https://ntrs.nasa.gov/search.jsp?R=19710013112 2020-03-23T17:21:01+00:00Z

NASA TECHNICAL MEMORANDUM

NASA TM X-2140

NASA TM X-2140

CASE FILE COPY

COMPUTER PROGRAM FOR COMPRESSIBLE LAMINAR OR TURBULENT NONSIMILAR BOUNDARY LAYERS

by Barbara A. Hixon, Ivan E. Beckwith, and Dennis M. Bushnell

Langley Research Center
Hampton, Va. 23365

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1971

1. Report No. NASA TM X-2140	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle COMPUTER PROGRAM FOR COMPRESSIBLE LAMINAR OR TURBULENT NONSIMILAR BOUNDARY LAYERS		5. Report Date April 1971
		6. Performing Organization Code
7. Author(s) Barbara A. Hixon, Ivan E. Beckwith, and Dennis M. Bushnell		8. Performing Organization Report No. L-7399
		10. Work Unit No. 129-04-20-02
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Va. 23365		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

A computer program is described which solves the two-dimensional and axisymmetric forms of the compressible-boundary-layer equations for continuity, mean momentum, and mean total enthalpy by an implicit finite-difference procedure. Turbulent flow is treated by the inclusion of an eddy viscosity model based upon a mixing-length formulation. The eddy conductivity is related to the eddy viscosity by the turbulent Prandtl number which may be an arbitrary function of the distance from the wall. The laminar-boundary-layer equations are recovered when the eddy viscosity is zero. Since a finite-difference procedure is used, the effects of variable wall and edge boundary conditions are easily included by modifying the program inputs.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Boundary-layer computer program Nonsimilar Compressible Laminar Turbulent Mixing-length model Implicit finite-difference procedure	Unclassified – Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	82	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

CONTENTS

COMPUTER PROGRAM FOR COMPRESSIBLE LAMINAR OR TURBULENT NONSIMILAR BOUNDARY LAYERS

By Barbara A. Hixon, Ivan E. Beckwith,
and Dennis M. Bushnell
Langley Research Center

SUMMARY

A computer program is described which solves the two-dimensional and axisymmetric forms of the compressible-boundary-layer equations for continuity, mean momentum, and mean total enthalpy by an implicit finite-difference procedure. Turbulent flow is treated by the inclusion of an eddy viscosity model based upon a mixing-length formulation. The eddy conductivity is related to the eddy viscosity by the turbulent Prandtl number which may be an arbitrary function of the distance from the wall. The laminar-boundary-layer equations are recovered when the eddy viscosity is zero. Since a finite-difference procedure is used, the effects of variable wall and edge boundary conditions are easily included by modifying the program inputs.

INTRODUCTION

Several finite-difference methods are currently available for computing the development of compressible turbulent boundary layers (for example, refs. 1 to 6). The numerical procedures used in these methods are generally different, but results are similar when a common eddy viscosity formulation is employed. Therefore, the main difference between the various methods is in the formulation of the eddy viscosity and turbulent Prandtl number functions used to model the turbulence flux terms appearing in the mean-flow equations.

This report describes a computer program developed to solve the compressible-nonsimilar-boundary-layer equations for continuity, mean momentum, and total mean enthalpy for an ideal gas with constant specific heat. This program was used to obtain the results reported in references 7 and 8. An implicit finite-difference procedure similar to the procedures described in references 9 and 10 is used. The program will solve problems with the following flow configurations: (1) two-dimensional, (2) axisymmetric where the boundary-layer thickness is much less than the body radius, and (3) swept infinite cylinders.

The eddy viscosity is taken as a function of the local boundary-layer thickness, the normal distance from the wall, and the mean velocity gradient in the boundary layer. The formulation for the eddy viscosity is based on the mixing-length models of references 2 and 11. The turbulent Prandtl number may be either a constant or a specified tabulated function of the ratio of the normal distance from the wall to the boundary-layer thickness. By setting the eddy viscosity equal to zero, nonsimilar-laminar-boundary-layer flows can be computed. Since a finite-difference procedure is used, the effects of variable wall and edge boundary conditions and wall blowing or suction are easily included by modifying the program inputs.

The governing partial differential equations and their finite-difference forms are discussed first. Then, the digital computer program is described in detail including the flow charts, program code, and instructions for use with sample input and output.

SYMBOLS

A,B,C — empirical constants in $\bar{f}_{max}$ expression (see eq. (56a))

A^*,B^*,C^*,D^* — coefficients in general difference equation for conservation of energy (see eq. (39))

$\hat{A},\hat{B},\hat{C},\hat{D}$ — coefficients in general difference equation for g-momentum (see eq. (38))

$\overline{A},\overline{B},\overline{C},\overline{D}$ — coefficients in general difference equation for F-momentum (see eq. (25))

A_b — damping constant (see eq. (59))

A_d — damping length (see eq. (59))

C_f — skin-friction coefficient, $\dfrac{\tau_w}{\frac{1}{2}\rho_e u_e^2}$

c_p — specific heat

F — dimensionless chordwise velocity profile, u/u_e

$\bar{f}$ — mixing-length ratio, function of y/δ (see eq. (51))

$\overline{G},\bar{g}$ — coefficients in formula for dependent variables (see eq. (31))

$\hat{G},\hat{g}$ — coefficients in formula for dependent variables (see eq. (42))

G^*, g^*	coefficients in formula for dependent variables (see eq. (43))
g	dimensionless spanwise velocity profile, w/w_e
H	total enthalpy, $h + \frac{u^2}{2} + \frac{w^2}{2}$
H^*	form factor, δ^*/θ^*
h	static enthalpy
j	body-shape index ($j = 0$ for two-dimensional and swept-cylinder flows; $j = 1$ for axisymmetric flows)
K	ratio of successive $\Delta\eta$ steps (see eqs. (22) to (24))
k	molecular conductivity
k^*	static eddy conductivity for turbulent flow, $-\frac{\overline{(\rho v)'h'}}{\partial h/\partial y}$
L	reference length; a constant
l	mixing length
M	Mach number
$\overline{M}$	effective viscosity function for F-momentum equation (see eq. (10b))
$\hat{M}$	effective viscosity function for g-momentum equation (see eq. (10c))
M^*, M'	effective viscosity and turbulent Prandtl number functions for total mean enthalpy equation (see eqs. (10d) and (10e))
m	computational grid index in x-direction (see fig. 2)
N	maximum value of n
N_{Pr}	molecular Prandtl number, $c_p\mu/k$
$N_{Pr,T}$	total turbulent Prandtl number, $c_p\epsilon/\kappa$

$N_{Pr,t}$	static turbulent Prandtl number, $c_p \epsilon / k^*$
N_{St}	Stanton number
n	computational grid index in y-direction (see fig. 2)
P	pressure gradient parameter (see eq. (10h))
p	pressure
$\overline{Q}$	heat-transfer parameter, $\dot{q}_w L / (\mu_s H_e)$
$\dot{q}$	heating rate
$R_{e,x}$	Reynolds number based on x, $\frac{\rho_e u_e x}{\mu_e}$
R_{e,δ^*}	Reynolds number based on displacement thickness, $\frac{\rho_e u_e \delta^*}{\mu_e}$
R_{e,θ^*}	Reynolds number based on momentum thickness, $\frac{\rho_e u_e \theta^*}{\mu_e}$
R_s	reference Reynolds number, $\frac{\rho_s \sqrt{2H_e} L}{\mu_s}$; a constant
r	dimensionless body radius (used only when $j = 1$), r_w/L
S	Sutherland's constant
T	temperature
U	velocity gradient parameter (see eq. (10g))
u	mean physical velocity in streamwise (or chordwise for swept-cylinder problems) direction (see fig. 1)
V	transformed normal velocity (see eq. (11))
v	mean physical velocity in direction normal to surface (see fig. 1)
w	mean physical velocity in spanwise direction (see fig. 1)

x	streamwise (or chordwise for swept-cylinder problems) direction in physical coordinate system (see fig. 1)
y	direction normal to surface in physical coordinate system (see fig. 1)
Z	wall temperature gradient parameter (see eq. (10i))
z	spanwise direction in physical coordinate system (see fig. 1)
δ	boundary-layer thickness in physical plane, taken where F or $g = 0.995$
δ^*	displacement thickness, $\int_0^{y_N}\left(1 - \frac{\rho u}{\rho_e u_e}\right)dy$
ϵ	eddy viscosity for turbulent flow, $-\frac{\overline{(\rho v)'u'}}{\partial\bar{u}/\partial y}$
ϵ_z	eddy viscosity in spanwise direction
ζ	dimensionless enthalpy ratio, H/H_e
η	transformed similarity coordinate (see eq. (8b))
η_δ	value of η when $F = 0.995$
θ	dimensionless enthalpy profile, $\frac{H - H_w}{H_e - H_w}$
θ^*	momentum thickness, $\int_0^{y_N}\left(\frac{\rho u}{\rho_e u_e}\right)\left(1 - \frac{u}{u_e}\right)dy$
κ	total eddy conductivity for turbulent flow, $-\frac{\overline{(\rho v)'H'}}{\partial\bar{H}/\partial y}$
Λ	sweep angle
μ	molecular viscosity
ξ	transformed similarity coordinate (see eq. (8a))
$\bar{\xi}$	transformed streamwise length parameter (see eq. (10f))
ρ	density

σ error or convergence criteria (see eqs. (34) and (46) to (48))

τ shear stress

φ_r density-viscosity product ratio (see eq. (10a))

Subscripts and superscripts:

a average value (see fig. 2 and eqs. (30a) and (30b))

e edge of boundary layer

F F profile

g g profile

i incompressible value

j body-shape index

max maximum value

N maximum value of n (see fig. 2 and eqs. (35), (44), and (45))

n index for points in y-direction

$\bar{n}$ similarity index (see eqs. (8b) and (9b))

o initial conditions at $x = x_o$

r variable reference value, evaluated at edge condition

res resultant

s constant reference value, usually taken as local isentropic stagnation conditions

t isentropic stagnation value

w wall

z		z-direction

∞		free stream

A prime on u, v, w, h, H, or ρ denotes a fluctuating quantity.

A bar over primed quantities indicates a time average.

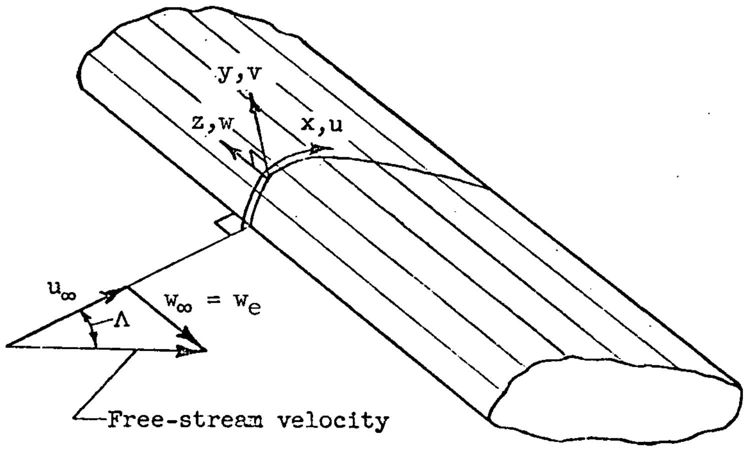


Figure 1.- Coordinate system for swept-leading-edge problem.

PROBLEM DISCUSSION

In this section, the partial differential equations and their finite-difference formulations are described. Also, details of the eddy-viscosity model are given. The basic physics of the problem is discussed, for example, in references 12 and 13.

Basic Partial Differential Equations

The partial differential equations in terms of physical coordinates and mean dimensional flow properties for boundary-layer flow are as follows (see refs. 12 and 13):

Continuity

$$\frac{\partial}{\partial x}\left(\rho u r^j\right) + \frac{\partial}{\partial y}\left(\rho v r^j\right) = 0 \tag{1}$$

where $j = 1$ for axisymmetric bodies and $j = 0$ for two-dimensional bodies and infinite swept cylinders.

x-momentum

$$\rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} = -\frac{dp}{dx} + \frac{\partial}{\partial y}\left[\mu\left(1 + \frac{\epsilon}{\mu}\right)\frac{\partial u}{\partial y}\right] \tag{2}$$

where it is assumed that $p = p(x)$.

In order to compute the flow on an infinite swept cylinder $\left(\text{i.e., } \frac{\partial}{\partial z}(\ ) = 0\right)$, it is necessary to incorporate the spanwise momentum equation. The spanwise coordinate is taken as z and the spanwise velocity as w; hence,

z-momentum

$$\rho u \frac{\partial w}{\partial x} + \rho v \frac{\partial w}{\partial y} = \frac{\partial}{\partial y}\left[\mu\left(1 + \frac{\epsilon_z}{\mu}\right)\frac{\partial w}{\partial y}\right] \tag{3}$$

Note that $w = 0$ for all cases except when a swept-cylinder problem is specified.

Total energy

$$\rho u \frac{\partial H}{\partial x} + \rho v \frac{\partial H}{\partial y} = \frac{\partial}{\partial y}\left\{\frac{\mu}{N_{Pr}}\left[\left(1 + \frac{\epsilon}{\mu}\frac{N_{Pr}}{N_{Pr,T}}\right)\frac{\partial H}{\partial y} - \left(1 - N_{Pr}\right)\left(u\frac{\partial u}{\partial y} + w\frac{\partial w}{\partial y}\right)\right]\right\} \tag{4}$$

The eddy viscosity ϵ and turbulent Prandtl number $N_{Pr,T}$ are supplied as simple functions of the local flow properties (within the boundary layer and/or in the free stream) and the distance from the wall. All flow quantities shown are time mean values. For an axisymmetric body, r is the dimensionless local radius of the body $\left(r = r_w/L\right)$, and the assumption is made that the boundary-layer thickness δ is much less than r_w.

