+0.24/AMACH1+NS(1))                                  GEOM 126
      ZZ=0.0                                                            GEOM 127
      DO 22 I=2,MI                                                      GEOM 128
      ZZ=ZZ+DXI                                                         GEOM 129
 22   NS(I)=NS(1)-DNS*((ZZ/DELS)**3)                                    GEOM 130
 23   DO 26 I=1,MAXI                                                    GEOM 131
 26   NS(I)=1.0                                                         GEOM 132
      IM=MI-1                                                           GEOM 133
      ZZ=K(MI)*DXI/2.0                                                  GEOM 134
      STH=SIN(TH(MI))                                                   GEOM 135
      RR(MAXI)=RR(MI)+DXI*(STH-ZZ*CTH(MI))                              GEOM 136
      ZR(MAXI)=ZR(MI)+DXI*(CTH(MI)+ZZ*STH)                              GEOM 137
      TH(MAXI)=TH(MI)+TH(MI)-TH(IM)                                     GEOM 138
      NS(MAXI)=NS(MI)+NS(MI)-NS(IM)                                     GEOM 139
      CTH(MAXI)=COS(TH(MAXI))                                           GEOM 140
      K(MAXI)=K(MI)+K(MI)-K(IM)                                         GEOM 141
      RETURN                                                            GEOM 142
      DO 25 I=2,MI                                                      GEOM 144
 21   ANS=K(1)                                                          GEOM 143
 25   ANS=ANS+K(I)                                                      GEOM 145
      ANS=ANS/FLOAT(MI)                                                 GEOM 146
      IF (ISEG.EQ.1) GO TO 21                                           GEOM 147
      ZZ=ANS**0.25                                                      GEOM 148
      DUM=XT0+0.5*DELS                                                  GEOM 149
      ANS=DNS/2.0                                                       GEOM 150
      DNS=DNS*(1.0+(2.0*ZZ-1.0)/DUM)                                    GEOM 151
      IF (DNS.GT.ANS) DNS=ANS                                           GEOM 152
      DO 27 I=2,MI                                                      GEOM 153
 27   NS(I)=NS(I-1)+DNS                                                 GEOM 154
      GO TO 23                                                          GEOM 155
 28   READ 63, M,NPTS,MAXI,IP,KSH,ANS,DX2,ZZ                            GEOM 156
      IF (M-ISEG) 29,32,31                                              GEOM 157
 29   DO 30 M=1,NPTS                                                    GEOM 158
 30   READ 61, DUM                                                      GEOM 159
      GO TO 28                                                          GEOM 160
 31   PRINT 64                                                          GEOM 161
      ISTOP=-1                                                          GEOM 162
      RETURN
32    IM=0
      IF (ANS.GT.0.0) BNOSE=ANS
      ANS=WS(IST)
      IF (ISEG.GT.1) DNS=(WS(IST)-WS(IST-1))/DXI
      IF (IMIN.EQ.1.OR.ZZ.NE.BLANK) GO TO 33
      ZB(1)=ZB(IST)/KZERO
      RB(1)=RB(IST)/KZERO
      TH(1)=TH(IST)
      K(1)=K(IST)*KZERO
      IM=1
33    IF (NPTS+IM.LE.IMAX) GO TO 34
      PRINT 62,NPTS,ISEG,IM,IMAX
      ISTOP=-1
      RETURN
34    IF (IGEOM.GT.0) MAXI=IGEOM
      IF(MAXI.LT.1) MAXI=NPTS
      MI=MAXI-1
      IF (MAXI.LE.IMAX) GO TO 35
      ISTOP=-1
      PRINT 65
      RETURN
35    DO 36 I=1,NPTS
      M=I+IM
36    READ 61, RB(M),ZB(M),TH(M),K(M),NST(M)
      XI0=(XI0+DELS)/KZERO
      WS(1)=XI0
      NPTS=NPTS+IM
      IM=IM+1
      DO 37 I=IM,NPTS
      IF (TH(I).GT.0.0) TH(I)=TH(I)*PI/180.0
37    TH(I)=ABS(TH(I))
      IF (IP.GT.0) MAXN=-IP
      IF (DX2.GT.0.0) GO TO 57
      IC=0
38    IC=IC+1
      IF (IC.LT.NPTS.AND.TH(IC).GT.0.1) GO TO 38
      IF (NPTS-IC.LT.2.AND.NPTS-IC.NE.0) IC=NPTS-2
      IF (IC.EQ.1) GO TO 42
      DO 39 I=1,IC
39    A(I)=TAN(PI2-TH(I))
      CALL COEFF (1,IC,RB,ZB,A)
      IF (ISTOP.LT.1) RETURN
      DO 40 I=1,IC
      WS(I)=SQRT(1.0+A(I)*A(I))
40    A(I)=A(I)*(2.0*C(3,I)+6.0*C(4,I)*RB(I))/WS(I)
      CALL COEFF (1,IC,RB,WS,A)
      IF (ISTOP.LT.1) RETURN
      DO 41 I=2,IC
      IM=I-1
      CALL INTEG (IM,I,RB,SUM)
41    WS(I)=WS(IM)+SUM
      IF (IC.EQ.NPTS) GO TO 46
42    DO 43 I=IC,NPTS
43    A(I)=TAN(TH(I))
```

```
      CALL COEFF (IC,NPTS,ZB,RB,A)                                      GEOM 218
      IF (ISTOP.LT.1) RETURN                                            GEOM 219
      DO 44 I=IC,NPTS                                                   GEOM 220
      NS(I)=SQRT(1.0+A(I)*A(I))                                         GEOM 221
   44 A(I)=A(I)*(2.0*C(3,I)+6.0*C(4,I)*ZB(I))/NS(I)                     GEOM 222
      CALL COEFF (IC,NPTS,ZB,NS,A)                                      GEOM 223
      IF (ISTOP.LT.1) RETURN                                            GEOM 224
      IP=IC+1                                                           GEOM 225
      DO 45 I=IP,NPTS                                                   GEOM 226
      IM=I-1                                                            GEOM 227
      CALL INTEG (IM,I,ZB,SUM)                                          GEOM 228
   45 NS(I)=NS(IM)+SUM                                                  GEOM 229
   46 DO 47 I=1,NPTS                                                    GEOM 230
   47 A(I)=-K(I)                                                        GEOM 231
      CALL COEFF (1,NPTS,NS,TH,A)                                       GEOM 232
      IF (ISTOP.LT.1) RETURN                                            GEOM 233
      ZZ=FLOAT(MI-1)                                                    GEOM 234
      DELS=NS(NPTS)-NS(1)                                               GEOM 235
      DXI=DELS/ZZ                                                       GEOM 236
      SI(1)=NS(1)                                                       GEOM 237
      DO 48 I=2,MI                                                      GEOM 238
      SI(I)=SI(I-1)+DXI                                                 GEOM 239
   48 CALL INTERP (NS,SI(I),A(I),K(I),NPTS)                             GEOM 240
      IF (ISTOP.LT.1) RETURN                                            GEOM 241
      DO 49 I=2,MI                                                      GEOM 242
      K(I)=-K(I)                                                        GEOM 243
   49 TH(I)=A(I)                                                        GEOM 244
      NS(1)=-COS(TH(1))*K(1)                                            GEOM 245
      A(1)=SIN(TH(1))                                                   GEOM 246
      DO 50 I=2,MI                                                      GEOM 247
      A(I)=SIN(TH(I))                                                   GEOM 248
      DUM=COS(TH(I))*K(I)                                               GEOM 249
   50 NS(I)=-DUM                                                        GEOM 250
      CALL COEFF (1,MI,SI,A,NS)                                         GEOM 251
      IF (ISTOP.LT.1) RETURN                                            GEOM 252
      DO 51 I=2,MI                                                      GEOM 253
      IM=I-1                                                            GEOM 254
      CALL INTEG (IM,I,SI,SUM)                                          GEOM 255
   51 RB(I)=RB(IM)+SUM                                                  GEOM 256
      NS(1)=COS(TH(1))                                                  GEOM 257
      A(1)=A(1)*K(1)                                                    GEOM 258
      DO 52 I=2,MI                                                      GEOM 259
      NS(I)=COS(TH(I))                                                  GEOM 260
   52 A(I)=A(I)*K(I)                                                    GEOM 261
      CALL COEFF (1,MI,SI,NS,A)                                         GEOM 262
      IF (ISTOP.LT.1) RETURN                                            GEOM 263
      DO 53 I=2,MI                                                      GEOM 264
      IM=I-1                                                            GEOM 265
      CALL INTEG (IM,I,SI,SUM)                                          GEOM 266
   53 ZB(I)=ZB(IM)+SUM                                                  GEOM 267
   54 IF (ISEG.GT.1) GO TO 55                                           GEOM 268
      KZERO=K(1)                                                        GEOM 269
      XI=-DXI                                                           GEOM 270
   55 DO 56 I=1,MI                                                      GEOM 271
      CTH(I)=COS(TH(I))                                                 GEOM 272
      (I)=S(I)/KZERO                                                    GEOM 273
      ZZ(I)=KZERO*ZZ(I)                                                 GEOM 274
      (I)=KZERO*ZZ(I)                                                   GEOM 275
      DXI=KZERO*DXI                                                     GEOM 276
      DELS=DELS*KZERO                                                   GEOM 277
      XT=X(I)*KZERO                                                     GEOM 278
      IF (KSH.NE.0) GO TO 58                                            GEOM 279
      ANS=DNS*DXI                                                       GEOM 280
      WS(1)=ANS                                                         GEOM 281
      GO TO 11                                                          GEOM 282
      DXI=DX2                                                           GEOM 283
      MT=NPTS                                                           GEOM 284
      MAXI=MT+1                                                         GEOM 285
      XI2=(XI1+DELS)/KZERO                                              GEOM 286
      DELS=DXI*FLOAT(MT-1)                                              GEOM 287
      IF (KSH.NE.0) KSH=-5                                              GEOM 288
      IF (MAXI.LE.IMAX) GO TO 54                                        GEOM 289
      ISTOP=-1                                                          GEOM 290
      PRINT 65                                                          GEOM 291
      RETURN                                                            GEOM 292
   54 DX2=DXI+DXI                                                       GEOM 293
      ZZ=XI2                                                            GEOM 294
      IP=MT-1                                                           GEOM 295
      DUM=0.0                                                           GEOM 296
      DO 60 I=IMIN,IP                                                   GEOM 297
      IM=I                                                              GEOM 298
      IF (KSH.EQ.-5) GO TO 60                                           GEOM 299
      ZZ=ZZ+DXI                                                         GEOM 300
      M=1                                                               GEOM 301
   59 M=M+1                                                             GEOM 302
      IM=M-1                                                            GEOM 303
      IF (WS(M)-ZZ.LE.0.0) GO TO 59                                     GEOM 304
      DUM=(WST(M)-WST(IM))*(ZZ-WS(M))/(WS(M)-WS(IM))                    GEOM 305
   60 WS(I)=-ABS(WST(IM)+DUM)*KZERO                                     GEOM 306
      WS(MT)=-ABS(WST(NPTS))*KZERO                                      GEOM 307
      IF (ISEG.GT.1) WS(1)=ANS                                          GEOM 308
      GO TO 23                                                          GEOM 309
C                                                                       GEOM 310
   61 FORMAT (8E10.4)                                                   GEOM 311
   62 FORMAT (//30X,65HTHE NUMBER OF POINTS PROVIDED FOR THE BODY SHAPE GEOM 312
     *EXCEEDED STORAGE/30X,I10,1X,35HPOINTS WERE READ FOR SEGMENT NUMBERGEOM 313
     *,I5/30X,I10,1X,58HPOINTS WERE REQUIRED TO STORE VALUES OF PRECEEDIGEOM 314
     *NG SEGMENT/30X,I10,1X,27HIS THE TOTAL NUMBER ALLOWED/30X,15HCASE TEGEOM 315
     *RMINATED)                                                         GEOM 316
   63 FORMAT (5I10,2E10.4,A10)                                          GEOM 317
   64 FORMAT (1H1//////20X,37HERROR IN DATA INPUT FOR BODY GEOMETRY/20X,GEOM 318
     *57HBODY SEGMENT NUMBER READ WAS GREATER THAN THE SEGMENT BEING TREGEOM 319
     *ATED/20X,15HCASE TERMINATED)                                      GEOM 320
   65 FORMAT (//30X,47HEXCEEDED THE MAXIMUM NUMBER OF NODES ALONG THE BOGEOM 321
     *DY/30X,15HCASE TERMINATED)                                        GEOM 322
      END                                                               GEOM 323
```

10. SUBROUTINE INITIAL

A. PURPOSE

This subroutine provides initial conditions at all nodes to start the solution. The shock jump conditions are applied to the shock shape point parameters. At body points, the surface boundary condition, an empirical pressure distribution (similar to modified Newtonian theory), conservation of total enthalpy and the known body streamline entropy supply the parameters. Intermediate points are treated by interpolation between the shock and body based on a variation proportional to η^2.

B. VARIABLE LIST

All variables previously defined in the description of the main program have been omitted from this list.

AMN - $(M_\infty \sin\beta)^2$

CBET - $\cos\beta$

DUM - dummy variable used for intermediate computations.

M - dummy index.

MAXNST - specifies the number of nodes to be used across the shock layer (may be negative).

PB - pressure on the body.

SBET - $\sin\beta$

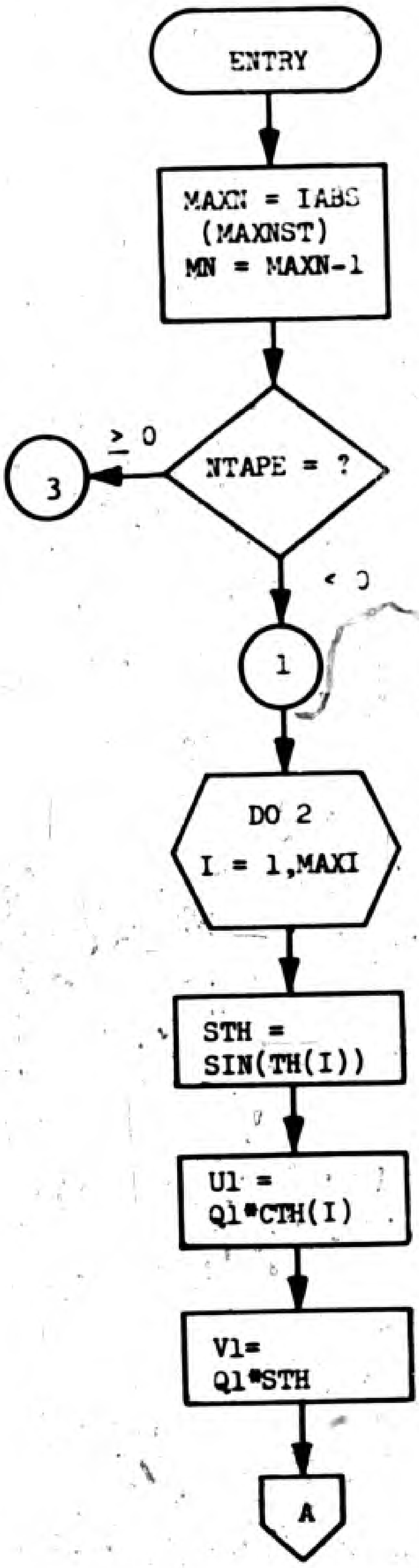
U1 - component of the free-stream velocity parallel to the shock.

V1 - component of the free-stream velocity normal to the shock.

V2 - component of the velocity at a shock point normal to the shock.

ZZ - dummy variable used for intermediate computations.

C. FLOW CHART



ENTER SUBROUTINE INITIAL

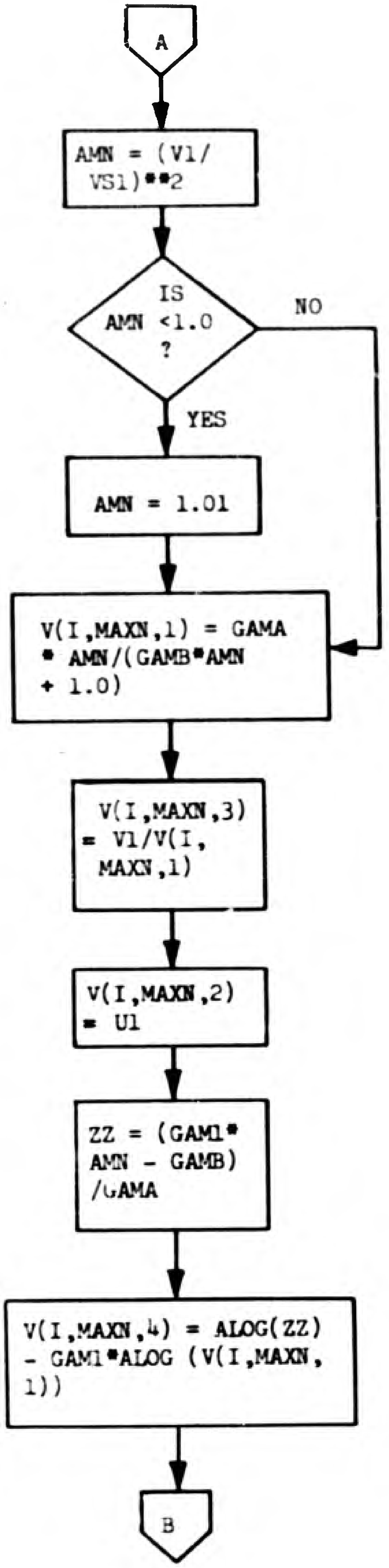
INSURE THAT THE INDEX OF THE SHOCK NODES IS POSITIVE.

UNSTEADY FLOW INITIAL CONDITIONS.

COMPUTE SIN θ.

COMPUTE U_∞

COMPUTE V_∞


A
AMN = (V1/ VS1)**2
IS AMN <1.0 ?
NO
YES
AMN = 1.01
V(I,MAXN,1) = GAMA * AMN/(GAMB*AMN + 1.0)
V(I,MAXN,3) = V1/V(I, MAXN,1)
V(I,MAXN,2) = U1
ZZ = (GAM1* AMN - GAMB) /GAMA
V(I,MAXN,4) = ALOG(ZZ) - GAM1*ALOG (V(I,MAXN, 1))
B

COMPUTE THE SQUARE OF THE MACH NUMBER NORMAL TO THE SHOCK.

IS AMN < 1.0 ?

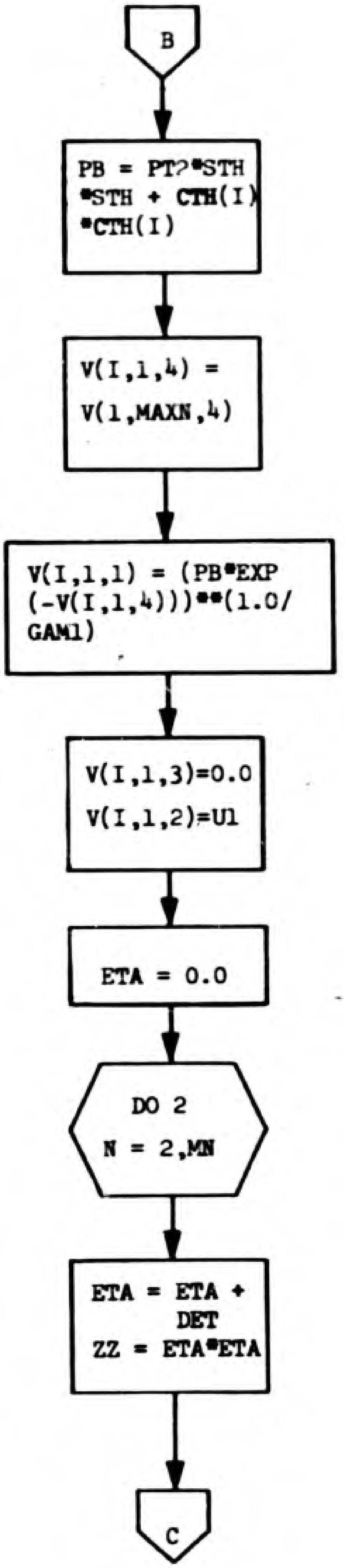
SET AMN = 1.01.

COMPUTE ρ AT THE SHOCK.

COMPUTE V AT THE SHOCK.

COMPUTE U AT THE SHOCK.

COMPUTE S AT THE SHOCK.



MODIFIED NEWTONIAN BODY PRESSURE.

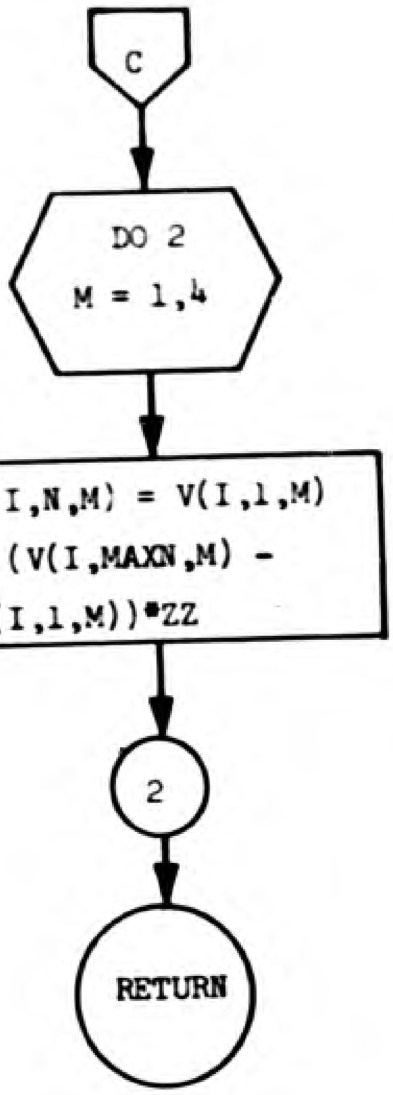
COMPUTE S AT THE BODY.

COMPUTE ρ AT THE BODY.

COMPUTE V AND U AT THE BODY.

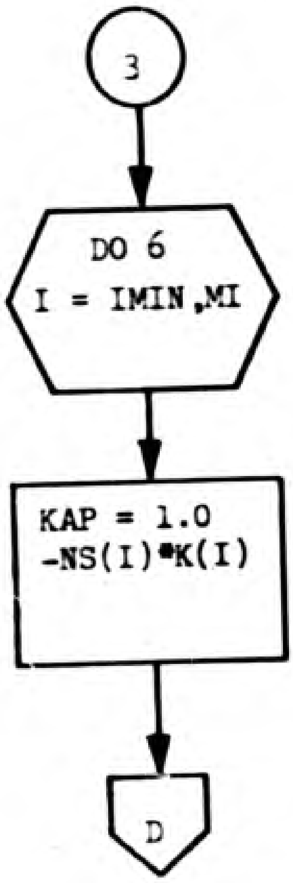
INITIALIZE η.

INCREMENT η AND COMPUTE A SCALE FACTOR, η^2.



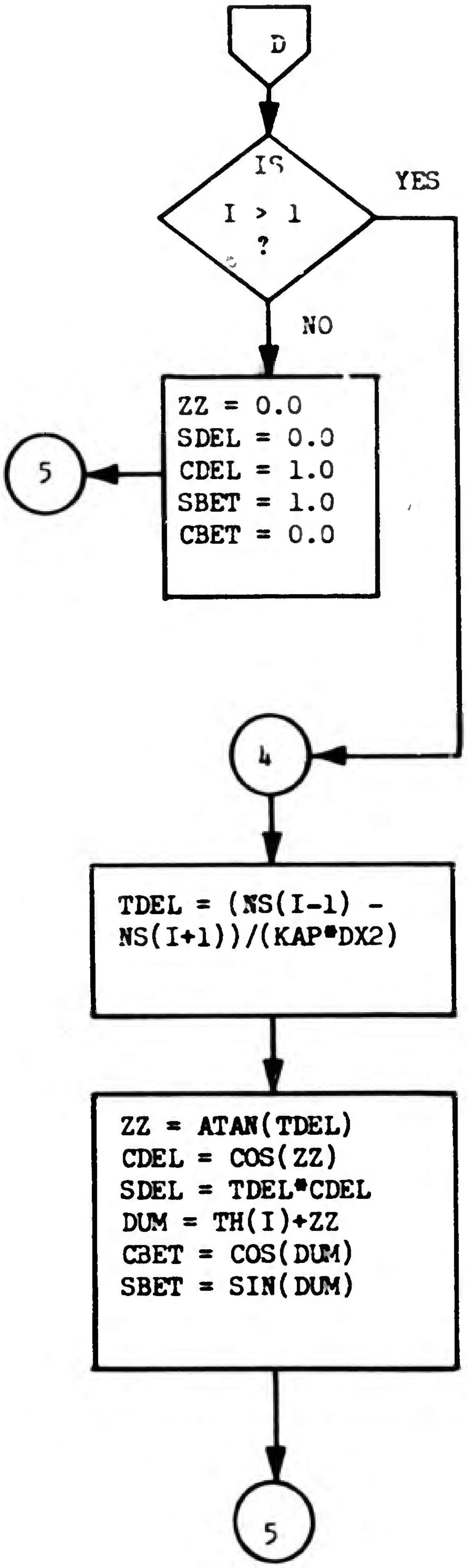
COMPUTE ρ, U, V AND S AT REMAINING NODES ACROSS THE SHOCK LAYER.

RETURN



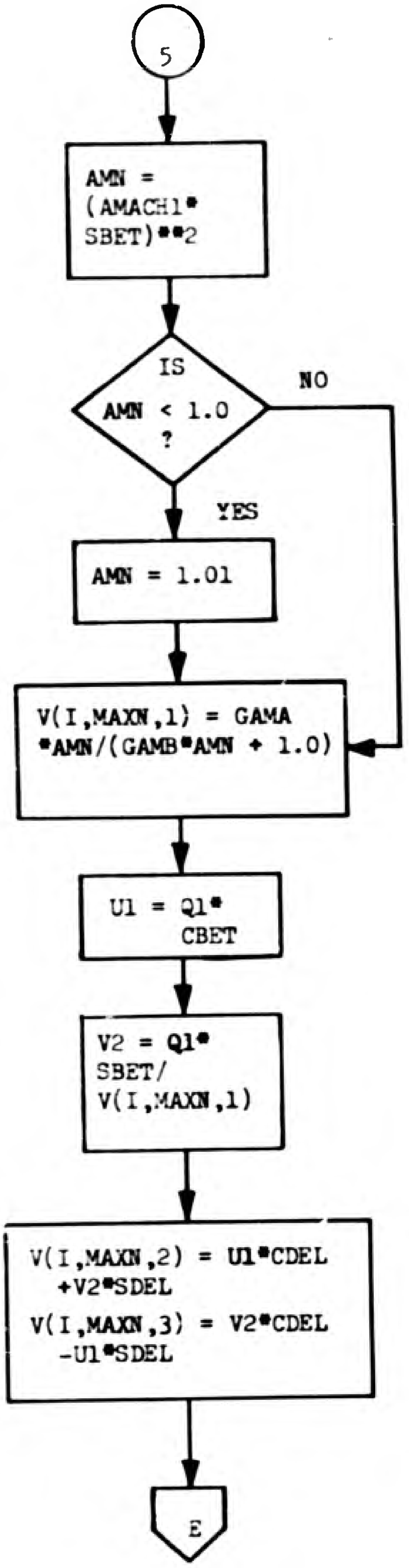
STEADY STATE INITIAL CONDITIONS

COMPUTE κ AT THE SHOCK.



COMPUTE SHOCK GEOMETRY FOR STAGNATION POINT.

COMPUTE SHOCK GEOMETRY FOR OTHER POINTS.


5
AMN = (AMACH1* SBET)**2
IS AMN < 1.0 ?
NO
YES
AMN = 1.01
V(I,MAXN,1) = GAMA *AMN/(GAMB*AMN + 1.0)
U1 = Q1* CBET
V2 = Q1* SBET/ V(I,MAXN,1)
V(I,MAXN,2) = U1*CDEL +V2*SDEL
V(I,MAXN,3) = V2*CDEL -U1*SDEL
E

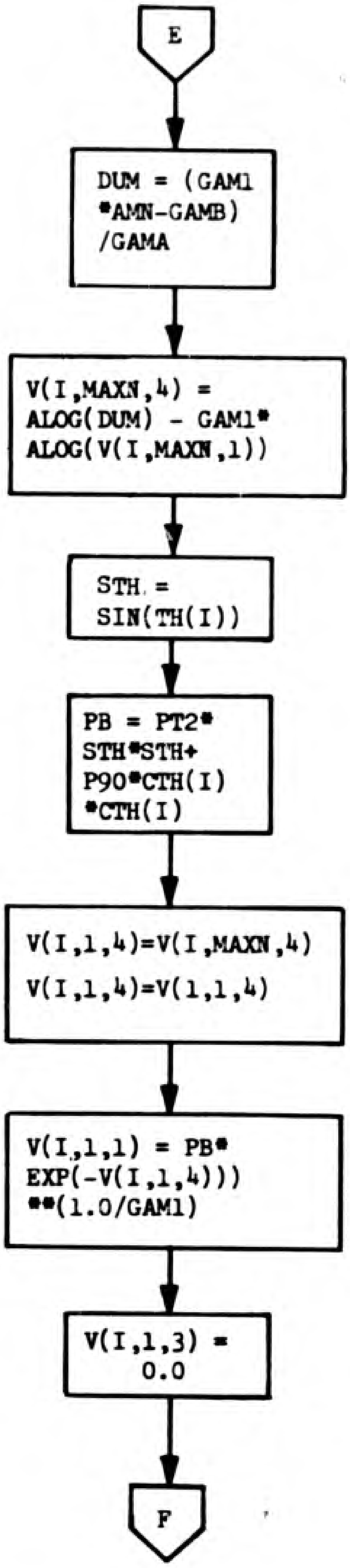
COMPUTE THE SQUARE OF THE MACH NUMBER NORMAL TO THE SHOCK.

AMN < 1.0 ?

COMPUTE ρ AT THE SHOCK.

COMPUTE VELOCITY COMPONENT TANGENT TO THE SHOCK.

COMPUTE VELOCITY COMPONENT NORMAL TO THE SHOCK.



COMPUTE S AT THE SHOCK.

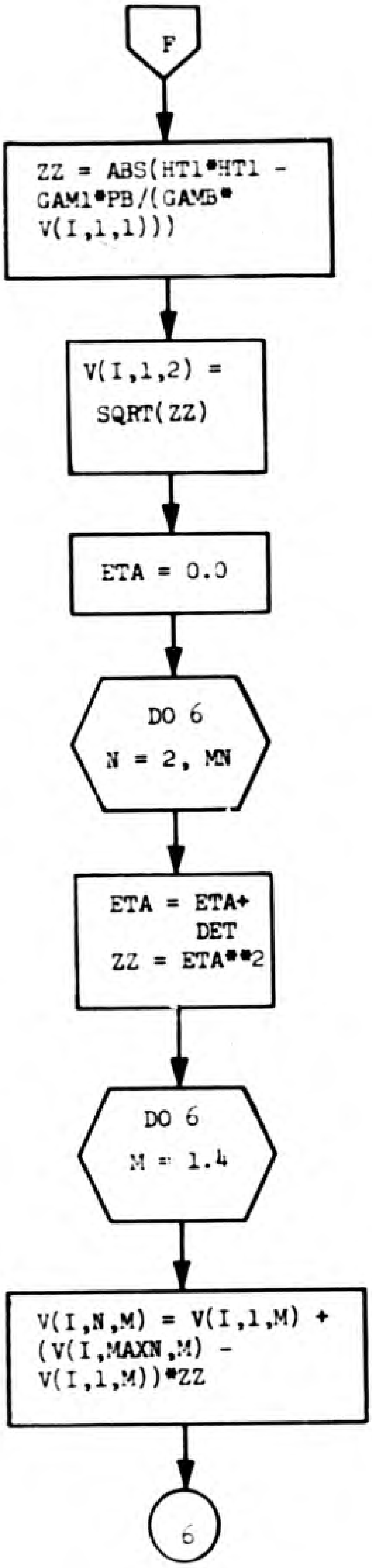
COMPUTE SIN θ.

ESTIMATE BODY PRESSURE.

CHOOSE MAXIMUM ENTROPY
FOR BODY POINTS.

COMPUTE ρ AT THE BODY.

SET V = 0 AT THE BODY.



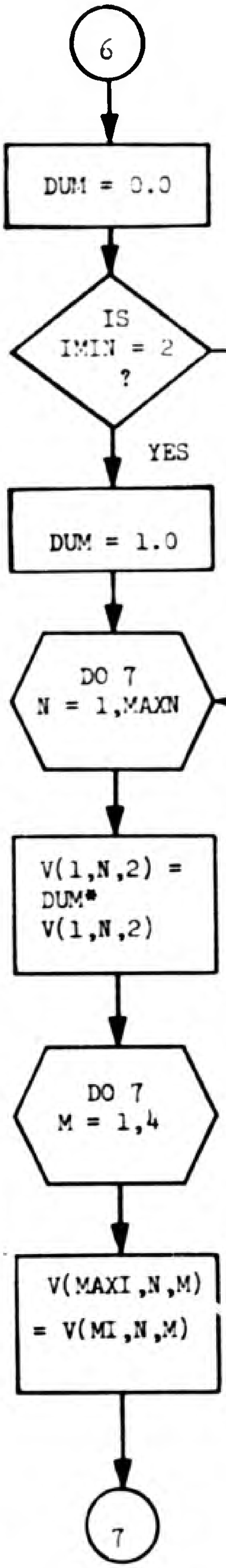
```
F
ZZ = ABS(HT1*HT1 -
GAM1*PB/(GAMB*
V(I,1,1)))
V(I,1,2) =
 SQRT(ZZ)
ETA = 0.0
DO 6
N = 2, MN
ETA = ETA+
      DET
ZZ = ETA**2
DO 6
M = 1.4
V(I,N,M) = V(I,1,M) +
(V(I,MAXN,M) -
V(I,1,M))*ZZ
6
```

SET U AT THE BODY TO CONSERVE TOTAL ENTHALPY.

INITIALIZE η.

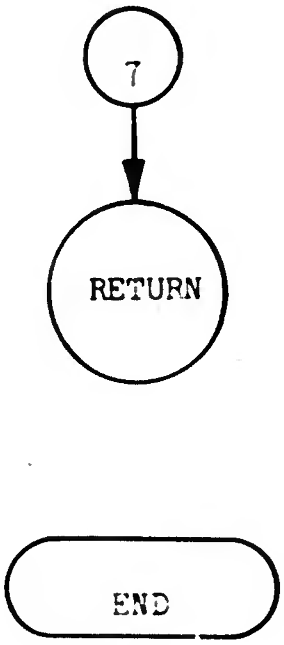
INCREMENT η AND COMPUTE SCALE FACTOR, η^2.

COMPUTE ρ, U, V AND S AT REMAINING POINTS ACROSS THE SHOCK LAYER.



ZERO U AT I = 1 IF IT IS THE STAGNATION STREAMLINE.

SET VALUES OF ρ, U, V AND S AT LAST BODY STATION NODES.



RETURN.

END OF SUBROUTINE INITIAL.

LISTING