Solutions to equations (1) to (4) are sought in terms of u, v, w, and H. Auxiliary equations for the other variables are shown subsequently. Besides initial conditions $\left(\text{at } x = x_o\right)$, the boundary conditions are:

$y = 0$ (wall or surface of body)

$$\left.\begin{aligned} u &= w = 0 \\ v &= v_w(x) \\ H &= H_w(x) \end{aligned}\right\} \tag{5a}$$

(Note that the adiabatic-wall boundary condition is not included in the program described herein and only air-to-air blowing can be considered.)

$$\left.\begin{aligned} u &\rightarrow u_e(x) \\ w &\rightarrow w_e \quad \text{(a constant)} \\ H &\rightarrow H_e \quad \text{(a constant)} \end{aligned}\right\} \tag{5b}$$

For convenience, dimensionless profile functions are introduced:

$$\left.\begin{aligned} F &= \frac{u}{u_e} \\ g &= \frac{w}{w_e} \\ \theta &= \frac{H - H_w}{H_e - H_w} \equiv \frac{\zeta - \zeta_w}{1 - \zeta_w} \end{aligned}\right\} \tag{6}$$

where $u_e = u_e(x)$, $\zeta_w = \zeta_w(x)$, and w_e and H_e are constants. Because of conditions (5a) and (5b), the boundary conditions for these new variables are

<u>$y = 0$</u>

$$F_w = g_w = \theta_w = 0 \tag{7a}$$

<u>$y \to \infty$</u>

$$F_e = g_e = \theta_e = 1 \tag{7b}$$

Equations (1) to (4) may then be written as

<u>Continuity</u>

$$\frac{\partial}{\partial x}\left(\rho u_e F r^j\right) + \frac{\partial}{\partial y}\left(\rho v r^j\right) = 0 \tag{1a}$$

<u>F-momentum</u>

$$\rho u \frac{\partial F}{\partial x} + \rho v \frac{\partial F}{\partial y} = -\rho F^2 \frac{du_e}{dx} - \frac{1}{u_e}\frac{dp}{dx} + \frac{\partial}{\partial y}\left[\mu\left(1 + \frac{\epsilon}{\mu}\right)\frac{\partial F}{\partial y}\right] \tag{2a}$$

<u>g-momentum</u>

$$\rho u \frac{\partial g}{\partial x} + \rho v \frac{\partial g}{\partial y} = \frac{\partial}{\partial y}\left[\mu\left(1 + \frac{\epsilon_z}{\mu}\right)\frac{\partial g}{\partial y}\right] \tag{3a}$$

<u>Total energy</u>

$$\rho u \frac{\partial \theta}{\partial x} + \rho v \frac{\partial \theta}{\partial y} = -\rho u \frac{1 - \theta}{1 - \zeta_w}\frac{d\zeta_w}{dx} + \frac{\partial}{\partial y}\left\{\frac{\mu}{N_{Pr}}\left[\left(1 + \frac{\epsilon}{\mu}\frac{N_{Pr}}{N_{Pr,T}}\right)\frac{\partial \theta}{\partial y}\right.\right.$$

$$\left.\left. - \frac{1 - N_{Pr}}{1 - \zeta_w}\left(\frac{u_e^2}{2H_e}\frac{\partial F^2}{\partial y} + \frac{w_e^2}{2H_e}\frac{\partial g^2}{\partial y}\right)\right]\right\} \tag{4a}$$

Transformation to Conventional Similarity Coordinates

A finite-difference computing procedure will be used to solve equations (1a) to (4a). In order to keep the number of steps across the boundary layer (i.e., in the y-direction) approximately constant and also to take advantage of similar profile shapes which exist for certain conditions, a similarity transformation of x and y is introduced as follows:

$$\xi\left(\frac{x}{L}\right) = R_s \int_0^{x/L} \frac{(\rho\mu)_e}{(\rho\mu)_s} \frac{u_e}{\sqrt{2H_e}} r^{2j} \, d\frac{x}{L} \tag{8a}$$

$$\eta\left(\frac{x}{L}, \frac{y}{L}\right) = R_s \frac{u_e/\sqrt{2H_e}}{(2\xi)^{\bar{n}}} r^j \int_0^{y/L} \frac{\rho}{\rho_s} \, d\frac{y}{L} \tag{8b}$$

The quantity $\bar{n}$ is taken as a constant, generally equal to about 0.8 for turbulent flows and 0.5 for laminar flows.

Since $\partial\xi/\partial y = 0$, the general transformation formulas from equations (8a) and (8b) are

$$\frac{\partial}{\partial x} = \frac{(\rho\mu)_e u_e}{\mu_s^2} r^{2j} \frac{\partial}{\partial\xi} + \frac{\partial\eta}{\partial x}\frac{\partial}{\partial\eta} \tag{9a}$$

$$\frac{\partial}{\partial y} = \frac{\rho u_e}{\mu_s} \frac{r^j}{(2\xi)^{\bar{n}}} \frac{\partial}{\partial\eta} \tag{9b}$$

For the sake of brevity, several special definitions are introduced:

$$\varphi_r = \frac{\rho\mu}{(\rho\mu)_e} \tag{10a}$$

$$\overline{M} = \varphi_r\left(1 + \frac{\epsilon}{\mu}\right) \tag{10b}$$

$$\hat{M} = \varphi_r\left(1 + \frac{\epsilon_z}{\mu}\right) \tag{10c}$$

$$M^* = \frac{\varphi_r}{N_{Pr}}\left(1 + \frac{\epsilon}{\mu} \frac{N_{Pr}}{N_{Pr,T}}\right) \tag{10d}$$

$$M' = \frac{\varphi_r}{N_{Pr}} \frac{1 - N_{Pr}}{1 - \zeta_w} \tag{10e}$$

$$\bar{\xi} = (2\xi)^{2\bar{n}} \tag{10f}$$

$$U = \frac{1}{u_e/\sqrt{2H_e}} \frac{d\left(u_e/\sqrt{2H_e}\right)}{d\xi} \tag{10g}$$

$$P = \frac{1}{\rho_e u_e^2} \frac{dp}{d\xi} \tag{10h}$$

$$Z = \frac{1}{1 - \zeta_w} \frac{d\zeta_w}{d\xi} \tag{10i}$$

In the present computer program $P = -U$, which requires constant entropy flow external to the boundary layer. When the entropy is not constant, $P \neq -U$ and the external flow conditions would be determined from an iteration to balance the mass flow in the boundary layer with that of the upstream inviscid flow. (See ref. 14.)

The transformed normal velocity is definéd as

$$V = \frac{\mu_s 2\bar{\xi}}{(\rho\mu)_e u_e r^{2j}} \left[F \frac{\partial \eta}{\partial x} + \frac{\rho v r^j}{\mu_s (2\xi)^{\bar{n}}}\right] \tag{11}$$

Application of equations (9a) and (9b) to equations (1a) to (4a) then results in (see ref. 9)

<u>Continuity</u>

$$\bar{\xi} \frac{\partial F}{\partial \xi} + \frac{\partial V}{\partial \eta} + \bar{\xi} F \frac{\bar{n}}{\xi} = 0 \tag{12}$$

<u>F-momentum</u>

$$\bar{\xi} F \frac{\partial F}{\partial \xi} + V \frac{\partial F}{\partial \eta} = -\bar{\xi}\left(UF^2 + P \frac{\rho_e}{\rho}\right) + \frac{\partial}{\partial \eta}\left(\overline{M} \frac{\partial F}{\partial \eta}\right) \tag{13}$$

<u>g-momentum</u>

$$\bar{\xi} F \frac{\partial g}{\partial \xi} + V \frac{\partial g}{\partial \eta} = \frac{\partial}{\partial \eta}\left(\hat{M} \frac{\partial g}{\partial \eta}\right) \tag{14}$$

<u>Total energy</u>

$$\bar{\xi} F \frac{\partial \theta}{\partial \xi} + V \frac{\partial \theta}{\partial \eta} = -\bar{\xi} Z F(1 - \theta) + \frac{\partial}{\partial \eta}\left[M^* \frac{\partial \theta}{\partial \eta} - M'\left(\frac{u_e^2}{2H_e} \frac{\partial F^2}{\partial \eta} + \frac{w_e^2}{2H_e} \frac{\partial g^2}{\partial \eta}\right)\right] \tag{15}$$

The boundary conditions on the profile functions (eqs. (7a) and (7b)) are restated as follows:

$\underline{\eta = 0}$

$$\left.\begin{aligned} & F_w = g_w = \theta_w = 0 \\ & v_w = \begin{cases} 0 \quad \text{or} \\ v_w(x), \text{ specified function} \end{cases} \end{aligned}\right\} \tag{16a}$$

$\underline{\eta = \eta_e}$

$$F_e = g_e = \theta_e = 1.0 \pm \text{Error criteria} \tag{16b}$$

To make the system given by equations (12) to (15) determinate, several auxiliary functions are required. The following are given functions of x:

$$\left.\begin{aligned} & \frac{u_e}{\sqrt{2H_e}} = \frac{u_e}{\sqrt{2H_e}}(x) \\ & \frac{\rho_e}{\rho_s} = \frac{\rho_e}{\rho_s}(x) \\ & r = r(x) \\ & \zeta_w = \zeta_w(x) \\ & \frac{p_e}{p_s} = \frac{p_e}{p_s}(x) \end{aligned}\right\} \tag{17a}$$

To avoid problems in numerical differentiation for sparse input data, the following derivatives can also be specified:

$$\left.\begin{aligned} & \frac{d\left(u_e/\sqrt{2H_e}\right)}{d\left(\frac{x}{L}\right)} = \frac{d\left(u_e/\sqrt{2H_e}\right)}{d\left(\frac{x}{L}\right)}(x) \\ & \frac{d\zeta_w}{d\left(\frac{x}{L}\right)} = \frac{d\zeta_w}{d\left(\frac{x}{L}\right)}(x) \end{aligned}\right\} \tag{17b}$$

For a perfect gas with constant specific heats, the static enthalpy ratio, computed from profile functions, is

$$\frac{h}{h_e} \equiv \frac{\rho_e}{\rho} = \frac{\left(1 - \zeta_w\right)\theta + \zeta_w - \frac{u_e^2}{2H_e} F^2 - \frac{w_e^2}{2H_e} g^2}{1 - \frac{u_e^2}{2H_e} - \frac{w_e^2}{2H_e}} \tag{18}$$

Sutherland's viscosity relation is used in the density-viscosity product ratio (eq. (10a)) to give

$$\varphi_r = \left(\frac{h/H_e}{h_e/H_e}\right)^{1/2} \frac{\frac{h_e}{H_e} + \frac{S}{H_e}}{\frac{h}{H_e} + \frac{S}{H_e}} \tag{19}$$

which can be utilized for various diatomic gases. The value of Sutherland's constant S would depend on the gas and the temperature range of the problem. The enthalpy ratios in equation (19) are

$$\frac{h}{H_e} = \left(1 - \zeta_w\right)\theta + \zeta_w - \frac{u_e^2}{2H_e} F^2 - \frac{w_e^2}{2H_e} g^2 \tag{20}$$

and

$$\frac{h_e}{H_e} = 1 - \frac{u_e^2}{2H_e} - \frac{w_e^2}{2H_e} \tag{21}$$

where

N_{Pr} = Constant

$N_{Pr,T}$ either a constant or a specified function of y/δ

$\frac{\epsilon}{\mu}$ function of profile parameters and y/δ

δ value of y where F or $g = 0.995$ (note that this is not the asymptotic boundary condition on the computed profiles)

$\xi = \xi\left(\frac{x}{L}\right)$ (see eq. (8a))

Numerical Computational Procedure

The system of equations (12) to (15) is solved by an implicit finite-difference procedure. The grid to be used and the node notation is illustrated in figure 2. To increase the accuracy and efficiency, a variable grid size in the η- and ξ-directions may be used. The initial distribution of $\Delta\eta$ will generally be held fixed for a given problem.

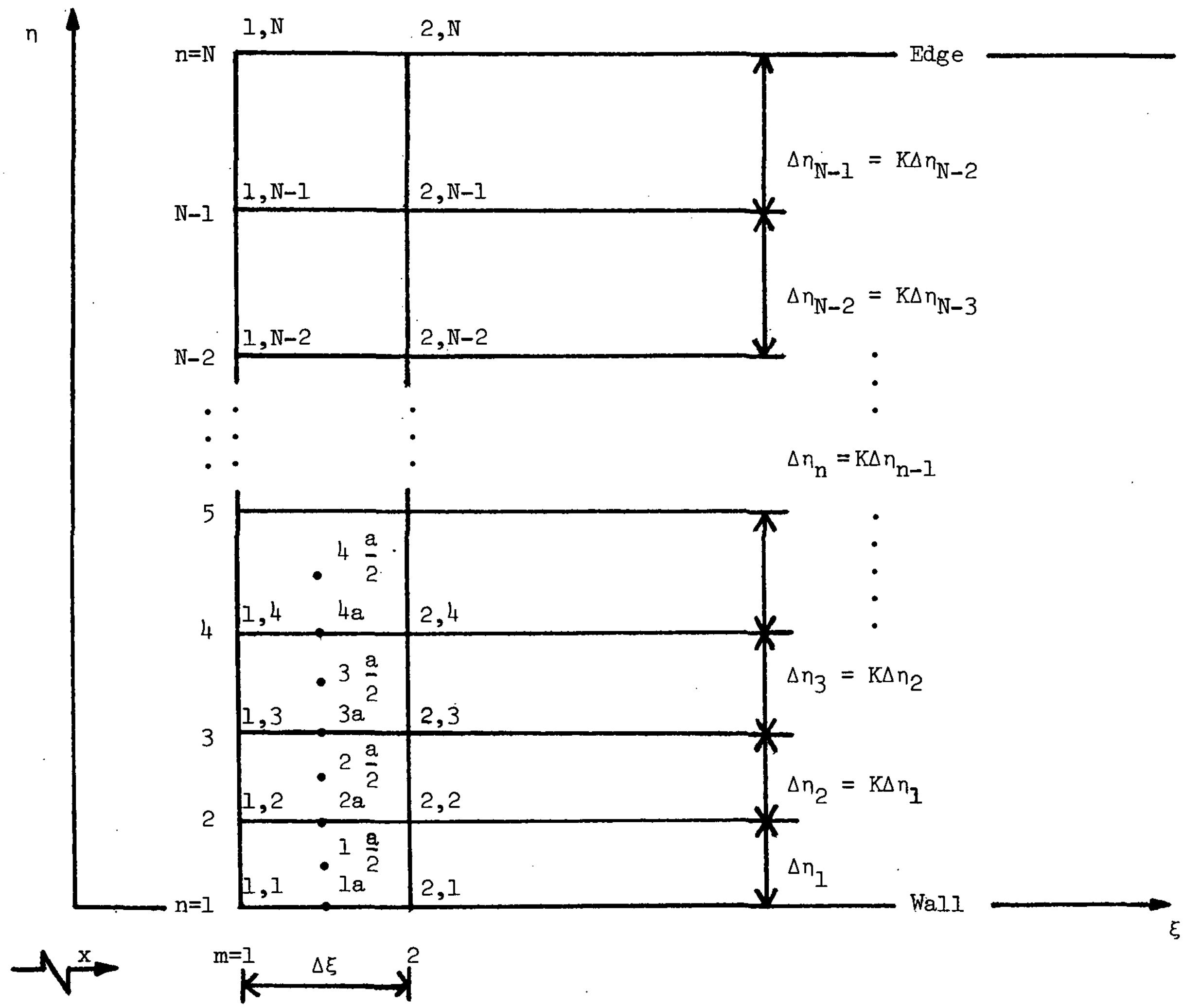


Figure 2.- Grid notation.

As illustrated in figure 2, the letter "a" indicates the average value of a quantity between the two x-locations under consideration and the notation "a/2" indicates the average value of a quantity at the four adjacent node points.

In the present procedure, the solution is advanced downstream in the x- or ξ-direction by redefining the $m = 2$ values as $m = 1$ values and a new $m = 2$ station is chosen a distance $\Delta\xi$ downstream; that is, a two-point difference scheme in the x- or ξ-direction is used.

It can be seen that if K is a constant, the successive values of $\Delta\eta$ form a geometric progression. Hence

$$\Delta\eta_n = K^{n-1}\Delta\eta_1 \tag{22}$$

Since the total number of $\Delta\eta$ steps across the boundary layer is N - 1, the size of the last step at n = N - 1 is

$$\Delta\eta_{N-1} = K^{N-2}\Delta\eta_1 \tag{23}$$

The thickness of the boundary layer η_e is given by

$$\eta_e = \Delta\eta_1 \frac{1 - K^{N-1}}{1 - K} \tag{24}$$

Thus, if η_e, $\Delta\eta_1$, and the number of steps (N - 1) are specified, K and $\Delta\eta_{N-1}$ can be determined. Note that if $\Delta\eta$ = Constant is desired, K = 1.0. Generally, the value of K will be a constant slightly greater than 1.0. (The value K = 1.02 is usually used.)

The input is specified at station m = 1 from n = 1 to n = N from which values of all variables are to be computed at the next station (m = 2). Equation (12) with $\frac{\partial F}{\partial \xi} = 0$ is used to obtain initial values of V. The various derivatives in equations (12) to (15) are replaced by linear difference quotients (see refs. 9 and 10) and the equations are evaluated at the intermediate station. Consider first the F-momentum equation (eq. (13)). At point 3a, the difference equation becomes

$$\bar{\xi}_a F_{3a} \frac{F_{2,3} - F_{1,3}}{\Delta\xi} + \frac{V_{3,a}}{2} \frac{F_{1,4} + F_{2,4} - F_{1,2} - F_{2,2}}{\Delta\eta_2 + \Delta\eta_3}$$

$$= -\bar{\xi}_a\left[U_a F_{3a}{}^2 + P_a\left(\frac{\rho_e}{\rho}\right)_{3a}\right] + \frac{\bar{M}_{\frac{3a}{2}} \dfrac{F_{2,4} + F_{1,4} - F_{1,3} - F_{2,3}}{2\Delta\eta_3} - \bar{M}_{\frac{2a}{2}} \dfrac{F_{2,3} + F_{1,3} - F_{2,2} - F_{1,2}}{2\Delta\eta_2}}{\frac{1}{2}\left(\Delta\eta_3 + \Delta\eta_2\right)}$$

The coefficients of the unknown quantities $F_{2,4}$, $F_{2,3}$, and $F_{2,2}$ are collected and equation (13) evaluated at point 3a may be written as

$$\bar{A}_3 F_{2,4} + \bar{B}_3 F_{2,3} + \bar{C}_3 F_{2,2} = \bar{D}_3$$

The general form of this equation is then

$$\bar{A}_n F_{2,n+1} + \bar{B}_n F_{2,n} + \bar{C}_n F_{2,n-1} = \bar{D}_n \tag{25}$$

where

$$\bar{A}_n = \frac{V_{na}}{2\left(\Delta\eta_n + \Delta\eta_{n-1}\right)} \frac{\bar{M}_{n\frac{a}{2}}\left(\frac{2}{1 + K}\right)}{2\Delta\eta_n \, \Delta\eta_{n-1}} \tag{26}$$

$$\bar{B}_n = \frac{\bar{\xi}_a}{\Delta\xi} F_{na} + \frac{\bar{M}_{n\frac{a}{2}}\left(\frac{2}{1 + K}\right) + \bar{M}_{(n-1)\frac{a}{2}}\left(\frac{2K}{1 + K}\right)}{2\Delta\eta_n \, \Delta\eta_{n-1}} \tag{27}$$

$$\bar{C}_n = -\left[\frac{V_{na}}{2\left(\Delta\eta_n + \Delta\eta_{n-1}\right)} + \frac{\bar{M}_{n\frac{a}{2}}\left(\frac{2K}{1 + K}\right)}{2\Delta\eta_n \, \Delta\eta_{n-1}}\right] \tag{28}$$

$$\bar{D}_n = -\bar{A}_n F_{1,n+1} + \left[\frac{\bar{\xi}_a}{\Delta\xi} F_{na} - \frac{\bar{M}_{n\frac{a}{2}}\left(\frac{2}{1 + K}\right) + \bar{M}_{(n-1)\frac{a}{2}}\left(\frac{2K}{1 + K}\right)}{2\Delta\eta_n \, \Delta\eta_{n-1}}\right] F_{1,n}$$

$$- \bar{C}_n F_{1,n-1} - (\bar{\xi}U)_a F_{na}^{\;2} - (\bar{\xi}P)_a \left(\frac{\rho_e}{\rho}\right)_{na} \tag{29}$$

In equations (26) to (29), any function that depends only on ξ is denoted by subscript a, which indicates the average value of the function between stations 1 and 2. Conditions at stations 1 and 2 are updated as the solution proceeds in the x-direction by a general iteration procedure. For the first iteration in this procedure, the "a" or average values of quantities in the $\bar{A}_n$, $\bar{B}_n$, $\bar{C}_n$, and $\bar{D}_n$ functions are evaluated at the $m = 1$ station. On successive iterations, the "a" values are updated using the latest values from the $m = 2$ station; that is, the "a" values are taken as simple numerical averages of the appropriate node points. Examples of these are as follows:

$$\left.\begin{aligned} F_{na} &= \frac{1}{2}\left(F_{1,n} + F_{2,n}\right) \\ F_{n\frac{a}{2}} &= \frac{1}{2}\left[F_{(n+1)a} + F_{na}\right] \end{aligned}\right\} \tag{30a}$$

Hence,

$$F_{n\frac{a}{2}} = \frac{1}{4}\left(F_{1,n+1} + F_{2,n+1} + F_{1,n} + F_{2,n}\right) \tag{30b}$$

Since boundary conditions are specified at $n = 1$ and $n = N$, equation (25) represents a system of $N - 2$ equations with $N - 2$ unknowns (the values of $F_{2,n}$ from $n = 2$ to $n = N - 1$). Since the matrix of the system given by equation (25) is tridiagonal, the unknown F_2 values are easily obtained by successive elimination of unknowns with the formula (see ref. 15)

$$F_{2,n} = \bar{G}_n F_{2,n+1} + \bar{g}_n \tag{31}$$

which is applied by starting at the outer boundary ($n = N$) where $F_e \approx 1.0$ and proceeding down to the wall ($n = 1$) where $F = 0$. The functions $\bar{G}_n$ and $\bar{g}_n$ are computed from the following recursion formulas:

$$\left.\begin{aligned} \bar{G}_n &= \frac{-\bar{A}_n}{\bar{B}_n + \bar{C}_n \bar{G}_{n-1}} \\ \bar{g}_n &= \frac{\bar{D}_n - \bar{C}_n \bar{g}_{n-1}}{\bar{B}_n + \bar{C}_n \bar{G}_{n-1}} \end{aligned}\right\} \tag{32}$$

From the wall boundary condition $\left(F_{2,1} = 0\right)$ applied to equation (31), $\bar{G}_1 = \bar{g}_1 = 0$. Therefore, from the recursion relations (32),

$$\left.\begin{aligned} \bar{G}_2 &= \frac{-\bar{A}_2}{\bar{B}_2} \\ \bar{g}_2 &= \frac{\bar{D}_2}{\bar{B}_2} \end{aligned}\right\} \tag{33}$$

The functions $\bar{G}_n$ and $\bar{g}_n$ are computed from equations (32) by starting at the wall (actually at $n = 2$) and working out to the outer boundary ($n = N - 1$). Equation (31) then supplies the required values of $F_{2,n}$ by starting at the outer edge ($n = N - 1$) and proceeding down to the wall ($n = 2$). Before this procedure can be completed, it is necessary to know the number of equations (or $\Delta\eta$ steps) to be used. The correct number of steps is determined from the physical requirement that

$$\left(\frac{\partial F}{\partial \eta}\right)_{\eta=\eta_e} \leqq \sigma_e \tag{34}$$

where σ_e is some small specified error criterion. Equation (34) is written in finite-difference form as

$$F_{2,N} - F_{2,N-1} \leqq \Delta\eta_{N-1}\sigma_e$$

From equation (31),

$$F_{N-1} = \bar{G}_{N-1}F_N + \bar{g}_{N-1}$$

Then, since $F_N = F_e \approx 1.0$, the values of $\bar{G}_{N-1}$ and $\bar{g}_{N-1}$ must satisfy the inequality

$$\left|F_e\left(1 - \bar{G}_{N-1}\right) - \bar{g}_{N-1}\right| \leqq \left|\Delta\eta_{N-1}\sigma_e\right| \tag{35}$$

where F_e and σ_e are specified and $\Delta\eta_{N-1}$ is computed from equation (23). Hence the value of $n = N - 1$ is obtained by inserting successive values of $\bar{G}_n$ and $\bar{g}_n$ from equations (32) (as the suspected neighborhood of $n \approx N - 10$, say, is approached) into inequality (35).

After the set of $F_{2,n}$ values is obtained, the values of V_{na} are updated by using equation (12), which is written in the appropriate finite-difference form evaluated at $n\frac{a}{2}$ (average of four adjacent node points). For example, at $2\frac{a}{2}$ there is obtained

$$V_{3a} = V_{2a} - \frac{\Delta\eta_2\bar{\xi}_a}{2\Delta\xi}\left(F_{2,3} + F_{2,2} - F_{1,3} - F_{1,2}\right) - \Delta\eta_2\bar{\xi}_a\frac{\bar{n}}{\xi_a}F_{2\frac{a}{2}} \tag{36}$$

As noted previously, input values of V_{na} for the first iteration at the input ξ station are computed from equation (12) by dropping derivatives with respect to ξ and evaluating at $m = 1$. The result is

$$V_{3a} = V_{2a} - \frac{1}{2}\Delta\eta_2\bar{\xi}_1\left(F_{1,3} + F_{1,2}\right)\frac{\bar{n}}{\xi_1} \tag{37}$$

General expressions for equations (36) and (37) follow from inspection. The values of V_{na} are computed by starting at the known boundary condition V_{1a} (that is, $V_{w,a}$) and applying the general form of equation (36) (or eq. (37) for the first iteration at the input station) successively out to $n = N$.