```
      SUBROUTINE INITIAL (MAXM)
C
C     THIS ROUTINE PROVIDES THE INITIAL GUESSES FOR FLOW FIELD
C     PARAMETERS.
C
      COMMON /1/ V(15,41,4),C(41,4),P(3,4))
      COMMON /2/ RB(15),ZB(15),TH(15),CTH(15),K(15),AC(15),NS(15)
      COMMON /3/ MAXI,MI,IMIN,MAXNST,MN,NMIN
      COMMON /4/ DXI,DET,DX2,DE2,DT
      COMMON /5/ Q1,VS1,HT1,S1,R1
      COMMON /INPUT/ GAM1,AMACH1,THETC,ZPN,TCRIT(2),P1,RHO1,RN,AMUP,T1,(
     *PP(2),IPP(5),INTAP,OTAP,IOPT,NCASE,METHD,NSEC,IGEOM,NGEOM,MTAP
      COMMON /CONST/ PI,PI2,GAMA,GAMB
      COMMON /PRES/ PT2,P31
      REAL NS,K,KAP
      MAXN=IABS(MAXNST)
      MN=MAXN-1
      IF (MTAP) 1,3,3
    1 DO 2 I=1,MAXI
      STH=SIN(TH(I))
      U1=Q1*CTH(I)
      V1=Q1*STH
      AMN=(V1/VS1)**2
      IF (AMN.LT.1.0) AMN=1.01
      V(I,MAXN,1)=GAMA*AMN/(GAMB*AMN+1.0)
      V(I,MAXN,3)=V1/V(I,MAXN,1)
      V(I,MAXN,2)=U1
      ZZ=(GAM1*AMN-GAMB)/GAMA
      V(I,MAXN,4)=ALOG(ZZ)-GAM1*ALOG(V(I,MAXN,1))
      RR=PT2*STH*STH+CTH(I)*CTH(I)
      V(I,1,4)=V(1,MAXN,4)
      V(I,1,1)=(RR*EXP(-V(I,1,4)))**(1.0/GAM1)
      V(I,1,3)=0.0
      V(I,1,2)=U1
      ETA=0.0
      DO 2 N=2,MN
      ETA=ETA+DET
      ZZ=ETA*ETA
      DO 2 M=1,4
    2 V(I,N,M)=V(I,1,M)+(V(I,MAXN,M)-V(I,1,M))*ZZ
      RETURN
    3 DO 6 I=IMIN,MI
      KAP=1.0-NS(I)*K(I)
      IF (I.GT.1) GO TO 4
      ZZ=0.0
      SDEL=0.0
      CDEL=1.0
      SBET=1.0
      CBET=0.0
      GO TO 5
    4 TDEL=(NS(I-1)-NS(I+1))/(KAP*DX2)
      ZZ=ATAN(TDEL)
```

NOT REPRODUCIBLE

```
      CDEL=COS(ZZ)
      SDEL=TDEL*CDEL
      DUM=TH(I)+ZZ
      CBET=COS(DUM)
      SBET=SIN(DUM)
      AMN=(AMACH1*SBET)**2
      IF (AMN.LT.1.01) AMN=1.01
      V(I,MAXN,1)=GAMA*AMN/(GAMB*AMN+1.0)
      U1=Q1*CBET
      V2=Q1*SBET/V(I,MAXN,1)
      V(I,MAXN,2)=U1*CDEL+V2*SDEL
      V(I,MAXN,3)=V2*CDEL-U1*SDEL
      DUM=(GAM1*AMN-GAMB)/GAMA
      V(I,MAXN,4)=ALOG(DUM)-GAM1*ALOG(V(I,MAXN,1))
      STH=SIN(TH(I))
      PP=PT2*STH*STH+P90*CTH(I)*CTH(I)
      V(I,1,4)=V(1,MAXN,4)
      V(I,1,4)=V(1,1,4)
      V(I,1,1)=(P0*EXP(-V(I,1,4)))**(1.0/GAM1)
      V(I,1,3)=0.0
      ZZ=ABS(HT1+HT1-GAM1*PP/(GAMB*V(I,1,1)))
      V(I,1,2)=SQRT(ZZ)
      ETA=0.0
      DO 6 N=2,MN
      ETA=ETA+DET
      ZZ=ETA**2
      DO 6 M=1,4
6     V(I,N,M)=V(I,1,M)+(V(I,MAXN,M)-V(I,1,M))*ZZ
      DUM=1.0
      IF (IMIN.EQ.2) DUM=1.0
      DO 7 N=1,MAXN
      V(1,N,2)=DUM*V(1,N,2)
      DO 7 M=1,4
7     V(MAXI,N,M)=V(MI,N,M)
      RETURN
      END
```

11. SUBROUTINE INTAPE

A. PURPOSE

This subroutine loads data and initial conditions from a magnetic tape unit. This type of initialization is used for restarting an unfinished case or for performing additional iterations on a completed case.

B. VARIABLE LIST

Most of the variables in this routine are read from magnetic tape, only. Their definitions can be determined by matching their locations in the various common blocks with the corresponding variables in the main program.

IRD - control variable specifying the type of read operation.

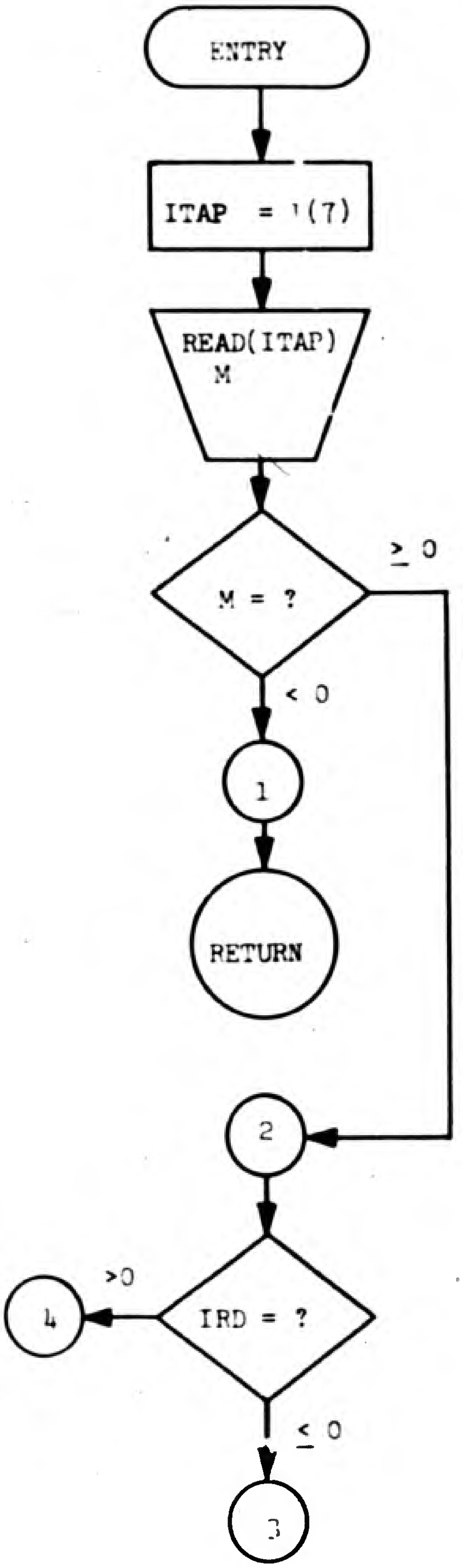
=0 - read data in COMMON/INPUT/.

>0 - read data required for a specific segment.

ITAP - variable used for temporary storage of the input tape unit number.

M - variable indicating the end of a case, when a negative value is read.

C. FLOW CHART



ENTER SUBROUTINE INTAPE.

SAVE THE INPUT TAPE NUMBER.

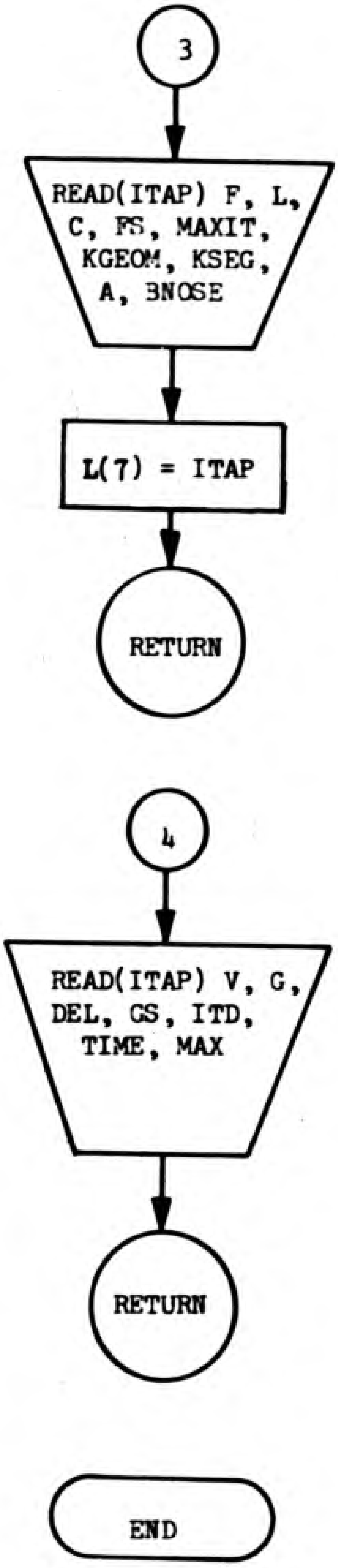
READ END OF CASE INDICATOR

END OF CASE.

RETURN.

THIS FILE CONTAINS DATA FOR ANOTHER SEGMENT.

WHAT TYPE OF DATA IS TO BE READ ?


3
READ(ITAP) F, L, C, FS, MAXIT, KGEOM, KSEG, A, BNOSE
L(7) = ITAP
RETURN
4
READ(ITAP) V, G, DEL, GS, ITD, TIME, MAX
RETURN
END

READ INPUT DATA AND CONTROL VARIABLES.

RECOVER INPUT TAPE NUMBER.

RETURN.

READ FLOW FIELD AND SEGMENT GEOMETRY.

RETURN.

END OF SUBROUTINE INTAPE.

D. LISTING

```
      SUBROUTINE INTAPE (IRD,M)
C
C      THIS ROUTINE READS THE FLOW FIELD DATA FROM TAPE
C
      COMMON /1/ V(15,40,4),D(40,4),P(3,40)
      COMMON /2/ G(113)
      COMMON /3/ MAX(6)
      COMMON /4/ IDUM(2),KGFCM,A(6),KSEG
      COMMON /5/ DEL(5)
      COMMON /6/ FS(3)
      COMMON /9/ ISEG,GS(3),BNOSE
      COMMON /11/ MAXIT,ISTOP,ITD,TIME
      COMMON /INPUT/ F(13),L(15)
      COMMON /CONST/ DUM(2),C(3)
      ITAP=L(7)
      READ (ITAP) M
      IF (M) 1,2,2
1     RETURN
2     IF (IRD) 3,3,4
3     READ (ITAP) F,L,C,FS,MAXIT,KGFCM,KSEG,A,BNOSE
      L(7)=ITAP
      RETURN
4     READ (ITAP) V,G,DEL,GS,ITD,TIME,MAX
      RETURN
      END
```

12. SUBROUTINE INTEG

A. PURPOSE

This routine uses the coefficients for a cubic fit generated by subroutine COEFF to integrate the function between 2 adjacent values of the independent variable.

B. VARIABLE LIST

C - array of cubic coefficients generated in subroutine COEFF.

I - index of the point in the X array corresponding to the upper limit of integration.

IM - index of the point in the X array corresponding to the lower limit of integration.

SUM - value of the integral.

X - array containing the values of the independent variable.

XM2 - $[X(IM)]^2$

XM3 - $[X(IM)]^3$

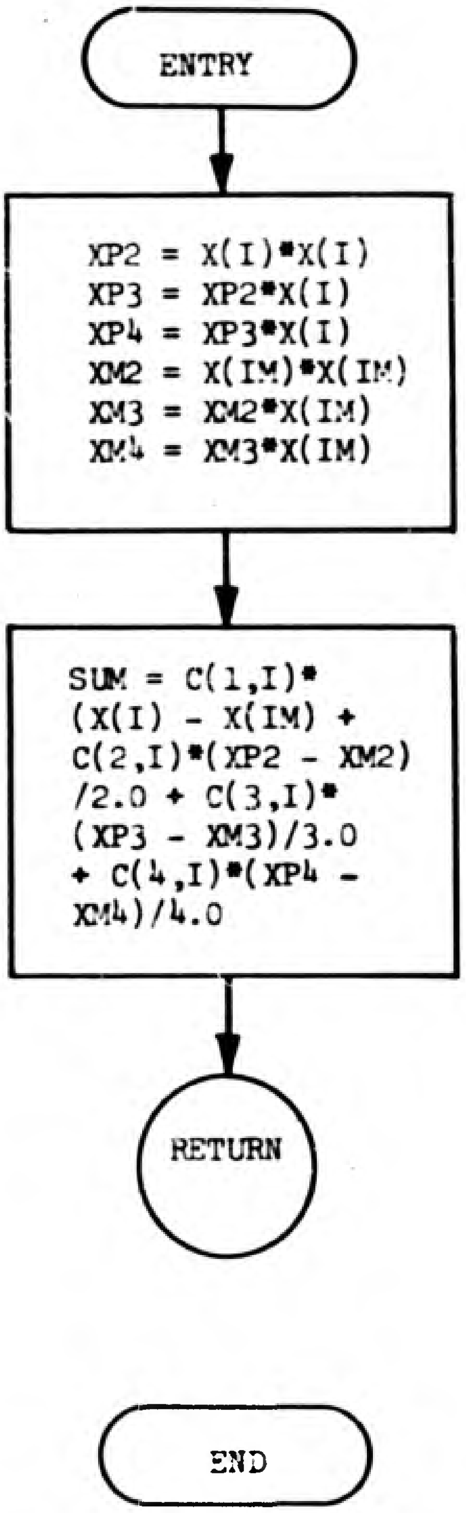
XM4 - $[X(IM)]^4$

XP2 - $[X(I)]^2$

XP3 - $[X(I)]^3$

XP4 - $[X(I)]^4$

C. FLOW CHART



ENTER SUBROUTINE INTEG.

COMPUTE X^2, X^3 AND X^4 AT THE POINTS IM AND I.

INTEGRATE THE FUNCTION DEFINED BY THE CUBIC COEFFICIENTS FROM X(IM) TO X(I).

RETURN.

END OF SUBROUTINE INTEG.

D. LISTING

```
      SUBROUTINE INTEG (IM,I,X,SUM)
C
C      THIS ROUTINE INTEGRATES A FUNCTION USING THE SPLINE COEFFICIENTS
C     FROM COEFF.
C
      COMMON /SPLINE/ C(4,15)
      DIMENSION X(15)
      XP2=X(I)*X(I)
      XP3=XP2*X(I)
      XP4=XP3*X(I)
      XM2=X(IM)*X(IM)
      XM3=XM2*X(IM)
      XM4=XM3*X(IM)
      SUM=C(1,I)*(X(I)-X(IM))+C(2,I)*(XP2-XM2)/2.0+C(3,I)*(XP3-XM3)/3.0+
     *C(4,I)*(XP4-XM4)/4.0
      RETURN
      END
```

13. SUBROUTINE INTERP

A. PURPOSE

This subroutine interpolates between data points for the values of a function and it's derivative, using the cubic fit coefficients generated by subroutine COEFF. An input table of independent variable points is searched to locate the point closest to the desired point of interpolation. If the table is exceeded by a significant amount, an error message is printed and the current case is terminated.

B. VARIABLE LIST

C - array of cubic fit coefficients.

DEL - difference between the values of the independent variable at the last 2 points.

DYDX - $\frac{dy}{dx}$

I - index specifying the location in the input array, X.

IM - I - 1.

ISTOP - error indicator.

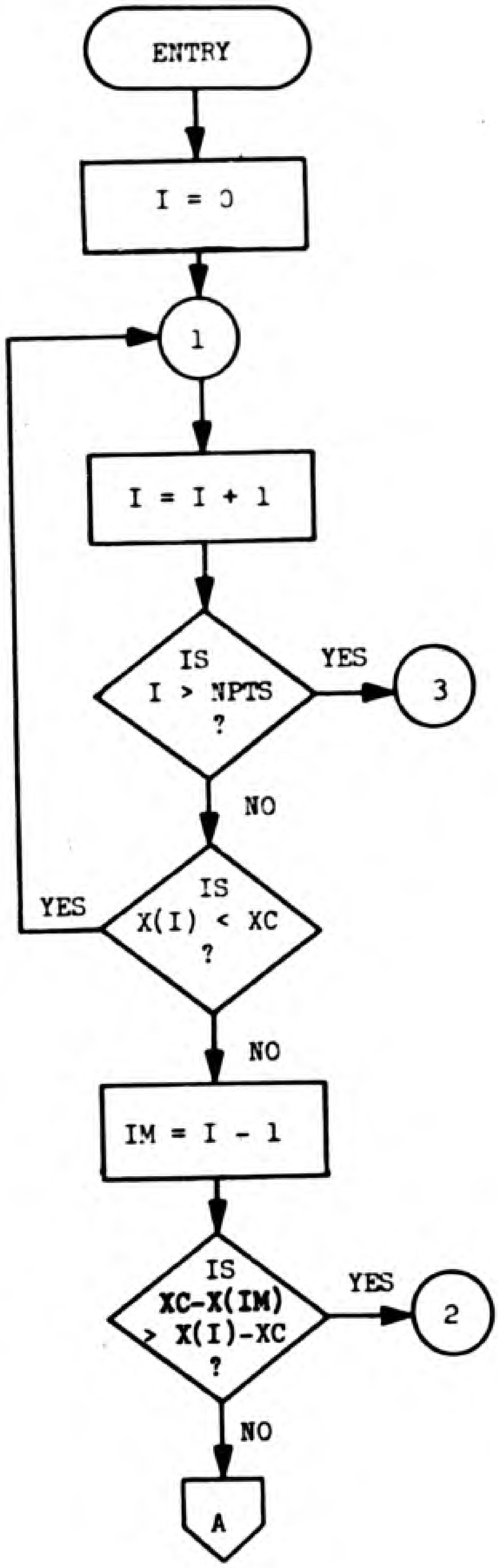
NPTS - number of points in the input array, X.

X - array containing the input table.

CX - value of the independent variable at which y and $\frac{dy}{dx}$ are to be determined.

YC - value of y at XC.

C. FLOW CHART



ENTER SUBROUTINE INTERP.

INITIALIZE I.

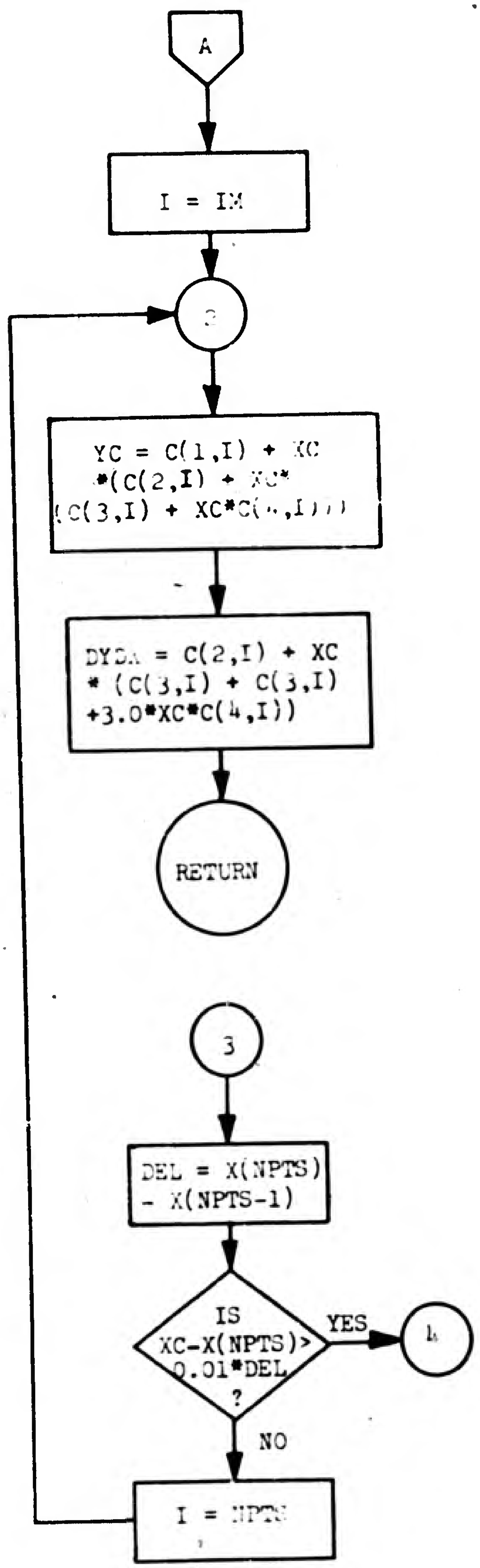
INCREMENT I.

DOES I EXCEED THE
TABLE SIZE ?

IS X(I) < XC ?

STORE I - 1.

IS XC CLOSER TO X(I)
THAN TO X(IM) ?


A
I = IM
2
YC = C(1,I) + XC *(C(2,I) + XC* (C(3,I) + XC*C(4,I)))
DYDX = C(2,I) + XC * (C(3,I) + C(3,I) +3.0*XC*C(4,I))
RETURN
3
DEL = X(NPTS) - X(NPTS-1)
IS XC-X(NPTS)> 0.01*DEL ?
YES
4
NO
I = NPTS

COMPUTE YC.

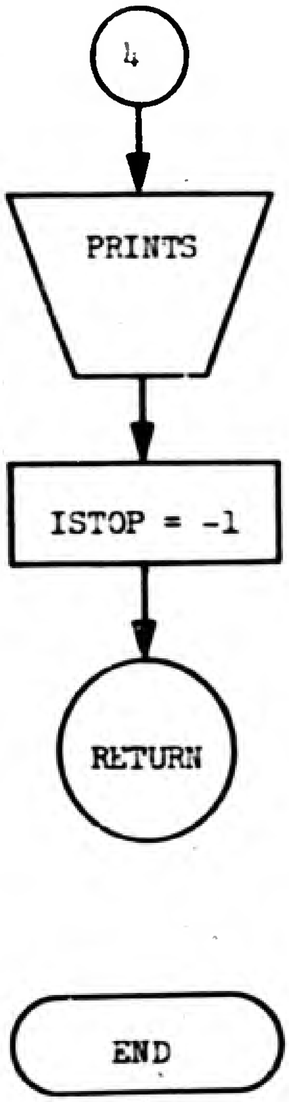
COMPUTE DYDX.

RETURN.

EXCEEDED TABLE SIZE.

CHECK TO SEE IF XC EXCEEDS THE TABLE BY A SIGNIFICANT AMOUNT.

SET I TO USE THE LAST POINT IN THE TABLE.



PRINTS ERROR MESSAGE.

SET ERROR INDICATOR.

RETURN.

END OF SUBROUTINE INTERP.

D. LISTING

```
      SUBROUTINE INTERP (X,XC,YC,DYDX,NPTS)
C
C     THIS ROUTINE INTERPOLATES FOR THE VALUE OF A FUNCTION USING
C     THE SPLINE COEFFICIENTS FROM COEFF.
C
      COMMON /11/ IDUMMY,ISTOP
      COMMON /SPLINE/ C(4,15)
      DIMENSION X(15)
      I=0
    1 I=I+1
      IF (I.GT.NPTS) GO TO 3
      IF (X(I).LT.XC) GO TO 1
      IM=I-1
      IF (XC-X(IM).GT.X(I)-XC) GO TO 2
      I=IM
    2 YC=C(1,I)+XC*(C(2,I)+XC*(C(3,I)+XC*C(4,I)))
      DYDX=C(2,I)+XC*(C(3,I)+C(3,I)+3.0*XC*C(4,I))
      RETURN
    3 DEL=X(NPTS)-X(NPTS-1)
      IF (XC-X(NPTS).GT.0.01*DEL) GO TO 4
      I=NPTS
      GO TO 2
    4 PRINT 5
      ISTOP=-1
      RETURN
C
    5 FORMAT (1H1,10X,26HERROR IN SUBROUTINE INTERP/10X,19HDATA RANGE EX
     *CEEDED/10X,14HJOB TERMINATED)
      END
```

14. SUBROUTINE LOAD

A. PURPOSE

This subroutine loads data from a magnetic tape for use as initial conditions. The principal purpose of this option is to allow additional iterations to be performed on a previous case which did not achieve a steady state. It is also useful when unsteady flows are being considered, allowing the solution to be advanced in time in stages to obtain printed outputs at the desired intervals. Also, if the supplied initial conditions are inadequate for a particular problem, this option will allow different initial conditions to be supplied by the user without requiring alterations of the computer code. Case data variables that may be changed from the values used on the previous case are saved before the read operation and are recovered before returning to the main program. Two error checks are accomplished when this option is used. First, the number of iterations to be performed must have been specified (the steady-state option is not permitted - see section VII). Also, the input and output magnetic tape unit numbers must be different. This requirement has been included to prevent an accidental write operation which could erase data acquired on a previous run, possibly a multicase run which required extensive computer time. When either error is sensed, an error message is printed and the case is terminated. The case and segment numbers where computation should start are supplied in the input data (see section VII).

B. VARIABLE LIST

Variables previously described in the description of the main program have been omitted from this list.

CH - array containing values of floating point variables in COMMON block /INPUT/ that are to be used rather than the values read from magnetic tape.

I - DO loop index.

ICH - same application as CH for integer variables.

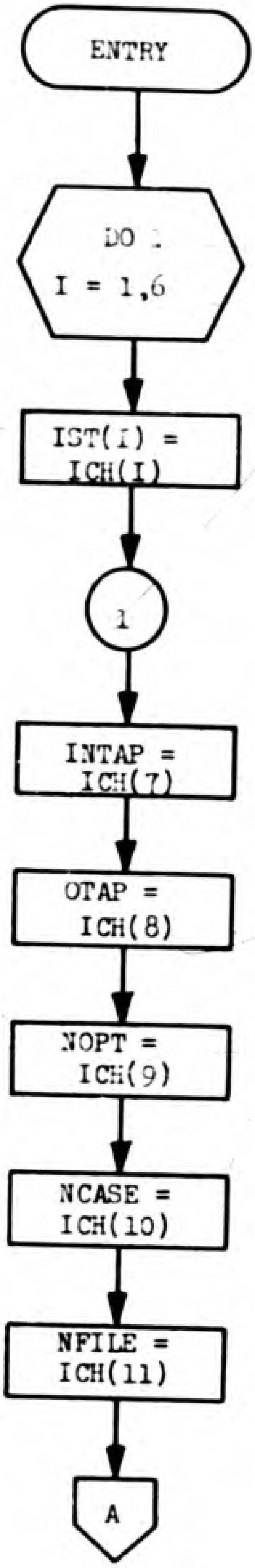
IST - array used for temporary storage of the ICH array.

M - indicator written on the tape that specified when the last segment has been completed.

N - variable specifying the number of cases and the number of files to be skipped.

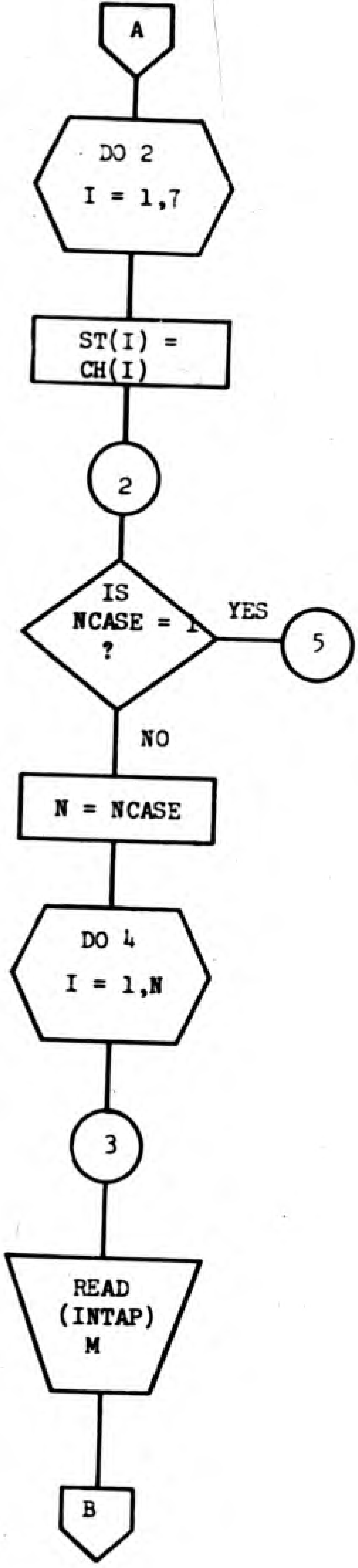
ST - array used for temporary storage of the CH array.

C. FLOW CHART



ENTER SUBROUTINE LOAD.

SAVE ALL INTEGER CASE DATA VARIABLES WHICH MAY BE CHANGED UNDER THIS OPTION



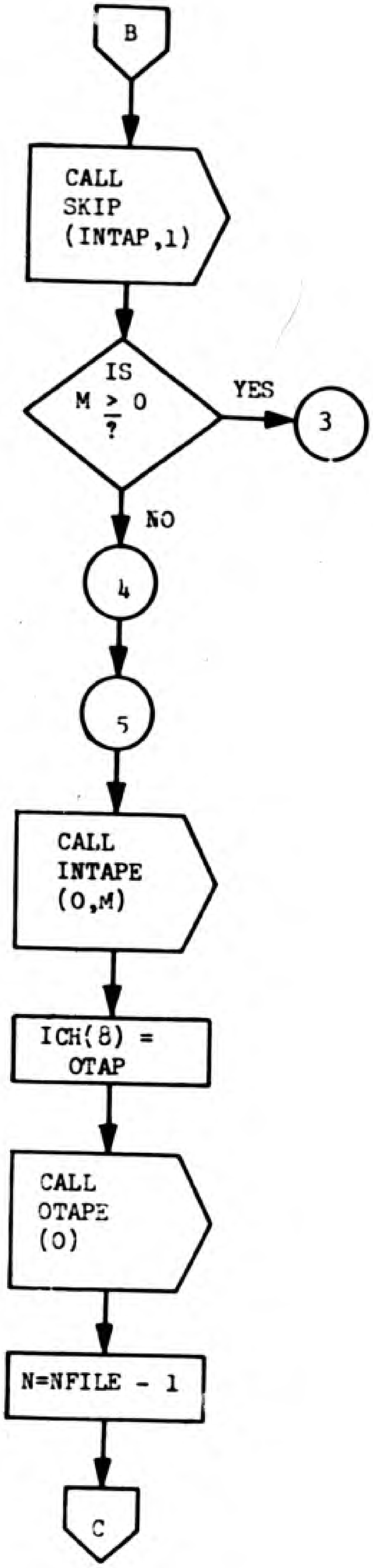
SAVE ALL FLOATING POINT VARIABLES
WHICH MAY BE CHANGED UNDER THIS
OPTION.

IS THE CASE TO BE LOADED THE
FIRST CASE STORED ON THE TAPE ?

COMPUTE THE NUMBER OF CASES
TO BE SKIPPED.

SKIP OVER N CASES.

READ VARIABLE USED TO SIGNAL
THE END OF A CASE.



SKIP 1 FILE ON THE TAPE.

ARE THERE MORE FILES FOR THE CURRENT CASE ?

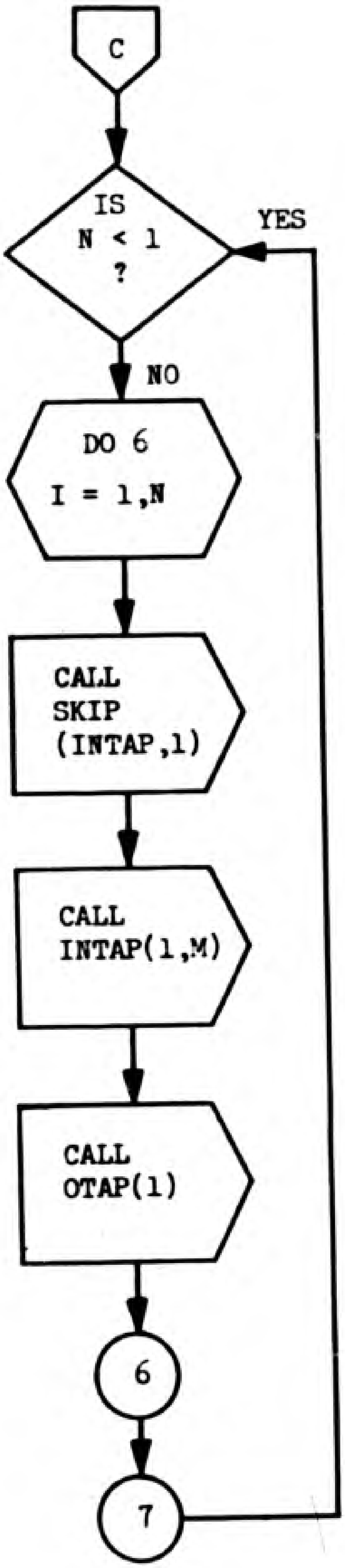
CORRECT CASE HAS BEEN LOCATED.

READ CASE DATA PREVIOUSLY USED FOR THE CASE.

RECOVER THE OUTPUT MAGNETIC TAPE UNIT NUMBER.

WRITE THE CASE DATA ON MAGNETIC TAPE.

COMPUTE THE NUMBER OF SEGMENTS TO BE SKIPPED.



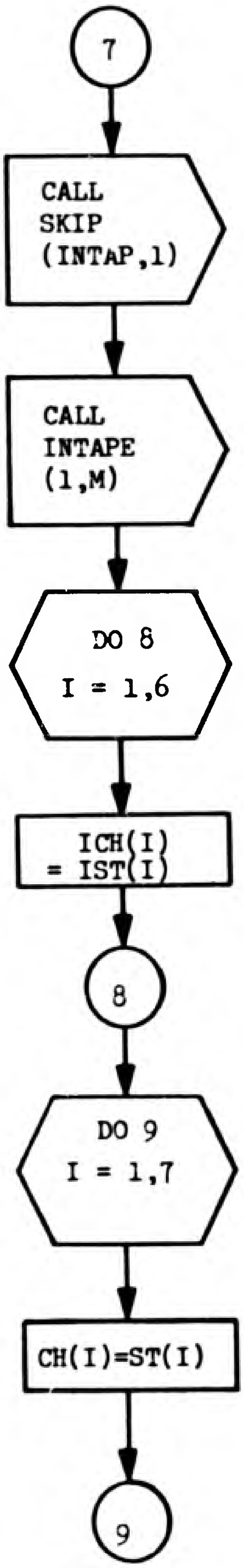
SHOULD COMPUTATIONS START AT THE FIRST SEGMENT ?

SKIP AN END OF FILE MARK.

READ FLOW FIELD DATA.

STORE DATA ON THE OUTPUT TAPE.

FIRST SEGMENT TO BE TREATED HAS BEEN LOCATED.

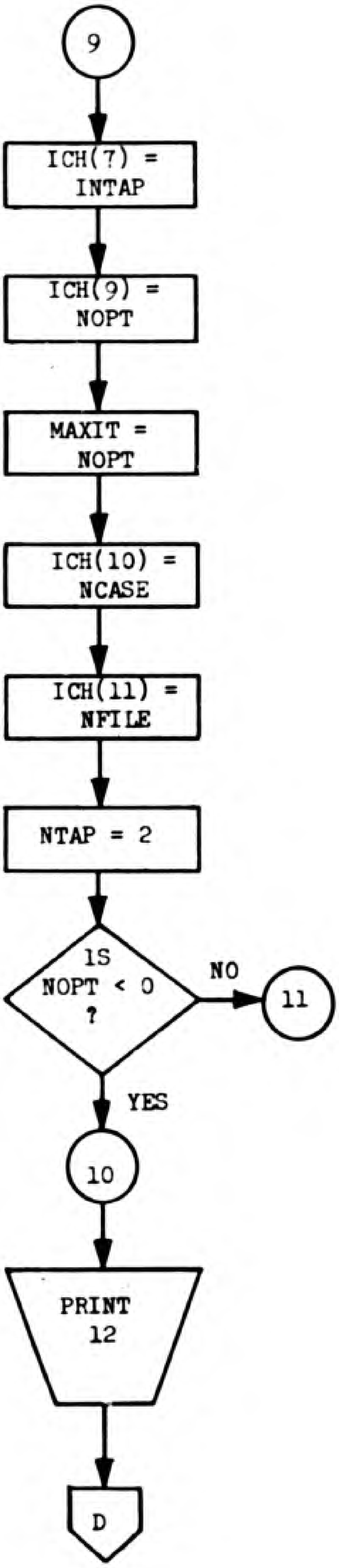


SKIP AN END OF FILE MARK.

READ FLOW FIELD DATA.

RECOVER INTEGER CASE DATA VARIABLES.

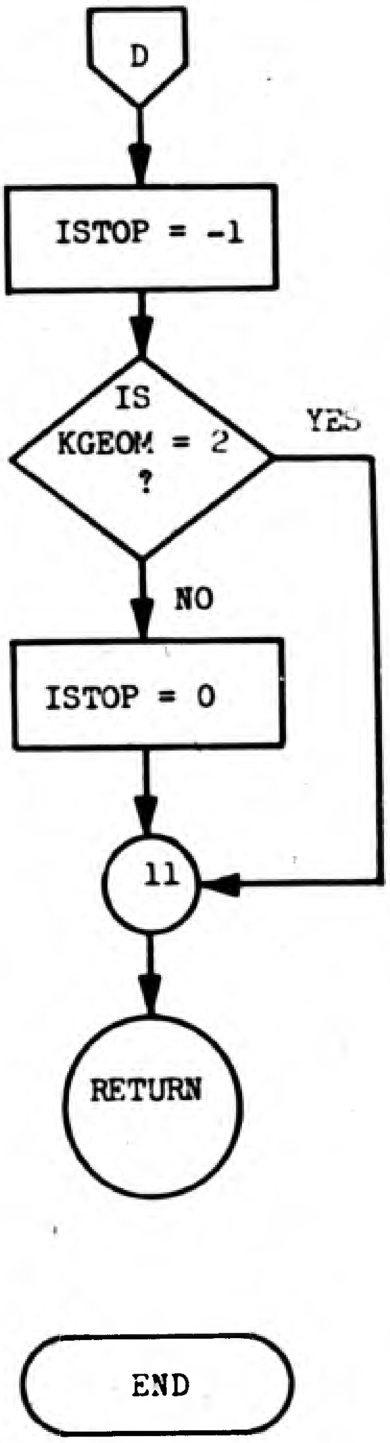
RECOVER FLOATING POINT CASE DATA VARIABLES.



RECOVER REMAINING CASE DATA VARIABLES AND SET REQUIRED CONSTANTS.

ERROR IN CASE DATA ?

PRINT ERROR MESSAGE.



SET ERROR INDICATOR.

ARBITRARY BODY OPTION ?

ALL DATA FOR THIS CASE WAS READ. CHANGE ERROR INDICATOR.

RETURN.

END OF SUBROUTINE LOAD.

```
      SUBROUTINE LOAD
C     THIS ROUTINE LOADS THE FLOW FIELD DATA FROM TAPE TO PERFORM
C       AN ITERATION ON A PREVIOUS CASE.
      COMMON /1/ U(15,41,4),R(41,4),F(7,41)
      COMMON /2/ G(105)
      COMMON /3/ MAX(6)
      COMMON /4/ ICUM(2),KGEOM,A(6),KSEC
      COMMON /5/ DEL(5)
      COMMON /6/ FS(3)
      COMMON /7/ ICT(6)
      COMMON /9/ TSEC,CC(4)
      COMMON /10/ ST(7)
      COMMON /11/ MAXIT,ISTOP,ITC,TIME
      COMMON /INPUT/ DUMM(6),CH(7),ICH(11),NSEG,ISTOP,NGEOM,NTAP
      COMMON /CONST/ D,X(3)
      INTEGER OTAP
      DO 1 I=1,6
1     IST(I)=ICH(I)
      INTAP=ICH(7)
      OTAP=ICH(8)
      NOPT=ICH(9)
      NCASE=ICH(10)
      NFILE=ICH(11)
      DO 2 I=1,7
2     ST(I)=CH(I)
      IF (NCASE.EQ.1) GO TO 5
      M=NCASE-1
      DO 4 I=1,M
3     READ (INTAP)
      CALL SKIP (INTAP,1)
      IF (I.GE.M) GO TO 3
4     CONTINUE
5     CALL INTAPE (0,M)
      ICH(8)=OTAP
      CALL OTAPE (0)
      N=NFILE-1
      IF (N.LT.1) GO TO 7
      DO 6 I=1,N
      CALL SKIP (INTAP,1)
      CALL INTAPE (1,M)
6     CALL OTAPE (1)
7     CALL SKIP (INTAP,1)
      CALL INTAPE (1,M)
      DO 8 I=1,6
8     ICH(I)=IST(I)
      DO 9 I=1,7
9     CH(I)=ST(I)
      ICH(7)=INTAP
      ICH(9)=NOPT
      MAXIT=NOPT
      ICH(10)=NCASE                                                     LCAD  53
      ICH(11)=NFILE                                                     LCAD  54
      NTAP=2                                                            LCAD  55
      IF (NOPT) 10,11,11                                                LCAD  56
10    PRINT 12                                                          LCAD  57
      ISTOP=-1                                                          LCAD  58
11    DUM(5)=CH(1)/(CH(2)*CH(5))                                        LCAD  59
      RETURN                                                            LCAD  60
C                                                                       LCAD  61
12    FORMAT (//30X,19HERROR IN INPUT DATA/30X,25HNTAP=2 REQUIRES NOPT.GLCAD  62
     *E.0/30X,15HCASE TERMINATED)                                       LCAD  63
      END                                                               LCAD  64
```

15. SUBROUTINE NMESH

A. PURPOSE

This subroutine establishes the nodal spacing across the shock layer. The nodal spacing used is selected according to the following order of priortiy:

(1) The value, if any, specified in the case data.

(2) The value, if any, specified in the body geometry data.

(3) Empirical relations depending on the flow conditions, the initial guess for shock shape and the body bluntness parameter.

The number of nodes selected is checked against the maximum number allowed by the dimension statements. If too many have been selected, the maximum number allowed is used. If this occurs, it will be indicated by a printed message in the output for the segment. The flow field variables are interpolated from the old nodal geometry to the new nodal geometry using first and second control differences in a truncated Taylor series.

B. VARIABLE LIST

Variables previously defined in the description of the main program have been omitted from this list.

ANS - value of n_s to be used to establish the nodal geometry under option (1).

D - array used for temporary storage of the interpolated flow field variables.

DE - variable used to store the value of $\Delta\eta$ used in the old nodal spacing.

DEL - $\frac{\eta-\eta_o}{2\Delta\eta}$, used for interpolation.

DEL2 - $\frac{1}{2}\left[\frac{\eta\ \eta_o}{\Delta\eta}\right]^2$, used for interpolation.

I1 - index of the first streamwise station to be considered.

I2 - index of the last streamwise station to be considered.

K - index for nodes across the shock layer in the old nodal geometry.

KM - K-1

KP - K+1

M - index specifying the proper flow parameters in the D array.

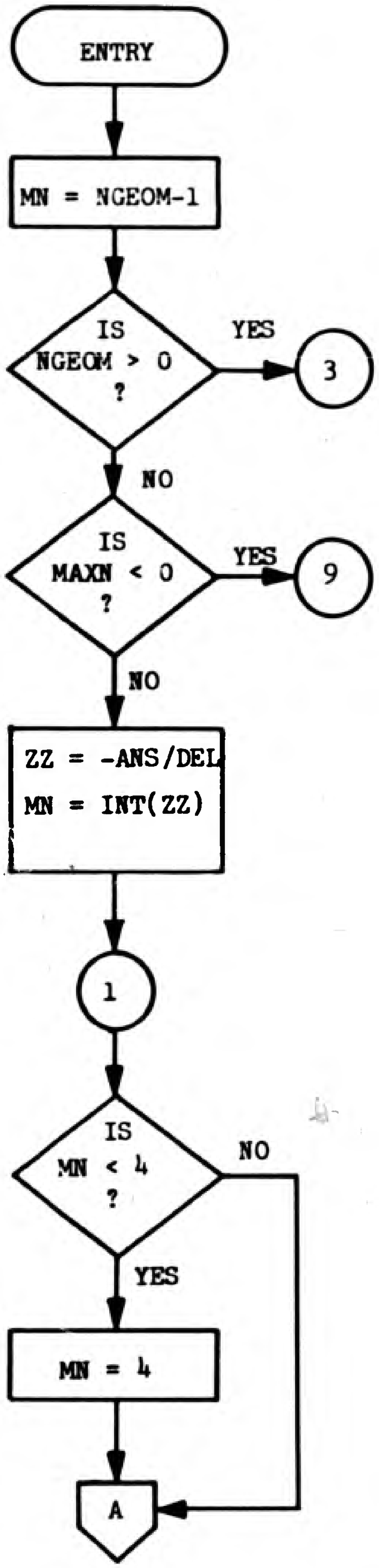
NST - number of nodes across the shock layer used in the old nodal geometry.

VE - variable containing a term required for the first central difference.

VE2 - variable containing a term required for the second central difference.

ZZ - dummy variable used for intermediate calculations.

C. FLOW CHART



ENTER SUBROUTINE NMESH

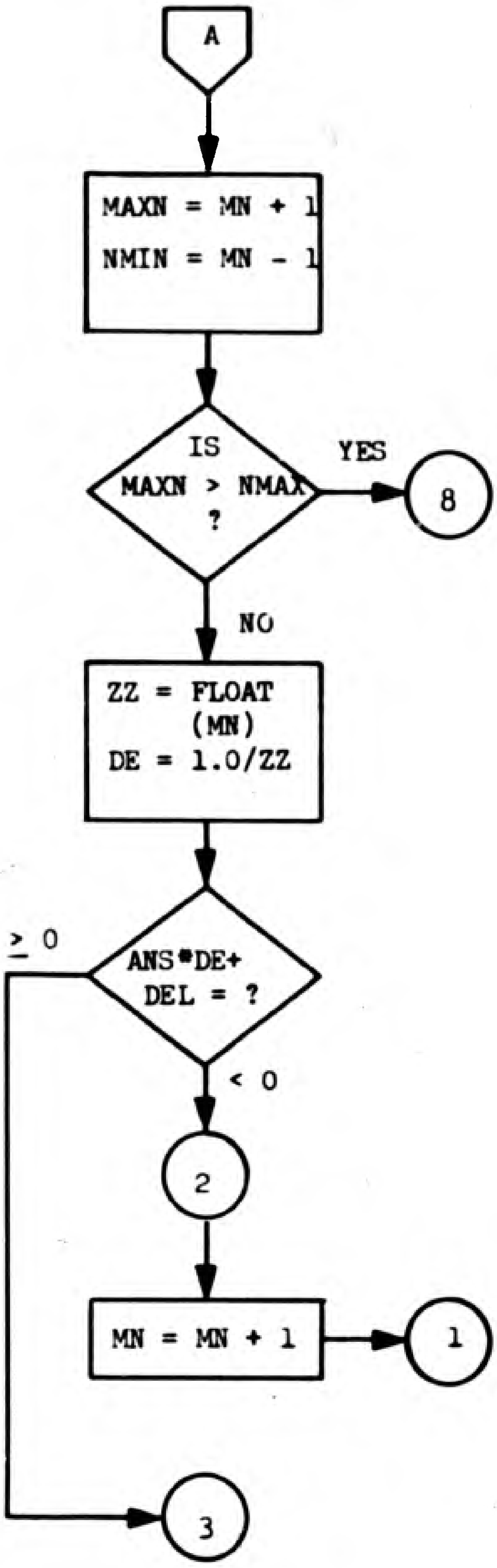
WAS MAXN SPECIFIED IN THE CASE DATA ?

WAS MAXN SPECIFIED WHEN THE BODY GEOMETRY DATA WAS READ ?

ESTIMATE MN.

IS MN < 4 ?

REQUIRE MN ≥ 4.



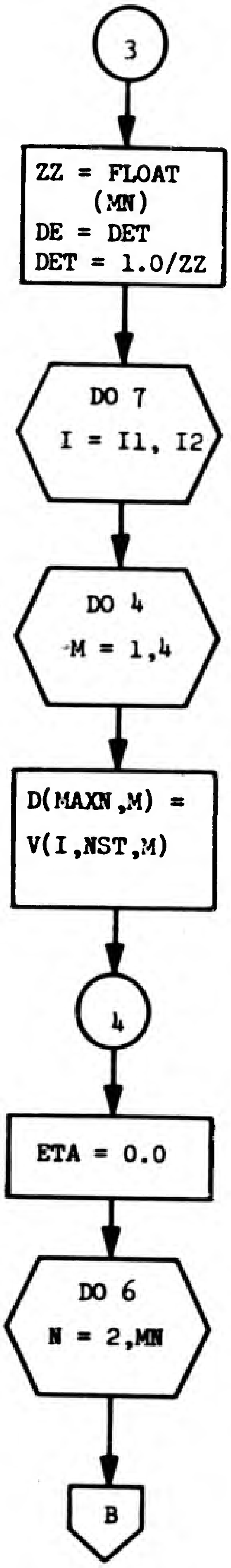
COMPUTE MAXN AND NMIN.

HAVE DIMENSIONS BEEN EXCEEDED ?

COMPUTE A NEW Δη.

IS THIS Δη SMALL ENOUGH TO PROVIDE THE REQUIRED Δη ?

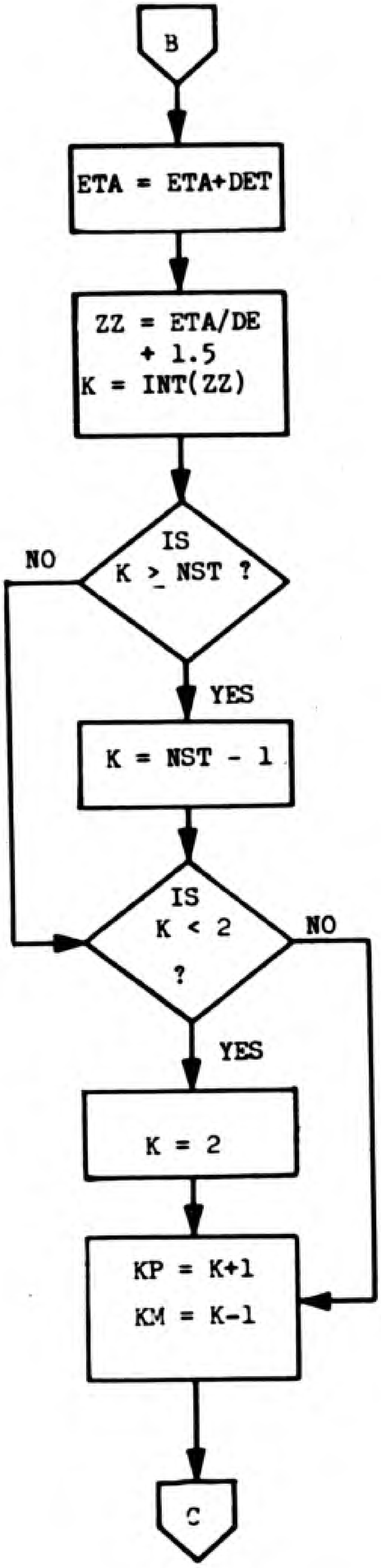
INCREMENT MN.



SAVE THE RESIDENT VALUE OF $\Delta\eta$ AND COMPUTE THE NEW VALUE.

STORE SHOCK CONDITIONS AT THE NEW N = MAXN LOCATION.

INITIALIZE η.



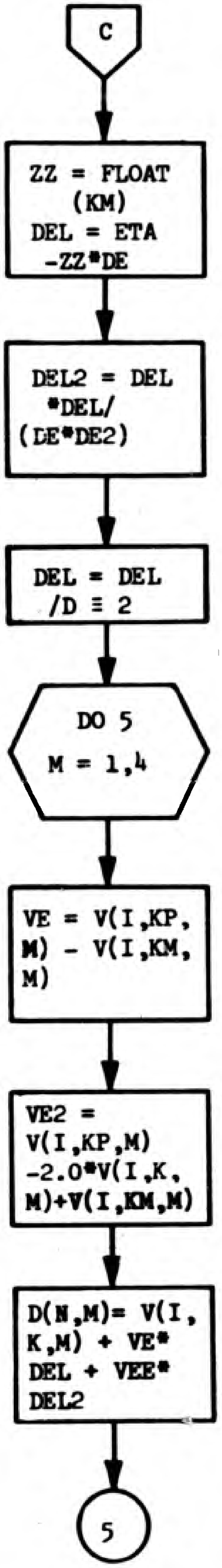
INCREMENT η.

ESTIMATE THE VALUE OF THE INDEX OF THE NODE IN THE RESIDENT NODAL GEOMETRY CLOSEST TO THE REQUIRED VALUE OF η.

IS K $\geq$ NST ?

IS K < 2 ?

COMPUTE KP AND KM.


C
ZZ = FLOAT (KM) DEL = ETA -ZZ*DE
DEL2 = DEL *DEL/ (DE*DE2)
DEL = DEL /D ≡ 2
DO 5 M = 1,4
VE = V(I,KP, M) - V(I,KM, M)
VE2 = V(I,KP,M) -2.0*V(I,K, M)+V(I,KM,M)
D(N,M)= V(I, K,M) + VE* DEL + VEE* DEL2
5

COMPUTE $\eta_N - \eta_K$

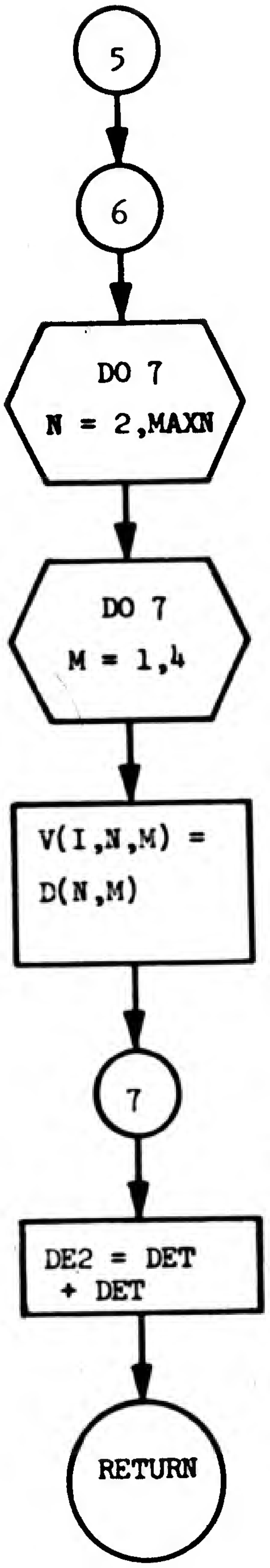
COMPUTE $\frac{(\eta_N - \eta_K)^2}{2\ (\Delta\eta)^2}$

COMPUTE $\frac{\eta_N - \eta_K}{2\Delta\eta}$

COMPUTE $f(\eta+\Delta\eta) - f(\eta-\Delta\eta)$

COMPUTE $f(\eta+\Delta\eta) - 2f(\eta)$
$+ f(\eta-\Delta\eta)$

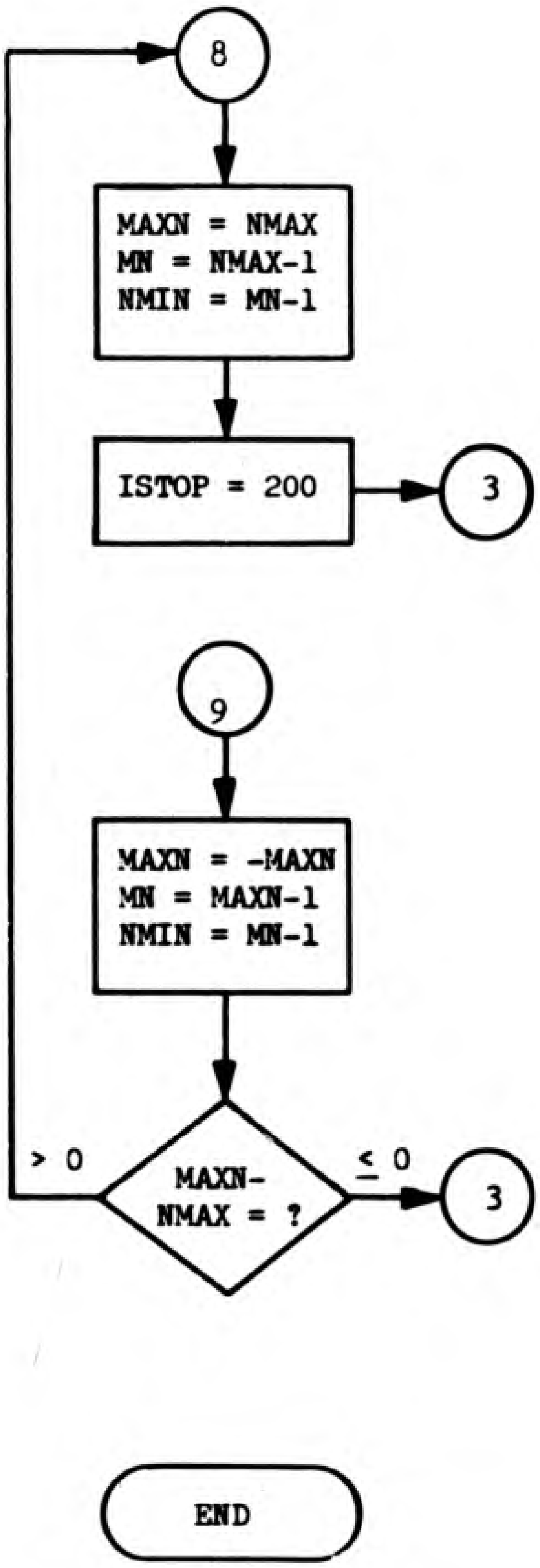
$f = f_o + f_\eta\ \Delta\eta + \frac{1}{2}\ f_{\eta\eta}\ (\Delta\eta)^2$



TRANSFER ρ, U, V AND S FROM TEMPORARY TO PERMANENT STORAGE.

COMPUTE 2Δη.

RETURN.



DIMENSIONS EXCEEDED.

USE MAXIMUM NUMBER OF
NODES AVAILABLE.

SET ERROR INDICATOR TO
PROVIDE AN ERROR MESSAGE
DURING PRINTOUT.

MAXN WAS SPECIFIED WHEN
BODY GEOMETRY WAS READ.

USE THE VALUE READ.

EXCEEDED DIMENSIONS ?

END OF SUBROUTINE NMESH.

D. LISTING

```
      SUBROUTINE NMESH (ANS,I1,I2,DEL,NST)
C
C      THIS ROUTINE COMPUTES THE PROPER NUMBER OF NODES IN THE NORMAL
C     DIRECTION TO BE EMPLOYED BASED ON DATA FROM RESET.
C
      COMMON /1/ V(15,40,4),D(40,4),P(3,40)
      COMMON /3/ MAXI,MI,IMIN,MAXN,MN,NMIN
      COMMON /5/ DXI,DET,DX2,DE2,DT
      COMMON /11/ MAXIT,ISTOP,ITD,TIME
      COMMON /INPUT/ A(13),IV(13),NGEOM
      COMMON /MESH/ IMAX,NMAX
      MN=NGEOM-1
      IF (NGEOM.GT.0) GO TO 3
      IF (MAXN.LT.0) GO TO 9
      ZZ=-ANS/DEL
      MN=INT(ZZ)
1     IF (MN.LT.4) MN=4
      MAXN=MN+1
      NMIN=MN-1
      IF (MAXN.GT.NMAX) GO TO 8
      ZZ=FLOAT(MN)
      DE=1.0/ZZ
      IF (ANS*DE+DEL) 2,3,3
2     MN=MN+1
      GO TO 1
3     ZZ=FLOAT(MN)
      DE=DET
      DET=1.0/ZZ
      DO 7 I=I1,I2
      DO 4 M=1,4
4     D(MAXN,M)=V(I,NST,M)
      ETA=1.0
      DO 6 N=2,MN
      ETA=ETA+DET
      ZZ=ETA/DE+1.5
      K=INT(ZZ)
      IF (K.GE.NST) K=NST-1
      IF (K.LT.2) K=2
      KP=K+1
      KM=K-1
      ZZ=FLOAT(KM)
      DEL=ETA-ZZ*DE
      DEL2=DEL*DEL/(DE*DE2)
      DEL=DEL/DE2
      DO 5 M=1,4
      VE=V(I,KP,M)-V(I,KM,M)
      VE2=V(I,KP,M)-2.0*V(I,K,M)+V(I,KM,M)
5     D(N,M)=V(I,K,M)+VE*DEL+VE2*DEL2
6     CONTINUE
      DO 7 N=2,MAXN
      DO 7 M=1,4
7     V(I,N,M)=D(N,M)
```

```
          N=NET+1                                                          NMESH  53
          RETURN                                                           NMESH  54
    4     MAXM=NMAX                                                        NMESH  55
          MM=MAXM-1                                                        NMESH  56
          MITM=MM-1                                                        NMESH  57
          IERCD=211                                                        NMESH  58
          GO TO 3                                                          NMESH  59
    2     MAXM=-MAXN                                                       NMESH  60
          MM=MAXM-1                                                        NMESH  61
          MXTM=MM-1                                                        NMESH  62
          IF (MAXM-NMAX) 3,3,3                                             NMESH  63
          END                                                              NMESH  64
```

16. SUBROUTINE OTAPE

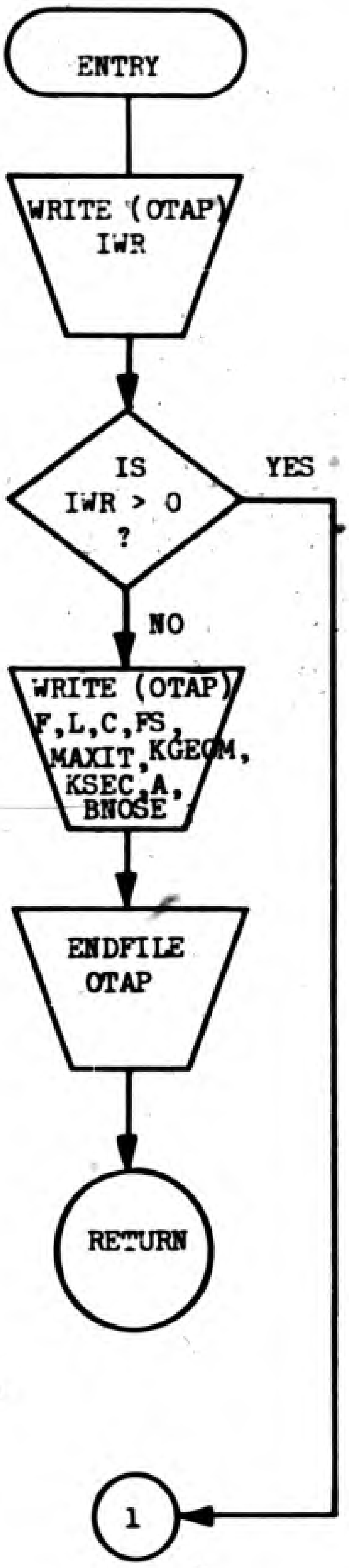
A. PURPOSE

This subroutine performs all output operations to the magnetic tape unit specified for storing the flow field data. By isolating the READ and WRITE statements in subroutines, access to this logic from several locations in the computer code is possible, and it is easily assured the binary read and write operations are consistent. A control variable, IWR, specifies whether case data or flow field data are to be written on tape. This argument should not be negative since it is written on the tape. A negative value would be interpreted as the end of a case if the data were later recovered from tape.

B. VARIABLE LIST

Most variables in this routine are written on tape only. Their definitions can be found in the description of the main program by matching locations in the various COMMON blocks. The control variable, IWR, has been described above. OTAP is the magnetic tape unit number on which the data is to be stored.

C. FLOW CHART



ENTER SUBROUTINE OTAPE.

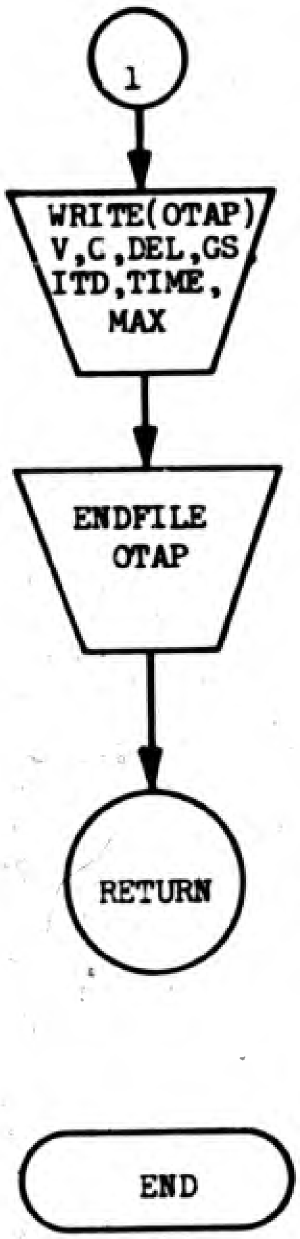
WRITE INDICATOR ON TAPE.

IS THIS ENTRY TO WRITE FLOW FIELD DATA FOR A SEGMENT ?

WRITE CASE DATA AND CONSTANTS.

ENDFILE OTAP.

RETURN.



WRITE FLOW FIELD DATA
FOR THIS SEGMENT.

ENDFILE OTAP.

RETURN.

END OF SUBROUTINE OTAPE.

D. LISTING

```
      SUBROUTINE OTAPE (IWR)
C
C      THIS ROUTINE WRITES THE FLOW DATA ON TAPE.
C
      COMMON /1/ V(15,40,4),D(40,4),P(3,40)
      COMMON /2/ G(103)
      COMMON /3/ MAX(6)
      COMMON /4/ IDUM(2),KGEOM,A(6),KSEG
      COMMON /5/ DEL(3)
      COMMON /6/ FS(3)
      COMMON /9/ ISEG,GS(3),BNOSE
      COMMON /11/ MAXIT,ISTOP,ITD,TIME
      COMMON /INPUT/ F(13),L(15)
      COMMON /CONST/ DUM(2),C(3)
      INTEGER OTAP
      EQUIVALENCE (L(8),OTAP)
      WRITE (OTAP) IWR
      IF (IWR.GT.1) GO TO 1
      WRITE (OTAP) F,L,C,FS,MAXIT,KGEOM,KSEG,A,BNOSE
      END FILE OTAP
      RETURN
1     WRITE (OTAP) V,G,DEL,GS,ITD,TIME,MAX
      END FILE OTAP
      RETURN
      END
```

17. SUBROUTINE RESET

A. PURPOSE

This subroutine schedules the calculation of initial conditions and body geometry parameters for all segments except the first. This data may be generated by calls to other routines or read from magnetic tape, depending on the option specified in the input data (see section VII). A maximum value of $\Delta\eta$ is computed for use in subroutine NMESH. It is based on the signs of the pressure gradient and the normal velocity component of the first node above the body on the previous segment. If the number of nodes across the shock layer has been specified in the input data, this quantity will be ignored.

B. VARIABLE LIST

Variables previously defined in the description of the main program have been omitted from this list.

DUM - array containing the value of ξ at the starting line of a segment and the segment length, measured along the body.

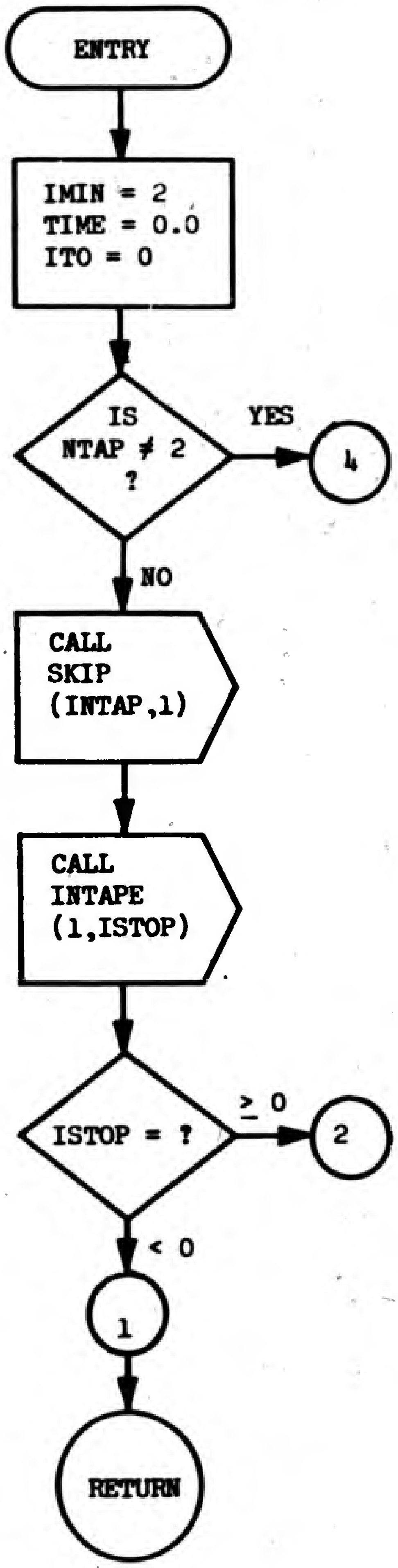
IST - index of the last streamwise node on the previous segment.

M - dummy index.

NST - number of nodes across the shock layer used on the previous segment.

ZZ - dummy variable used for intermediate computations.

C. FLOW CHART



ENTER SUBROUTINE RESET.

INITIALIZE VARIABLES FOR
A NEW SEGMENT.

NEW CASE ?

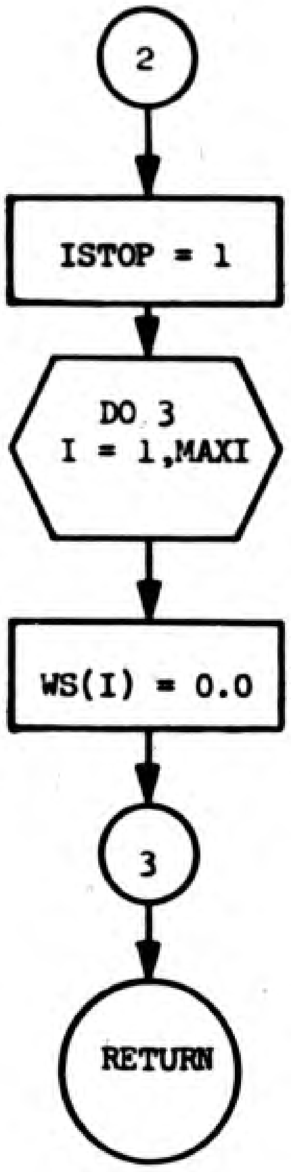
SKIP AN END OF FILE MARK.

READ FLOW FIELD DATA
FROM TAPE.

END OF CASE ?

YES.

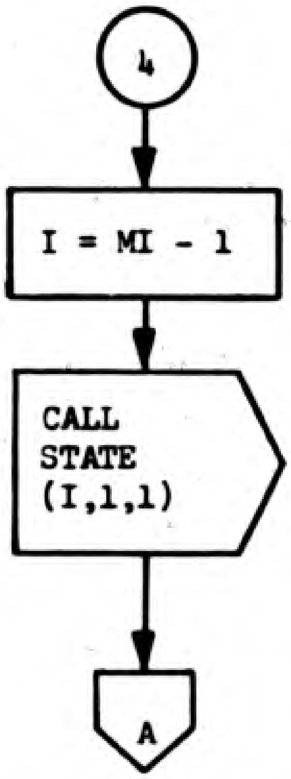
RETURN.



SET ERROR INDICATOR.

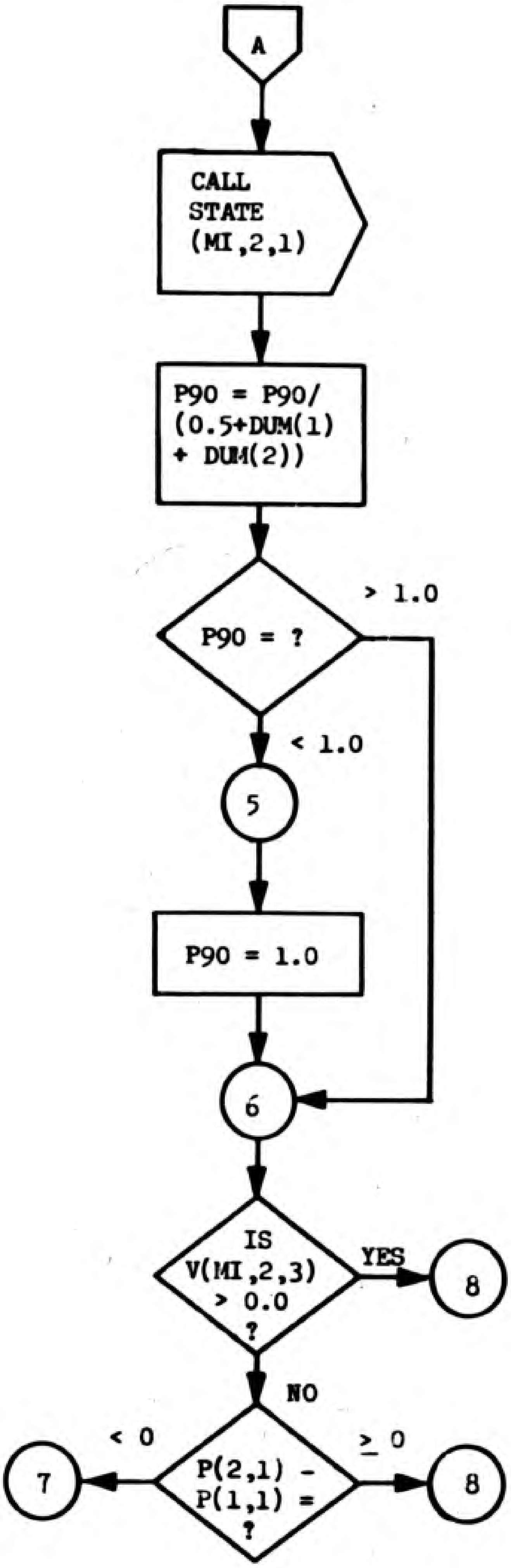
SET SHOCK VELOCITY TO ZERO.

RETURN.



NEW CASE.

COMPUTE BODY PRESSURE
AT MI - 1.



COMPUTE BODY PRESSURE AT MI.

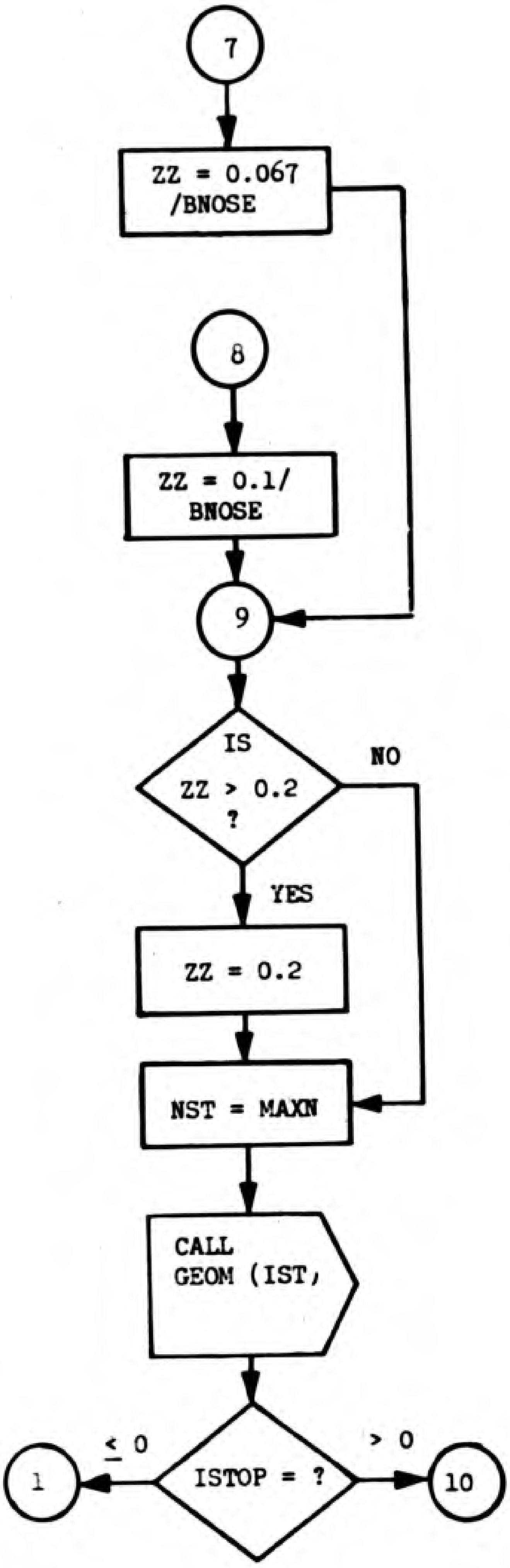
COMPUTE P90.

IS P90 < 1.0 ?

REQUIRE P90 ≥ 1.0

IS V POSITIVE AT THE FIRST NODE ABOVE THE BODY AT THE END OF THE LAST SEGMENT ?

WHAT IS THE SIGN OF THE PRESSURE GRADIENT ?



FINE NODAL SPACING REQUIRED.

COARSE NODAL SPACING ALLOWABLE.

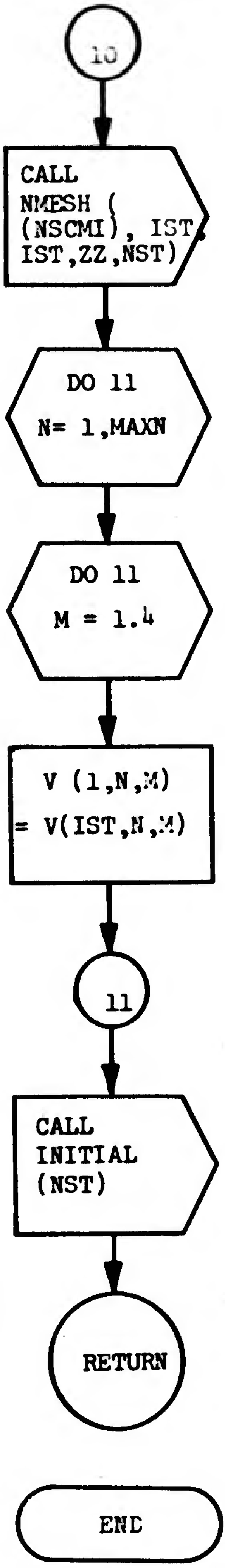
WILL Δη BE > 0.2 ?

REQUIRE Δη ≤ 0.2

SAVE THE VALUE OF MAXN

COMPUTE BODY GEOMETRY FOR THE NEXT SEGMENT.

ERROR SENSED IN SUBROUTINE GEOM ?



NO ERRORS.

ESTABLISH NODAL SPACING ACROSS THE SHOCK LAYER.

TRANSFER THE VALUES OF THE FLOW PARAMETERS AT THE END OF THE PREVIOUS SEGMENT TO THE I = 1 LOCATION IN THE ARRAY, V.

COMPUTE INITIAL CONDITIONS FOR OTHER NODES.

RETURN.

END OF SUBROUTINE RESET.

9. LISTING