With these improved values of V_{na} from equation (36), the first approximations of $g_{2,n}$ and $\theta_{2,n}$ are computed from equations (14) and (15). These equations are written in finite-difference form by a procedure identical to that illustrated for the F-momentum equation. The results for equations (14) and (15) are

$$\hat{A}_n g_{2,n+1} + \hat{B}_n g_{2,n} + \hat{C}_n g_{2,n-1} = \hat{D}_n \tag{38}$$

$$A_n^* \theta_{2,n+1} + B_n^* \theta_{2,n} + C_n^* \theta_{2,n-1} = D_n^* \tag{39}$$

The expressions for $\hat{A}_n, \hat{B}_n, \hat{C}_n$ and A_n^*, B_n^*, C_n^* are identical to those for equation (13), as given by equations (26), (27), and (28), except that the $\overline{M}$ values are replaced by $\hat{M}$ and M^*, respectively. The "D" quantities are computed from the following equations:

$$\hat{D}_n = -\hat{A}_n g_{1,n+1} + \left[\frac{\xi_a}{\Delta\xi} F_{na} - \frac{\hat{M}_{n\frac{a}{2}}\left(\frac{2}{1+K}\right) + \hat{M}_{(n-1)\frac{a}{2}}\left(\frac{2K}{1+K}\right)}{2\Delta\eta_n \, \Delta\eta_{n-1}}\right]_{g_{1,n}} - \hat{C}_n g_{1,n-1} \tag{40}$$

$$\begin{aligned}
D_n^* = {} & -A_n^* \theta_{1,n+1} + \left[\frac{\bar{\xi}_a}{\Delta\xi} F_{na} - \frac{M^*_{n\frac{a}{2}}\left(\frac{2}{1+K}\right) + M^*_{(n-1)\frac{a}{2}}\left(\frac{2K}{1+K}\right)}{2\Delta\eta_n \, \Delta\eta_{n-1}}\right]_{\theta_{1,n}} - C_n^* \theta_{1,n-1} \\
& + (\bar{\xi}Z)_a F_{na}\left(\theta_{na} - 1\right) - \frac{\left(u_e^2\right)_a}{2H_e \, 2\Delta\eta_n \, \Delta\eta_{n-1}}\left[M'_{n\frac{a}{2}}\left(\frac{2}{1+K}\right)\left(F^2_{1,n+1} + F^2_{2,n+1} - F^2_{1,n} - F^2_{2,n}\right)\right. \\
& \left. - M'_{(n-1)\frac{a}{2}}\left(\frac{2K}{1+K}\right)\left(F^2_{1,n} + F^2_{2,n} - F^2_{1,n-1} - F^2_{2,n-1}\right)\right] \\
& - \frac{w_e^2}{2H_e \, 2\Delta\eta_n \, \Delta\eta_{n-1}}\left[M'_{n\frac{a}{2}}\left(\frac{2}{1+K}\right)\left(g^2_{1,n+1} + g^2_{2,n+1} - g^2_{1,n} - g^2_{2,n}\right)\right. \\
& \left. - M'_{(n-1)\frac{a}{2}}\left(\frac{2K}{1+K}\right)\left(g^2_{1,n} + g^2_{2,n} - g^2_{1,n-1} - g^2_{2,n-1}\right)\right]
\end{aligned} \tag{41}$$

The formulas for g_2 and θ_2 are of the same form as equation (31) (since the wall boundary conditions for g and θ are the same as those for F) and are written as

$$g_{2,n} = \hat{G}_n g_{2,n+1} + \hat{g}_n \tag{42}$$

$$\theta_{2,n} = G_n^* \theta_{2,n+1} + g_n^* \tag{43}$$

where the recursion formulas for the $\overline{G}$ and $\tilde{g}$ functions are of the same form as equation (32) with the appropriate superscript notation supplied. The number of equations in the systems given by equations (42) and (43) is again determined from

$$\left|g_e\left(1 - \hat{G}_{N-1}\right) - \hat{g}_{N-1}\right| \leqq \left|\Delta\eta_{N-1}\sigma_e\right| \tag{44}$$

$$\left|\theta_e\left(1 - G^*_{N-1}\right) - g^*_{N-1}\right| \leqq \left|\Delta\eta_{N-1}\sigma_e\right| \tag{45}$$

where, as before, g_e, σ_e, and θ_e are specified. An iteration procedure is thus used where equations (25), (36), (38), and (39) are solved, in that order. Equation (37) is used only on the first iteration. When the following convergence criteria are satisfied, the iterations are stopped and the entire procedure is repeated at the next ξ step. The convergence criteria are written as

$$\frac{F_{2,2,i+1} - F_{2,2,i}}{F_{2,2,i}} \leqq \sigma_w \tag{46}$$

$$\frac{g_{2,2,i+1} - g_{2,2,i}}{g_{2,2,i}} \leqq \sigma_w \tag{47}$$

$$\frac{\theta_{2,2,i+1} - \theta_{2,2,i}}{\theta_{2,2,i}} \leqq \sigma_w \tag{48}$$

where the index i denotes the number of the iteration cycle at a given ξ station.

Eddy Viscosity and Turbulent Prandtl Number Formulations

In this section, the models used for the turbulence correlation terms appearing in the conservation equations for the mean flow are described. The eddy-viscosity formulations are discussed first and then the turbulent Prandtl number expressions are discussed. For a more detailed discussion of these models, see reference 7.

<u>Mixing-length relations and eddy-viscosity function for two-dimensional and axisymmetric flows.</u>- The turbulent shear for two-dimensional flow is given by

$$\tau_t = \epsilon \frac{\partial u}{\partial y} \tag{49}$$

A mixing length l may be defined by the relation (see refs. 2 and 11)

$$\tau_t = \rho l^2 \left|\frac{\partial u}{\partial y}\right| \frac{\partial u}{\partial y} \tag{50}$$

From the calculations in reference 11 (based on experimental data), it was shown that l/δ tends to be a nearly universal function of y/δ and, in the outer portion of the boundary layer, l/δ is approximately constant and equal to a typical incompressible value of 0.09 (at least for adiabatic flows up to $M_e = 5.0$). Hence, it is assumed herein that

$$\frac{l}{\delta} = \bar{f}\left(\frac{y}{\delta}\right) \tag{51}$$

where

$$\frac{y}{\delta} = \frac{\int_0^{\eta} \frac{\rho_e}{\rho} \, d\eta}{\int_0^{\eta_\delta} \frac{\rho_e}{\rho} \, d\eta} \tag{52}$$

The eddy viscosity is then given by the equation

$$\epsilon = \rho \delta^2 \left(\frac{l}{\delta}\right)^2 \left| \frac{\partial u}{\partial y} \right| \tag{53}$$

Transformation to the η coordinate then gives

$$\frac{\epsilon}{\mu} = (2\xi)^{\bar{n}} \left(\frac{l}{\delta}\right)^2 \left| \frac{\partial F}{\partial \eta} \right| \frac{\mu_s}{\mu} \left(\frac{\rho}{\rho_e}\right)^2 r^{-j} \left(\int_0^{\eta_\delta} \frac{\rho_e}{\rho} \, d\eta\right)^2 \tag{54}$$

where η_δ is the value of η when $F = 0.995$. The appropriate finite-difference form for equation (54) is illustrated by evaluating the equation at point (1,n) as follows:

$$\left(\frac{\epsilon}{\mu}\right)_{1,n} = \left[\frac{(2\xi)^{\bar{n}}}{r^j}\right]_1 \left[\left(\frac{l}{\delta}\right)^2 \frac{\mu_s}{\mu} \left(\frac{\rho}{\rho_e}\right)^2 \left(\int_0^{\eta_\delta} \frac{\rho_e}{\rho} \, d\eta\right)^2\right]_{1,n} \left| \frac{F_{1,n+1} - F_{1,n-1}}{\Delta\eta_{n-1} + \Delta\eta_n} \right| \tag{55}$$

Note that the quantity ϵ/μ enters the viscosity functions, which are always evaluated at either $n\frac{a}{2}$ or $(n-1)\frac{a}{2}$ as illustrated for $\overline{M}$ by equations (26) to (29). These values of $\overline{M}_{n\frac{a}{2}}$ or $\overline{M}_{(n-1)\frac{a}{2}}$ are calculated from the arithmetical average of the four surrounding mesh points as illustrated by equation (30b). The $\bar{f}$ function is either a tabulated function of y/δ or it may be computed from the following expressions:

$$\left.\begin{aligned}
\bar{f} &= 0.4 \frac{y}{\delta} && \left(\frac{y}{\delta} \leqq 0.1\right) \\
\bar{f} &= 0.04 + \frac{\frac{y}{\delta} - 0.1}{0.2}\left(\bar{f}_{max} - 0.04\right) && \left(0.1 \leqq \frac{y}{\delta} \leqq 0.3\right) \\
\bar{f} &= \bar{f}_{max} && \left(\frac{y}{\delta} \geqq 0.3\right)
\end{aligned}\right\} \tag{56}$$

where

$$\bar{f}_{max} = A + BH_i^* + CH_i^{*2} \tag{56a}$$

and

$$H_i^* \equiv \frac{\int_0^{\delta} (1 - F)dy}{\int_0^{\delta} F(1 - F)dy} \tag{57}$$

Recommended values of A, B, and C are given in reference 7.

In the wall region, the basic mixing-length function (eqs. (56)) is modified to account for the turbulent damping effect of the wall by multiplying $\bar{f}$ by the Van Driest exponential damping function (see ref. 7). The final mixing length then becomes

$$\frac{l}{\delta} = \left[1 - \exp\left(- \frac{y/L}{A_d/L}\right)\right]\bar{f} \tag{58}$$

where, if A_d is based on wall properties,

$$A_d = \frac{A_b \mu_w}{\rho_w \sqrt{\tau_w/\rho_w}}$$

or

$$\left(\frac{A_d}{L}\right)_m = \frac{A_b(2\xi)^{\bar{n}/2}}{R_s\left(\frac{\mu_s}{\mu_w}\right)^{1/2} \frac{\rho_w}{\rho_e} \frac{\rho_e}{\rho_s} \frac{u_e}{\sqrt{2H_e}} r^{j/2}\left(\frac{F_{m,2}}{\Delta\eta_1}\right)^{1/2}} \tag{59}$$

For $v_w = 0$, $A_b = 26$; and for $v_w \neq 0$ (blowing or suction), the A_b value is obtained from an input table of A_b values as a function of $\frac{\rho_w v_w}{\rho_e u_e} \frac{2}{C_f}$. (See ref. 7.) Computations can also be carried out by using the program with $A_d = \frac{A_b \mu}{\rho\sqrt{\tau_w/\rho}}$; that is, A_d can be based on wall shear and local properties in the boundary layer.

Eddy-viscosity function for flows on infinite swept cylinders.- An eddy-viscosity formulation suitable for flows on infinite swept cylinders was derived in reference 8. This formulation is similar to that for two-dimensional and axisymmetric flows except that $\left|\frac{\partial u}{\partial y}\right|$ is replaced by $\sqrt{\left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial y}\right)^2}$ in equation (53). The eddy-viscosity relation evaluated at the point $1,n$ then becomes

$$\left(\frac{\epsilon}{\mu}\right)_{1,n} = \left[\frac{(2\xi)^{\bar{n}}}{r^j}\right]_1 \left[\left(\frac{l}{\delta}\right)^2 \frac{\mu_s}{\mu}\left(\frac{\rho}{\rho_e}\right)^2 \left(\int_0^{\eta_\delta} \frac{\rho_e}{\rho}\, d\eta\right)^2\right]_{1,n} \left[\left(\frac{F_{1,n+1} - F_{1,n-1}}{\Delta\eta_{n-1} + \Delta\eta_n}\right)^2 + \left(\frac{g_{1,n+1} - g_{1,n-1}}{\Delta\eta_{n-1} + \Delta\eta_n}\right)^2 \left(\frac{w_e/\sqrt{2H_e}}{u_e/\sqrt{2H_e}}\right)^2\right]^{1/2} \tag{60}$$

This formulation for ϵ/μ is used in both the chordwise and spanwise momentum equations (eqs. (13) and (14)); that is, $\epsilon_z = \epsilon$. The wall damping function (eq. (58)) is applied to the swept-cylinder problem by assuming that $\sqrt{\tau_w/\rho_w}$ is based on the total shear at the wall. This total or resultant shear is the vector sum of the shear in the chordwise and spanwise directions. Equation (59) then becomes

$$\left(\frac{A_d}{L}\right)_m = \frac{A_b(2\xi)^{\bar{n}/2}}{R_s\left(\frac{\mu_s}{\mu_w}\right)^{1/2} \frac{\rho_w}{\rho_e}\frac{\rho_e}{\rho_s}\frac{u_e}{2H_e} r^{j/2}\left[\left(\frac{F_{m,2}}{\Delta\eta_1}\right)^2 + \left(\frac{g_{m,2}}{\Delta\eta_1}\right)^2\left(\frac{w_e\sqrt{2H_e}}{u_e\sqrt{2H_e}}\right)^2\right]^{1/4}} \tag{61}$$

Also, for $v_w \neq 0$, the A_b value is taken as a function of $\dfrac{\rho_w v_w}{\rho_e\left(u_e^2 + w_e^2\right)^{1/2}} \dfrac{2}{C_{f,res}}$. (See ref. 8.)

Turbulent Prandtl number formulations.- The term involving eddy conductivity which appears in the conservation equation for total energy (eq. (4a)) has been modeled in the program by using two approaches. The first of these involves modeling the term

$$-\overline{(\rho v)'H'} = \kappa \frac{\partial H}{\partial y} \tag{62}$$

which results in the definition of a "total turbulent Prandtl number"

$$N_{Pr,T} \equiv \frac{\overline{(\rho v)'u'}\ \dfrac{\partial \bar{H}}{\partial y}}{\overline{(\rho v)'H'}\ \dfrac{\partial \bar{u}}{\partial y}} = \frac{c_p \epsilon}{\kappa} \tag{63}$$

The mean total enthalpy equation using this approach is as previously given (eq. (4) or 4(a)).

The second approach involves modeling the term

$$-\overline{(\rho v)'h'} = k^* \frac{\partial h}{\partial y}$$

which results in the static turbulent Prandtl number

$$N_{Pr,t} \equiv \frac{\overline{(\rho v)'u'}\ \dfrac{\partial \bar{h}}{\partial y}}{\overline{(\rho v)'h'}\ \dfrac{\partial \bar{u}}{\partial y}} = \frac{c_p \epsilon}{k^*} \tag{64}$$

For this case the last term in the bracket of equation (4a) is modified to read

$$-\frac{1 - N_{Pr}}{1 - \zeta_w}\left(1 + \frac{\epsilon}{\mu}\frac{N_{Pr}}{N_{Pr,t}}\frac{1 - N_{Pr,t}}{1 - N_{Pr}}\right)\left(\frac{u_e^2}{2H_e}\frac{\partial F^2}{\partial y} + \frac{w_e^2}{2H_e}\frac{\partial g^2}{\partial y}\right)$$

The definition of M' (eq. (10e)) then becomes

$$M' = \frac{\varphi_r}{N_{Pr}}\frac{1 - N_{Pr}}{1 - \zeta_w}\left(1 + \frac{\epsilon}{\mu}\frac{N_{Pr}}{N_{Pr,t}}\frac{1 - N_{Pr,t}}{1 - N_{Pr}}\right) \tag{65}$$

For swept-leading-edge flows, the $N_{Pr,t}$ definition for turbulent Prandtl number must be used in order for the present system of equations to be conceptually correct. (See ref. 8.)