```
      SUBROUTINE RESET
C
C      THIS ROUTINE SETS UP THE INITIAL DATA FOR ALL SEGMENTS
C     AFTER THE FIRST.
C
      COMMON /1/ V(15,40,4),D(40,4),P(3,40)
      COMMON /2/ RF(15),ZR(15),TH(15),CTH(15),K(15),NS(15),KS(15)
      COMMON /3/ MAXT,MT,IMIN,MAXN,MN,NMIN
      COMMON /4/ ICUM(2),KGEOM,A(6),KSEG
      COMMON /5/ DEL(3)
      COMMON /6/ ES(3)
      COMMON /9/ ISEG,DUM(2),KZERO,RNOSE
      COMMON /11/ MAXIT,ISTOP,ITC,TIME
      COMMON /INPUT/ GAM1,AMACH1,THETC,ZRN,ICRIT(2),P1,RHO1,PN,AMUP,T1,F
     *RR(2),IPR(5),INTAP,OTAP,NOPT,NCASE,NFILE,NSEG,IGEOM,NGEOM,NTAE
      COMMON /CONST/ DUMM(3)
      COMMON /PRES/ PT2,P02
      REAL NS,K,KZERO
      IMIN=2
      TIME=0.0
      ITC=0
      IF (INTAP.NE.0) GO TO 4
      CALL SKIP (INTAP,1)
      ZZ=NS(MT)
      CALL INTAPE (1,ISTOP)
      IF (ISTOP) 1,2,2
1     RETURN
2     DO 3 I=1,MAXT
3     NS(I)=0.0
      GO TO 12
4     I=MT-1
      CALL STATE (I,1,1)
      CALL STATE (MT,2,1)
      P02=P02/(0.5*DUM(1)+DUM(2))
      IF (P02-1.0) 5,6,6
5     P02=1.0
6     IF (V(MT,2,3).GT.0.1) GO TO 8
      IF (D(2,1)-D(1,1)) 7,8,8
7     ZZ=0.057/RNOSE
      GO TO 9
8     ZZ=0.1/RNOSE
9     IF (ZZ.GT.0.2) ZZ=0.2
      NST=MAXN
      CALL GEOM (IST)
      IF (ISTOP) 1,1,10
10    CALL NMESH (NS(MT),IST,IST,ZZ,NST)
      DO 11 N=1,MAXN
      DO 11 M=1,4
11    V(1,N,M)=V(IST,N,M)
      CALL INITIAL (NST)
      RETURN
12    ISTOP=1
      NS(1)=ZZ
      RETURN
      END
```

18. SUBROUTINE RESTART

A. PURPOSE

This subroutine recovers data stored on magnetic tape to restart an unfinished case. Since it is difficult to estimate the computation time required for a particular problem, this capability is necessary. Should the job be terminated prior to completion, computations can be resumed with very little loss of computer time. The magnetic tape is searched for the correct case, the case data input is read and files are skipped until the flow field data for the last completed segment is read. Computations resume for the next segment.

B. VARIABLE LIST

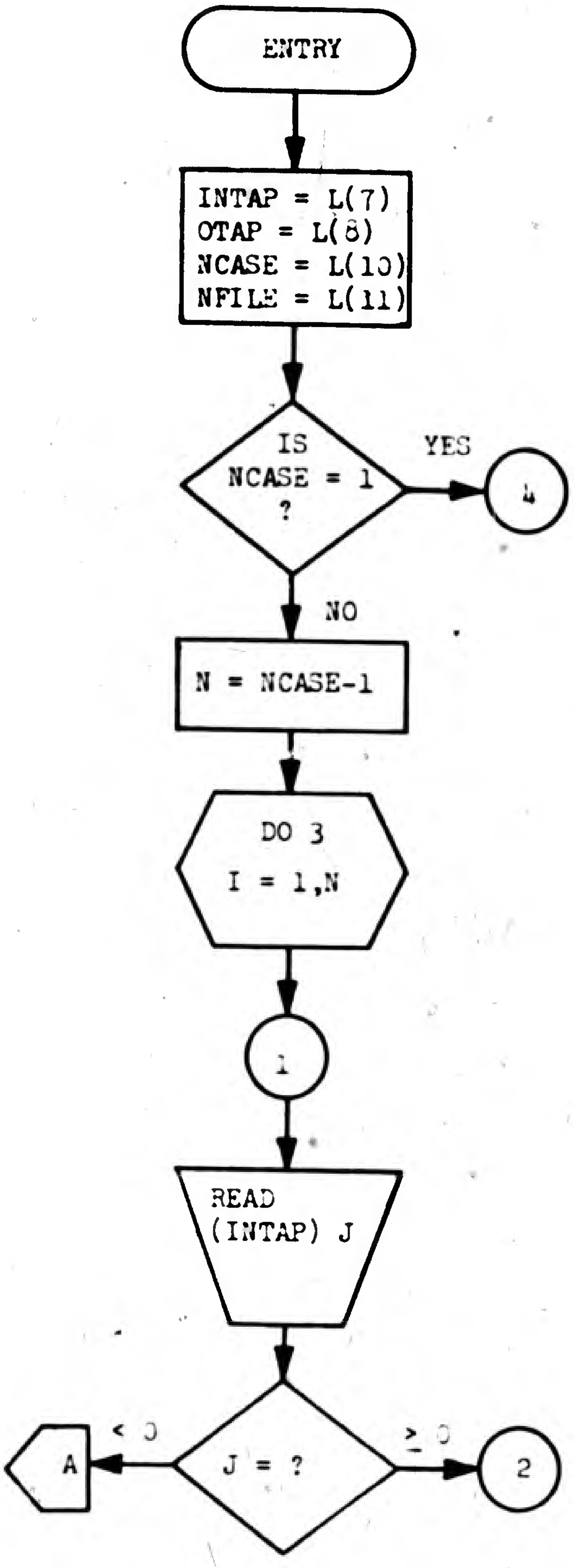
Variables previously defined in the description of the main program have been omitted from this list.

J - variable read from magnetic tape. A negative value indicates the end of a case.

L - array containing all integer variables in COMMON block /INPUT/.

M - dummy variable in a subroutine call statement.

C. FLOW CHART



ENTER SUBROUTINE RESTART.

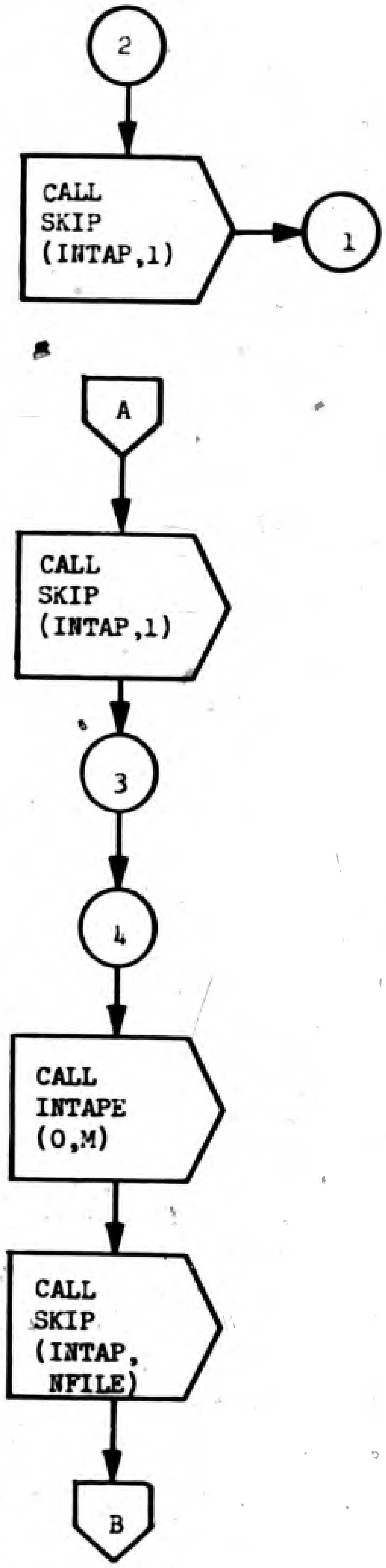
SAVE CASE DATA VARIABLES WHICH ARE TO BE CHANGED FROM THEIR VALUES ON TAPE.

IS THE UNFINISHED CASE THE FIRST ONE STORED ON THE TAPE ?

DO LOOP TO SKIP ALL CASES PRIOR TO THE UNFINISHED CASE.

READ END OF CASE INDICATOR.

WAS THIS THE LAST FILE OF THE CASE ?



SKIP END OF FILE MARK AND RETURN TO 1 TO READ ADDITIONAL FILES.

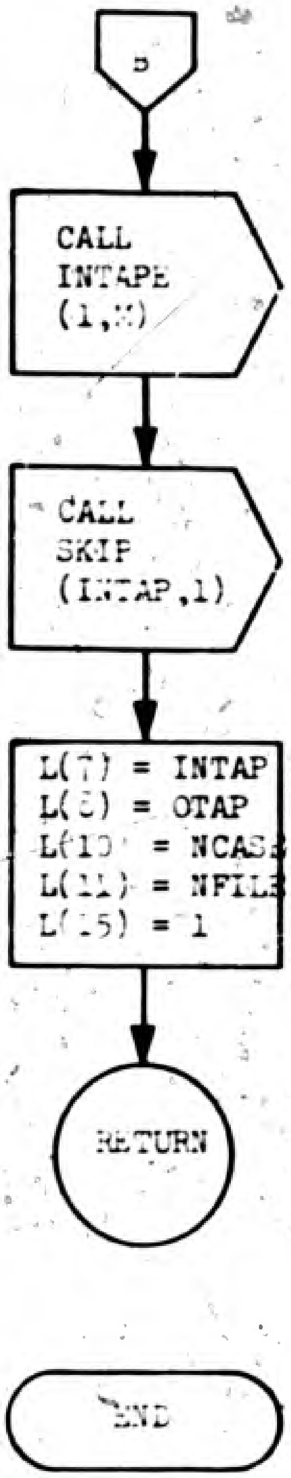
LAST FILE IN THIS CASE.

SKIP END OF FILE MARK.

CORRECT CASE HAS BEEN LOCATED.

READ CASE DATA AND CONSTANTS FROM TAPE.

SKIP FILES TO THE ONE CONTAINING THE LAST SEGMENT COMPLETED.



READ FLOW FIELD DATA.

SKIP END OF FILE MARK.

RECOVER CASE DATA
VARIABLES SAVED.

RETURN.

END OF SUBROUTINE RESTART.

D. LISTING

```
      SUBROUTINE RESTART
C
C      THIS ROUTINE READS THE DATA FROM TAPE TO RESTART AN
C     UNFINISHED CASE.
C
      COMMON /1/ V(15,40,4),D(40,4),P(3,40)
      COMMON /2/ G(105)
      COMMON /3/ MAX(6)
      COMMON /4/ IDUM(2),KGEOM,A(6),KSEG
      COMMON /5/ DEL(5)
      COMMON /6/ FS(3)
      COMMON /9/ ISEG,DUM(4)
      COMMON /11/ MAXIT,ISTOP,ITD,TIME
      COMMON /INPUT/ F(13),L(15)
      COMMON /CONST/ DUMM(5)
      INTEGER OTAP
      INTAP=L(7)
      OTAP=L(8)
      NCASE=L(10)
      NFILE=L(11)
      IF (NCASE.EQ.1) GO TO 4
      N=NCASE-1
      DO 3 I=1,N
1     READ (INTAP) J
      IF (J) 3,2,2
2     CALL SKIP (INTAP,1)
      GO TO 1
3     CALL SKIP (INTAP,1)
4     CALL INTAPE (0,M)
      CALL SKIP (INTAP,NFILE)
      CALL INTAPE (1,M)
      CALL SKIP (INTAP,1)
      L(7)=INTAP
      L(8)=OTAP
      L(10)=NCASE
      L(11)=NFILE
      L(15)=1
      RETURN
      END
```

19. SUBROUTINE SHOCK

A. PURPOSE

This subroutine performs all computations for nodes at the shock using the method presented in section IV.

B. VARIABLE LIST

Variables previously defined in the description of the main program have been omitted from this list.

A - weighting function, equation (45).

AA - sound speed at the node defined by N.

AE - $\frac{\partial a}{\partial \xi}$ at the node defined by N.

ANS - $W_s \Delta r$

AS - average of the sound speeds at points 2 and 4 (Fig. 4).

ATSTAG- Assigned a value of 2.0 at the symmetry axis, and a value of 1.0 at all other points.

AX - $\frac{\partial a}{\partial \xi}$ at the node defined by N.

A1 - value of A at the shock.

A2 - sound speed at point 2 (Fig. 4).

A4 - sound speed at point 4 (Fig. 4).

B - weighting function, equation (46).

B1 - value of B at the shock.

DE - difference in η between the node defined by N and other points at which interpolation is required.

DS - distance parallel to the shock, used to compute A and B.

DSO - intermediate value of DS, used to ensure that DS is non-zero.

DTZ - $\Delta r/2$

DUM - dummy variable used for intermediate computations.

DX - difference in ξ between the node defined by N and points at which interpolation is required.

N - NP-1

NP - number of nodes across the shock layer. Identifies the node at the shock.

NM - NP-2

NPLUS - distance between points 2 and 4 (Fig. 4).

PE - $\frac{\partial P}{\partial \eta}$ at the node defined by N.

PX - $\frac{\partial P}{\partial \xi}$ at the node defined by N.

P2 - pressure at point 2 (Fig. 4).

P3 - intermediate value of P2.

P4 - pressure at point 4 (Fig. 4).

RE - $\frac{\partial \rho}{\partial \eta}$ at the node defined by N.

RHO - average of the values of ρ at points 2 and 4 (Fig. 4).

RX - $\frac{\partial \rho}{\partial \xi}$ at the node defined by N.

RO - value of ρ used to compute A.

UE - $\frac{\partial U}{\partial \eta}$ at the node defined by N.

UU - U at the node defined by N.

UX - $\frac{\partial U}{\partial \xi}$ at the node defined by N.

UO - value of U used to compute A.

U1 - free-stream value of U.

U4 - value of U at point 4 (figure 4).

VE - $\frac{\partial V}{\partial \eta}$ at the node defined by N.

VV - value of V at the node defined by N.

VX - $\frac{\partial V}{\partial \xi}$ at the node defined by N.

VO - value of V used to compute B.

V1 - free-stream value of V.

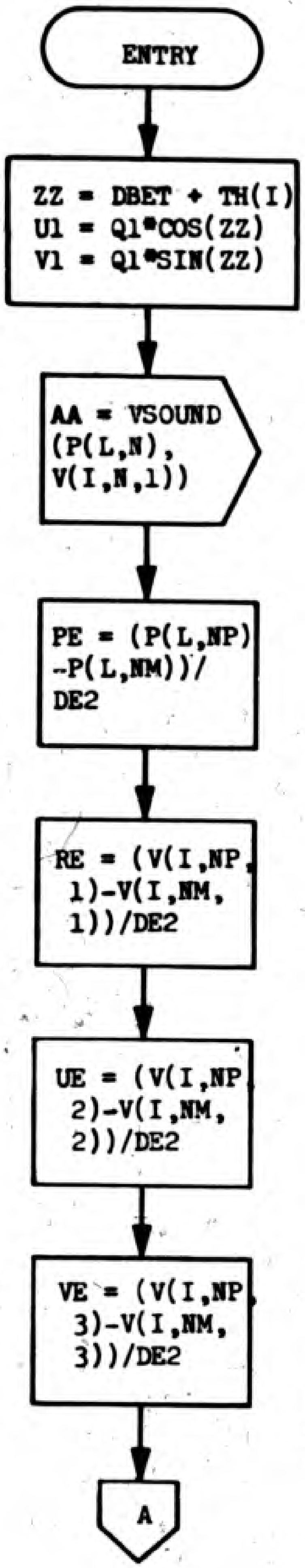
V2 - value of V at point 2 (Fig. 4).

V4 - value of V at point 4 (Fig. 4).

WSH - local shock velocity.

ZZ - dummy variable used for intermediate computations.

C. FLOW CHART



COMPUTE FREE STREAM VELOCITY COMPONENTS PARALLEL TO AND NORMAL TO THE SHOCK.

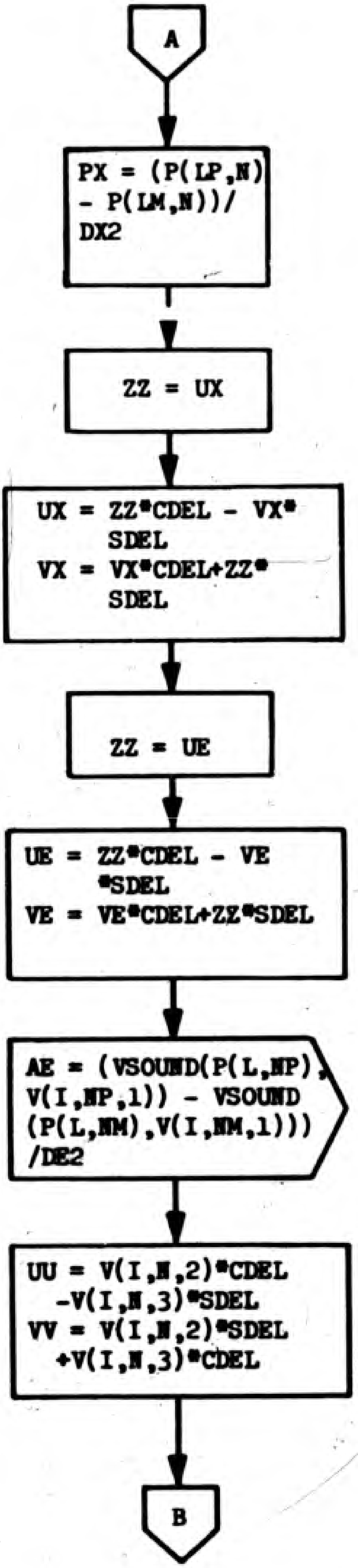
COMPUTE SOUND SPEED AT N.

$\frac{\partial P}{\partial \eta}$ AT N.

$\frac{\partial \rho}{\partial \eta}$ AT N.

$\frac{\partial U}{\partial \eta}$ AT N.

$\frac{\partial V}{\partial \eta}$ AT N.



```
PX = (P(LP,N)
- P(LM,N))/
DX2
```

$\frac{\partial P}{\partial \xi}$ AT N.

```
ZZ = UX
```

```
UX = ZZ*CDEL - VX*
     SDEL
VX = VX*CDEL+ZZ*
     SDEL
```

$\frac{\partial U}{\partial \xi}$ AND $\frac{\partial V}{\partial \xi}$ AT N.

```
ZZ = UE
```

```
UE = ZZ*CDEL - VE
     *SDEL
VE = VE*CDEL+ZZ*SDEL
```

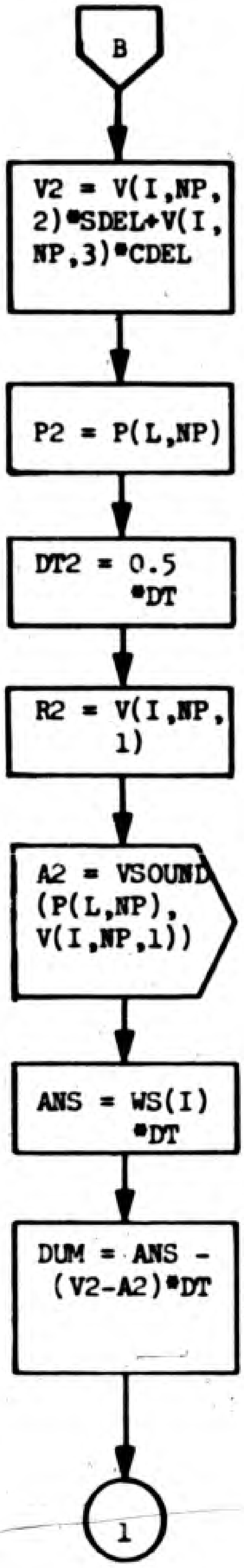
$\frac{\partial U}{\partial \eta}$ AND $\frac{\partial V}{\partial \eta}$ AT N.

```
AE = (VSOUND(P(L,NP),
V(I,NP,1)) - VSOUND
(P(L,NM),V(I,NM,1)))
/DE2
```

$\frac{\partial a}{\partial \eta}$ AT N.

```
UU = V(I,N,2)*CDEL
  -V(I,N,3)*SDEL
VV = V(I,N,2)*SDEL
  +V(I,N,3)*CDEL
```

U AND V AT N.



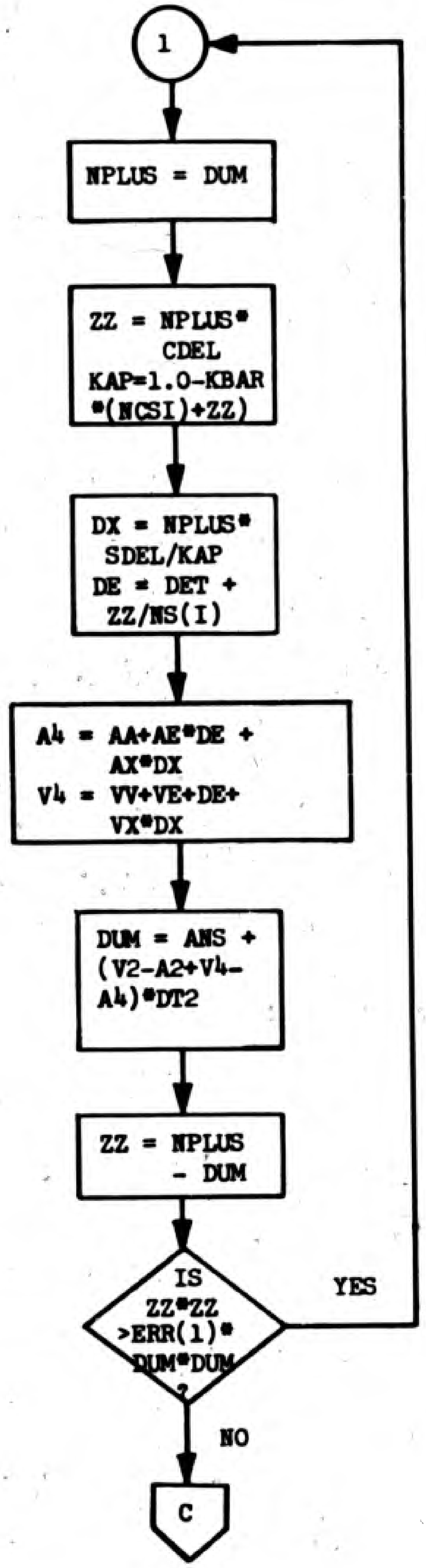
COMPUTE V AT THE SHOCK.

STORE P AT THE SHOCK.

STORE ρ AT THE SHOCK.

COMPUTE a AT THE SHOCK.

COMPUTE X_N.



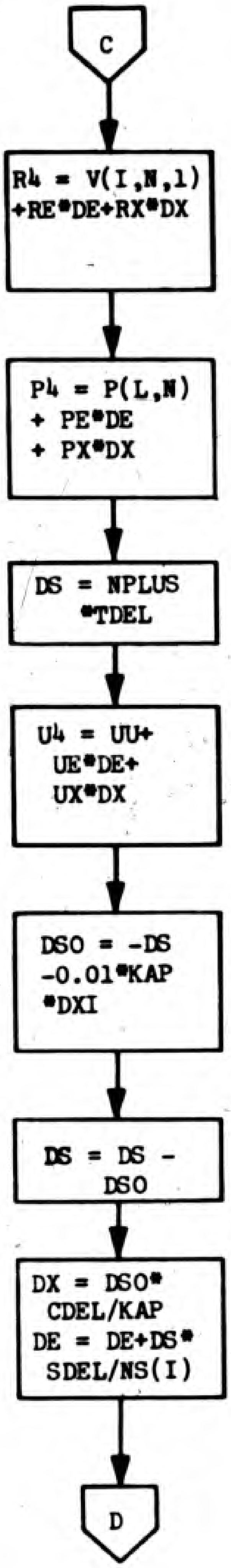
STORE X_N.

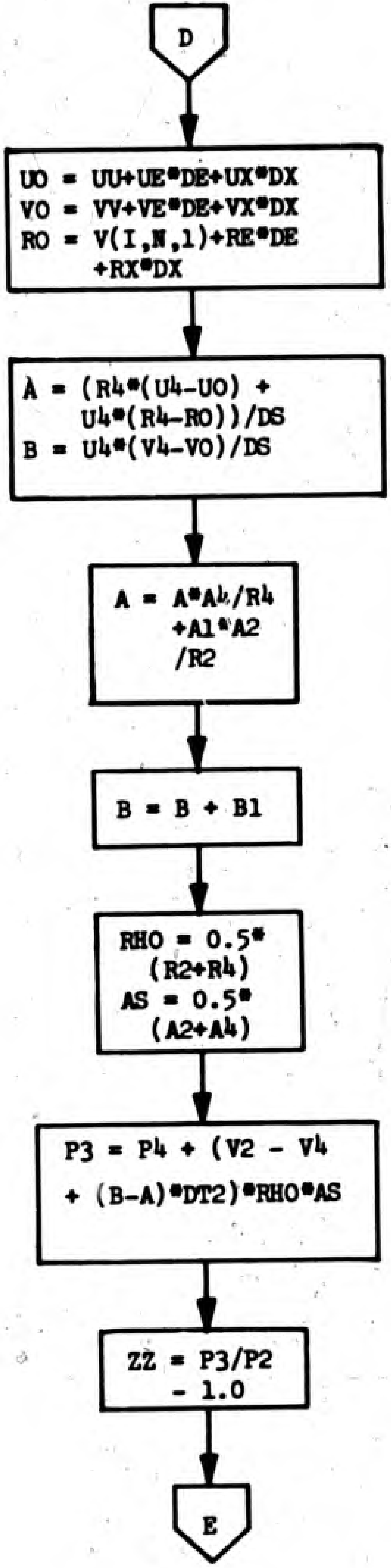
COMPUTE $\Delta\xi$ AND $\Delta\eta$ BETWEEN X_N AND THE POINT (I,N)

COMPUTE a_4 AND V_4.

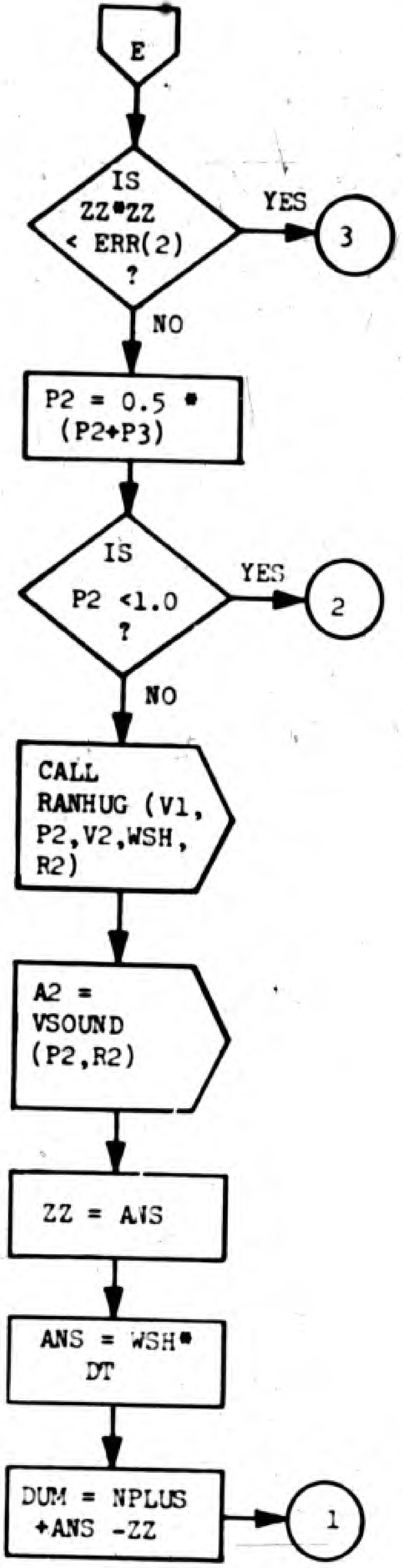
COMPUTE NEW X_N.

IS THE DIFFERENCE BETWEEN SUCCESSIVE VALUES OF X_N TOO LARGE ?


C
R4 = V(I,N,1) +RE*DE+RX*DX
P4 = P(L,N) + PE*DE + PX*DX
DS = NPLUS *TDEL
U4 = UU+ UE*DE+ UX*DX
DSO = -DS -0.01*KAP *DXI
DS = DS - DSO
DX = DSO* CDEL/KAP DE = DE+DS* SDEL/NS(I)
D


D
UO = UU+UE*DE+UX*DX
VO = VV+VE*DE+VX*DX
RO = V(I,N,1)+RE*DE
+RX*DX
A = (R4*(U4-UO) +
U4*(R4-RO))/DS
B = U4*(V4-VO)/DS
A = A*A4/R4
+A1*A2
/R2
B = B + B1
RHO = 0.5*
(R2+R4)
AS = 0.5*
(A2+A4)
P3 = P4 + (V2 - V4
+ (B-A)*DT2)*RHO*AS
ZZ = P3/P2
- 1.0
E

EQUATION (49)



DO TWO SUCCESSIVE VALUES OF SHOCK PRESSURE AGREE WITHIN THE ALLOWED ERROR ?

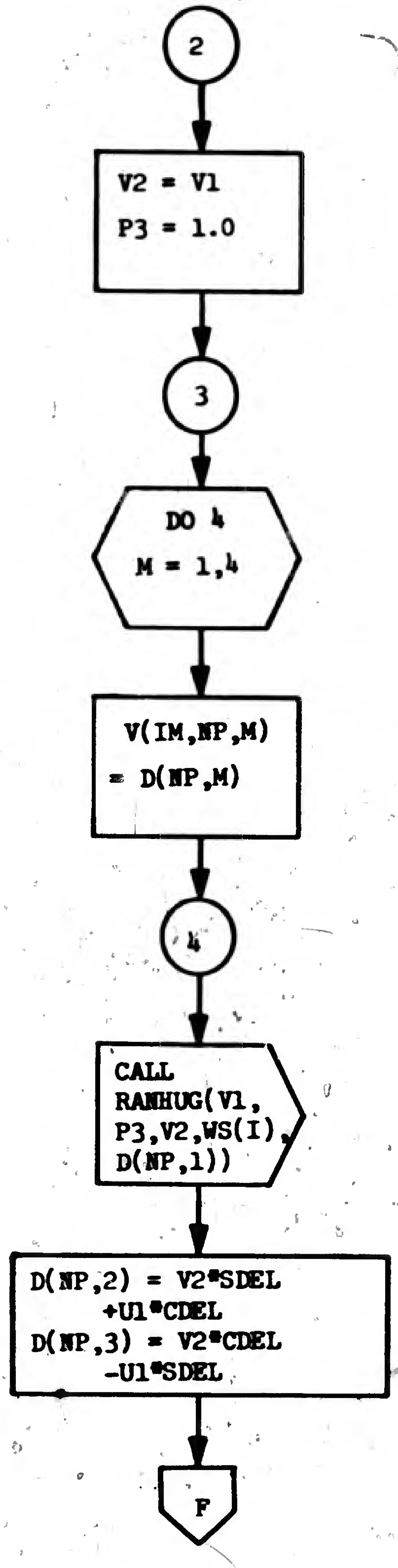
AVERAGE THE TWO VALUES.

IS THE SHOCK PRESSURE < 1.0 ?

SOLVE RANKINE-HUGONIOT RELATIONS.

COMPUTE NEW SOUND SPEED AT THE SHOCK.

COMPUTE NEW VALUE OF NPLUS.



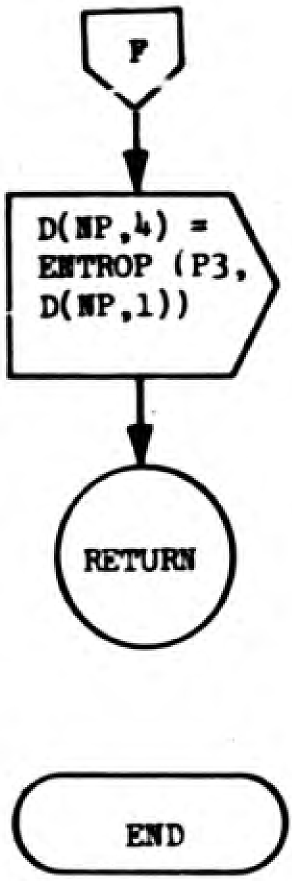
PRESSURE AT THE SHOCK WAS < 1.0. SHOCK IS A MACH WAVE.

SET NORMAL VELOCITY AND PRESSURE AT THE SHOCK EQUAL TO FREE STREAM VALUES.

TRANSFER ρ, U, V AND S FROM TEMPORARY TO PERMANENT STORAGE AT IM.

SOLVE RANKINE-HUGONIOT RELATIONS.

COMPUTE U AND V.



COMPUTE S AT THE SHOCK.

RETURN.

D. LISTING

```
      SUBROUTINE SHOCK (ATSTAG)
C
C      THIS ROUTINE TREATS THE SHOCK NODES USING THE METHOD SUGGESTED
C     BY MORETTI AND ABBETT, AIAA J. NO. 4, 2136-2141 (1966).
C
      COMMON /1/ V(15,40,4),Q(40,4),P(3,40)
      COMMON /2/ RS(15),ZB(15),TH(15),CTH(15),K(15),NS(15),WS(15)
      COMMON /3/ MAXI,MI,IMIN,NP,N,NM
      COMMON /5/ DXI,DET,DX2,DE2,DT
      COMMON /6/ Q1,VS1,HT1,A1,R1
      COMMON /7/ I,IP,IM,L,LP,LM
      COMMON /8/ PX,UX,VX,AX,KBAR,C2D
      COMMON /12/ DBET,CDEL,SDEL,TDEL
      COMMON /INPUT/ GAM1,AMACH1,THETC,ZRN,TCRIT(2),P1,RHO1,RN,AMU0,T1,E
     *RR(2),IPR(6),INTAP,OTAP,NCPT,NCASE,NFILE,NSEG,IGEOM,NGEOM,NTAP
      COMMON /CONST/ PI,PI2,GAMA,GAMB
      REAL NS,K,KAP,NPLUS,KBAR
      ZZ=DBET+TH(I)
      U1=Q1*COS(ZZ)
      V1=Q1*SIN(ZZ)
      AA=VSOUND(P(L,N),V(I,N,1))
      PE=(P(L,NP)-P(L,NM))/DE2
      RE=(V(I,NP,1)-V(I,NM,1))/DE2
      UE=(V(I,NP,2)-V(I,NM,2))/DE2
      VE=(V(I,NP,3)-V(I,NM,3))/DE2
      PX=(P(LP,N)-P(LM,N))/DX2
      ZZ=UX
      UX=ZZ*CDEL-VX*SDEL
      VX=VX*CDEL+ZZ*SDEL
      ZZ=UE
      UE=ZZ*CDEL-VE*SDEL
      VE=VE*CDEL+ZZ*SDEL
      AE=(VSOUND(P(L,NP),V(I,NP,1))-VSOUND(P(L,NM),V(I,NM,1)))/DE2
      UU=V(I,N,2)*CDEL-V(I,N,3)*SDEL
      VV=V(I,N,2)*SDEL+V(I,N,3)*CDEL
      V2=V(I,NP,2)*SDEL+V(I,NP,3)*CDEL
      P2=P(L,NP)
      DT2=0.5*DT
      R2=V(I,NP,1)
      A2=VSOUND(P2,R2)
      ANS=NS(I)*DT
      DUM=ANS-(V2-A2)*DT
1     NPLUS=DUM
      ZZ=NPLUS*CDEL
      KAP=1.0-KBAR*(NS(I)+ZZ)
      DX=NPLUS*SDEL/KAP
      DE=DET+ZZ/NS(I)
      A4=AA+AE*DE+AX*DX
      V4=VV+VE*DE+VX*DX
      DUM=ANS-(V2-A2+V4-A4)*DT2
      ZZ=NPLUS-DUM
      IF (ZZ*ZZ.GT.ERR(1)*DUM*DUM) GO TO 1
      R4=V(I,M,1)+RE*DE+RX*DX                                          SHOCK 53
      P4=P(L,M)+PE*DE+PX*DX                                            SHOCK 54
      DS=NPLUS*TDEL                                                    SHOCK 55
      U4=UM+UE*DE+UX*DX                                                SHOCK 56
      DS1=-DS-0.01*KAP*DXI                                             SHOCK 57
      DS=DS-DS1                                                        SHOCK 58
      DX=DS1*CDEL/KAP                                                  SHOCK 59
      DE=DE+DS*SDEL/NS(I)                                              SHOCK 60
      U2=UU+UE*DE+UX*DX                                                SHOCK 61
      V2=VV+VE*DE+VX*DX                                                SHOCK 62
      R2=V(I,M,1)+RE*DE+RX*DX                                          SHOCK 63
      A=(R4*(U4-U2)+U4*(R4-R2))/DS                                     SHOCK 64
      B=U4*(V4-V2)/DS                                                  SHOCK 65
      B=(A*A4/R4+B1*A2/R2)*ATSTAG                                      SHOCK 66
      B=B+B1                                                           SHOCK 67
      RHO=0.5*(R4+R2)                                                  SHOCK 68
      AS=0.5*(A4+A2)                                                   SHOCK 69
      P3=P4+(V2-V4+(F-B)*DT2)*RHO*AS                                   SHOCK 70
      ZZ=P3/P2-1.0                                                     SHOCK 71
      IF (ZZ*ZZ.LT.EPS(2)) GO TO 3                                     SHOCK 72
      P2=0.5*(P2+P3)                                                   SHOCK 73
      IF (P2.LT.1.0) GO TO 2                                           SHOCK 74
      CALL RANHUG (V1,P2,V2,WSH,R2)                                    SHOCK 75
      A2=VSOUND(P2,R2)                                                 SHOCK 76
      ZZ=ANS                                                           SHOCK 77
      ANS=WSH*DT                                                       SHOCK 78
      DUM=NPLUS+ANS-ZZ                                                 SHOCK 79
      GO TO 1                                                          SHOCK 80
2     V2=V1                                                            SHOCK 81
      P3=1.0                                                           SHOCK 82
3     DO 4 M=1,4                                                       SHOCK 83
4     V(IM,NP,M)=D(NP,M)                                               SHOCK 84
      CALL RANHUG (V1,P3,V2,WS(I),D(NP,1))                             SHOCK 85
      D(NP,2)=V2*SDEL+U1*CDEL                                          SHOCK 86
      D(NP,3)=V2*CDEL-U1*SDEL                                          SHOCK 87
      D(NP,4)=ENTROP(P3,D(NP,1))                                       SHOCK 88
      RETURN                                                           SHOCK 89
      END                                                              SHOCK 90
```

20. SUBROUTINE SKIP

A. PURPOSE

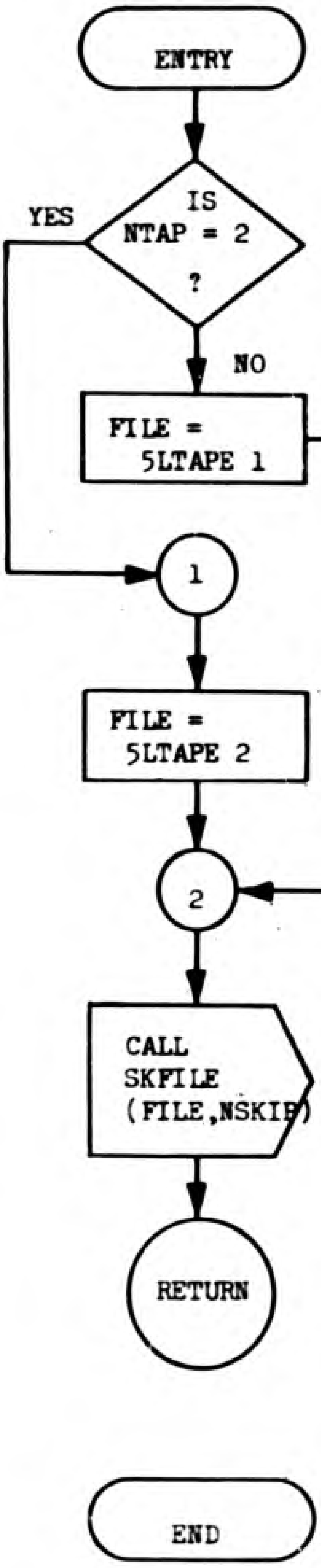
This subroutine has been included to isolate a call to a CDC machine routine which skips files on a magnetic tape. This machine routine requires a Hollerith argument which might not be accepted by the equivalent routine available to the user. Provisions for skipping files only on tapes 1 and 2, have been provided. This could easily be generalized to apply to other tape numbers.

B. VARIABLE LIST

NTAP - magnetic tape unit number.

NSKIP - number of files to be skipped.

C. FLOW CHART



ENTER SUBROUTINE SKIP.

IS NTAP = 2 ?

SET ARGUMENT TO SKIP
FILES ON TAPE 1.

SET ARGUMENT TO SKIP
FILES ON TAPE 2.

SKIP NSKIP FILES.

RETURN.

END OF SUBROUTINE SKIP.

D. LISTING

```
      SUBROUTINE SKIP (NTAP,NSKIP)
C
C      THIS IS A DUMMY ROUTINE. IT IS USES A CDC MACHINE ROUTINE
C     TO SKIP FILES ON A MAGNETIC TAPE UNIT. NTAP=TAPE NUMBER, NSKIP=
C     NUMBER OF FILES TO BE SKIPPED.
C
      IF (NTAP.EQ.2) GO TO 1
      FILE=5LTAPE1
      GO TO 2
    1 FILE=5LTAPE2
    2 CALL SKFILE (FILE,NSKIP)
      RETURN
      END
```

21. EQUATION OF STATE ROUTINES

A. PURPOSE

These routines provide the required equation of state computations. Perfect gas equations of state are used. Real gas properties could be incorporated into the various routines. The routines available are:

SUBROUTINE STATE - computes the sound speed from values of density and entropy.

FUNCTION VSOUND - computes the sound speed from values of density and pressure.

FUNCTION TEMP - computes the temperature from values of density and pressure. Used only for printed output.

FUNCTION HSP - computes the enthalpy from values of density and pressure. Used only for printed output.

FUNCTION ENTROP - computes the entropy from values of density and pressure.

SUBROUTINE RANHUG - solves the Rankine-Hugonoit relations for the shock.

B. VARIABLE LIST

Variables previously defined in the description of the main program have been omitted from this list.

ENTROP - entropy.

HSP - enthalpy.

MDOT - mass flux, $\dot{m}$

P - pressure.

P2 - pressure behind a shock.

RGAS - gas constant.

RHO - density.

RHO2 - Density behind a shock.

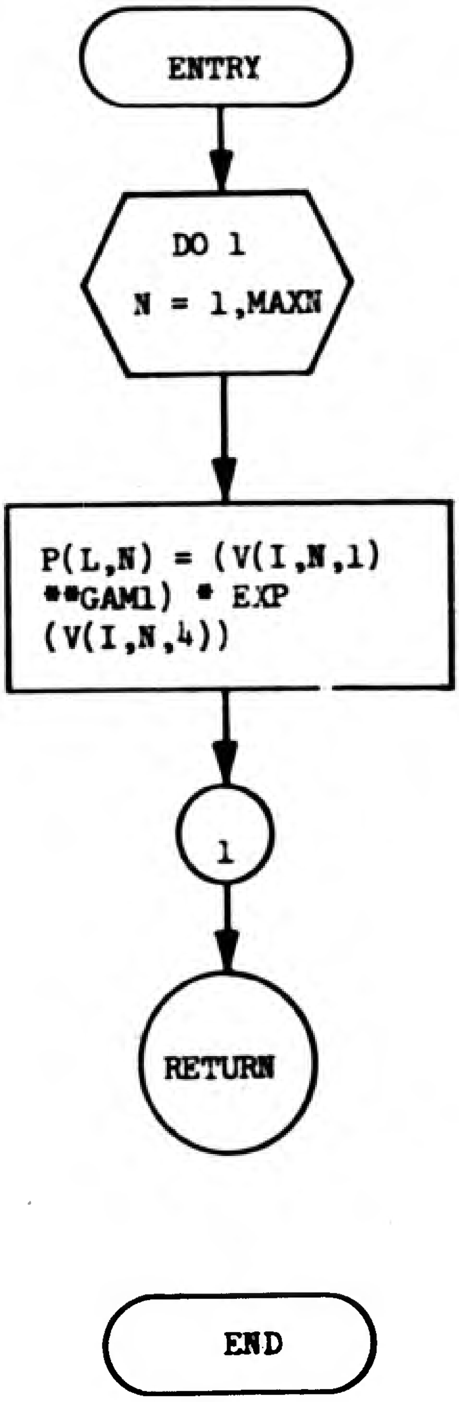
TEMP - temperature.

VSOUND - sound speed.

V1 - free-stream velocity component normal to the shock.

V2 - velocity component normal to the shock and behind the shock.

C. FLOW CHART

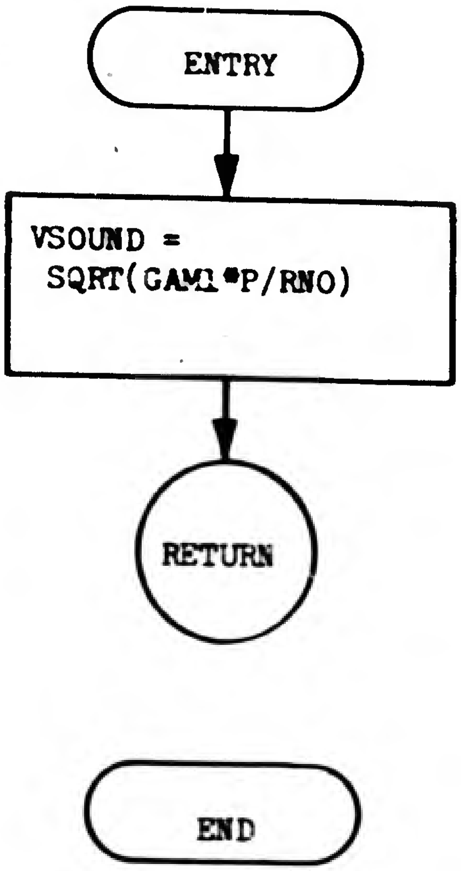


ENTER SUBROUTINE STATE.

COMPUTE PRESSURE.

RETURN.

END OF SUBROUTINE STATE.

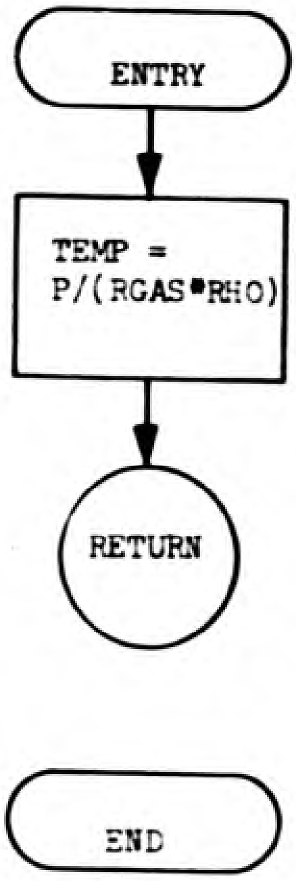


ENTER FUNCTION VSOUND.

COMPUTE SOUND SPEED.

RETURN.

END OF FUNCTION VSOUND.

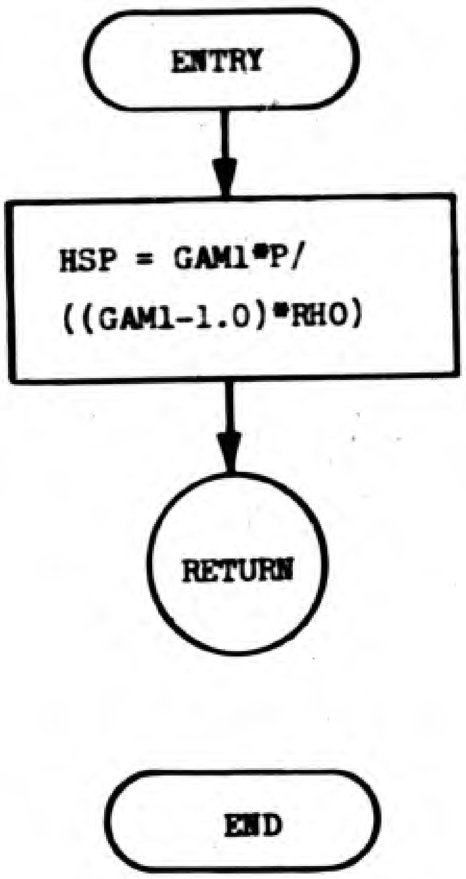


ENTER FUNCTION TEMP.

COMPUTE TEMPERATURE.

RETURN.

END OF FUNCTION TEMP.

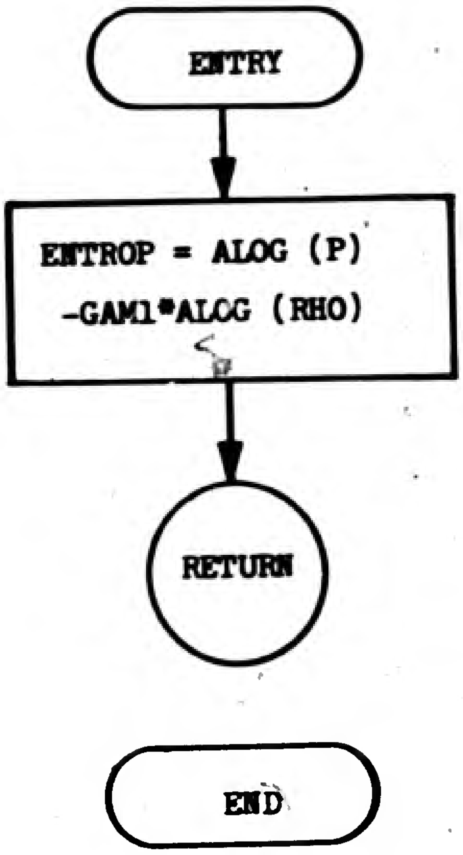


ENTER FUNCTION HSP.

COMPUTE SPECIFIC ENTHALPY.

RETURN.

END OF FUNCTION HSP.

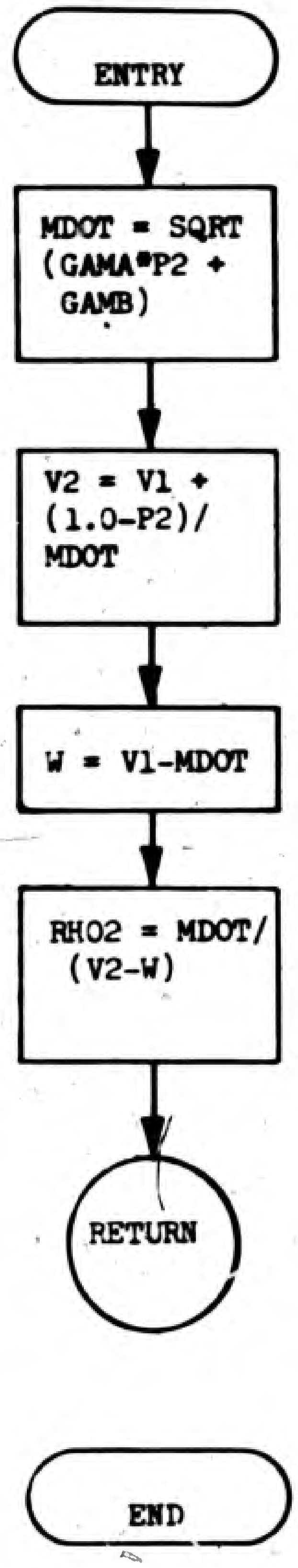


ENTER FUNCTION ENTROP.

COMPUTE ENTROPY.

RETURN.

END OF FUNCTION ENTROP.



ENTER SUBROUTINE RANHUG.

COMPUTE $\overset{\circ}{m}$.

COMPUTE V AT THE SHOCK.

COMPUTE W_s.

COMPUTE ρ AT THE SHOCK.

RETURN.

END OF SUBROUTINE RANHUG.

D. LISTING

```
      SUBROUTINE STATE (I,L,MAXN)
C
C      THIS ROUTINE COMPUTES PRESSURE, GIVEN DENSITY AND ENTROPY,
C     USING A PERFECT GAS EQUATION OF STATE.
C
      COMMON /1/ V(15,41,4),C(41,4),P(3,41)
      COMMON /INPUT/ GAM1
      DO 1 N=1,MAXN
    1 P(L,N)=(V(I,N,1)**GAM1)*EXP(V(I,N,4))
      RETURN
      END


      FUNCTION VSOUND (P,RHO)
C
C      THIS ROUTINE COMPUTES THE SOUND SPEED USING THE PERFECT GAS
C     RELATION.
C
      COMMON /INPUT/ GAM1
      VSOUND=SQRT(GAM1*P/RHO)
      RETURN
      END


      FUNCTION TEMP (P,RHO)
C
C      THIS ROUTINE COMPUTES THE TEMPERATURE USING A PERFECT GAS
C     RELATION (USED ONLY FOR PRINTOUT).
C
      COMMON /CONST/ PI,PI2,GAMA,GAMP,RGAS
      TEMP=P/(RGAS*RHO)
      RETURN
      END


      FUNCTION HSP (P,RHO)
C
C      THIS ROUTINE COMPUTES THE STATIC ENTHALPY USING A PERFECT GAS
C     RELATION (USED ONLY FOR PRINTOUT).
C
      COMMON /INPUT/ GAM1
      HSP=GAM1*P/((GAM1-1.)*RHO)
      RETURN
      END
      FUNCTION ENTROP (P,RHO)
C
C      THIS ROUTINE COMPUTES THE ENTROPY, GIVEN THE PRESSURE AND
C     DENSITY, USING A PERFECT GAS EQUATION (USED IN SHOCK).
C
      COMMON /INPUT/ GAM1
      ENTROP=ALOG(P)-GAM1*ALOG(RHO)
      RETURN
      END


      SUBROUTINE RANHUG (V1,P2,V2,W,RHO2)
C
C      THIS ROUTINE SOLVES THE RANKINE-HUGONIOT RELATIONS FOR A
C     NORMAL SHOCK USING PERFECT GAS RELATIONS.
C
      COMMON /CONST/ PI,PI2,GAMA,GAMB
      REAL MDOT
      MDOT=SQRT(GAMA*P2+GAMB)
      V2=V1+(1.0-P2)/MDOT
      W=V1-MDOT
      RHO2=MDOT/(V2-W)
      RETURN
      END
```

SECTION VII

COMPUTER CODE USER'S INSTRUCTIONS

1. INPUT DATA

The computer code provided in section VI performs most of the decision making and scheduling required to apply the numerical method developed in section IV. Since a certain degree of uncertainty is always involved when applying a time-dependent method to the blunt body problem, a very general input capability has been incorporated. Should the internal logic or empirical constants prove inadequate for a particular problem, the user has several options available to over-ride the internal logic with direct input parameters. It should be noted that the internal logic has been adequate for all problems considered in this study. To avoid unnecessarily complicated data input while retaining the general input capability, all problem constants, empirical constants and control variables are loaded by a sorting routine, SUBROUTINE DATA. These data will be referred to as case data variables. An aribtrary number of these variables may be input in any order. The principal advantage of this data loading procedure is that initial resident values supplied for most of the case data variables by a DATA statement in the main program can be used. Consequently, the case data input required is usually quite brief, typically 2 to 4 variables, while up to 23 variables can be input if necessary. Case data input is also simplified for multicase runs, since only variables whose resident values are to be changed need be included in the case data input. Some caution is required when relying on resident values from a previous case. The computer code accomplishes numerous error checks while loading the input data. When an error is sensed, the case under consideration is terminated with a printed error message and a search for the first data card of the next case is initiated. This will usually salvage a multicase run, but some case data variables may not have been loaded. Thus, reliance on a resident value loaded in a previous case could result in solving the wrong problem. Since the case data input is usually quite brief, some repetitions of data input for subsequent cases is advisable on long multi-case runs. Also, when relying on resident values from a previous case, the user must be aware of the source of these values. Usually, the resident values originate from the DATA statement mentioned previously and card input. However, under certain options described later in this section, many case data variables are loaded from magnetic tape. When sphere-cone vehicles are to be considered, the case data completely specify the problem. Other body shapes are considered by inputting body geometry data. Basically, this data consists of values of r, z, Θ and K at points on the body surface. The body geometry data input is described in more detail later in this section.

2. CASE DATA INPUT

The computer code first loads the case data in SUBROUTINE DATA. The case data packet organization is as follows:

First card: FORMAT (8A10)

Field 1 (columns 1-10) - the word TITLE, left justified in the field. This signals the start of the case data.

Fields 2-8 (columns 11-80) - an alphanumeric title which will appear on the printed output.

Cards 2, 3, . . . : FORMAT (A10, E10.4, 6I10)

Field 1 (columns 1-10) - the variable name, left justified in the field. The order in which variables are input is arbitrary.

Field 2 (columns 11-20) - the value of the variable if it is floating point form.

Field 3 (columns 21-30) - the value of the variable if it is integer form.

Fields 4-8 (columns 31-80), 10 per field) - additional integer values for the dimensional variable, IPR(6).

Last card: FORMAT (A10, E10.4, 6I10)

Field 1 (columns 1-10) - the word LAST, left justified in the field. This signals the end of the case data.

The complete list of case data variable names and their definitions will now be given. An (I) and (F) will be used to signify integer and floating point variables, respectively. Initial resident values will be specified when they have been provided.

M - (F).

>0 - free-stream Mach number.

<0 - free-stream velocity. The absolute value will be used with the values of P1 and RHO1 (below) to compute the free stream Mach number. Consequently, the units of these 3 variables must be compatible.

Z/RN - (F). Specifies the body geometry option to be used. Initialized to 0.

>0 - Specifies the axial length, in nose radii, for a sphere-cone vehicle. No body geometry data will be read.

≤ 0 - Arbitrary body geometry data will be read.

THETA - (F) cone half-angle (degrees). Used only when Z/RN>0.

GAMMA - (F) ratio of specific heats. Initialized to 1.40.

NTAP - (I) specifies the type of initial conditions to be used. Initialized to 0.

>0 - initial shock shape assigned close to the body and at a constant distance from the body. Used to model a transient flow essentially starting impulsively from rest. Should not be used when NSEG >1.

=0 - initial shock shape selected empirically. Attempts to approximate the steady state shock shape as the initial profile.

=1 - restart option (discussed below).

=2 - load option (discussed below).

MAXIT - (I). specifies the iteration procedure to be used. Initialized to -1.

<0 - steady-state solution using equation (63) and the requirement that the non-dimensional shock velocities be $<10^{-3}$ as criteria for a steady state.

=0 - no iterations are performed, but a complete printed output is provided. Useful for checking initial conditions or for obtaining dimensional output for a completed case.

>0 - specifies the number of iterations to be performed for all segments. This option is required when NTAP=2.

NSEG - (I). specifies the number of segments to be considered. Must be input when the body geometry is read in. A value greater than 0 will over-ride the axial length specification provided for the sphere-cone vehicle option. Normally should be ≤ 0 for this option. Initialized to 0.

IPR - (I).-dimensional variable providing up to 6 additional printed outputs. Since a printed output is provided after the last iteration, specifying that iteration number here will give 2 identical printed outputs and should be avoided. A negative value, or a value less than a preceding value will suppress further printed output except for the final. IPR(1)=0 will result in printing the initial conditions. All values initialized to -5.

MAXIT 0 - iteration numbers, in ascending order, at which printed outputs should be provided.

MAXIT=0 - no effect

MAXIT<0 - numbers between 0 and 10, in ascending order, specifying times when additional printed outputs should be supplied, where a value of 10 corresponds to the predicted time for steady state generated by equation (63).

P1 - (F).- free-stream pressure. Initialized to 1.0 (thus will give nondimensional output).

M>0 - used only to supply dimensional printed output.

M<0 - same as above plus used to compute the free-stream Mach number (see discussion of M, above).

RHO1 - (F).- free-stream density. Same application as P1 and units must be compatible with P1. Initialized to 1.0 (thus, will give nondimensional output).

T1 - (F).- free-stream temperature. Used only to provide dimensional printed output. Units must be compatible with P1 and RHO1. Initialized to 1.0 (thus, will give nondimensional output).

RN - (F).- all length parameters are multiplied by this scale factor to provide dimensional printed output. Units are arbitrary. Normally, RN would be equal to the radius of curvature at the stagnation point. Initialized to 1.0 (thus, will give nondimensional output).

MU - (F).- constant, μ_o, in equation (62) specifying the magnitude of the stabilizing terms. Initialized to 0.04.

MAXI - (I). specifies the number of nodes to be used per segment in the streamwise direction. A value >0 will over-ride other possible specifications for this quantity (see body geometry data for an exception). Normally, an empirical specification (sphere-cone vehicles) or a specification supplied in the body geometry input would be used. Initialized to 0.

MAXN - (I). specifies the number of nodes to be used across the shock layer. A value >0 will over-ride all other specifications of this quantity and the empirical specification generated by the computer code. Normally, this specification is not used. Initialized to 0.

INTAP - (I). specifies the magnetic tape unit number to be used for input of initial conditions. Used only for NTAP>0. See discussion of tape handling, below. Initialized to 2.

OTAP - (I). specifies the magnetic tape unit number to be used for storing the completed solutions for all segments. See discussion of tape handling, below. Initialized to 1.

NCASE - (I). specifies the case number to be used in supplying initial conditions, where the cases are considered to be numbered in ascending order starting with 1 as they apear on the tape. Used only when NTAP>0 (see discussion of Restart and Lead options). Initialized to 1.

NFILE - (I). must be equal to 1 unless NTAP>0. Initialized to 1.

NTAP=1 : Specifies the number of the last segment completed for an incomplete case. Restart initiates on the next segment.

NTAP=2 : Specifies the segment number at which computation should start. All segments following this one will also be treated.

TCRIT1 - (F). constant, C_1, in Equation (63). Initialized to 2.0.

TCRIT2 - (F). constant, C_2, in Equation (63). Initialized to 10.0.

ERRORN - (F). convergence criterion for the iteration on n in the shock point computations. Initialized to 10^{-5}.

ERRORP - (F). convergence criterion for the iteration on P in the shock point computations. Initialized to 10^{-8}.

3. BODY GEOMETRY DATA INPUT

The body geometry data are loaded by SUBROUTINE GEOM when the case data variable Z/RN ≤ 0. These data cards are arranged in packets, one for each segment considered, and are supplied immediately after the case data cards. An exact specification of the body geometry at each point for the present coordinate system is supplied by values for r, z, θ and K. It is recognized that not all of these parameters will be known for some problems of interest. However, it has been left to the user to select approximate numerical techniques for generating these parameters when necessary. This computer code supplies an exact solution for known body geometrics. Since explicit expressions for arc length are seldom available, even for analytical body profiles, numerical procedures have been incorporated to generate the required parameters at equal spaced intervals along the body, based on their values at a representative series of points. For example, it is easy to define all of the required parameters at any point on an ellipse, but considerably more difficult to compute their values at equally spaced intervals along the body. Since the geometry specifications for the representative series of points are exact, extremely accurate numerical integrations and interpolations are possible to obtain the equal spaced data. When computations start at a segment other than the first, NTAP>0, all body geometry data packets can be included. The computer code skips the data for all segments up to the one at which computations start. Specifications for the number of nodes to be used along each coordinate direction can be specified for each segment considered. An initial shock shape can be read in. This option has been included since the user could encounter a problem for which the empirical relation used to generate initial shock shape is inadequate. The required parameters can be input at equal spaced intervals along the body, for direct use in the numerical solution, or at a representative series of points on the body. Each body geometry packet is arranged as follows:

Card 1 : FORMAT (5I10, 2E10.4, A10)

Field 1 (columns 1-10) - M. The segment number.

Field 2 (columns 11-20) - NPTS. The number of data cards to follow for the current segment. If no additional specifications for the number of nodes to be used along the body are provided, the value of NPTS will be used as this specification. Also, see field 7 of this card.

Field 3 (columns 21-30) - MAXI. The total number of streamwise nodes to be used. Specifications of this variable in the case data will over-ride the present input. Also, see field 7 of this card.

Field 4 (columns 31-40) - IP. The number of nodes to be used across the shock layer. Specification of MAXN in the case data will over-ride the present input. If no specification for this parameter is supplied, the computer code will determine one empirical based on the variable input into field 6 of this card.

Field 5 (cloumns 41-50) - KSH.

KSH = 0. Empirical relations used to determine initial shock shape.

KSH ≠ 0. Initial shock shape supplied on the following data cards.

Field 6 (columns 51-60) - BNOSE. A nose bluntness parameter. Used to establish the nodal spacing across the shock layer when it is determined empirically. The usual definition applies, e.g., for an elliptical nose BNOSE is the ratio of the semi-major axes squared. A value $\leq$ 0 is ignored. For each case, BNOSE is initialized to 1.0. For downstream segments where nose bluntness effects have damped out, BNOSE should be set to 1.0.

Field 7 (columns 61-70) - DX2.

DX2 $\leq$ - The body geometry data to follow will be specified for a representative series of body points, not necessarily equally spaced along the body. Numerical procedures will be used to obtain equally spaced data.

DX2 >0 - The body geometry data to follow will be specified for points equally spaced along the body with a nodal spacing equal to the value read in this field. The number of streamwise nodes to be used will be established by the value of NPTS. Other specifications for this parameter will be ignored. Thus, the body geometry data will be used directly in the numerical solution.

Field 8 (columns 71-80) - ZZ. Any entry into this alphanumeric field will signify a discontinuity in body curvature at the upstream boundary of the segment. In this case, the body geometry data on card 2 will be identical to the data on the last card for the previous segment, except for the new value of body curvature. If this field is blank, no data for the upstream boundary should be input since all parameters will be saved from the previous segment.

Cards 2, 3 . . . , (NPTS+1): FORMAT (8E10.4)

Field 1 (columns 1-10) - RB(I). The body radius, measured normal to the symmetry axis, for the Ith point.

Field 2 (columns 11-20) - ZB(I). The value of Z, measured from the stagnation point along the symmetry axis, for the Ith point.

Field 3 (columns 21-30) - TH(I). The angle, Θ, between the symmetry axis and a tangent to the body for the Ith point.

TH(I) >0 - units are degrees.

TH(I) <0 - units are radians.

Field 4 (columns 31-40) - K(I). The body curvature at the Ith point.

Field 5 (columns 41-50) - NST(I). The initial value of n_s at the Ith point (must be <0 when used). Ignored if KSH = 0.

The above data packets are repeated for each segment, 1 to NSEG. The maximum values of NPTS and MAXI (Fields 2 and 3 of Card 1) are limited by the dimensioned variable sizes (currently 15 streamwise nodes are available). MAXI must not exceed this value since it includes the additional downstream node used for linear extrapolation and, for segments other than the first, a node for the constant upstream starting line. In establishing NPTS, more care is required. When DX2 $\leq$ 0 (card 1, Field 7), the total number of body geoemtry data points (including the upstream boundary point for segments other than the first) must not exceed 15. When DX2 >0, the value of MAXI will be established from NPTS. Therefore, the total number of body geometry data points (including the upstream boundary point for segments other than the first) must not exceed 14. That is, all body geometry points input will be actual computation points. Thus, an additional point will be required for the linear extrapolation at the downstream boundary. Should the user exceed the dimension sizes, the case will be terminated with a printed error message.

4. MAGNETIC TAPES

The computer code employs two magnetic tape units. The tape unit number specified by the case data variable INTAP is used as the source of initial conditions when NTAP >0. The tape unit number specified by OTAP is used to store the flow field data for each segment as soon as the solution at that segment is completed. The computer code rewinds tape unit number INTAP immediately after loading the case data. Tape unit number OTAP is never rewound by the computer code. Thus, for multi-case runs, the 2 tape unit numbers normally should not be equal. However, when NTAP $\leq$ 0, the computer code does not treat this situation as an error since it is often useful to have INTAP = OTAP. When NTAP >0, the computer code terminates the case if the following restrictions are violated:

NTAP = 1 : Requires INTAP = OTAP

NTAP = 2 : Requires INTAP $\neq$ OTAP

5. RESTART OPTION

When the case data variable NTAP = 1, the computer code will restart an unfinished case using the data stored on magnetic tape. The case data variables

NTAP, INTAP, OTAP, NCASE and NFILE will retain the values present after the case data was loaded. All other case data variables will be read from magnetic tape for the unfinished case. As previously noted, INTAP = OTAP is required. This means the data up to and including the last segment completed (NFILE) will be skipped on the tape and the remainder of the solution will be loaded on the same magnetic tape. The option to restart an unfinished case is a necessity for a time-dependent method. When a steady state solution is being generated, it is difficult to estimate the computation time required, since it is quite dependent on the body geometry and reentry conditions. Recognizing that the computation time allotted by the user or available to him may not be sufficient, the restart option has been incorporated to prevent excessive waste of computer time.

6. LOAD OPTIONS

When the case data variables NTAP = 2, the computer code loads all initial conditions from magnetic tape unit INTAP. All data for the case up to but excluding segment number NFILE is transferred to tape unit number OTAP with no computation. As previously noted, INTAP ≠ OTAP is required. The case data variables NTAP, MAXIT, IPR, INTAP, OTAP, NCASE, NFILE, P1, RHO1, T1, RN, MU, ERRORN and ERRORP retain the values present from the case data input operation. All other case data variables are read from magnetic tape. The above list of variables includes all of the case data variables the user can change without altering the basic problem stored on magnetic tape. The variable MAXIT ≥ 0 is required, i.e., the number of iterations to be performed on each segment must be specified. The principal purpose of this option is to allow the user to perform additional iterations on a case when a satisfactory approximation to the steady-state flow was not realized on a previous attempt. Rather than starting the case over with new values of TCRIT1 and TCRIT2, the user can use the previous solution as an initial guess for additional iterations. It should be noted that none of the cases attempted in this study required use of this option. Other applications for this option can be identified. By setting MAXIT = 0 and specifying P1, RHO1, T1 and RN, dimensional output can be obtained from the nondimensional data stored on magnetic tape for a completed case while no iterations will be performed. Also, this option could be used to input initial conditions, should the empirical methods provided for this purpose prove inadequate for a particular problem.

7. SEGMENT LENGTHS

The choice of the length of the segment to be used can significantly affect the computation time required for the problem. The length of the first segment will, generally, be specified by the location of the sonic line, since the downstream segment boundary must lie in supersonic flow. The lengths of other segments are arbitrary. It should be apparent that the use of shorter segments will reduce the computation time required for the problem. If a segment is treated with a single computational node (MAXI = 3), the length of the segment will have little effect on the computation time. It is easily shown that this is the optimum choice for the number of nodes per segment. Even for this optimum choice, reasonably short segments are recommended to provide greater numerical accuracy. Also, should the case require more computation time than was allotted by the user, shorter segments will result in more efficient restarts. That is, shorter segments require less computation time per segment. Consequently, less computation time will be lost when a restart is necessary.

8. PRINTED OUTPUT

The computer code provides a comprehensive printed output. After the case data is loaded, the case title and all case data variables are printed. After computations for a segment are completed, and at intermediate iterations when requested by the user, complete flow field outputs are provided by SUBROUTINE DMPOUT. This printed output may be nondimensional or dimensional (see case data variables P1, RHO1, T1 and RN). The following nomenclature applies to the printed output:

TIME	- nondimensional time.
S	- streamwise distance.
N	- distance normal to the body, directed away from the body.
RHO	- density.
U	- streamwise velocity component.
V	- velocity component normal to the body and directed outward from the body.
(S-S1)/CV	- nondimensional entropy.
P	- pressure.
M	- Mach number.
T	- temperature.
DH/H1	- error in total enthalpy.
RB	- body radius.
THETA	- θ
K	- body curvature.
ZB	- value of Z at a body point.

RS - shock radius.

ZS - value of z at the shock point.

WS - shock velocity, directed outward along the normal coordinate.

CP - pressure coefficient at a body point.

P/PO - ratio of the body pressure to the stagnation pressure behind the normal shock.

When the nodal spacing normal to the body is computed by the computer code, the computed number of nodes can exceed the dimension sizes of the arrays. When this occurs, the maximum number allowed will be used and a printed message will appear in the standard output. Currently, up to 40 nodes across the shock layer can be used. This number has been found to be adequate for typical problems even when the empirical relations predict that more are required. When this number of nodes is exceeded by the empirical determination, the shock layer is quite thick and the gradients of the flow parameters normal to the body are, generally, quite small.

9. SAMPLE PROBLEMS

Input data and computer outputs for 2 sample problems are provided on the following pages. The first problem considers a Mach 6 flow about an elliptically blunted cylinder with a nose bluntness parameter of 2.25. This case utilizes the arbitrary body shape option. The second case considers a Mach 4.0 flow about a sphere-cone vehicle with a cone half-angle of 40°. The internal logic for sphere-cone geometry is used for this case. These 2 cases were run as a single multicase job.