PROGRAM DESCRIPTION

The program is written in FORTRAN IV. A FORTRAN variable list is given first. Then, the various portions of the program, the main program and subroutines, each accompanied by flow charts, are given.

The main program controls the finite-difference solution of the boundary-layer equations, reads the input, calls the various subroutines, and writes the output. CALCM is called to compute the viscosity functions. During the iteration procedure, ABCDGS and COMPUTE are called successively for each variable to calculate the coefficients of the solution matrix and to compute new values for the variable, respectively. Library subroutines FTLUP and DISCOT are used for interpolation. A description of these subroutines is included in the appendix.

FORTRAN Variable List

AB	value used from ABTAB
ABTAB	table of A_b values in Van Driest damping function (see eq. (59))
ADL	constant A_d in Van Driest damping function $\left(A_d/L\right)$ (see eq. (58))

AP constant A in expression for $\bar{f}_{max}$ (see eq. (56a))

BCG temporary storage used in computing G and g in ABCDGS

BP constant B in expression for $\bar{f}_{max}$ (see eq. (56a))

CAPA, CAPB, CAPC, CAPD coefficients of solution matrix (see eqs. (25), (38), and (39))

CAPG coefficient in formula for dependent variables (see eq. (31), for example)

CAPMA2 dummy in ABCDGS for $\overline{M}_{n\frac{a}{2}}$, $\hat{M}_{n\frac{a}{2}}$, and $M^*_{n\frac{a}{2}}$

CAPP1 $P_{m=1}$ (see eq. (10h))

CAPP2 $P_{m=2}$ (see eq. (10h))

CAPRS reference Reynolds number, $\rho_s\sqrt{2H_e}\,L/\mu_s$

CAPU1 $\left[\frac{1}{u_e\sqrt{2H_e}}\frac{d\left(u_e/\sqrt{2H_e}\right)}{d(x/L)}\right]_{m=1}$ (see eq. (10g))

CAPU2 $\left[\frac{1}{u_e\sqrt{2H_e}}\frac{d\left(u_e/\sqrt{2H_e}\right)}{d(x/L)}\right]_{m=2}$ (see eq. 10g))

CAPZ1 $\left[\frac{1}{1-\zeta_w}\frac{d\zeta_w}{d(x/L)}\right]_{m=1}$ (see eq. (10i))

CAPZ2 $\left[\frac{1}{1-\zeta_w}\frac{d\zeta_w}{d(x/L)}\right]_{m=2}$ (see eq. (10i))

CFF wall-skin-friction coefficient for chordwise profile

CFG	wall-skin-friction coefficient for spanwise profile
CKVAL	check value used in COMPUTE for left side of equations (35), (44), and (45)
CP	constant C in expression for $\bar{f}_{max}$ (see eq. (56a))
DELDEL2	temporary storage in equations for CAPA, CAPC, and CAPD in ABCDGS
DELETA	$\Delta\eta$
DELXI	$\Delta\xi$
DELXI0	initial $\Delta\xi$ step size
DSDLF	$(\delta^*/L)_F$
DSDLG	$(\delta^*/L)_g$
DUDXTAB	$\dfrac{d\left(u_e/\sqrt{2H_e}\right)}{d(x/L)}$ table
DUEDX1	$\left[\dfrac{d\left(u_e/\sqrt{2H_e}\right)}{d(x/L)}\right]_{m=1}$ (see eqs. (17b))
DUEDX2	$\left[\dfrac{d\left(u_e/\sqrt{2H_e}\right)}{d(x/L)}\right]_{m=2}$ (see eqs. (17b))
DUM2	dummy in COMPUTE for F, g, and θ
DZDXTAB	$\dfrac{d\zeta_w}{d(x/L)}$ table
DZWDX1	$\left[\dfrac{d\zeta_w}{d(x/L)}\right]_{m=1}$ (see eqs. (17b))
DZWDX2	$\left[\dfrac{d\zeta_w}{d(x/L)}\right]_{m=2}$ (see eqs. (17b))

EMUSDMU	μ_s/μ
EPSDMU	ϵ/μ
EPSLONE	accuracy criteria for $\partial(F,G,\theta)/\partial\eta$ at outer edge of profile (σ_e) (see eqs. (35), (44), and (45))
EPSLONW	convergence criteria for iterations on F, g, and θ profiles, allowable percent change in wall slope between iterations (σ_w) (see eqs. (46) to (48))
ETA	η
ETATAB	input η table
FA, FA2	average values of F (see fig. 2)
FBAR	l/δ (see eq. (51))
FBARMAX	maximum value of $\bar{f}$
FBARTAB	$\bar{f}$ values corresponding to YDDFB values
FCFTAB	$\frac{\rho_w v_w}{\rho_e u_e} \frac{2}{C_f}$ values corresponding to ABTAB values
FCT1	2K/(1 + K), used in ABCDGS to account for variable η step size
FCT2	2/(1 + K), used in ABCDGS to account for variable η step size
FDCFD2	$\frac{\rho_w v_w}{\rho_e u_e} \frac{2}{C_f}$
FNCDEL1	intermediate variable used in calculation of H_i^*
FNCDEL2	intermediate variable used in calculation of H_i^*
FNCFRR1	intermediate variable used in calculation of $(\delta^*/L)_F$

FNCFRR2	intermediate variable used in calculation of $(\delta^*/L)_F$
FNCGRR1	intermediate variable used in calculation of $(\delta^*/L)_g$
FNCGRR2	intermediate variable used in calculation of $(\delta^*/L)_g$
FNCRER1	intermediate variable used in calculation of y/L
FNCRER2	intermediate variable used in calculation of y/L
FNCTHE1	intermediate variable used in calculation of H_i^*
FNCTHE2	intermediate variable used in calculation of H_i^*
FPREV	$F_{1,2}$
FR0	temperature recovery factor, used to compute Stanton number
FTAB	input u/u_e profile
FTEST	value of F at outer edge of boundary layer
FUNCF1	intermediate variable used in calculation of $(\theta^*/L)_F$
FUNCF2	intermediate variable used in calculation of $(\theta^*/L)_F$
FUNCG1	intermediate variable used in calculation of $(\theta^*/L)_g$
FUNCG2	intermediate variable used in calculation of $(\theta^*/L)_g$
FUNCX1	intermediate variable used in calculation of table of ξ values, XITAB
FUNCX2	intermediate variable used in calculation of table of ξ values, XITAB
FUNCY1	intermediate variable used in calculation of table of η values, ETATAB

FUNCY2	intermediate variable used in computation of table of η values, ETATAB
F1	$F_{1,n}$
F2	$F_{2,n}$
GEE2	input $g_{1,2}$
GPREV	$g_{1,2}$
GTAB	input w/w_e profile
GTEST	value of g at outer edge of boundary layer
G1	$g_{1,n}$
G2	$g_{2,n}$
HDCAPHE	h/H_e
HEDCPHE	h_e/H_e
HIS	$H_i^* = (\delta^*/\theta^*)_i$ (see eq. (57))
HISF	$\left(H_i^*\right)_F$
HISG	$\left(H_i^*\right)_g$
HRDCPHE	h_e/H_e
HSHE	$1 - \frac{{w_e}^2}{2H_e}$
I	a subscript
ICHD	indicates which D to compute in ABCDGS
ICOUNT	a count of number of times output is printed

IFBLU	= 0 for computing $\bar{f}$ = 1 for table lookup of $\bar{f}$
INIT	= 0 for $\Delta\xi_{initial}$ = DELXI0 = 1 for small $\Delta\xi_{initial}$
ITERATE	indicates number -1 of iterations to get an acceptable set of values for all variables at a particular step
ITHETA	code for input temperature profile ITHETA = 1 if RHOTAB used ITHETA = 2 if ZETATAB used
IUSEEMU	= 0 for laminar ($\epsilon/\mu = 0$) solution = 1 for turbulent solution
IVAL	a variable used to denote outer boundary (number of equations (or $\Delta\eta$ steps) to be used at a particular step for a particular variable)
IVEG	= 1 for printout of V profile = 2 for printout of ϵ/μ profile = 3 for printout of g profile
IWLDMP	= 0 for using wall properties in wall damping function = 1 for using local properties in wall damping function
J	body shape index j = 0 for two-dimsional flows j = 1 for axisymmetric flows
JJ	a subscript used in COMPUTE
KF	a subscript used in COMPUTE
KMAX	N-10 (Number of values computed across boundary layer - 10)
M	denotes order of interpolation for library subroutine FTLUP

MAX	indicates number of values to be printed
MPWEMU	= 0 for PRTTAB = $N_{Pr,T}$ values = 1 for PRTTAB = $N_{Pr,t}$ values
M2	-2, indexing parameter in COMPUTE
NBACK	indexing parameter used in COMPUTE to denote edge
NF	subscript used in COMPUTE
NFBY	number of values in FBARTAB
NFCFAB	number of values in ABTAB
NMAXF	number of η steps to outer edge of F profile
NMAXG	number of η steps to outer edge of g profile
NMAXTH	number of η steps to outer edge of θ profile
NPRINT	counter used to determine when to print
NSTEPS	number of ξ steps between profile printouts
NUMDELE	NUMETA - 1
NUMETA	maximum number of steps in η-direction
NUMX	number of values in XL table
NUMY	number of values in YL table
NYP	number of values in PRTTAB
OL	reference length
PHIR	$(\rho\mu)/(\rho\mu)_e$ (see eq. (19))

PR	molecular Prandtl number
PRT	turbulent Prandtl number
PRTTAB	turbulent Prandtl number table
QBAR	output heat-transfer parameter, $\frac{\dot{q}_w L}{\mu_e H_e}$
R	r, body radius divided by L
RA	average value of r
RCURJ	a combination of terms used in computing $\bar{\theta}/L$
RDXL	$R_{e,x}/L$
REDSF	$\frac{\rho_e u_e \delta_F^*}{\mu_e}$
REDSG	$\frac{\rho_e u_e \delta_g^*}{\mu_e}$
RERS	ρ_e/ρ_s
RERSA	$(\rho_e/\rho_s)_a$
RERSTAB	ρ_e/ρ_s table
RERS1	$(\rho_e/\rho_s)_1$
RERS2	$(\rho_e/\rho_s)_2$
RETSF	$\frac{\rho_e u_e \theta_F^*}{\mu_e}$
RETSG	$\frac{\rho_e u_e \theta_g^*}{\mu_e}$
REX	$\frac{\rho_e u_e x}{\mu_e}$
RHOEROA	$(\rho_e/\rho)_a$

RHOERO1	$(\rho_e/\rho)_{1,n}$
RHOERO2	$(\rho_e/\rho)_{2,n}$
RHORHOE	ρ/ρ_e
RHOTAB	input ρ_e/ρ table
RMRRMSA	average value of $(\rho\mu)_e/(\rho\mu)_s$
RMUTAB	table of $(\rho\mu)_e/(\rho\mu)_s$
RRUUER	$R_s \frac{(\rho\mu)_e}{(\rho\mu)_s} \frac{u_e}{\sqrt{2H_e}} r^{2j}$
RTAB	r/l table
RURDRUS	$(\rho\mu)_e/(\rho\mu)_s$
RUSDRUR	$(\rho\mu)_s/(\rho\mu)_e$
R1	r_1
R2	r_2
SHE	Sutherland's constant divided by H_e
SMDEL	summation used in computing H_i^*
SMLG	coefficient in formula for dependent variables (see eq. (31), for example)
SMRER	summation used in computing y/δ
SMTHE	summation used in computing H_i^*
ST	Stanton number
SUMF	summation used in computing $(\theta^*/L)_F$
SUMFETA	summation used in computing η

SUMFRR	summation used in computing $(\delta^*/L)_F$
SUMFXI	summation used in computing ξ
SUMG	summation used in computing $(\theta^*/L)_g$
SUMGRR	summation used in computing $(\delta^*/L)_g$
SUMRER	summation used in computing y/L and y/δ in CALCM
TAFDTAG	ratio of chordwise to spanwise shear at wall
TEST	value of F, g, or θ at edge of boundary layer
THETAA	average value of θ
THETAPR	$\theta_{1,2}$
THETA1	$\theta_{1,n}$
THETA2	$\theta_{2,n}$
THTEST	value of θ at edge of boundary layer
TIMES	temporary storage used in computing V_a
TMDLDL2	temporary storage used in computing CAPB and CAPD in ABCDGS
TSDLF	$(\theta^*/L)_F$
TSDLG	$(\theta^*/L)_g$
TSTVAL	test value used in COMPUTE for right side of equations (35), (44), and (45)
UEDSTAB	$u_e/\sqrt{2H_e}$
UES2HEA	$\left(u_e/\sqrt{2H_e}\right)_a$
UES2HE1	$\left(u_e/\sqrt{2H_e}\right)_1$