```
TITLE      SAMPLE PROBLEM 1 - ELLIPSE-CYLINDER - B/A=1.5
M           6.0
NSEG                              3
LAST
           1          15          10                         2.25
0.          0.        9.0000E 014.4444E-01
9.4538E-021.9881E-038.7599E+014.4592E-01
1.8908E-017.9762E-038.5159E+014.5039E-01
2.8361E-011.8037E-028.2695E+014.5802E-01
3.7815E-013.2299E-028.0147E+014.6907E-01
4.7269E-015.0950E-027.7518E+014.8393E-01
5.6723E-017.4255E-027.4766E+015.0320E-01
6.6176E-011.0258E-017.1954E+015.2767E-01
7.5630E-011.3641E-016.8732E+015.5849E-01
8.5084E-011.7644E-016.5337E+015.9718E-01
9.4538E-012.2361E-016.1579E+016.4601E-01
1.0399E+002.7933E-015.7327E+017.0924E-01
1.1345E+003.4579E-015.2373E+017.8872E-01
1.2290E+004.2667E-014.6397E+018.9507E-01
1.3235E+005.2941E-013.8650E+011.0397E+00
           2          14          6
1.3475E+005.6078E-013.6254E+011.0949E+00
1.3696E+005.9216E-013.3923E+011.1299E+00
1.3896E+006.2353E-013.1364E+011.1745E+00
1.4078E+006.5490E-012.8879E+011.2181E+00
1.4243E+006.8627E-012.6363E+011.2604E+00
1.4390E+007.1765E-012.3821E+011.3008E+00
1.4520E+007.4902E-012.1252E+011.3388E+00
1.4634E+007.8039E-011.8657E+011.3739E+00
1.4732E+008.1176E-011.6040E+011.4056E+00
1.4814E+008.4314E-011.3401E+011.4334E+00
1.4881E+008.7451E-011.0743E+011.4568E+00
1.4933E+009.0588E-018.0711E+001.4754E+00
1.4970E+009.3725E-015.3877E+001.4890E+00
1.5000E+001.0000E+000.        1.5000E+00
           3          15          6                                            CHANGE K
1.5000E+001.0000E+000.          0.
1.5000E+001.0286E+000.          0.
1.5000E+001.0571E+000.          0.
1.5000E+001.0857E+000.          0.
1.5000E+001.1143E+000.          0.
1.5000E+001.1429E+000.          0.
1.5000E+001.1714E+000.          0.
1.5000E+001.2000E+000.          0.
1.5000E+001.2286E+000.          0.
1.5000E+001.2571E+000.          0.
1.5000E+001.2857E+000.          0.
1.5000E+001.3143E+000.          0.
1.5000E+001.3429E+000.          0.
1.5000E+001.3714E+000.          0.
1.5000E+001.4000E+000.          0.
TITLE      SAMPLE PROBLEM 2 - SPHERE-CONE VEHICLE
M             4.0
Z/RN          1.4
THETA        40.0
NSEG                              0
LAST
```

```
AFWL TIME-DEPENDENT FLOW FIELD PROGRAM FOR AXISYMMETRIC BLUNT BODIES
AFWL-TR-70-16, R.H. AUNGIER

CASE TITLE - SAMPLE PROBLEM 1 - ELLIPSE-CYLINDER - B/A=1.5

********************************** INPUT DATA **********************************

INTAP               2      OTAP                1      MAXIT              -1

NCASE               1      NFILE               1      NSEG                3

MAXI                0      MAXN                0      NTAP                0

IPR(1)             -5      IPR(2)             -5      IPR(3)             -5

IPR(4)             -5      IPR(5)             -5      IPR(6)             -5

GAMMA     1.400E+00        M         6.000E+00        THETA     0.

Z/RN      0.               TCRIT1    2.000E+00        TCRIT2    1.000E+01

P1        1.000E+00        RHO1      1.000E+00        RN        1.000E+00

MU        4.000E-02        T1        1.000E+00        ERRORN    1.000E-05

ERRORP    1.000E-08
```

```
                     SEGMENT NUMBER   1          ITERATION NUMBER 2025          TIME=  2.5315E+00

      S=     0.         N            RHO          U            V          (S-S1)/CV     P            M            T           DM/M1
     RB=     0.     0.           5.7224E+00   0.          -0.           1.4073E+00   4.6967E+01   0.           8.2175E+00   .00043
     ZB=     0.     1.1313E-02   5.7162E+00   0.          -1.0973E-01   1.4073E+00   4.6896E+01   3.2374E-02   8.2040E+00   .00075
  THETA=    90.00   2.2627E-02   5.7090E+00   0.          -2.2143E-01   1.4073E+00   4.6817E+01   6.5354E-02   8.1933E+00   .00054
      K=     1.00   3.3940E-02   5.6936E+00   0.          -3.3069E-01   1.4073E+00   4.6636E+01   9.7653E-02   8.1913E+00   .00051
                    4.5254E-02   5.6731E+00   0.          -4.3994E-01   1.4073E+00   4.6402E+01   1.3041E-01   8.1743E+00   .00054
                    5.6567E-02   5.6464E+00   0.          -5.4875E-01   1.4073E+00   4.6096E+01   1.6232E-01   8.1636E+00   .00063
                    6.7881E-02   5.6138E+00   0.          -6.5730E-01   1.4073E+00   4.5723E+01   1.9453E-01   8.1443E+00   .00062
                    7.9191E-02   5.5749E+00   0.          -7.6767E-01   1.4073E+00   4.5291E+01   2.2765E-01   8.1223E+00   .00079
                    9.0504E-02   5.5297E+00   0.          -8.7654E-01   1.4073E+00   4.4768E+01   2.6096E-01   8.0935E+00   .00075
                    1.0182E-01   5.4777E+00   0.          -9.9093E-01   1.4073E+00   4.4180E+01   2.9449E-01   8.0654E+00   .00069
                    1.1313E-01   5.4187E+00   0.          -1.1053E+00   1.4073E+00   4.3515E+01   3.2964E-01   8.0335E+00   .00061
                    1.2444E-01   5.3515E+00   0.          -1.2224E+00   1.4073E+00   4.2761E+01   3.6549E-01   7.9935E+00   .00049
                    1.3576E-01   5.2683E+00   0.          -1.3476E+00   1.4073E+00   4.1833E+01   4.0416E-01   7.9436E+00   .00000

      S=      .03       N            RHO          U            V          (S-S1)/CV     P            M            T           DM/M1
     RB=      .08   0.           5.6716E+00   4.0851E-01  -0.           1.4073E+00   4.6354E+01   1.2073E-01   8.1743E+00   .00020
     ZB=      .00   1.1400E-02   5.6675E+00   4.1613E-01  -1.0606E-01   1.4069E+00   4.6319E+01   1.2695E-01   8.1726E+00   .00012
  THETA=    45.30   2.2799E-02   5.6633E+00   4.2386E-01  -2.1514E-01   1.4065E+00   4.6252E+01   1.4058E-01   8.1670E+00   .00009
      K=     1.01   3.4199E-02   5.6506E+00   4.3133E-01  -3.2236E-01   1.4061E+00   4.6049E+01   1.5949E-01   8.1565E+00   .00024
                    4.5599E-02   5.6329E+00   4.4045E-01  -4.3035E-01   1.4058E+00   4.5871E+01   1.8238E-01   8.1434E+00   .00030
                    5.6998E-02   5.6087E+00   4.4950E-01  -5.3862E-01   1.4054E+00   4.5579E+01   2.0799E-01   8.1206E+00   .00035
                    6.8398E-02   5.5783E+00   4.5907E-01  -6.4802E-01   1.4051E+00   4.5219E+01   2.3574E-01   8.1062E+00   .00045
                    7.9797E-02   5.5414E+00   4.6908E-01  -7.5889E-01   1.4047E+00   4.4785E+01   2.6523E-01   8.0820E+00   .00053
                    9.1197E-02   5.4974E+00   4.7944E-01  -8.7180E-01   1.4044E+00   4.4274E+01   2.9630E-01   8.0530E+00   .00061
                    1.0260E-01   5.4459E+00   4.9003E-01  -9.8729E-01   1.4041E+00   4.3680E+01   3.2892E-01   8.0230E+00   .00071
                    1.1400E-01   5.3862E+00   5.0074E-01  -1.1060E+00   1.4037E+00   4.2996E+01   3.6318E-01   7.9826E+00   .00083
                    1.2540E-01   5.3185E+00   5.1149E-01  -1.2281E+00   1.4034E+00   4.2227E+01   3.9901E-01   7.9395E+00   .00092
                    1.3680E-01   5.2650E+00   5.2185E-01  -1.3396E+00   1.4030E+00   4.1618E+01   4.3215E-01   7.9047E+00   .00000

      S=      .10       N            RHO          U            V          (S-S1)/CV     P            M            T           DM/M1
     RB=      .16   0.           5.5589E+00   7.9451E-01  -0.           1.4073E+00   4.5099E+01   2.3575E-01   8.1123E+00   .00034
     ZB=      .01   1.1470E-02   5.5627E+00   8.1369E-01  -1.0551E-01   1.4056E+00   4.5068E+01   2.4369E-01   8.1017E+00   .00025
  THETA=    43.48   2.2940E-02   5.5666E+00   8.3392E-01  -2.1382E-01   1.4040E+00   4.5038E+01   2.5579E-01   8.0903E+00   .00040
      K=     1.05   3.4410E-02   5.5619E+00   8.5496E-01  -3.1985E-01   1.4024E+00   4.4915E+01   2.7148E-01   8.0754E+00   .00066
                    4.5880E-02   5.5524E+00   8.7726E-01  -4.2644E-01   1.4009E+00   4.4739E+01   2.9042E-01   8.0576E+00   .00079
                    5.7349E-02   5.5365E+00   9.0087E-01  -5.3305E-01   1.3994E+00   4.4493E+01   3.1207E-01   8.0363E+00   .00088
                    6.8819E-02   5.5147E+00   9.2575E-01  -6.4060E-01   1.3979E+00   4.4181E+01   3.3615E-01   8.0115E+00   .00091
                    8.0289E-02   5.4865E+00   9.5173E-01  -7.4945E-01   1.3964E+00   4.3797E+01   3.6237E-01   7.9828E+00   .00092
                    9.1759E-02   5.4515E+00   9.7887E-01  -8.6020E-01   1.3948E+00   4.3339E+01   3.9061E-01   7.9449E+00   .00092
                    1.0323E-01   5.4092E+00   1.0069E+00  -9.7344E-01   1.3932E+00   4.2799E+01   4.2079E-01   7.9123E+00   .00092
                    1.1470E-01   5.3588E+00   1.0356E+00  -1.0899E+00   1.3915E+00   4.2171E+01   4.5295E-01   7.8695E+00   .00093
                    1.2617E-01   5.3001E+00   1.0651E+00  -1.2098E+00   1.3898E+00   4.1454E+01   4.8708E-01   7.8213E+00   .00092
                    1.3764E-01   5.2532E+00   1.0948E+00  -1.3206E+00   1.3880E+00   4.0868E+01   5.1977E-01   7.7797E+00   .00000

      S=      .25       N            RHO          U            V          (S-S1)/CV     P            M            T           DM/M1
     RB=      .24   0.           5.3618E+00   1.2136E+00  -0.           1.4073E+00   4.2876E+01   3.6272E-01   7.9966E+00   .00085
     ZB=      .03   1.1658E-02   5.3774E+00   1.2404E+00  -1.0304E-01   1.4041E+00   4.2912E+01   3.7239E-01   7.9801E+00   .00017
  THETA=    75.40   2.3316E-02   5.3933E+00   1.2682E+00  -2.0845E-01   1.4009E+00   4.2952E+01   3.8490E-01   7.9639E+00   .00001
      K=     1.12   3.4973E-02   5.4013E+00   1.2975E+00  -3.1126E-01   1.3977E+00   4.2904E+01   4.0013E-01   7.9432E+00   .00030
                    4.6631E-02   5.4053E+00   1.3289E+00  -4.1448E-01   1.3944E+00   4.2808E+01   4.1805E-01   7.9197E+00   .00042
                    5.8289E-02   5.4037E+00   1.3625E+00  -5.1764E-01   1.3910E+00   4.2647E+01   4.3847E-01   7.8922E+00   .00053
                    6.9947E-02   5.3970E+00   1.3982E+00  -6.2168E-01   1.3875E+00   4.2424E+01   4.6127E-01   7.8607E+00   .00059
                    8.1605E-02   5.3846E+00   1.4360E+00  -7.2692E-01   1.3839E+00   4.2133E+01   4.8628E-01   7.8247E+00   .00064
                    9.3263E-02   5.3662E+00   1.4755E+00  -8.3390E-01   1.3800E+00   4.1771E+01   5.1340E-01   7.7842E+00   .00067
                    1.0492E-01   5.3410E+00   1.5165E+00  -9.4312E-01   1.3761E+00   4.1332E+01   5.4257E-01   7.7387E+00   .00070
                    1.1658E-01   5.3080E+00   1.5589E+00  -1.0553E+00   1.3720E+00   4.0807E+01   5.7380E-01   7.6878E+00   .00073
                    1.2824E-01   5.2665E+00   1.6022E+00  -1.1709E+00   1.3677E+00   4.0190E+01   6.0713E-01   7.6313E+00   .00074
                    1.3990E-01   5.2336E+00   1.6462E+00  -1.2794E+00   1.3634E+00   3.9665E+01   6.4004E-01   7.5790E+00   .00000
```

```
    S=     .33       N            RHO           U           V            (S-S1)/CV       P             M              T            DM/M1
   RU=     .32    0.            5.0731E+00  1.6255E+00  -0.              1.4073E+00   3.9679E+01   4.9123E-01    7.7214E+00    .00102
   ZU=     .05    1.1887E-02    5.1057E+00  1.6595E+00  -1.0292E-01      1.4010E+00   3.9809E+01   5.0325E-01    7.7471E+00    .00147
THETA=   69.92    2.3773E-02    5.1399E+00  1.6961E+00  -2.0733E-01      1.3958E+00   3.9948E+01   5.1303E-01    7.7723E+00    .00129
    K=    1.23    3.5660E-02    5.1672E+00  1.7364E+00  -3.0900E-01      1.3890E+00   4.0032E+01   5.3573E-01    7.7414E+00    .00173
                  4.7547E-02    5.1918E+00  1.7803E+00  -4.0958E-01      1.3833E+00   4.0013E+01   5.5629E-01    7.7036E+00    .00150
                  5.9434E-02    5.2117E+00  1.8234E+00  -5.0870E-01      1.3767E+00   3.9904E+01   5.7930E-01    7.6631E+00    .00120
                  7.1320E-02    5.2271E+00  1.8734E+00  -6.0727E-01      1.3700E+00   3.9860E+01   6.0459E-01    7.6257E+00    .00120
                  8.3207E-02    5.2372E+00  1.9330E+00  -7.0552E-01      1.3631E+00   3.9696E+01   6.3108E-01    7.5796E+00    .00139
                  9.5094E-02    5.2419E+00  1.9879E+00  -8.0414E-01      1.3562E+00   3.9470E+01   6.6045E-01    7.5237E+00    .00104
                  1.0698E-01    5.2404E+00  2.0441E+00  -9.0383E-01      1.3491E+00   3.9174E+01   6.9038E-01    7.4755E+00    .00135
                  1.1887E-01    5.2321E+00  2.1017E+00  -1.0055E+00      1.3417E+00   3.8802E+01   7.2307E-01    7.4162E+00    .00101
                  1.3075E-01    5.2159E+00  2.1607E+00  -1.1104E+00      1.3342E+00   3.8342E+01   7.5726E-01    7.3513E+00    .00073
                  1.4264E-01    5.2029E+00  2.2207E+00  -1.2119E+00      1.3262E+00   3.7906E+01   7.9212E-01    7.2457E+00    .00000

    S=     .41       N            RHO           U           V            (S-S1)/CV       P             M              T            DM/M1
   RU=     .40    0.            4.6854E+00  2.1237E+00  -0.              1.4073E+00   3.5500E+01   6.5360E-01    7.5706E+00    .00242
   ZU=     .09    1.2385E-02    4.7490E+00  2.1641E+00  -9.6449E-02      1.3975E+00   3.5812E+01   6.6665E-01    7.5420E+00    .00104
THETA=   63.82    2.4770E-02    4.8101E+00  2.2030E+00  -1.9215E-01      1.3878E+00   3.6120E+01   6.8201E-01    7.5092E+00    .00095
    K=    1.39    3.7155E-02    4.8644E+00  2.2436E+00  -2.8230E-01      1.3785E+00   3.6351E+01   6.9912E-01    7.4731E+00    .00040
                  4.9540E-02    4.9144E+00  2.2868E+00  -3.7023E-01      1.3694E+00   3.6540E+01   7.1800E-01    7.4354E+00    .00024
                  6.1925E-02    4.9589E+00  2.3323E+00  -4.5565E-01      1.3604E+00   3.6673E+01   7.3852E-01    7.3955E+00    .00027
                  7.4310E-02    4.9992E+00  2.3802E+00  -5.4009E-01      1.3513E+00   3.6756E+01   7.6076E-01    7.3573E+00    .00041
                  8.6694E-02    5.0357E+00  2.4313E+00  -6.2440E-01      1.3418E+00   3.6782E+01   7.8498E-01    7.3042E+00    .00054
                  9.9079E-02    5.0688E+00  2.4863E+00  -7.0980E-01      1.3317E+00   3.6748E+01   8.1160E-01    7.2497E+00    .00050
                  1.1146E-01    5.0987E+00  2.5460E+00  -7.9734E-01      1.3207E+00   3.6646E+01   8.4107E-01    7.1873E+00    .00051
                  1.2385E-01    5.1248E+00  2.6112E+00  -8.8619E-01      1.3087E+00   3.6469E+01   8.7362E-01    7.1161E+00    .00034
                  1.3623E-01    5.1452E+00  2.6819E+00  -9.4379E-01      1.2956E+00   3.6195E+01   9.1027E-01    7.0347E+00    .00007
                  1.4862E-01    5.1641E+00  2.7581E+00  -1.0815E+00      1.2815E+00   3.5871E+01   9.5001E-01    6.9462E+00    .00000

    S=     .49       N            RHO           U           V            (S-S1)/CV       P             M              T            DM/M1
   RU=     .47    0.            4.1053E+00  2.6495E+00  -0.              1.4073E+00   2.9502E+01   8.3529E-01    7.1804E+00    .00132
   ZU=     .13    1.3135E-02    4.1927E+00  2.6670E+00  -8.0298E-02      1.3959E+00   3.0041E+01   8.4243E-01    7.1652E+00    .00217
THETA=   56.33    2.6269E-02    4.2809E+00  2.6849E+00  -1.6208E-01      1.3844E+00   3.0574E+01   8.5185E-01    7.1423E+00    .00251
    K=    1.61    3.9404E-02    4.3618E+00  2.7180E+00  -2.3850E-01      1.3727E+00   3.1024E+01   8.6404E-01    7.1126E+00    .00292
                  5.2538E-02    4.4414E+00  2.7559E+00  -3.1409E-01      1.3604E+00   3.1429E+01   8.8124E-01    7.0763E+00    .00301
                  6.5673E-02    4.5194E+00  2.8039E+00  -3.8800E-01      1.3469E+00   3.1774E+01   9.0222E-01    7.0306E+00    .00303
                  7.8808E-02    4.5986E+00  2.8629E+00  -4.6177E-01      1.3320E+00   3.2073E+01   9.2803E-01    6.9745E+00    .00294
                  9.1942E-02    4.6798E+00  2.9333E+00  -5.3549E-01      1.3152E+00   3.2322E+01   9.5891E-01    6.9068E+00    .00260
                  1.0508E-01    4.7635E+00  3.0146E+00  -6.0964E-01      1.2966E+00   3.2524E+01   9.9460E-01    6.8277E+00    .00255
                  1.1821E-01    4.8495E+00  3.1054E+00  -6.8413E-01      1.2762E+00   3.2677E+01   1.0353E+00    6.7341E+00    .00212
                  1.3135E-01    4.9366E+00  3.2035E+00  -7.5919E-01      1.2544E+00   3.2776E+01   1.0793E+00    6.6334E+00    .00149
                  1.4448E-01    5.0254E+00  3.3063E+00  -8.3337E-01      1.2313E+00   3.2839E+01   1.1273E+00    6.5345E+00    .00050
                  1.5762E-01    5.0953E+00  3.4133E+00  -9.1804E-01      1.2073E+00   3.2838E+01   1.1794E+00    6.4152E+00    .00000

    S=     .57       N            RHO           U           V            (S-S1)/CV       P             M              T            DM/M1
   RU=     .53    0.            3.5926E+00  3.1356E+00  -0.              1.4073E+00   2.4476E+01   1.0153E+00    6.8130E+00    .00213
   ZU=     .18    1.4041E-02    3.7230E+00  3.1479E+00  -4.7253E-02      1.3908E+00   2.5308E+01   1.0205E+00    6.7973E+00    .00105
THETA=   44.59    2.8081E-02    3.8574E+00  3.1683E+00  -1.0072E-01      1.3739E+00   2.6114E+01   1.0293E+00    6.7703E+00    .00150
    K=    1.93    4.2122E-02    3.9742E+00  3.2026E+00  -1.4071E-01      1.3564E+00   2.6793E+01   1.0436E+00    6.7413E+00    .00124
                  5.6162E-02    4.0949E+00  3.2499E+00  -1.9038E-01      1.3376E+00   2.7420E+01   1.0634E+00    6.6901E+00    .00124
                  7.0203E-02    4.2149E+00  3.3100E+00  -2.4133E-01      1.3173E+00   2.7976E+01   1.0887E+00    6.6375E+00    .00134
                  8.4243E-02    4.3371E+00  3.3810E+00  -2.8505E-01      1.2952E+00   2.8494E+01   1.1190E+00    6.5674E+00    .00147
                  9.8284E-02    4.4609E+00  3.4603E+00  -3.2740E-01      1.2716E+00   2.8937E+01   1.1534E+00    6.4853E+00    .00154
                  1.1232E-01    4.5958E+00  3.5448E+00  -3.6899E-01      1.2468E+00   2.9341E+01   1.1903E+00    6.3983E+00    .00156
                  1.2637E-01    4.7397E+00  3.6316E+00  -4.1071E-01      1.2213E+00   2.9689E+01   1.2302E+00    6.3033E+00    .00140
                  1.4041E-01    4.8294E+00  3.7188E+00  -4.5414E-01      1.1954E+00   2.9965E+01   1.2711E+00    6.2043E+00    .00121
                  1.5445E-01    4.9486E+00  3.8044E+00  -4.9713E-01      1.1695E+00   3.0212E+01   1.3123E+00    6.1052E+00    .00099
                  1.6849E-01    5.0313E+00  3.8942E+00  -5.0445E-01      1.1436E+00   3.0131E+01   1.3587E+00    5.9563E+00    .00000
```

```
     S=     .65        N            RHO           U            V          (S-S1)/CV        P              H              T          DH/H1
    RB=     .59    0.            2.6492E+00   3.9022E+00  -0.            1.4073E+00   1.5978E+01   1.3429E+00   6.0314E+00   .00081
    ZB=     .24    1.6713E-02    2.8461E+00   3.8443E+00   5.4750E-02    1.3885E+00   1.7336E+01   1.3106E+00   6.0913E+00   .00036
 THETA=   38.66    3.3544E-02    3.0428E+00   3.8024E+00   7.0755E-02    1.3685E+00   1.8659E+01   1.2974E+00   6.1322E+00   .00019
     K=    2.33    5.0376E-02    3.2224E+00   3.7837E+00   8.1818E-02    1.3469E+00   1.9748E+01   1.2907E+00   6.1403E+00   .00160
                   6.7168E-02    3.4024E+00   3.7874E+00   9.1369E-02    1.3232E+00   2.0852E+01   1.2933E+00   6.1217E+00   .00259
                   8.3960E-02    3.5807E+00   3.8112E+00   7.3955E-02    1.2972E+00   2.1823E+01   1.3050E+00   6.0945E+00   .00301
                   1.0075E-01    3.7632E+00   3.8549E+00   7.2695E-02    1.2688E+00   2.2742E+01   1.3245E+00   6.0413E+00   .00444
                   1.1754E-01    3.9506E+00   3.9053E+00   6.5955E-02    1.2386E+00   2.3619E+01   1.3500E+00   5.9737E+00   .00512
                   1.3434E-01    4.1416E+00   3.9694E+00   5.6013E-02    1.2070E+00   2.4447E+01   1.3809E+00   5.9023E+00   .00560
                   1.5113E-01    4.3422E+00   4.0407E+00   4.7223E-02    1.1742E+00   2.5279E+01   1.4154E+00   5.8217E+00   .00555
                   1.6792E-01    4.5392E+00   4.1238E+00   3.0210E-02    1.1404E+00   2.6004E+01   1.4562E+00   5.7269E+00   .00507
                   1.8471E-01    4.7647E+00   4.2129E+00   2.3066E-02    1.1056E+00   2.6879E+01   1.4991E+00   5.6412E+00   .00234
                   2.0150E-01    4.9500E+00   4.3296E+00  -1.4401E-02    1.0690E+00   2.7334E+01   1.5572E+00   5.5223E+00   .00000


                                                        SHOCK AND BODY PARAMETERS

      S             RB            ZB             K           THETA          RS            ZS            HS            CP           P/PO
 0.            0.            0.            1.0000E+00   9.0000E+01  -1.1808E-14  -1.3576E-01  -2.1505E-06   1.8241E+00   1.0032E+00
 8.1706E-02    8.1614E-02    3.3431E-03    1.0126E+00   8.5239E+01   9.2825E-02  -1.3299E-01  -2.2110E-06   1.8010E+00   9.9079E-01
 1.6341E-01    1.6267E-01    1.3434E-02    1.0518E+00   8.0477E+01   1.8544E-01  -1.2231E-01  -2.2794E-06   1.7500E+00   9.6334E-01
 2.4512E-01    2.4255E-01    3.0462E-02    1.1214E+00   7.5404E+01   2.7781E-01  -1.0492E-01  -2.4307E-06   1.6614E+00   9.1586E-01
 3.2682E-01    3.2053E-01    5.4745E-02    1.2231E+00   6.9919E+01   3.6951E-01  -7.9223E-02  -2.5736E-06   1.5349E+00   8.4756E-01
 4.0853E-01    3.9567E-01    8.6746E-02    1.3604E+00   6.3820E+01   4.6124E-01  -4.6026E-02  -2.8317E-06   1.3690E+00   7.5930E-01
 4.9023E-01    4.6667E-01    1.2716E-01    1.6107E+00   5.6833E+01   5.5290E-01  -4.8753E-03  -3.1874E-06   1.1310E+00   6.3018E-01
 5.7194E-01    5.3173E-01    1.7637E-01    1.9250E+00   4.8595E+01   6.4316E-01   4.9998E-02  -3.5425E-06   9.3159E-01   5.2282E-01
 6.5365E-01    5.8820E-01    2.3528E-01    2.3333E+00   3.8660E+01   7.4555E-01   1.0940E-01  -4.5911E-06   5.9437E-01   3.4130E-01
```

```
                    SEGMENT NUMBER  2          ITERATION NUMBER  541          TIME=  7.8971E-01

   S=     .71      N          RHO         U           V          (S-S1)/CV    P           H           T          DH/H1
  RB=     .62   0.           2.1549E+00  4.3465E+00 -0.          1.4073E+00  1.1196E+01  1.5930E+00  5.4443E+00  [illegible]
  ZB=     .24   1.5240E-02   2.2548E+00  4.3020E+00  1.2805E-01  1.3900E+00  1.2533E+01  1.5423E+00  5.5512E+00  .00006
THETA=  30.43   3.0432E-02   2.4510E+00  4.2323E+00  1.7794E-01  1.3714E+00  1.3826E+01  1.5077E+00  5.5403E+00  .00005
   K=    2.65   4.5734E-02   2.6299E+00  4.1852E+00  2.5280E-01  1.3512E+00  1.4953E+01  1.4801E+00  5.6350E+00  .00035
                6.0944E-02   2.8052E+00  4.1582E+00  2.9191E-01  1.3289E+00  1.6006E+01  1.4744E+00  5.7306E+00  .00103
                7.6230E-02   2.9792E+00  4.1509E+00  3.2302E-01  1.3045E+00  1.6993E+01  1.4733E+00  5.7133E+00  .00024
                9.1475E-02   3.1545E+00  4.1594E+00  3.4983E-01  1.2780E+00  1.7928E+01  1.4793E+00  5.6172E+00  .00031
                1.0672E-01   3.3334E+00  4.1807E+00  3.7366E-01  1.2498E+00  1.8829E+01  1.4926E+00  5.6410E+00  .00021
                1.2197E-01   3.5148E+00  4.2110E+00  3.9587E-01  1.2204E+00  1.9699E+01  1.5160E+00  5.6040E+00  .00049
                1.3721E-01   3.6944E+00  4.2474E+00  4.1690E-01  1.1918E+00  2.0550E+01  1.5302E+00  5.5505E+00  .00000
                1.5246E-01   3.8811E+00  4.2863E+00  4.3592E-01  1.1635E+00  2.1372E+01  1.5524E+00  5.5103E+00  .00047
                1.6770E-01   4.0650E+00  4.3339E+00  4.5412E-01  1.1357E+00  2.2179E+01  1.5767E+00  5.4506E+00  .01341
                1.8295E-01   4.2522E+00  4.3874E+00  4.6995E-01  1.1070E+00  2.2952E+01  1.6052E+00  5.3977E+00  .00255
                1.9820E-01   4.4468E+00  4.4549E+00  4.8285E-01  1.0751E+00  2.3669E+01  1.6415E+00  5.3227E+00  .00103
                2.1344E-01   4.6813E+00  4.5324E+00  5.0561E-01  1.0378E+00  2.4504E+01  1.6947E+00  5.2344E+00  .00069
                2.2869E-01   4.8598E+00  4.5405E+00  4.8437E-01  9.9329E-01  2.4697E+01  1.7515E+00  5.0413E+00  .00002

   S=     .77      N          RHO         U           V          (S-S1)/CV    P           H           T          DH/H1
  RB=     .65   0.           1.4698E+00  4.9069E+00 -0.          1.4073E+00  7.0036E+00  1.8998E+00  4.7651E+00  .00053
  ZB=     .33   1.5249E-02   1.7097E+00  4.7451E+00  2.9653E-01  1.3906E+00  8.5118E+00  1.9009E+00  4.9794E+00  .00092
THETA=  21.03   3.0597E-02   1.9370E+00  4.6159E+00  4.5198E-01  1.3718E+00  9.9441E+00  1.7297E+00  5.1359E+00  .00103
   K=    3.02   5.4896E-02   2.1322E+00  4.5218E+00  5.6710E-01  1.3510E+00  1.1146E+01  1.6840E+00  5.2273E+00  .00070
                7.3145E-02   2.3195E+00  4.4579E+00  6.4605E-01  1.3279E+00  1.2254E+01  1.6503E+00  5.2423E+00  .00225
                9.1494E-02   2.4993E+00  4.4194E+00  7.0873E-01  1.3025E+00  1.3262E+01  1.6422E+00  5.3062E+00  .00383
                1.0979E-01   2.6771E+00  4.4014E+00  7.5974E-01  1.2751E+00  1.4207E+01  1.6366E+00  5.3063E+00  .00529
                1.2809E-01   2.8540E+00  4.3996E+00  8.0363E-01  1.2466E+00  1.5101E+01  1.6432E+00  5.2913E+00  .00025
                1.4639E-01   3.0305E+00  4.4108E+00  8.4266E-01  1.2176E+00  1.5956E+01  1.6540E+00  5.2651E+00  .00060
                1.6469E-01   3.2083E+00  4.4348E+00  8.7819E-01  1.1379E+00  1.6740E+01  1.6709E+00  5.2293E+00  .00621
                1.8299E-01   3.3946E+00  4.4735E+00  9.1206E-01  1.1558E+00  1.7542E+01  1.6955E+00  5.1794E+00  .00524
                2.0129E-01   3.5973E+00  4.5311E+00  9.4580E-01  1.1184E+00  1.8369E+01  1.7312E+00  5.1063E+00  .00401
                2.1958E-01   3.8335E+00  4.6089E+00  9.8440E-01  1.0730E+00  1.9187E+01  1.7804E+00  5.0052E+00  .00265
                2.3788E-01   4.1013E+00  4.7097E+00  1.0221E+00  1.0144E+00  1.9969E+01  1.8459E+00  4.8643E+00  .00155
                2.5618E-01   4.4510E+00  4.8139E+00  1.0845E+00  9.5632E-01  2.1046E+01  1.9179E+00  4.7233E+00  .00043
                2.7448E-01   4.7214E+00  4.9525E+00  1.0888E+00  8.8924E-01  2.1374E+01  2.0142E+00  4.5271E+00  .00004

   S=     .83      N          RHO         U           V          (S-S1)/CV    P           H           T          DH/H1
  RB=     .60   0.           3.0955E-01  5.3473E+00 -0.          1.4073E+00  3.9135E+00  2.2666E+00  4.0351E+00  .00221
  ZB=     .33   2.1776E-02   1.2491E+00  5.1455E+00  5.0013E-01  1.3897E+00  5.4795E+00  2.0360E+00  4.3869E+00  .00059
THETA=  10.95   4.3551E-02   1.4972E+00  4.9638E+00  7.6824E-01  1.3687E+00  6.9148E+00  1.9741E+00  4.6196E+00  .00227
   K=    3.27   6.5327E-02   1.7123E+00  4.8224E+00  9.5353E-01  1.3445E+00  8.1458E+00  1.9046E+00  4.7572E+00  .00113
                8.7102E-02   1.9190E+00  4.7258E+00  1.0826E+00  1.3169E+00  9.2947E+00  1.8618E+00  4.8436E+00  .00017
                1.0888E-01   2.1180E+00  4.6623E+00  1.1861E+00  1.2863E+00  1.0349E+01  1.8393E+00  4.8904E+00  .00089
                1.3065E-01   2.3165E+00  4.6263E+00  1.2723E+00  1.2530E+00  1.1349E+01  1.8321E+00  4.8941E+00  .00141
                1.5242E-01   2.5166E+00  4.6125E+00  1.3496E+00  1.2176E+00  1.2302E+01  1.8371E+00  4.8841E+00  .00152
                1.7420E-01   2.7229E+00  4.6176E+00  1.4216E+00  1.1798E+00  1.3225E+01  1.8528E+00  4.8571E+00  .00049
                1.9598E-01   2.9410E+00  4.6401E+00  1.4917E+00  1.1382E+00  1.4133E+01  1.8791E+00  4.8053E+00  .00012
                2.1776E-01   3.1788E+00  4.6764E+00  1.5624E+00  1.0916E+00  1.5040E+01  1.9165E+00  4.7515E+00  .00053
                2.3953E-01   3.4421E+00  4.7296E+00  1.6346E+00  1.0394E+00  1.5954E+01  1.9642E+00  4.6505E+00  .00106
                2.6131E-01   3.7343E+00  4.7877E+00  1.7085E+00  9.8298E-01  1.6904E+01  2.0193E+00  4.5205E+00  .00225
                2.8309E-01   4.0452E+00  4.8477E+00  1.7779E+00  9.2506E-01  1.7843E+01  2.0778E+00  4.4103E+00  .00239
                3.0440E-01   4.3865E+00  4.9770E+00  1.8501E+00  8.6904E-01  1.8897E+01  2.1316E+00  4.3073E+00  .00277
                3.2663E-01   4.6134E+00  4.9746E+00  1.9613E+00  9.1651E-01  1.9241E+01  2.1980E+00  4.1757E+00  .00010
```

```
                              N            RHO          U            V         (S-S1)/CV       P            M            T         DM/M1
     S=      .83     0.              6.1649E-01   5.8431E+00  -0.           1.4073E+00   2.0753E+00   2.6916E+00   3.3062E+00   .00533
    RB=      .67     2.9460E-02      9.2022E-01   5.4444E+00   2.6129E-01   1.3907E+00   3.5760E+00   2.3632E+00   3.8601E+00   .00323
    ZB=      .45     5.8920E-02      1.1656E+00   5.1654E+00   1.2613E+00   1.3691E+00   4.8729E+00   2.1979E+00   4.1835E+00   .00230
 THETA=     -.00     8.8391E-02      1.3756E+00   4.9637E+00   1.5064E+00   1.3432E+00   5.9869E+00   2.1014E+00   4.3523E+00   .00345
     K=     3.38     1.1784E-01      1.5730E+00   4.8228E+00   1.6727E+00   1.3127E+00   7.0066E+00   2.0441E+00   4.4544E+00   .00282
                     1.4730E-01      1.7616E+00   4.7261E+00   1.8029E+00   1.2783E+00   7.9324E+00   2.0146E+00   4.5033E+00   .00533
                     1.7676E-01      1.9495E+00   4.6644E+00   1.9115E+00   1.2404E+00   8.8016E+00   2.0050E+00   4.5143E+00   .00671
                     2.0622E-01      2.1404E+00   4.6237E+00   2.0090E+00   1.1991E+00   9.6266E+00   2.0112E+00   4.4975E+00   .00773
                     2.3568E-01      2.3406E+00   4.6174E+00   2.1004E+00   1.1540E+00   1.0428E+01   2.0311E+00   4.4554E+00   .00837
                     2.6514E-01      2.5568E+00   4.6241E+00   2.1895E+00   1.1037E+00   1.1223E+01   2.0639E+00   4.3894E+00   .00865
                     2.9460E-01      2.7966E+00   4.6403E+00   2.2783E+00   1.0477E+00   1.2031E+01   2.1080E+00   4.3020E+00   .00883
                     3.2406E-01      3.0647E+00   4.6804E+00   2.3669E+00   9.8656E-01   1.2865E+01   2.1635E+00   4.1978E+00   .00864
                     3.5352E-01      3.3669E+00   4.7211E+00   2.4558E+00   9.2182E-01   1.3755E+01   2.2252E+00   4.0854E+00   .00841
                     3.8298E-01      3.6970E+00   4.7703E+00   2.5395E+00   8.5496E-01   1.4665E+01   2.2932E+00   3.9608E+00   .00744
                     4.1244E-01      4.0939E+00   4.8193E+00   2.6341E+00   7.8618E-01   1.5792E+01   2.3633E+00   3.8574E+00   .00412
                     4.4190E-01      4.4384E+00   4.9280E+00   2.6881E+00   7.1167E-01   1.6413E+01   2.4671E+00   3.6930E+00   .00005


                                              SHOCK AND BODY PARAMETERS

      S            RB           ZB            K          THETA          RS           ZS           WS           CP          P/P0
 7.1092E-01   6.2073E-01   2.8237E-01   2.6772E+00   3.0429E+01   8.1792E-01   1.6655E-01  -1.6403E-04   4.0460E-01   2.3415E-01
 7.6820E-01   6.4569E-01   3.3396E-01   3.0187E+00   2.1078E+01   9.0180E-01   2.3514E-01   2.5619E-04   2.3924E-01   1.4960E-01
 8.2548E-01   6.6158E-01   3.8842E-01   3.2674E+00   1.0953E+01   9.8227E-01   3.2675E-01   7.7719E-04   1.1561E-01   8.3594E-02
 8.8276E-01   6.6707E-01   4.4574E-01   3.3750E+00  -1.3344E-06   1.1090E+00   4.4574E-01  -4.5285E-04   4.2069E-02   4.4329E-02
```

```
              SEGMENT NUMBER  3            ITERATION NUMBER  216            TIME=  4.9330E-11

     S=     .93        N           RHO          U            V         (S-S1)/CV       P            H            T          DH/H1.
    RJ=     .67   0.            5.3109E-01   5.3323E+00  -0.           1.4073E+00   1.6842E+00   2.8154E+00   3.1780E+00   .01015
    ZJ=     .43   2.2716E-12    7.1000E-01   5.7057E+00   2.6579E-01   1.3961E+00   2.5009E+00   2.5721E+00   3.5224E+00   .01205
 THETA=    0.     4.5432E-12    9.6004E-01   5.5272E+00   6.4278E-01   1.3833E+00   3.2294E+00   2.4290E+00   3.7543E+00   .00153
     K=    0.     6.8148E-12    1.0157E+00   5.3539E+00   9.7326E-01   1.3679E+00   4.0135E+00   2.3101E+00   3.9515E+00   .00111
                  9.0863E-12    1.1591E+00   5.2153E+00   1.1925E+00   1.3500E+00   4.7430E+00   2.2353E+00   4.0920E+00   .00235
                  1.1358E-11    1.2947E+00   5.0967E+00   1.3624E+00   1.3295E+00   5.4439E+00   2.1768E+00   4.1957E+00   .00343
                  1.3630E-11    1.4350E+00   4.9998E+00   1.5023E+00   1.3067E+00   6.1247E+00   2.1357E+00   4.2692E+00   .00407
                  1.5901E-11    1.5694E+00   4.9232E+00   1.6191E+00   1.2816E+00   6.7734E+00   2.1038E+00   4.3142E+00   .00534
                  1.8173E-11    1.7040E+00   4.8646E+00   1.7210E+00   1.2545E+00   7.3937E+00   2.0955E+00   4.3310E+00   .00704
                  2.0444E-11    1.8474E+00   4.8213E+00   1.8107E+00   1.2253E+00   7.9947E+00   2.0873E+00   4.3451E+00   .00743
                  2.2716E-11    1.9903E+00   4.7917E+00   1.8938E+00   1.1940E+00   8.5900E+00   2.0934E+00   4.3377E+00   .00855
                  2.4987E-11    2.1260E+00   4.7744E+00   1.9716E+00   1.1604E+00   9.1735E+00   2.1016E+00   4.3140E+00   .00504
                  2.7259E-11    2.2801E+00   4.7680E+00   2.0461E+00   1.1239E+00   9.7554E+00   2.1200E+00   4.2735E+00   .00425
                  2.9531E-11    2.4458E+00   4.7712E+00   2.1184E+00   1.0843E+00   1.0345E+01   2.1454E+00   4.2235E+00   .00940
                  3.1802E-11    2.6252E+00   4.7827E+00   2.1902E+00   1.0416E+00   1.0945E+01   2.1773E+00   4.1632E+00   .00943
                  3.4074E-11    2.8204E+00   4.8011E+00   2.2606E+00   9.9611E-01   1.1562E+01   2.2151E+00   4.0935E+00   .00940
                  3.6345E-11    3.0335E+00   4.8245E+00   2.3302E+00   9.4840E-01   1.2207E+01   2.2573E+00   4.0120E+00   .00916
                  3.8617E-11    3.2658E+00   4.8523E+00   2.3992E+00   8.9890E-01   1.2842E+01   2.3035E+00   3.9444E+00   .00851
                  4.0889E-11    3.5163E+00   4.8852E+00   2.4659E+00   8.4808E-01   1.3578E+01   2.3536E+00   3.8614E+00   .00733
                  4.3160E-11    3.8010E+00   4.9201E+00   2.5388E+00   7.9588E-01   1.4372E+01   2.4004E+00   3.7813E+00   .00455
                  4.5432E-11    4.1003E+00   4.9653E+00   2.6063E+00   7.4104E-01   1.5128E+01   2.4677E+00   3.6894E+00   .00211
                  4.7703E-11    4.3470E+00   5.0284E+00   2.6574E+00   6.8349E-01   1.5699E+01   2.5411E+00   3.5735E+00   .00000

     S=     .97        N           RHO          U            V         (S-S1)/CV       P            H            T          DH/H1
    RJ=     .67   0.            5.2032E-01   5.9325E+00  -0.           1.4073E+00   1.6366E+00   2.8271E+00   3.1454E+00   .00325
    ZJ=     .53   2.4317E-12    6.5024E-01   5.7754E+00   2.5890E-02   1.3955E+00   2.2099E+00   2.6477E+00   3.3485E+00   .00444
 THETA=    0.     4.8634E-12    7.5229E-01   5.5531E+00   4.1775E-01   1.3833E+00   2.6772E+00   2.5422E+00   3.5513E+00   .00504
     K=    0.     7.2952E-12    8.9062E-01   5.5175E+00   7.0312E-01   1.3694E+00   3.3403E+00   2.4272E+00   3.7510E+00   .00355
                  9.7269E-12    1.0222E+00   5.3849E+00   9.4803E-01   1.3509E+00   3.9817E+00   2.3415E+00   3.8952E+00   .00412
                  1.2159E-11    1.1541E+00   5.2687E+00   1.1455E+00   1.3308E+00   4.6252E+00   2.2703E+00   4.0075E+00   .00443
                  1.4590E-11    1.2850E+00   5.1639E+00   1.3097E+00   1.3042E+00   5.2553E+00   2.2248E+00   4.0411E+00   .00571
                  1.7022E-11    1.4160E+00   5.1865E+00   1.4482E+00   1.2831E+00   5.8713E+00   2.1959E+00   4.1402E+00   .00673
                  1.9454E-11    1.5494E+00   5.0230E+00   1.5680E+00   1.2556E+00   6.4735E+00   2.1753E+00   4.1807E+00   .00767
                  2.1885E-11    1.6836E+00   4.9736E+00   1.6745E+00   1.2254E+00   7.0649E+00   2.1652E+00   4.1902E+00   .00846
                  2.4317E-11    1.8234E+00   4.9374E+00   1.7711E+00   1.1936E+00   7.6494E+00   2.1645E+00   4.1951E+00   .00904
                  2.6749E-11    1.9699E+00   4.9139E+00   1.8606E+00   1.1588E+00   8.2316E+00   2.1724E+00   4.1786E+00   .00945
                  2.9181E-11    2.1259E+00   4.9016E+00   1.9451E+00   1.1208E+00   8.8167E+00   2.1805E+00   4.1474E+00   .00974
                  3.1612E-11    2.2940E+00   4.8936E+00   2.0263E+00   1.0795E+00   9.4118E+00   2.2163E+00   4.1027E+00   .00994
                  3.4044E-11    2.4771E+00   4.9005E+00   2.1052E+00   1.0349E+00   1.0023E+01   2.2453E+00   4.0401E+00   .00990
                  3.6476E-11    2.6772E+00   4.9209E+00   2.1824E+00   9.8729E-01   1.0655E+01   2.2800E+00   3.9747E+00   .00984
                  3.8907E-11    2.8964E+00   4.9412E+00   2.2583E+00   9.3720E-01   1.1315E+01   2.3251E+00   3.9005E+00   .00940
                  4.1339E-11    3.1365E+00   4.9604E+00   2.3334E+00   8.8521E-01   1.2008E+01   2.3703E+00   3.8284E+00   .00843
                  4.3771E-11    3.3986E+00   4.9982E+00   2.4074E+00   8.3137E-01   1.2731E+01   2.4225E+00   3.7401E+00   .00690
                  4.6203E-11    3.6919E+00   5.0343E+00   2.4852E+00   7.7534E-01   1.3514E+01   2.4795E+00   3.6614E+00   .00435
                  4.8634E-11    4.0122E+00   5.0774E+00   2.5611E+00   7.1631E-01   1.4312E+01   2.5447E+00   3.5671E+00   .00160
                  5.1066E-11    4.3275E+00   5.1376E+00   2.6192E+00   6.5216E-01   1.4927E+01   2.6242E+00   3.4432E+00   .00000
```

```
                         N          RHO          U           V        (S-S1)/CV       P            M            T        DM/M1
   S=      1.02     0.           5.4060E-01  5.9001E+00  -0.          1.4073E+00  1.7269E+00  2.7901E+00  3.1941E+00  .00433
   RB=      .67     2.5970E-02   6.2220E-01  5.8345E+00  -9.4109E-02  1.3945E+00  2.0755E+00  2.6804E+00  3.3357E+00  .00655
   ZB=      .53     5.1739E-02   6.8232E-01  5.7383E+00   2.3107E-01  1.3825E+00  2.3335E+00  2.6246E+00  3.4139E+00  .00635
THETA=     0.       7.7634E-02   8.0331E-01  5.6242E+00   4.4767E-01  1.3679E+00  2.4900E+00  2.5155E+00  3.5975E+00  .00605
    K=     0.       1.0348E-01   9.2000E-01  5.5032E+00   7.4258E-01  1.3508E+00  3.4350E+00  2.4315E+00  3.7337E+00  .00625
                    1.2935E-01   1.0424E+00  5.4014E+00   9.5532E-01  1.3309E+00  4.0111E+00  2.3633E+00  3.8473E+00  .00650
                    1.5522E-01   1.1600E+00  5.3006E+00   1.1365E+00  1.3044E+00  4.5878E+00  2.3120E+00  3.9347E+00  .00714
                    1.8104E-01   1.2917E+00  5.2244E+00   1.2915E+00  1.2832E+00  5.1632E+00  2.2750E+00  3.9973E+00  .00795
                    2.0630E-01   1.4201E+00  5.1573E+00   1.4266E+00  1.2555E+00  5.7346E+00  2.2535E+00  4.0342E+00  .00871
                    2.3243E-01   1.5524E+00  5.1040E+00   1.5404E+00  1.2253E+00  6.3029E+00  2.2304E+00  4.0630E+00  .00930
                    2.5870E-01   1.6904E+00  5.0633E+00   1.6547E+00  1.1924E+00  6.8713E+00  2.2333E+00  4.0647E+00  .00947
                    2.8457E-01   1.8361E+00  5.0303E+00   1.7544E+00  1.1565E+00  7.4423E+00  2.2386E+00  4.0533E+00  .01020
                    3.1044E-01   1.9922E+00  5.0201E+00   1.8479E+00  1.1172E+00  8.0215E+00  2.2531E+00  4.0205E+00  .01043
                    3.3631E-01   2.1613E+00  5.0144E+00   1.9369E+00  1.0744E+00  8.6143E+00  2.2756E+00  3.9456E+00  .01054
                    3.6218E-01   2.3463E+00  5.0178E+00   2.0229E+00  1.0281E+00  9.2263E+00  2.3058E+00  3.9323E+00  .01051
                    3.8805E-01   2.5494E+00  5.0292E+00   2.1065E+00  9.7860E-01  9.8629E+00  2.3429E+00  3.8684E+00  .01020
                    4.1392E-01   2.7726E+00  5.0472E+00   2.1886E+00  9.2046E-01  1.0529E+01  2.3854E+00  3.7976E+00  .00963
                    4.3979E-01   3.0183E+00  5.0712E+00   2.2699E+00  8.7201E-01  1.1230E+01  2.4344E+00  3.7206E+00  .00847
                    4.6566E-01   3.2889E+00  5.1015E+00   2.3537E+00  8.1526E-01  1.1966E+01  2.4889E+00  3.6382E+00  .00665
                    4.9153E-01   3.5922E+00  5.1372E+00   2.4341E+00  7.5561E-01  1.2754E+01  2.5498E+00  3.5506E+00  .00402
                    5.1739E-01   3.9306E+00  5.1786E+00   2.5168E+00  6.9175E-01  1.3573E+01  2.6187E+00  3.4531E+00  .00133
                    5.4326E-01   4.2688E+00  5.2376E+00   2.5747E+00  6.2257E-01  1.4217E+01  2.7038E+00  3.3304E+00  .00000

                         N          RHO          U           V        (S-S1)/CV       P            M            T        DM/M1
   S=      1.06     0.           5.6801E-01  5.8627E+00  -0.          1.4073E+00  1.6504E+00  2.7452E+00  3.2577E+00  .00391
   RB=      .67     2.7347E-02   6.1046E-01  5.8232E+00  -1.3464E-01  1.3934E+00  2.0187E+00  2.7057E+00  3.3008E+00  .00626
   ZB=      .62     5.4693E-02   6.4052E-01  5.7918E+00   1.1929E-01  1.3813E+00  2.1333E+00  2.6828E+00  3.3306E+00  .00917
THETA=     0.       8.2040E-02   7.4485E-01  5.7028E+00   3.3119E-01  1.3666E+00  2.5966E+00  2.5857E+00  3.4861E+00  .00636
    K=     0.       1.0939E-01   8.4587E-01  5.6063E+00   5.8498E-01  1.3496E+00  3.0505E+00  2.5086E+00  3.6063E+00  .00605
                    1.3673E-01   9.5869E-01  5.5088E+00   7.9999E-01  1.3294E+00  3.5638E+00  2.4401E+00  3.7173E+00  .00683
                    1.6408E-01   1.0742E+00  5.4131E+00   9.9064E-01  1.3074E+00  4.0863E+00  2.3872E+00  3.8039E+00  .00740
                    1.9143E-01   1.1938E+00  5.3401E+00   1.1563E+00  1.2821E+00  4.6185E+00  2.3477E+00  3.8659E+00  .00810
                    2.1877E-01   1.3173E+00  5.2731E+00   1.3022E+00  1.2541E+00  5.1547E+00  2.3205E+00  3.9132E+00  .00881
                    2.4612E-01   1.4458E+00  5.2166E+00   1.4321E+00  1.2234E+00  5.6949E+00  2.3045E+00  3.9369E+00  .00945
                    2.7347E-01   1.5810E+00  5.1765E+00   1.5497E+00  1.1898E+00  6.2408E+00  2.2986E+00  3.9473E+00  .00995
                    3.0081E-01   1.7249E+00  5.1462E+00   1.6577E+00  1.1530E+00  6.7951E+00  2.3022E+00  3.9394E+00  .01033
                    3.2816E-01   1.8799E+00  5.1272E+00   1.7567E+00  1.1125E+00  7.3616E+00  2.3150E+00  3.9159E+00  .01053
                    3.5551E-01   2.0489E+00  5.1187E+00   1.8546E+00  1.0683E+00  7.9452E+00  2.3366E+00  3.8777E+00  .01072
                    3.8285E-01   2.2345E+00  5.1135E+00   1.9468E+00  1.0204E+00  8.5510E+00  2.3603E+00  3.8208E+00  .01068
                    4.1020E-01   2.4390E+00  5.1264E+00   2.0364E+00  9.6920E-01  9.1837E+00  2.4033E+00  3.7653E+00  .01037
                    4.3755E-01   2.6648E+00  5.1444E+00   2.1244E+00  9.1509E-01  9.8479E+00  2.4409E+00  3.6956E+00  .00963
                    4.6489E-01   2.9146E+00  5.1670E+00   2.2113E+00  8.5834E-01  1.0548E+01  2.4909E+00  3.6191E+00  .00832
                    4.9224E-01   3.1920E+00  5.1906E+00   2.2983E+00  7.9881E-01  1.1288E+01  2.5535E+00  3.5362E+00  .00636
                    5.1959E-01   3.5047E+00  5.2307E+00   2.3869E+00  7.3578E-01  1.2090E+01  2.6174E+00  3.4467E+00  .00376
                    5.4693E-01   3.8582E+00  5.2700E+00   2.4748E+00  6.6820E-01  1.2916E+01  2.6894E+00  3.3477E+00  .00118
                    5.7428E-01   4.2133E+00  5.3260E+00   2.5410E+00  5.9574E-01  1.3589E+01  2.7770E+00  3.2253E+00  .00000
```

```
                                       SHOCK AND BODY PARAMETERS
     S            RB          ZB           K          THETA         RS          ZS           WS          CP          P/PO
 9.2721E-01  6.6666E-01  4.8889E-01  0.               0.       1.1437E+00  4.8889E-01  4.7019E-07  2.7153E-02  3.5976E-02
 9.7165E-01  6.6666E-01  5.3333E-01  0.               0.       1.1773E+00  5.3333E-01  3.0875E-06  2.5263E-02  3.4960E-02
 1.0161E+00  6.6666E-01  5.7777E-01  0.               0.       1.2099E+00  5.7777E-01 -1.9947E-08  2.4440E-02  3.6989E-02
 1.0605E+00  6.6666E-01  6.2222E-01  0.               0.       1.2409E+00  6.2222E-01 -4.1449E-06  3.3747E-02  3.9520E-02
```

```
CASE TITLE - SAMPLE PROBLEM 2 - SPHERE-CONE VEHICLE

********************************* INPUT DATA *********************************

INTAP              2      OTAP               1      MAXIT             -1

NCASE              1      NFILE              1      NSEG               0

MAXI               0      MAXN               0      NTAP               0

IPR(1)            -5      IPR(2)            -5      IPR(3)            -5

IPR(4)            -5      IPR(5)            -5      IPR(6)            -5

GAMMA     1.400E+00       M          4.000E+00      THETA      4.000E+01

Z/RN      1.400E+00       TCRIT1     2.000E+00      TCRIT2     1.000E+01

P1        1.000E+00       RHO1       1.000E+00      RN         1.000E+00

MU        4.000E-02       T1         1.000E+00      ERRORN     1.000E-05

ERRORP    1.000E-08
```

```
           SEGMENT NUMBER  1          ITERATION NUMBER  984          TIME=  4.4078E+00

    S=      0.         N         RHO          U            V        (S-S1)/CV       P            M            T         DM/M1
   RB=      0.     0.          5.0344E+00  0.          -0.           7.9001E-01  2.1177E+01  0.           4.2062E+00   .00149
   ZB=      0.     2.9632E-02  5.0122E+00  0.          -1.7524E-01   7.9001E-01  2.1045E+01  7.2260E-02   4.1937E+00   .00073
THETA=     90.00   5.9263E-02  4.9811E+00  0.          -3.5176E-01   7.9001E-01  2.0802E+01  1.4527E-01   4.1692E+00   .00140
    K=      1.00   8.8895E-02  4.9149E+00  0.          -5.1831E-01   7.9001E-01  2.0498E+01  2.1474E-01   4.1672E+00   .00135
                   1.1857E-01  4.8357E+00  0.          -6.9525E-01   7.9001E-01  2.0014E+01  2.8407E-01   4.1359E+00   .00142
                   1.4816E-01  4.7250E+00  0.          -8.5592E-01   7.9001E-01  1.9376E+01  3.5622E-01   4.1017E+00   .00114
                   1.7779E-01  4.5714E+00  0.          -1.0353E+00   7.9001E-01  1.8530E+01  4.3490E-01   4.0603E+00   .00000

    S=       .15       N         RHO          U            V        (S-S1)/CV       P            M            T         DM/M1
   RB=       .14   0.          4.9278E+00  4.6376E-01  -0.           7.9001E-01  2.0550E+01  1.9193E-01   4.1702E+00   .00022
   ZB=       .01   3.0573E-02  4.9140E+00  4.7454E-01  -1.6754E-01   7.8855E-01  2.0440E+01  2.0855E-01   4.1595E+00   .00104
THETA=     81.67   6.1146E-02  4.8941E+00  4.8618E-01  -3.4072E-01   7.8714E-01  2.0295E+01  2.4640E-01   4.1463E+00   .00000
    K=      1.00   9.1718E-02  4.8405E+00  4.9906E-01  -5.0600E-01   7.8581E-01  1.9958E+01  2.9622E-01   4.1232E+00   .00107
                   1.2229E-01  4.7630E+00  5.1608E-01  -6.7617E-01   7.8451E-01  1.9497E+01  3.5542E-01   4.0913E+00   .00125
                   1.5286E-01  4.6549E+00  5.3467E-01  -8.5100E-01   7.8322E-01  1.8846E+01  4.2214E-01   4.0447E+00   .00109
                   1.8344E-01  4.5583E+00  5.5441E-01  -1.0124E+00   7.8190E-01  1.8277E+01  4.8730E-01   4.0195E+00   .00000

    S=       .29       N         RHO          U            V        (S-S1)/CV       P            M            T         DM/M1
   RB=       .29   0.          4.6752E+00  8.9543E-01  -0.           7.9001E-01  1.9090E+01  3.7451E-01   4.0553E+00   .00059
   ZB=       .04   3.1775E-02  4.6929E+00  9.2059E-01  -1.5974E-01   7.8333E-01  1.9074E+01  3.9418E-01   4.0643E+00   .00223
THETA=     73.33   6.3549E-02  4.7069E+00  9.6228E-01  -3.2530E-01   7.7766E-01  1.9035E+01  4.2683E-01   4.0442E+00   .00231
    K=      1.00   9.5324E-02  4.6882E+00  1.0040E+00  -4.8196E-01   7.7152E-01  1.8813E+01  4.6985E-01   4.0133E+00   .00234
                   1.2710E-01  4.6479E+00  1.0522E+00  -6.4122E-01   7.6527E-01  1.8472E+01  5.2240E-01   3.9742E+00   .00211
                   1.5887E-01  4.5767E+00  1.1066E+00  -8.0488E-01   7.5881E-01  1.7961E+01  5.8377E-01   3.9243E+00   .00195
                   1.9065E-01  4.5088E+00  1.1648E+00  -9.5907E-01   7.5208E-01  1.7470E+01  6.4782E-01   3.8743E+00   .00000

    S=       .44       N         RHO          U            V        (S-S1)/CV       P            M            T         DM/M1
   RB=       .42   0.          4.2897E+00  1.3319E+00  -0.           7.9001E-01  1.6923E+01  5.6667E-01   3.9452E+00   .00035
   ZB=       .09   3.4035E-02  4.3548E+00  1.3744E+00  -1.4577E-01   7.7718E-01  1.7064E+01  5.9011E-01   3.9194E+00   .00206
THETA=     65.00   6.8071E-02  4.4174E+00  1.4228E+00  -2.9514E-01   7.6437E-01  1.7187E+01  6.2262E-01   3.8907E+00   .00182
    K=      1.00   1.0211E-01  4.4500E+00  1.4785E+00  -4.3373E-01   7.5147E-01  1.7142E+01  6.6348E-01   3.8522E+00   .00207
                   1.3614E-01  4.4657E+00  1.5425E+00  -5.7358E-01   7.3806E-01  1.6997E+01  7.1290E-01   3.8062E+00   .00165
                   1.7018E-01  4.4520E+00  1.6143E+00  -7.1727E-01   7.2391E-01  1.6637E+01  7.7116E-01   3.7481E+00   .00145
                   2.0421E-01  4.4336E+00  1.6918E+00  -8.5606E-01   7.0897E-01  1.6344E+01  8.3463E-01   3.6864E+00   .00000

    S=       .58       N         RHO          U            V        (S-S1)/CV       P            M            T         DM/M1
   RB=       .55   0.          3.7821E+00  1.7409E+00  -0.           7.9001E-01  1.4159E+01  7.6227E-01   3.7514E+00   .00302
   ZB=       .16   3.7379E-02  3.8997E+00  1.7914E+00  -1.2425E-01   7.6942E-01  1.4508E+01  7.8686E-01   3.7202E+00   .00455
THETA=     56.67   7.4758E-02  4.0208E+00  1.8475E+00  -2.5319E-01   7.4793E-01  1.4820E+01  8.2091E-01   3.6859E+00   .00412
    K=      1.00   1.1214E-01  4.1177E+00  1.9173E+00  -3.6910E-01   7.2535E-01  1.4931E+01  8.6514E-01   3.6391E+00   .00411
                   1.4952E-01  4.2060E+00  2.0006E+00  -4.8458E-01   7.0135E-01  1.5066E+01  9.1920E-01   3.5821E+00   .00299
                   1.8689E-01  4.2738E+00  2.0940E+00  -5.9886E-01   6.7603E-01  1.5022E+01  9.8161E-01   3.5159E+00   .00176
                   2.2427E-01  4.3226E+00  2.1945E+00  -7.1504E-01   6.4963E-01  1.4865E+01  1.0519E+00   3.4390E+00   .00000

    S=       .73       N         RHO          U            V        (S-S1)/CV       P            M            T         DM/M1
   RB=       .66   0.          3.2651E+00  2.1384E+00  -0.           7.9001E-01  1.1549E+01  9.6093E-01   3.5372E+00   .00229
   ZB=       .25   4.1857E-02  3.4461E+00  2.1849E+00  -9.1762E-02   7.5878E-01  1.2072E+01  9.8747E-01   3.5032E+00   .00324
THETA=     48.33   8.3714E-02  3.6301E+00  2.2458E+00  -1.8618E-01   7.2674E-01  1.2575E+01  1.0233E+00   3.4041E+00   .00248
    K=      1.00   1.2557E-01  3.7927E+00  2.3197E+00  -2.6566E-01   6.9405E-01  1.2940E+01  1.0683E+00   3.4119E+00   .00221
                   1.6743E-01  3.9493E+00  2.4044E+00  -3.4390E-01   6.6033E-01  1.3241E+01  1.1211E+00   3.3526E+00   .00110
                   2.0928E-01  4.0901E+00  2.4946E+00  -4.1901E-01   6.2568E-01  1.3433E+01  1.1797E+00   3.2842E+00   .00041
                   2.5114E-01  4.2012E+00  2.5927E+00  -5.0472E-01   5.9033E-01  1.3458E+01  1.2473E+00   3.2033E+00   .00000
   S=     .87       N*             RHO          U            V            (S-S1)/CV     P             M             T            DH/H1
  RB=     .77    0.            2.6398E+00   2.5585E+00  -0.             7.9001E-01   8.5763E+00   1.1997E+00   3.2493E+00   .00362
  ZB=     .36    4.9523E-02    2.8818E+00   2.5701E+00  -3.2730E-02     7.5048E-01   9.3211E+00   1.2079E+00   3.2344E+00   .00519
THETA=  40.00    9.9045E-02    3.1284E+00   2.6042E+00  -8.0467E-02     7.0917E-01   1.0033E+01   1.2296E+00   3.2071E+00   .00550
   K=    1.00    1.4857E-01    3.3605E+00   2.6590E+00  -1.1734E-01     6.6648E-01   1.0627E+01   1.2649E+00   3.1623E+00   .00611
                 1.9809E-01    3.5936E+00   2.7346E+00  -1.5973E-01     6.2144E-01   1.1159E+01   1.3134E+00   3.1053E+00   .00543
                 2.4761E-01    3.8335E+00   2.8253E+00  -1.9595E-01     5.7393E-01   1.1649E+01   1.3731E+00   3.0347E+00   .00367
                 2.9714E-01    4.0510E+00   2.9451E+00  -2.5116E-01     5.2285E-01   1.1939E+01   1.4540E+00   2.9519E+00   .00000