UES2HE2	$\left(u_e/\sqrt{2H_e}\right)_2$
VA	average values of V
VADLDL2	temporary storage used in computing CAPA and CAPC in ABCDGS
VWA	average value of VWTAB
VWTAB	v_w/u_e for axisymmetric or two-dimensional flow v_w/w_e for swept-leading-edge flow
WEDS2HE	$w_e/\sqrt{2H_e}$
XI	ξ
XIA	ξ_a
XIBARA	$(2\xi)_a^{2\bar{n}}$
XIBAR1	$(2\xi)_1^{2\bar{n}}$
XIBAR2	$(2\xi)_2^{2\bar{n}}$
XIBCPPA	$(\bar{\xi}P)_a$
XIBCPUA	$(\bar{\xi}U)_a$
XIBCPZA	$(\bar{\xi}Z)_a$
XIDELFA	temporary storage used in computing CAPB and CAPD in ABCDGS
XINB	$(2\xi)^{\bar{n}}$
XISTOP	value of ξ where solution is to be terminated
XITAB	table of ξ values where input is specified
XITEST	value of ξ where $\Delta\xi$ is increased by a factor of 10

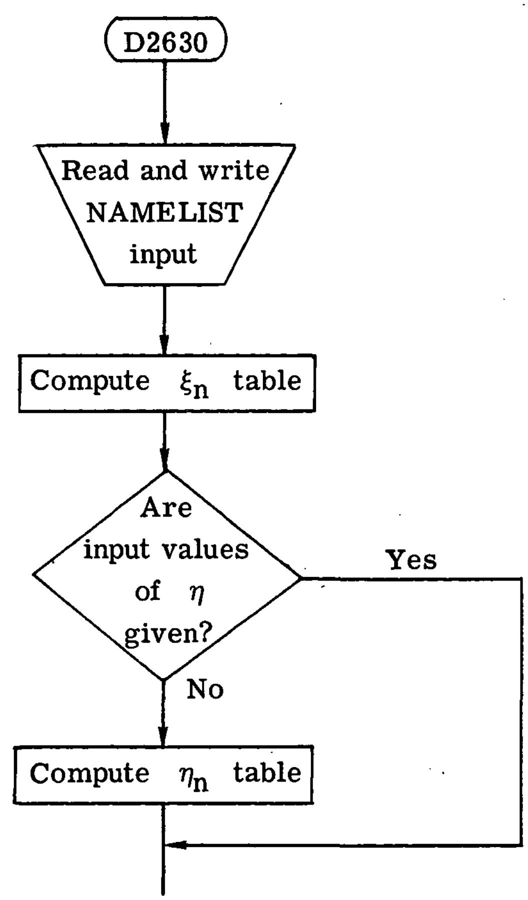
XI0	value of ξ at input station	
XI1	ξ_1	
XI2	ξ_2	
XK	$\Delta\eta_n/\Delta\eta_{n-1}$	
XL	table of x/L values	
XLPR	x/L values where profile printout is required	
XMBAR	$\overline{M}$	
XMBARA	$\overline{M}_a$	
XMBARA2	$\overline{M}_{a/2}$	
XMBAR1	$\overline{M}_{1,n}$	
XMBAR2	$\overline{M}_{2,n}$	(see eqs. (10b) and (10c))
XMCIRA2	$\hat{M}_{a/2}$	
XMCIRC	$\hat{M}$	
XMCIRCA	$\hat{M}_a$	
XMCIRC1	$\hat{M}_{1,n}$	
XMCIRC2	$\hat{M}_{2,n}$	
XMMEF	$(M/M_e)_F$	
XMMEG	$(M/M_e)_g$	

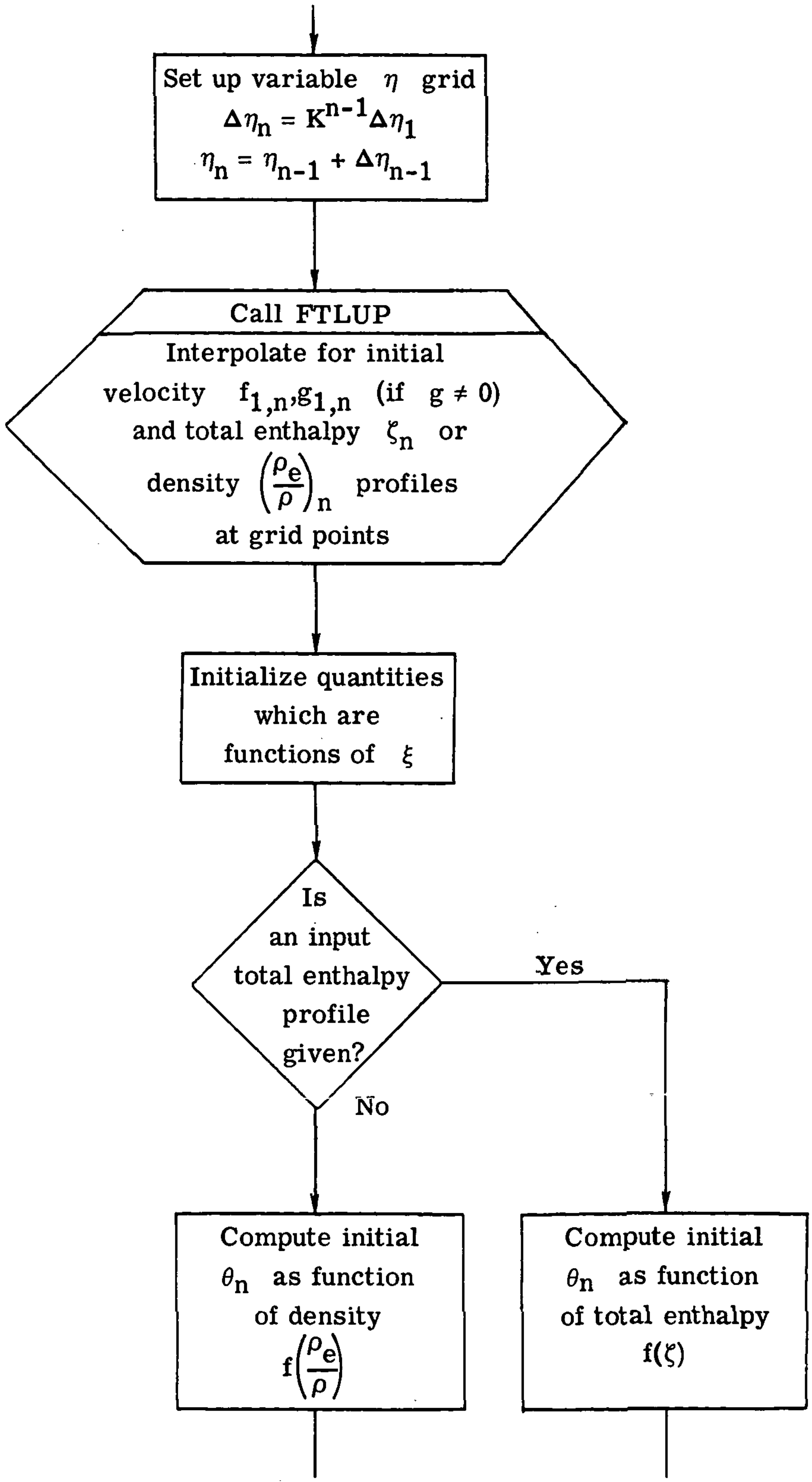
XMPRIM	M'	(see eqs. (10d) and (10e))
XMPRIMA	M'_a	
XMPRIM1	$M'_{1,n}$	
XMPRIM2	$M'_{2,n}$	
XMPRMA2	$M'_{a/2}$	
XMSTAR	M^*	
XMSTARA	M^*_a	
XMSTAR1	$M^*_{1,n}$	
XMSTAR2	$M^*_{2,n}$	
XMSTRA2	$M^*_{a/2}$	
XNBAR	$\bar{n}$	
XXL	x/L	
X0, X1	initial value of x/L	
YDD	y/δ	
YDDFB	y/δ values corresponding to FBARTAB	
YDDPRT	y/δ values corresponding to PRTTAB	
YDL	y/L	
YL	table of y/L values for initial profiles	
ZETA	ζ	

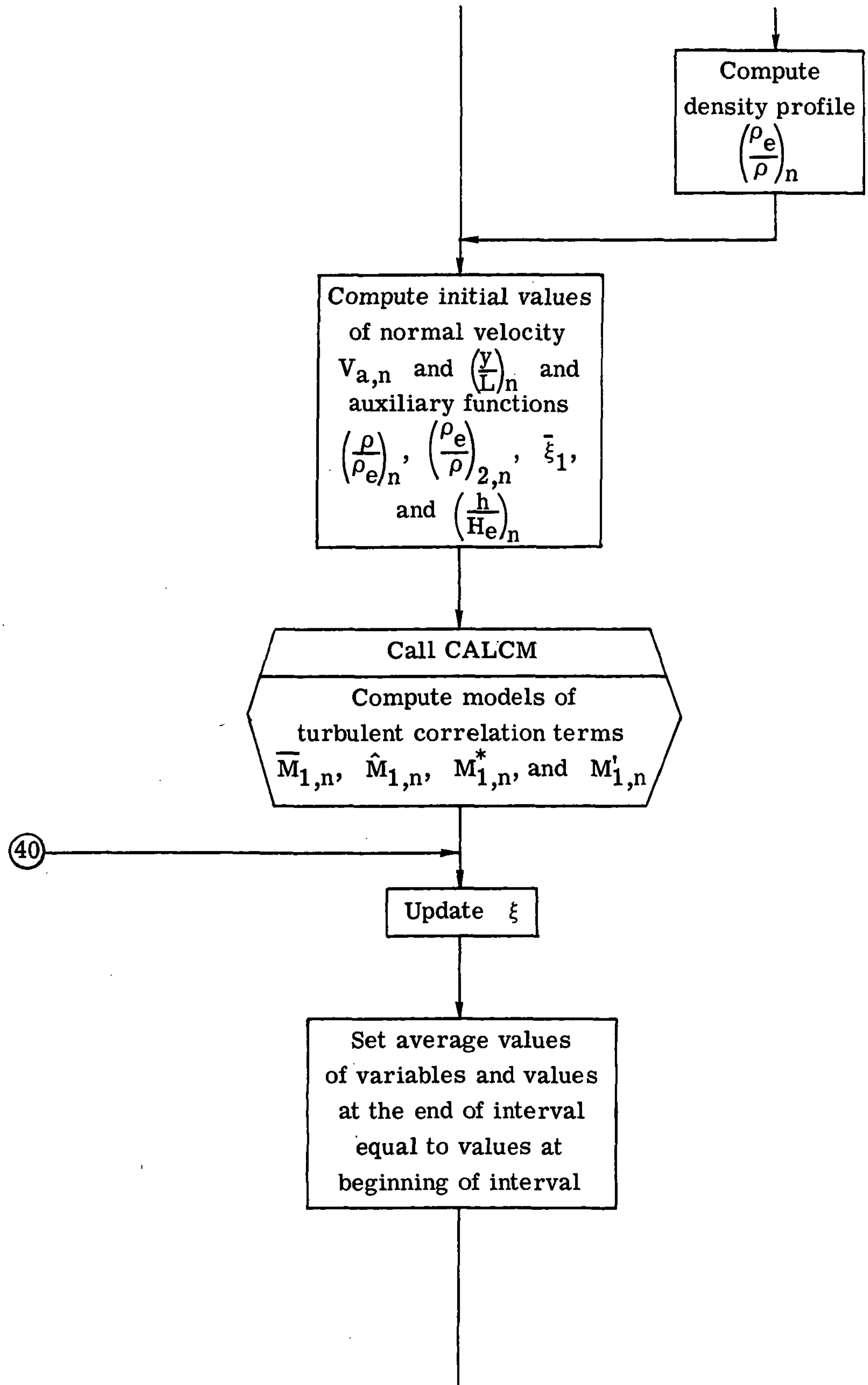
ZETAE	$\zeta_e = 1.0$
ZETATAB	input ζ profile
ZETAW1	ζ_{w_1}
ZETAW2	ζ_{w_2}
ZETWTAB	ζ_w table

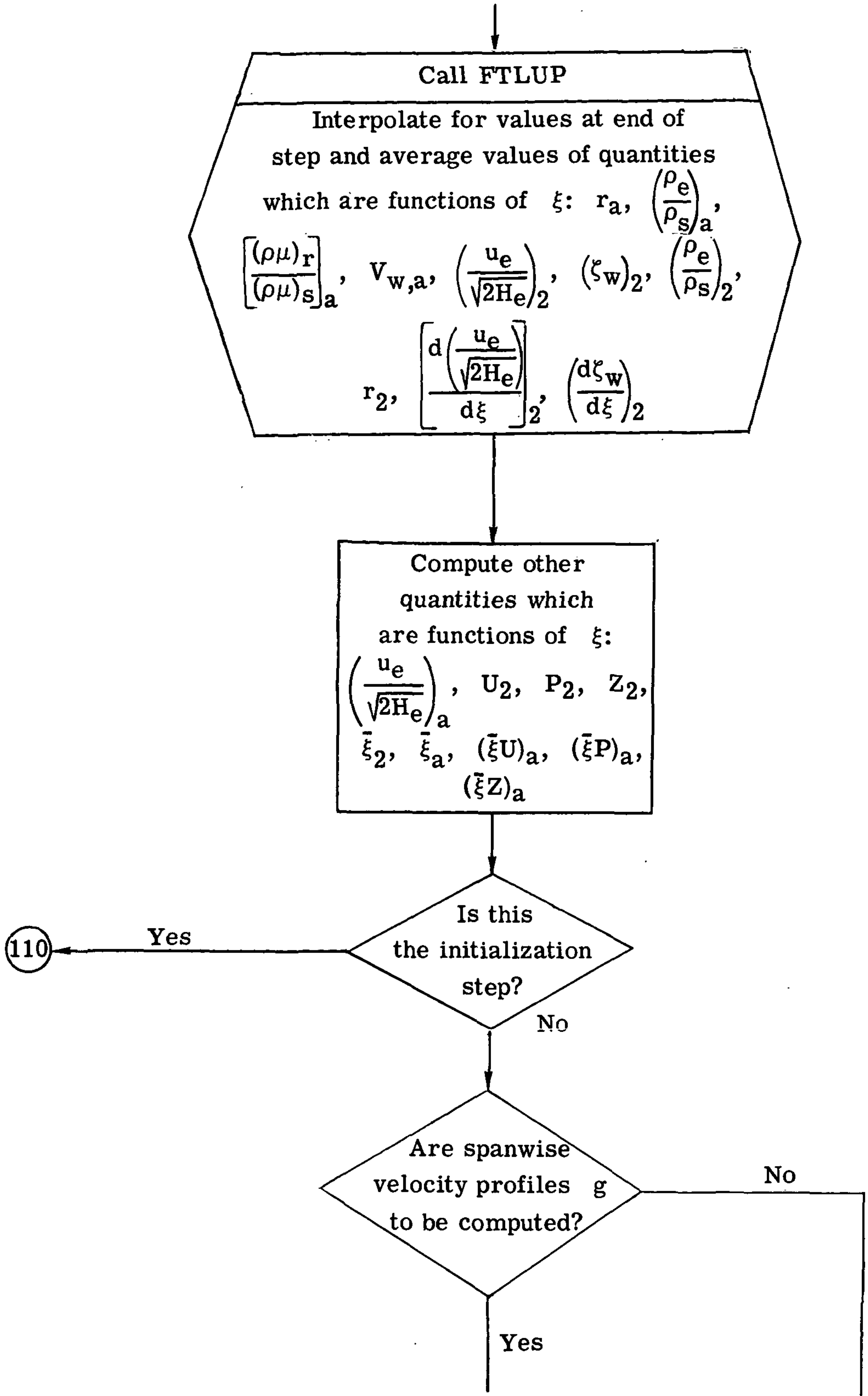
Description, Flow Charts, and Listings of the Main Program and Subprograms

Main program D2630.- The main program controls the finite-difference solution of the boundary-layer equations. It reads the input, computes other initial conditions, sets up the grid used in the solution, calls all the subroutines, and writes the output. The flow-diagram of main program D2630 is as follows:




Set up variable η grid
$\Delta\eta_n = K^{n-1}\Delta\eta_1$
$\eta_n = \eta_{n-1} + \Delta\eta_{n-1}$
Call FTLUP
Interpolate for initial velocity $f_{1,n}, g_{1,n}$ (if $g \neq 0$) and total enthalpy ζ_n or density $\left(\frac{\rho_e}{\rho}\right)_n$ profiles at grid points
Initialize quantities which are functions of ξ
Is an input total enthalpy profile given?
Yes
No
Compute initial θ_n as function of density $f\left(\frac{\rho_e}{\rho}\right)$
Compute initial θ_n as function of total enthalpy $f(\zeta)$


Compute density profile $\left(\frac{\rho_e}{\rho}\right)_n$
Compute initial values of normal velocity $V_{a,n}$ and $\left(\frac{y}{L}\right)_n$ and auxiliary functions $\left(\frac{\rho}{\rho_e}\right)_n$, $\left(\frac{\rho_e}{\rho}\right)_{2,n}$, $\bar{\xi}_1$, and $\left(\frac{h}{H_e}\right)_n$
Call CALCM
Compute models of turbulent correlation terms $\overline{M}_{1,n}$, $\hat{M}_{1,n}$, $M^*_{1,n}$, and $M'_{1,n}$
40
Update ξ
Set average values of variables and values at the end of interval equal to values at beginning of interval


Call FTLUP
Interpolate for values at end of step and average values of quantities which are functions of ξ: r_a, $\left(\frac{\rho_e}{\rho_s}\right)_a$, $\left[\frac{(\rho\mu)_r}{(\rho\mu)_s}\right]_a$, $V_{w,a}$, $\left(\frac{u_e}{\sqrt{2H_e}}\right)_2$, $(\zeta_w)_2$, $\left(\frac{\rho_e}{\rho_s}\right)_2$, r_2, $\left[\frac{d\left(\frac{u_e}{\sqrt{2H_e}}\right)}{d\xi}\right]_2$, $\left(\frac{d\zeta_w}{d\xi}\right)_2$
Compute other quantities which are functions of ξ: $\left(\frac{u_e}{\sqrt{2H_e}}\right)_a$, U_2, P_2, Z_2, $\bar{\xi}_2$, $\bar{\xi}_a$, $(\bar{\xi}U)_a$, $(\bar{\xi}P)_a$, $(\bar{\xi}Z)_a$
Is this the initialization step?
Yes
110
No
Are spanwise velocity profiles g to be computed?
No
Yes

Correct input normal velocity at wall $V_{w,a}$ for a swept-leading-edge situation

(37)

Compute average values and values at end of step for densities $\left(\frac{\rho_e}{\rho}\right)_{2,n}$, $\left(\frac{\rho_e}{\rho}\right)_{na}$, and $\left(\frac{\rho}{\rho_e}\right)_{2,n}$, for y and $\left(\frac{y}{L}\right)_n$, and for static enthalpy $\left(\frac{h}{H_e}\right)_n$

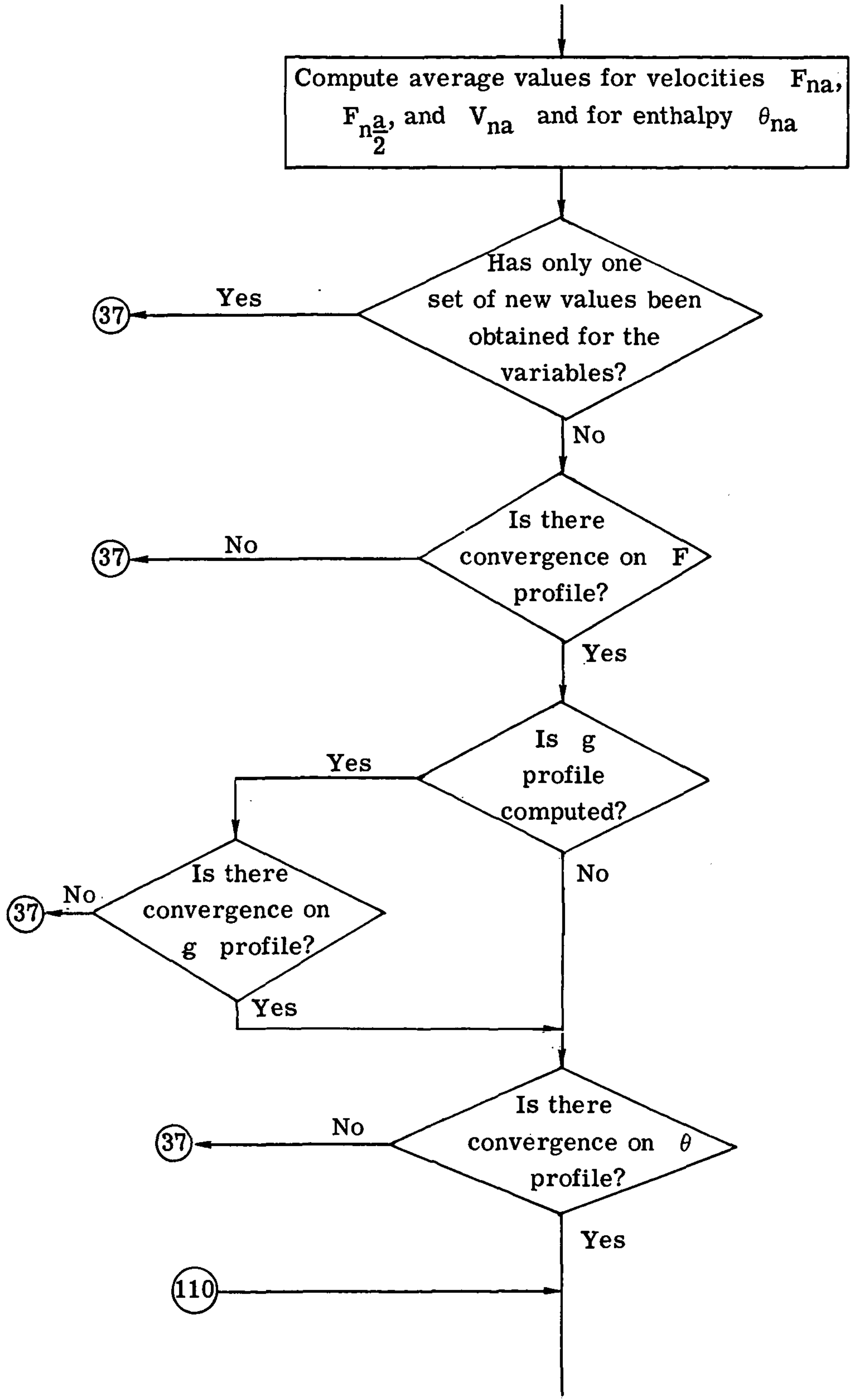
Call CALCM

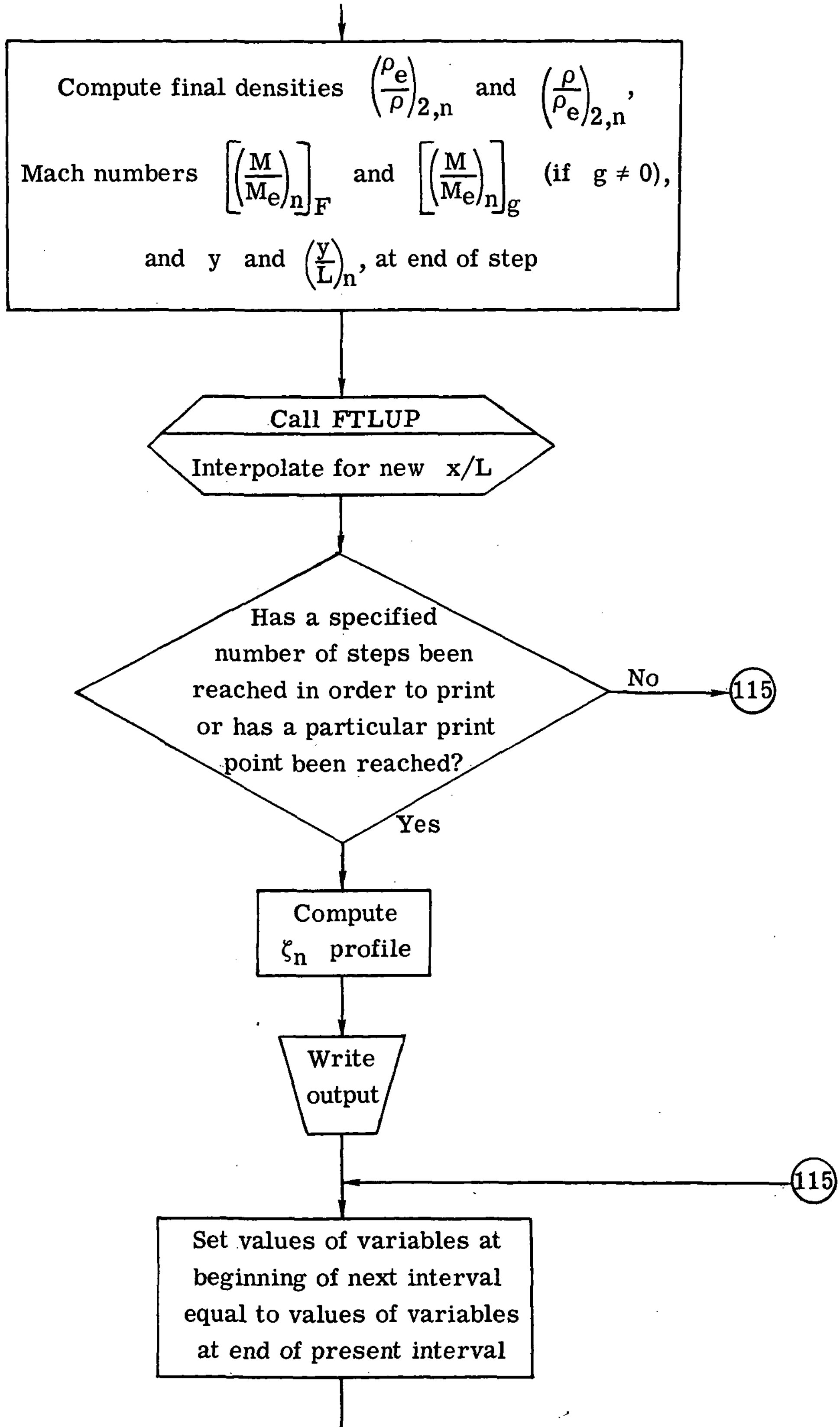
Compute models of turbulence correlations at end of step: $\overline{M}_{2,n}$, $\hat{M}_{2,n}$, $M^*_{2,n}$, and $M'_{2,n}$