                                                    SHOCK AND BODY PARAMETERS

     S            RB           ZB            K            THETA         RS            ZS           WS            CP           P/P0
0.           0.           0.           1.0000E+00   9.0000E+01   0.           -1.7779E-01  -6.8819E-07   1.9015E+00   1.0052E+00
1.4544E-01   1.4493E-01   1.0558E-02   1.0000E+00   8.1667E+01   1.7152E-01   -1.7094E-01  -7.4018E-07   1.7455E+00   9.7540E-01
2.9089E-01   2.8680E-01   4.2010E-02   1.0000E+00   7.3333E+01   3.4148E-01   -1.4063E-01  -8.0570E-07   1.6152E+00   9.0613E-01
4.3633E-01   4.2262E-01   9.3692E-02   1.0000E+00   6.5000E+01   5.0892E-01   -9.1387E-02  -9.2682E-07   1.4217E+00   8.0327E-01
5.8178E-01   5.4951E-01   1.6451E-01   1.0000E+00   5.6667E+01   6.7275E-01   -2.2866E-02  -1.1032E-06   1.1775E+00   6.7343E-01
7.2722E-01   6.6480E-01   2.5297E-01   1.0000E+00   4.8333E+01   8.3175E-01    6.5366E-02  -1.3157E-06   9.4189E-01   5.4818E-01
8.7266E-01   7.6604E-01   3.5721E-01   1.0000E+00   4.0000E+01   9.9366E-01    1.6622E-01  -1.7338E-06   6.7646E-01   4.0708E-01
```

```
                    SEGMENT NUMBER  2            ITERATION NUMBER  318          TIME=  1.7201E+00

     S=    1.02        N             RHO           U            V           (S-S1)/CV       P             H             T            DH/H1
    RB=     .86   0.             2.7978E+00   2.4536E+00 -0.             7.9001E-01   9.3032E+00   1.1372E+00   3.3252E+00    .00351
    ZB=     .47   8.1624E-02     2.9632E+00   2.6417E+00 -1.5262E-01     7.1626E-01   9.3653E+00   1.2579E+00   3.1600E+00    .00932
 THETA=   40.00   1.6325E-01     3.1946E+00   2.8063E+00 -2.5726E-01     6.4170E-01   9.6575E+00   1.3708E+00   3.0231E+00    .00971
     K=    0.     2.4487E-01     3.5295E+00   2.9672E+00 -3.2219E-01     5.6150E-01   1.0248E+01   1.4803E+00   2.9306E+00    .00566
                  3.2649E-01     3.9261E+00   3.1415E+00 -3.7264E-01     4.7191E-01   1.0877E+01   1.6063E+00   2.7703E+00    .00000

     S=    1.16        N             RHO           U            V           (S-S1)/CV       P             H             T            DH/H1
    RB=     .95   0.             2.9067E+00   2.3921E+00 -0.             7.9001E-01   9.6145E+00   1.1002E+00   3.3764E+00    .00145
    ZB=     .58   8.6726E-02     3.0379E+00   2.6443E+00 -1.3967E-01     7.0359E-01   9.5755E+00   1.2605E+00   3.1520E+00    .01102
 THETA=   40.00   1.7345E-01     3.1983E+00   2.8861E+00 -3.0994E-01     6.1609E-01   9.4297E+00   1.4208E+00   2.9410E+00    .01149
     K=    0.     2.6018E-01     3.4671E+00   3.1006E+00 -4.1018E-01     5.2419E-01   9.6297E+00   1.5861E+00   2.7774E+00    .00598
                  3.4691E-01     3.7998E+00   3.3011E+00 -5.0096E-01     4.2438E-01   9.4076E+00   1.7476E+00   2.6074E+00    .00000

     S=    1.31        N             RHO           U            V           (S-S1)/CV       P             H             T            DH/H1
    RB=    1.05   0.             2.9381E+00   2.3722E+00 -0.             7.9001E-01   9.9630E+00   1.0887E+00   3.3910E+00    .00123
    ZB=     .69   9.0449E-02     3.1279E+00   2.6431E+00 -1.1889E-01     6.9219E-01   9.8619E+00   1.2593E+00   3.1523E+00    .01121
 THETA=   40.00   1.8090E-01     3.2858E+00   2.9360E+00 -3.1446E-01     5.9017E-01   9.5411E+00   1.4645E+00   2.9038E+00    .01207
     K=    0.     2.7135E-01     3.4949E+00   3.1931E+00 -4.5254E-01     4.8878E-01   9.3990E+00   1.6620E+00   2.6893E+00    .00592
                  3.6180E-01     3.7173E+00   3.3836E+00 -5.8558E-01     3.9534E-01   9.3329E+00   1.8342E+00   2.5106E+00    .00000

     S=    1.45        N             RHO           U            V           (S-S1)/CV       P             H             T            DH/H1
    RB=    1.14   0.             2.9611E+00   2.3564E+00 -0.             7.9001E-01   1.0072E+01   1.0798E+00   3.4016E+00    .00124
    ZB=     .80   9.3527E-02     3.2023E+00   2.6501E+00 -1.0952E-01     6.8091E-01   1.0078E+01   1.2636E+00   3.1470E+00    .01143
 THETA=   40.00   1.8705E-01     3.3911E+00   2.9773E+00 -2.9883E-01     5.6661E-01   9.7398E+00   1.4922E+00   2.3722E+00    .01160
     K=    0.     2.8058E-01     3.5621E+00   3.2531E+00 -4.6137E-01     4.6177E-01   9.3959E+00   1.7098E+00   2.6377E+00    .00477
                  3.7411E-01     3.6842E+00   3.4235E+00 -6.1953E-01     3.8410E-01   9.1138E+00   1.8679E+00   2.4737E+00    .00000


                                                   SHOCK AND BODY PARAMETERS

       S             RB           ZB             K            THETA         RS           ZS           WS            CP           P/P0
  1.0181E+00    8.5953E-01   4.6403E-01   0.             4.0000E+01   1.1096E+00   2.5876E-01   1.4347E-06   7.4136E-01   4.4153E-01
  1.1636E+00    9.5302E-01   5.8005E-01   0.             4.0000E+01   1.2188E+00   3.5706E-01  -1.3495E-06   7.8701E-01   4.6585E-01
  1.3090E+00    1.0465E+00   6.9146E-01   0.             4.0000E+01   1.3237E+00   4.5890E-01   2.2044E-06   8.0027E-01   4.7290E-01
  1.4544E+00    1.1400E+00   8.0248E-01   0.             4.0000E+01   1.4266E+00   5.6241E-01  -6.4036E-06   8.1004E-01   4.7809E-01
```

```
                      SEGMENT NUMBER  3               ITERATION NUMBER  328             TIME=  1.7741E+00

    S=    1.60        N            RHO           U           V          (S-S1)/CV       P            M            T           DH/H1
   RB=    1.23    0.            2.9986E+00  2.3309E+00 -0.            7.9001E-01  1.0251E+01  1.0654E+00  3.4157E+00   .00125
   ZB=     .91    9.6446E-02    3.2665E+00  2.6636E+00 -1.0316E-01    6.6954E-01  1.0245E+01  1.2721E+00  3.1363E+00   .01160
THETA=   40.00    1.9239E-01    3.5045E+00  3.0141E+00 -2.8092E-01    5.4404E-01  9.9714E+00  1.5167E+00  2.8445E+00   .01095
    K=    0.      2.8934E-01    3.6509E+00  3.2962E+00 -4.5856E-01    4.3553E-01  9.5013E+00  1.7434E+00  2.6025E+00   .00362
                  3.8578E-01    3.6738E+00  3.4303E+00 -6.3027E-01    3.8059E-01  9.0457E+00  1.8795E+00  2.4623E+00   .00000

    S=    1.75        N            RHO           U           V          (S-S1)/CV       P            M            T           DH/H1
   RB=    1.33    0.            3.0376E+00  2.3034E+00 -0.            7.9001E-01  1.0439E+01  1.0502E+00  3.4304E+00   .00133
   ZB=    1.03    9.9439E-02    3.3233E+00  2.5738E+00 -9.7818E-02    6.5826E-01  1.0377E+01  1.2821E+00  3.1225E+00   .01212
THETA=   40.00    1.9898E-01    3.6094E+00  3.0481E+00 -2.6519E-01    5.2353E-01  1.0181E+01  1.5397E+00  2.8206E+00   .01001
    K=    0.      2.9832E-01    3.7382E+00  3.3246E+00 -4.4739E-01    4.2106E-01  9.6514E+00  1.7645E+00  2.5813E+00   .00251
                  3.9776E-01    3.6819E+00  3.4229E+00 -6.2197E-01    3.8330E-01  9.0983E+00  1.3703E+00  2.4711E+00   .00000

    S=    1.89        N            RHO           U           V          (S-S1)/CV       P            M            T           DH/H1
   RB=    1.42    0.            3.0763E+00  2.2782E+00 -0.            7.9001E-01  1.0625E+01  1.0300E+00  3.4573E+00   .00111
   ZB=    1.14    1.0259E-01    3.3812E+00  2.6957E+00 -9.3282E-02    6.4648E-01  1.0507E+01  1.2932E+00  3.1074E+00   .01263
THETA=   40.00    2.0519E-01    3.7103E+00  3.0839E+00 -2.5383E-01    5.0367E-01  1.0373E+01  1.5640E+00  2.7954E+00   .00867
    K=    0.      3.0778E-01    3.8217E+00  3.3446E+00 -4.3231E-01    4.0773E-01  9.8229E+00  1.7778E+00  2.5713E+00   .00119
                  4.1038E-01    3.7007E+00  3.4048E+00 -6.0264E-01    3.8967E-01  9.2271E+00  1.8512E+00  2.4920E+00   .00000

    S=    2.04        N            RHO           U           V          (S-S1)/CV       P            M            T           DH/H1
   RB=    1.51    0.            3.1069E+00  2.2571E+00 -0.            7.9001E-01  1.0773E+01  1.0244E+00  3.4670E+00   .00114
   ZB=    1.25    1.0615E-01    3.4357E+00  2.7133E+00 -8.7988E-02    6.3504E-01  1.0622E+01  1.3049E+00  3.0917E+00   .01319
THETA=   40.00    2.1231E-01    3.7923E+00  3.1172E+00 -2.4528E-01    4.8655E-01  1.0514E+01  1.5871E+00  2.7725E+00   .00734
    K=    0.      3.1846E-01    3.8821E+00  3.3540E+00 -4.1675E-01    4.0018E-01  9.9653E+00  1.7829E+00  2.5670E+00   .00027
                  4.2462E-01    3.7213E+00  3.3840E+00 -5.9148E-01    3.9671E-01  9.3598E+00  1.8301E+00  2.5152E+00   .00000


                                                 SHOCK AND BODY PARAMETERS

      S            RB           ZB                K         THETA         RS           ZS           WS            CP           P/PI
 1.5999E+00   1.2335E+00   9.1430E-01   0.                4.0000E+01  1.5290E+00  6.6672E-01 -1.5632E-06   4.2031E-01   4.3657E-01
 1.7453E+00   1.3270E+00   1.0257E+00   0.                4.0000E+01  1.6317E+00  7.7004E-01 -1.3942E-08   4.4273E-01   4.4947E-01
 1.8908E+00   1.4205E+00   1.1371E+00   0.                4.0000E+01  1.7348E+00  8.7334E-01 -3.2755E-06   4.5340E-01   5.0463E-01
 2.0362E+00   1.5140E+00   1.2435E+00   0.                4.0000E+01  1.8392E+00  9.7561E-01  5.0525E-07   4.7201E-01   5.1135E-01
```

```
                    SEGMENT NUMBER  4             ITERATION NUMBER  174           TIME=  3.4116E-01

   S=     2.23       N           RHO          U           V          (S-S1)/CV      P            H            T         DM/M1
  RB=     1.64   0.          3.1147E+00  2.2524E+00 -0.            7.9001E-01  1.0811E+01  1.0218E+00  3.4711E+00  .00100
  ZB=     1.40   8.9248E-02  3.4197E+00  2.6548E+00 -6.9417E-02   6.5152E-01  1.0728E+01  1.2672E+00  3.1372E+00  .01315
THETA=   40.00   1.7850E-01  3.7624E+00  3.0230E+00 -1.8323E-01   5.1744E-01  1.0724E+01  1.5161E+00  2.8504E+00  .00935
   K=     0.     2.6774E-01  3.9433E+00  3.2627E+00 -2.9964E-01   4.2837E-01  1.0477E+01  1.6988E+00  2.6570E+00  .00225
                 3.5699E-01  3.9261E+00  3.3640E+00 -4.3246E-01   3.9318E-01  1.0053E+01  1.7914E+00  2.5606E+00  .00095
                 4.4624E-01  3.7522E+00  3.3530E+00 -5.4975E-01   4.0744E-01  9.5710E+00  1.7980E+00  2.5507E+00  .00001


                                              SHOCK AND BODY PARAMETERS

      S            RB           ZB          K           THETA         RS           ZS           WS           CP          P/P0
 2.2339E+00   1.6410E+00   1.4000E+00   0.          4.0000E+01   1.9829E+00   1.1132E+00   3.6455E-05   6.7600E-01   5.1316E-01
```

SECTION VIII

CONCLUSIONS

1. THE DIFFERENCING SCHEME

The differencing scheme developed in this study incorporates a complete decoupling of the magnitude of the stabilizing terms from the finite difference mesh sizes used. Consequently, different mesh sizes for each coordinate direction and variations in mesh size can be employed. The mesh sizes can be selected based on finite difference accuracy requirements without regard to the stabilizing term influence. Since the stabilizing terms can be specified arbitrarily small, the ability to achieve a valid mathematical model of the inviscid flow problem is ensured. The stability arguments used to establish the differencing scheme should be applicable to any coordinate system. Thus, the differencing scheme is not limited to the blunt body problem. It should be useful for any problem for which the time-dependent technique is applicable.

2. SURFACE BOUNDARY CONDITIONS

The technique developed for treating points on the body surface is exact in the limit of standard finite difference approximations. This technique is applicable to any point for which a streamline and a coordinate line are coincident, in particular, the body points with a body oriented coordinate system. This technique has significantly improved the accuracy of the blunt body solution. Since the general class of body shapes required to realize the objectives of this study can be achieved with body oriented coordinates, this exact method has been employed. However, definite restrictions on the magnitude of a negative body curvature result from this technique. The restriction to smooth body contours is fairly academic, since any finite difference solution for a body with corners will necessarily assume a finite curvature at the discontinuities (e.g., see Ref. 11).

3. COMPUTER EFFICIENCY

The segmented computation procedure developed greatly enhances the computer efficiency obtainable with a time-dependent technique. The computer storage required is minimal regardless of the body length. Computation times are reduced by factors in excess of 1000 when typical high performance vehicles are considered. It has been shown that even the relatively simple axisymmetric flow problem requires the type of computation time reduction realized by this procedure if a unified time-dependent solution is to be economically feasible. An equivalent three-dimensional solution without this type of improved computer efficiency would certainly be impossible with the computers currently available. This procedure should enable the aerodynamicist to employ the time-dependent technique as a practical solution procedure for complex problems such as nonequilibrium and three-dimensional flows. Certainly, this procedure or another equally effective approach is required to transform the time-dependent method from an interesting theoretical solution to a useful computational method.

4. THE BLUNT BODY SOLUTION

A practical computational method has been presented to treat axisymmetric blunt body flows. The computer code presented can consider a very general class of body geometries, sufficient to satisfy the objectives of this study. Since additional stabilizing terms are introduced into the governing equations, reference to this technique as an exact solution may be properly questioned. However, the differencing scheme assures that the effect of these terms can be made negligible relative to the numerical errors. Consequently, this question is rather academic. For practical purposes, the present technique is a direct, exact and unified solution procedure. This is believed to be the first technique reported which contains all of these features and still may be employed as a practical computational method. Extensive comparison with other computational methods has been accomplished. The present method is clearly more accurate than other time-dependent techniques reported. In fact, the accuracy is comparable to other methods currently regarded as standard solution procedures. A calculation for a typical high performance reentry vehicle has been presented. The computation time and computer storage required was quite reasonable. This is believed to be the first reported solution for a problem of this magnitude generated with a unified time-dependent method.

APPENDIX I

STABILITY CONSIDERATIONS

1. RESULTS OF REFERENCE 1

In reference 1 it was shown that useful results could be obtained from a stability analysis of a simple linearized one-dimensional momentum equation of the form

$$u_t + Ku_x = \mu u_{xx} \qquad (71)$$

Employing a forward difference approximation for the partial derivative with respect to time and appropriate first and second central difference approximations for the spatial partial derivatives, the von Neumann (Ref. 5) stability analysis was used. In this analysis, u is replaced by

$$u \rightarrow u_o + (\delta u) \text{ EXP } (\alpha t - i\beta\Delta x)$$

where the second term introduces an error into u.

The differencing scheme will be stable if the errors decay exponentially in time, i.e.,

$$\left| e^{\alpha t} \right| \leq 1 \qquad (72)$$

The details of the analysis are given in reference 1. only the results of that analysis will be given here

$$\frac{K^2(\Delta t)^2}{(\Delta x)^2} \sin^2 \phi + \left[1 - \frac{2\mu\Delta t}{(\Delta x)^2} (1 - \cos \phi)\right]^2 \leq 1 \qquad (73)$$

where

$$\phi = \beta\Delta x \qquad (74)$$

It is easily seen that if μ vanishes, the central difference scheme must be unstable for a finite, positive Δt, i.e.,

$$\frac{K^2(\Delta t)^2}{(\Delta x)^2} \sin^2 \phi \leq 0 \qquad (75)$$

Using a special relation for μ

$$\mu = \nu \frac{(\Delta x)^2}{2\Delta t} \qquad (76)$$

It was found (Ref. 1) that

$$\Delta t \leq \frac{\sqrt{\nu}\Delta x}{K} \qquad (77)$$

should result in a stable differencing scheme where ν is a constant and

$$0 < \nu \leq 1 \tag{78}$$

K was replaced by (u+a) to agree with the Courant-Friedricks-Lewy (CFL) stability criterion, which must always be satisfied. Experience with this expression showed that is was extremely accurate when applied to the multidimensional flow problem. The impressive success of this analysis in reference 1 prompted the attempt to generalize equation (77) discussed in section IV. The computational results presented in section V clearly demonstrate the validity of using this analysis.

2. A STABILITY ANALYSIS FOR BACKWARD DIFFERENCES.

A similar stability analysis will now be presented where u_x is approximated by a backward difference

$$u_x \leftarrow \left[u(x,t) - u(x-\Delta x,t)\right] / \Delta X \tag{79}$$

Introducing this difference approximation into equation (71) it is easily shown that

$$\left| 1 - \frac{2\mu\Delta t}{(\Delta x)^2} (1-\cos \emptyset) - \frac{K\Delta t}{\Delta x} (1-\cos \emptyset + i \sin \emptyset)\right| \leq 1 \tag{80}$$

is the appropriate stability criterion.

By taking the absolute value of this complex expression, it is easily reduced to

$$\Delta t \leq \frac{\Delta x}{\frac{2\mu}{\Delta x} + K} \tag{81}$$

In this case, if μ vanishes, a stable differencing scheme is still possible when K is positive. This accounts for the success of linearly extending the solution in the supersonic region to terminate the solution domain (Refs. 1, 11-15). It is easily shown that this procedure results in a backward difference in the streamwise coordinate with no stabilizing terms relative to that direction, i.e., the second derivatives vanish. The question of stability is more complicated when K vanishes. However, to achieve agreement with the CFL stability criterion, equation (81) must be written

$$\Delta t \leq \frac{\Delta x}{\frac{2\mu}{\Delta x} + u + a} \tag{82}$$

If μ and u vanish, we expect

$$\Delta t \leq \frac{\Delta x}{a} \tag{83}$$

will result in a stable differencing scheme. This will permit a backward difference approximation, together with the requirement that the normal velocity component vanish on the body, to be used to impose surface boundary conditions. Similar arguments using a forward difference approximation for u_x result in

$$\Delta t \leq \frac{\Delta x}{\frac{2\mu}{\Delta x} - K} \tag{84}$$

which allows μ to vanish when K is negative. As a result, the artificial dissipative terms could vanish in the streamwise direction at the stagnation streamline for axisymmetric flow.

APPENDIX II

THE UNSTEADY SHOCK PROBLEM

1. THE METHOD OF GODUNOV

Time-dependent blunt body flows with discontinuous shock waves were considered by Godunov (Ref. 9) using the one-dimensional Riemann problem to describe the wave interactions near the shock surface. Reference 16 contains an excellent description of this approach. The relevant wave phenomena are illustrated in figure 21. W_s is the shock velocity and V_s is the flow velocity behind a normal shock wave. The left running disturbance is the bow shock. The right running disturbance definded by $\frac{dx}{dt}$ = V+a may be either a shock or a finite expansion wave. The wave defined by $\frac{dx}{dt} = V_s$ is an entropy line. The shock jump conditions are expressed by

$$\dot{m} = \left(\frac{\gamma+1}{2}P_s + \frac{\gamma-1}{2}\right)^{1/2} \tag{85}$$

$$\dot{m}V_s + P_s = \dot{m}V_\infty + 1 \tag{86}$$

where $\dot{m}$ is the mass flow across the wave, the subscripts s and ∞ refer to conditions behind the shock and in the free stream, respectively, and nondimensional variables with a perfect gas equation of state have been used. The Godunov method relies

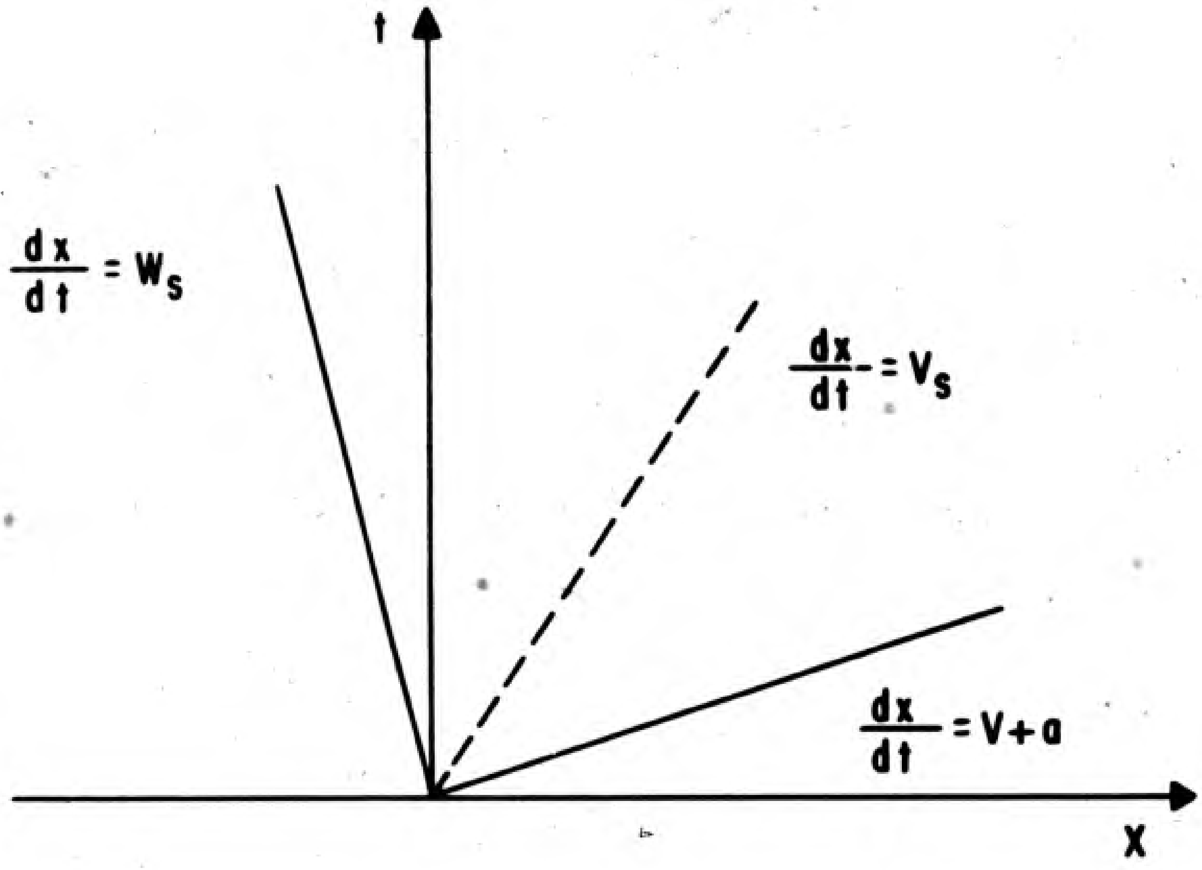


Figure 21. The One-Dimensional Riemann Problem

on two independent solutions for P_s based on independent jump relations for right and left running waves. An iteration procedure is used to obtain acceptable agreement between the two predictions for P_s. The validity of the method appears questionable for the general blunt body problem. Specifically, when the stagnation streamline for a spherical body is considered, this method does not model the problem correctly. This region is characterized by increasing pressure and decreasing velocity as the body is approached from the shock. In the limit of a steady state flow, the flow inside the shock is isentropic. Then, $\dot{m}$ across the right running disturbance is given by

$$\dot{m} = -\delta(\gamma P\rho)^{1/2} \ \frac{1 - \zeta}{1 - \zeta}\delta \tag{87}$$

where

$$\delta = \frac{\gamma - 1}{2\gamma} \tag{88}$$

ζ is the ratio of the pressures behind and in front of the wave and P and are the pressure and density in front of the wave. Since ζ is less than unity, $\dot{m}$ must be negative. Then, equation (86) requires that the velocity in front of the right running disturbance be greater than V_s. This is not in agreement with the physical situation where V decreases with increasing X. To avoid this difficulty, some attempt to consider multidimensional effects is required.

2. THE METHOD OF MORETTI AND ABBETT

A method that includes multidimensional effects in an approximate manner was suggested by Moretti and Abbett (Ref. 13). The governing equations in the shock fixed coordinate system shown in figure 3 are

$$\frac{\partial\rho}{\partial t} + \rho\frac{\partial V}{\partial x} + V\frac{\partial\rho}{\partial x} = -\frac{\partial\rho u}{\partial y} \tag{89}$$

$$\frac{\partial V}{\partial t} + V\frac{\partial V}{\partial x} + \frac{1}{\rho}\frac{\partial P}{\partial x} = -u\frac{\partial V}{\partial y} \tag{90}$$

and S is presumed constant inside the shock. The right hand terms in equations (89) and (90) are considered to be constant

$$A = \frac{\partial\rho u}{\partial y} \tag{91}$$

$$B = U\frac{\partial V}{\partial y} \tag{92}$$

For a rectilinear shock, A and B would vanish. It is reasonable to expect constant values for A and B to adequately model the curved shock problem. Multiplying both sides of equation (89) by $\frac{a}{\rho}$

$$\frac{a}{\rho}\frac{\partial\rho}{\partial t} + V\frac{a}{\rho}\frac{\partial\rho}{\partial x} + a\frac{\partial V}{\partial x} = - A\frac{a}{\rho} \qquad (93)$$

and rearranging equation (90)

$$\frac{\partial V}{\partial t} + V\frac{\partial V}{\partial x} + \frac{a^2}{\rho}\frac{\partial\rho}{\partial x} = -B \qquad (94)$$

where

$$\frac{\partial P}{\partial x} = \left(\frac{\partial P}{\partial\rho}\right)_S \frac{\partial\rho}{\partial x} = a^2\frac{\partial\rho}{\partial x} \qquad (95)$$

for isentropic flow. Adding and subtracting equations (94) and (95)

$$\frac{\partial V}{\partial t} + (V\pm a)\frac{\partial V}{\partial x} \pm \frac{a}{\rho}\left[\frac{\partial\rho}{\partial t} + (V\pm a)\frac{\partial\rho}{\partial x}\right] = - (B\pm A\frac{a}{\rho}) \qquad (96)$$

The ± notation should be used consistently through the remaining equations. Defining

$$\frac{d}{dt} = \frac{\partial}{\partial t} + (V\pm a)\frac{\partial}{\partial x} \qquad (97)$$

Equation (96) can be expressed as

$$\frac{dV}{dt} \pm \frac{a}{\rho}\frac{d\rho}{dt} = - \left[B \pm A\frac{a}{\rho}\right] \qquad (98)$$

This indicates that the directional derivative defined by equation (97) is the significant operator. This operator corresponds to a derivative along the characteristic directions

$$\frac{dx}{dt} = V \pm a \qquad (99)$$

Noting that

$$\frac{d\rho}{dt} = \left(\frac{\partial\rho}{\partial P}\right)_S \frac{dP}{dt} = \frac{1}{a^2}\frac{dP}{dt} \qquad (100)$$

for isentropic flow

$$\frac{dP}{dt} = \rho a \left[\frac{dV}{dt} \pm (B-A\frac{a}{\rho})\right] \qquad (101)$$

is obtained. Equation (101) represents the compatability relation associated with the characteristic directions specified by equation (99). The pressure predictions from the Rankine-Hugonoit shock jump relations and from equation (101) can be used in an iterative scheme to obtain the conditions behind a normal shock. It is easily seen that a clear parallel exists between this approach and the method of Godunov, the difference being that multi-dimensional effects have been considered in an approximate manner.

3. APPLICATION ON THE SYMMETRY AXIS

When the method of reference 13 is applied on the symmetry axis, equations (91) and (92) must be modified. The unsteady shock problem in axisymmetric flow becomes a three-dimensional problem at this point. It is easily shown that if

$$A = \frac{\partial \rho u}{\partial x} + \frac{\partial \rho W}{\partial z} \qquad (102)$$

$$B = U\frac{\partial V}{\partial y} + W\frac{\partial V}{\partial z} \qquad (103)$$

where z is perpendicular to x and y and W is the velocity component along z, all equations developed above are valid at the symmetry axis. For axisymmetric flow, the second term in equation (102) is equal to the first term and the second term in equation (103) vanishes. Consequently, the only modification required is to multiply equation (91) by 2 when computations are performed on the symmetry axis.

REFERENCES

1. Aungier, R. H., *A Time-Dependent Numerical Method For Calculating The Flow About Blunt Bodies*, AFWL-TR-68-52, AF Weapons Laboratory, Kirtland AFB, NM, August 1968.

2. Garabedian, P. R. and Lieberstein, H. M., "On The Numerical Calculation of Detached Bow Shock Waves In Hypersonic Flow," *J. Aeronautical Sci.* *25*, 109-118 (1958).

3. Inouye, M., Rakich, J. V., and Lomax, H., *A Description of Numerical Methods and Computer Programs for Two-dimensional and Axisymmetric Supersonic Flow over Blunt-nosed and Flared Bodies*, NASA TN D-2970 Moffett, Calif., August 1965.

4. Belotserkovskii, O. M., "The Calculation of Flows Past Axisymmetric Bodies with Detached Shock Waves," *Vychislitel'nyi Tsentr Akad. Nauk SSSR* (1961).

5. Von Neumann, J. and Richtmyer, R., "A Method for the Numerical Calculation of Hydrodynamic Shocks," *J. Appl. Phys*. *21*, 232-237 (1950).

6. Lax, P. D., "Weak Solutions of Nonlinear Hyperbolic Equations and Their Numerical Computation," *Commun. Pure Appli. Math*. *7*, 159-193 (1954).

7. Lax, P. D. and Wendroff, B., "Systems of Conservation Laws," *Commun. Pure Appli. Math*. *13*, 217-237 (1960).

8. Lax, P. D. and Wendroff, B., "Differencing Schemes for Hyperbolic Equations with High Order of Accuracy," *Commun. Pure Appli. Math.* *17*, 381-398 (1964).

9. Godunov, S. K., Zabrodin, A. V., and Prokopov, G. P., "A Computational Scheme for Two-dimensional Nonstationary Problems of Gasdynamics and of the Flow from a Shock Wave Approaching a Stationary State," *Zh. Vychislitelnoi Mat. i. Mat. Fiziki* *1*, 1020-1050 (1961).

10. Burstein, S. Z., "Numerical Methods in Multidimensional Shocked Flow," *AIAA J*. *2*, 2111-2117 (1964).

11. Bohachevsky, I. O. and Rubin, E. L., "A Direct Method for Computation of Nonequilibrium Flows with Detached Shock Waves," *AIAA J*. *4*, 600-607 (1966).

12. Bohachevsky, I. O. and Mates, R. S., "A Direct Method for the Calculation of the Flow About an Axisymmetric Blunt Body at Angle of Attack," AIAA J. 4, 776-782 (1966).

13. Moretti, G. and Abbett, M., "A Time-dependent Computational Method for Blunt Body Flows," AIAA J. 4, 2136-2141 (1966).

14. Moretti, G. and Bleick, G., "Three Dimensional Flow Around Blunt Bodies," AIAA J. 5, 1557-1562 (1967).

15. Abbett, M. J. and Fort, R., "Three-dimensional Inviscid Flow About Supersonic Blunt Cones At Angle of Attack," SC-CR-3728, Vol. III, Sandia Laboratories, Albuquerque, NM, 1968.

16. Masson, B. S., Two-dimensional Flow Field Calculations by the Godunov Method, Aeronutronic Report No. U-4137 (1967).

17. Hays, W. D. and Probstein, R. F., Hypersonic Flow Theory, Vol. I (Academic Press Inc., New York, 1966).

18. Courant, R., Friedricks, K. O. and Lewy, H., "Uber die Partiellen Differenzengleichungen der Mathematischen Physik," Math. Ann. 100, 32 (1928).

19. Bushnell, D. M., Jones, R. A., and Hoffman, J. A., Heat Transfer and Pressure Distribution on Spherically Blunted 25° Half-angle Cone at Mach 8 and Angles of Attack Up to 90°, NASA TN D-4792 (1968).

This page intentionally left blank.

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Air Force Weapons Laboratory (WLEE) Kirtland Air Force Base, New Mexico 87117	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE
A COMPUTATIONAL METHOD FOR EXACT, DIRECT, AND UNIFIED SOLUTIONS FOR AXISYMMETRIC FLOW OVER BLUNT BODIES OF ARBITRARY SHAPE (PROGRAM BLUNT)

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
May 1969 to December 1969

5. AUTHOR(S) (First name, middle initial, last name)
Ronald H. Aungier, Capt, USAF

6. REPORT DATE July 1970	7a. TOTAL NO. OF PAGES 318	7b. NO. OF REFS 19
8a. CONTRACT OR GRANT NO. b. PROJECT NO. 5791 c. Task 27 d.	9a. ORIGINATOR'S REPORT NUMBER(S) AFWL-TR-70-16 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

10. DISTRIBUTION STATEMENT This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of AFWL (WLEE), Kirtland AFB, NM 87117. Distribution is limited because of the technology discussed in the report.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY AFWL (WLEE) Kirtland AFB, NM 87117

13. ABSTRACT

(Distribution Limitation Statement No. 2)

A time-dependent numerical method is presented that provides direct, exact, and unified solutions for axisymmetric flows about blunt nosed bodies of essentially arbitrary shape. The differencing scheme used ensures that the required stabilizing terms can be specified arbitrarily small and completely independent of the finite difference mesh sizes used. The method is shown to be more accurate than other reported time-dependent techniques. Computational procedures are introduced to enhance the computer efficiency obtainable with the time-dependent method. Extensive comparison with standard computational methods shows that the present method is comparable in both numerical accuracy and computer efficiency. A FORTRAN IV computer code and instructions for its use are provided.

Unclassified

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Reentry vehicles						
Blunt body problem						
Axisymmetric flow						
Inviscid flow						
Fluid mechanics						
Hypersonic flow						
Time dependent solution						

AFSC-HOLLOMAN AFB, NMEX

Unclassified

Security Classification