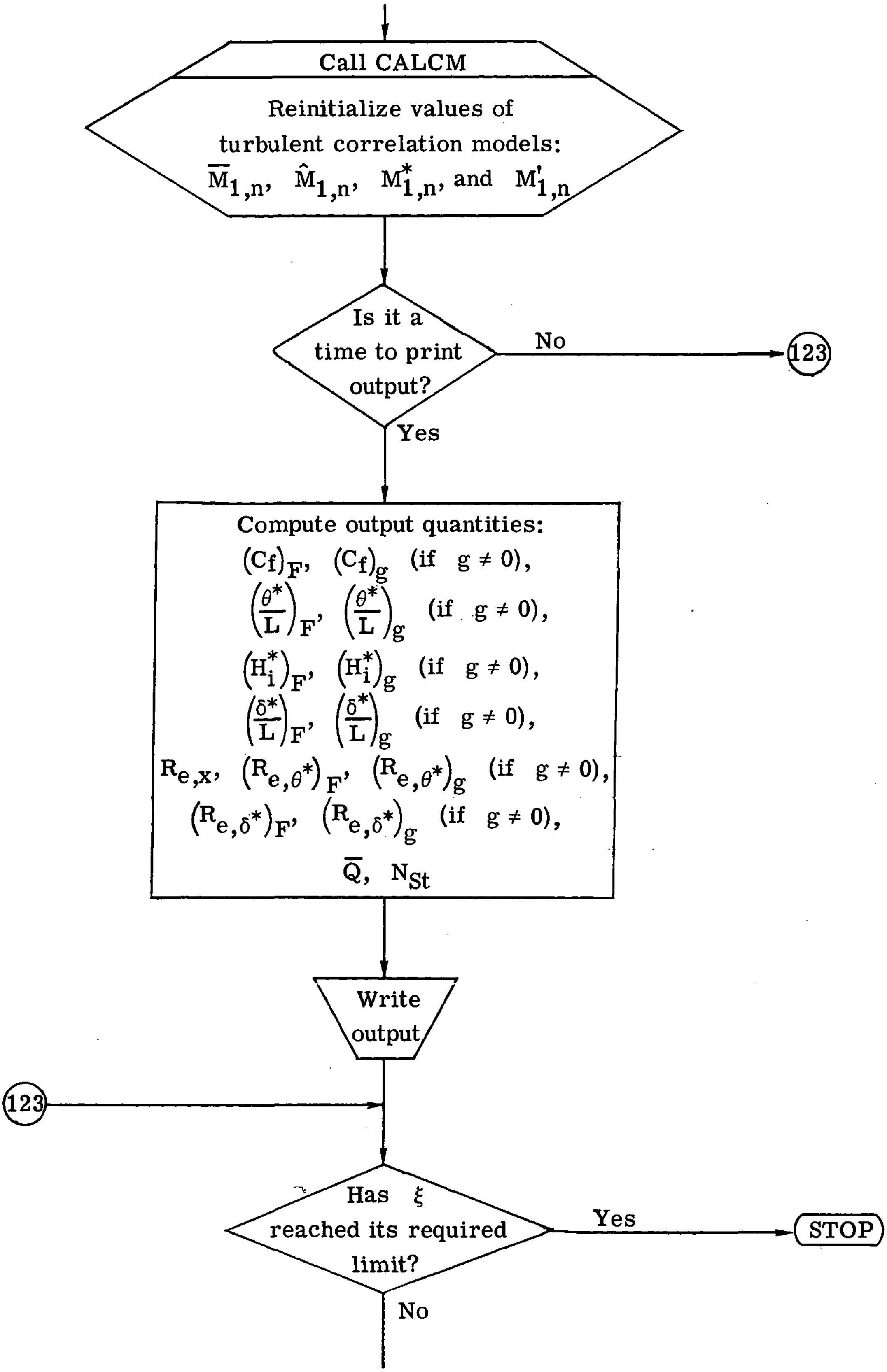
Compute average values of turbulence correlation models:
$\overline{M}_{na}$, M^*_{na}, M'_{na}, and $\hat{M}_{na}$ (if $g \neq 0$)
$\overline{M}_{n\frac{a}{2}}$, $M^*_{n\frac{a}{2}}$, $M'_{n\frac{a}{2}}$, and $\hat{M}_{n\frac{a}{2}}$ (if $g \neq 0$)

Call ABCDGS
Call COMPUTE

Calculate coefficients of solution matrix
Compute new values for velocities $F_{2,n}$ and $g_{2,n}$ (if $g \neq 0$) and for enthalpy $\theta_{2,n}$


Compute average values for velocities F_{na}, $F_{n\frac{a}{2}}$, and V_{na} and for enthalpy θ_{na}
Has only one set of new values been obtained for the variables?
Yes
37
No
Is there convergence on F profile?
No
37
Yes
Is g profile computed?
Yes
No
Is there convergence on g profile?
No
37
Yes
Is there convergence on θ profile?
No
37
Yes
110


Compute final densities $\left(\frac{\rho_e}{\rho}\right)_{2,n}$ and $\left(\frac{\rho}{\rho_e}\right)_{2,n}$,
Mach numbers $\left[\left(\frac{M}{M_e}\right)_n\right]_F$ and $\left[\left(\frac{M}{M_e}\right)_n\right]_g$ (if $g \neq 0$),
and y and $\left(\frac{y}{L}\right)_n$, at end of step
Call FTLUP
Interpolate for new x/L
Has a specified number of steps been reached in order to print or has a particular print point been reached?
No
115
Yes
Compute ζ_n profile
Write output
115
Set values of variables at beginning of next interval equal to values of variables at end of present interval


Call CALCM
Reinitialize values of turbulent correlation models: $\overline{M}_{1,n}$, $\hat{M}_{1,n}$, $M^*_{1,n}$, and $M'_{1,n}$
Is it a time to print output?
No
123
Yes
Compute output quantities:
$(C_f)_F$, $(C_f)_g$ (if $g \neq 0$),
$\left(\frac{\theta^*}{L}\right)_F$, $\left(\frac{\theta^*}{L}\right)_g$ (if $g \neq 0$),
$(H_i^*)_F$, $(H_i^*)_g$ (if $g \neq 0$),
$\left(\frac{\delta^*}{L}\right)_F$, $\left(\frac{\delta^*}{L}\right)_g$ (if $g \neq 0$),
$R_{e,x}$, $(R_{e,\theta^*})_F$, $(R_{e,\theta^*})_g$ (if $g \neq 0$),
$(R_{e,\delta^*})_F$, $(R_{e,\delta^*})_g$ (if $g \neq 0$),
$\overline{Q}$, N_{St}
Write output
123
Has ξ reached its required limit?
Yes
STOP
No

Reinitialize quantities which are functions of ξ:

$$\xi_1 = \xi_2$$
$$\bar{\xi}_1 = \bar{\xi}_2$$
$$U_1 = U_2$$
$$P_1 = P_2$$
$$Z_1 = Z_2$$

40

The program listing for Main program D2630 is as follows:

```
 PROGRAM D2630(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
 DIMENSION DELETA(350),FTAB(100),F1(350),FA(350),FA2(350),
1VA(350),GTAB(100),G1(350),G2(350),THETA1(350),THETA2(350),
2THETAA(350),RHOTAB(100),RHOERO1(350),RHOEROA(350),ZETATAB(100),
3ZETA(350),XMBAR1(350),XMCIRC1(350),XMSTAR1(350),XMPRIM1(350),
4PHIR(350),XMBARA2(350),XMCIRA2(350),XMSTRA2(350),XMPRMA2(350),
5CAPG(350),SMLG(350),ETATAB(100),DUDXTAB(75),DZDXTAB(75),
6XITAB(75),XMBAR2(350),XMCIRC2(350),XMSTAR2(350),XMPRIM2(350),
7XMBARA(350),XMCIRCA(350),XMSTARA(350),XMPRIMA(350),SUMRER(350),
8F2(350),XLPR(30),VWTAB(75),RMUTAB(75),ABTAB(20),FCFTAB(20),
9XMMEG(350),XMMEF(350),ETA(350),UEDSTAB(75),ZETWTAB(75),XL(75),
1RTAB(75),RERSTAB(75),YL(100)
 COMMON/EPSDMU/EPSDMU(350)
 COMMON/TABLE1/RHOERO2(350)
 COMMON/TABLE2/YDL(350),RHORHOE(350)
 COMMON/THREE/NUMETA,NMAXF
 COMMON/FEB12/VWA
 COMMON/FEB11/PRTTAB(20),YDDPRT(20),NYP
 COMMON/HDCAPHE/HDCAPHE(350),GEE2
 COMMON/MUUSE/IUSEEMU,MPWEMU
 COMMON/FBAR/FBARTAB(20),IFBLU,YDDFB(20),NFBY
 COMMON/IWLDMP/IWLDMP
 NAMELIST/NAM1/NUMETA,NMAXF,NMAXG,DELETA,XK,XI0,DELXI0,XITEST,
1XISTOP,FTAB,ETATAB,VWTAB,EPSLONE,EPSLONW,        GTAB,UEDSTAB,
2 WEDS2HE,PR,ZETWTAB,XNBAR,RERSTAB,CAPRS,RTAB,J,RHOTAB,
3SHE,ZETATAB,ITHETA,HSHE,XL,NUMX,YL,NUMY,
4X0,OL,DUDXTAB,DZDXTAB,FRO,NSTEPS,
5AP,BP,CP,ABTAB,FCFTAB,NFCFAB,PRTTAB,YDDPRT,NYP,
6          XLPR,IVEG,INIT,IUSEEMU,MPWEMU,
7FBARTAB,IFBLU,YDDFB,NFBY,IWLDMP
 ZETATAB(2)=0.0
 RHOTAB(2)=0.0
 ETATAB(2)=0.0
 XLPR(2)=0.0
 XI0=0.0
 NPRINT=1
 ICOUNT=1
 READ(5,NAM1)
```

```
      WRITE(6,NAM1)
C
C     OPTION FOR INITIAL DELTA XI
C
      IF(INIT.EQ.0)DELXI=DELXI0
      IF(INIT.EQ.1)DELXI=XI0*1.E-5
      GEE2=GTAB(2)
C
C     T R A N S F O R M   X   T O   X I
C
      HRDCPHE=1.-UEDSTAB(1)**2-WEDS2HE**2
      RMUTAB(1)=SQRT(HRDCPHE/HSHE)*(HSHE+SHE)/(HRDCPHE+SHE)
     1*RERSTAB(1)*HRDCPHE
     2/HSHE
      FUNCX1=RMUTAB(1)*UEDSTAB(1)*RTAB(1)**(2*J)
      X1=X0
      DO 22 I=1,NUMX
      IF(I.EQ.1)GO TO 15
      HRDCPHE=1.-UEDSTAB(I)**2-WEDS2HE**2
      RMUTAB(I)=SQRT(HRDCPHE/HSHE)*(HSHE+SHE)/(HRDCPHE+SHE)
     1*RERSTAB(I)*HRDCPHE
     2/HSHE
      FUNCX2=RMUTAB(I)*UEDSTAB(I)*RTAB(I)**(2*J)
      SUMFXI=SUMFXI+(FUNCX2+FUNCX1)/2.*(XL(I)-XL(I-1))
      FUNCX1=FUNCX2
      GO TO 95
   15 SUMFXI=0.
   95 RRUUER=CAPRS*RMUTAB(I)*UEDSTAB(I)*RTAB(I)**(2*J)
      IF(XI0.EQ.0.)GO TO 25
      GO TO 31
   25 XI0=RRUUER*X1
   31 XITAB(I)=XI0+CAPRS*SUMFXI
      DUDXTAB(I)=DUDXTAB(I)/RRUUER
      DZDXTAB(I)=DZDXTAB(I)/RRUUER
   22 CONTINUE
      XI1=XI0
C
C     I F   N O   E T A   T A B L E ,  T R A N S F O R M   Y   T O   E T A
C
      IF(ETATAB(2).NE.0.)GO TO 26
      IF(ZETATAB(2).EQ.0.)GO TO 32
      DO 202 I=1,NUMY
      THETA1(I)=(ZETATAB(I)-ZETWTAB(1))/(1.-ZETWTAB(1))
  202 CONTINUE
      DO 33 I=1,NUMY
      RHOTAB(I)=((1.-ZETWTAB(1))*THETA1(I)+ZETWTAB(1)-UEDSTAB(1)**2
     1*FTAB(I)**2-WEDS2HE    **2*GTAB(I)**2)/(1.-UEDSTAB(1)**2-WEDS2HE
     2**2)
   33 CONTINUE
   32 CONTINUE
      DO 34 I=1,NUMY
      RHORHOE(I)=1./RHOTAB(I)
   34 CONTINUE
      FUNCY1=RHORHOE(1)*RERSTAB(1)
      SUMFETA=0.0
      ETATAB(1)=0.0
      DO 27 I=2,NUMY
      FUNCY2=RHORHOE(I)*RERSTAB(1)
      SUMFETA=SUMFETA+(FUNCY2+FUNCY1)/2.*(YL(I)-YL(I-1))
```

```
      FUNCY1=FUNCY2
      ETATAB(I)=CAPRS*UEDSTAB(1)*RTAB(1)**J/(2*XI0)**XNBAR *SUMFETA
   27 CONTINUE
   26 CONTINUE
C
C     C O N S T A N T S
C
      ZETAE=1.
      FTEST=.99999
      GTEST=.99999
      THTEST=.99999
      ETA(1)=0.
      SUMRER(1)=0.
C
C     C O M P U T E   D E L T A   E T A S
C
      F1(1)=FTAB(1)
      G1(1)=GTAB(1)
      IF(ZETATAB(2).NE.0.)ZETA(1)=ZETATAB(1)
      IF(RHOTAB(2).NE.0.)RHOERO1(1)=RHOTAB(1)
      M=2
      DO 10 I=2,NUMETA
      DELETA(I)=XK**(I-1)*DELETA(1)
C
C     C O M P U T E   E T A   S
C     I N T E R P O L A T E   T O   G E T   F ( E T A ) S
C
      ETA(I)=ETA(I-1)+DELETA(I-1)
      IF(F1(I-1).GT..9)M=1
      CALL FTLUP(ETA(I),F1(I),M,NUMY,ETATAB,FTAB)
      M=2
      IF(GTAB(2).EQ.0.)GO TO 17
      IF(G1(I-1).GT..9)M=1
      CALL FTLUP(ETA(I),G1(I),M,NUMY,ETATAB,GTAB)
      M=2
      GO TO 16
   17 G1(I)=0.0
   16 IF(ZETATAB(2).EQ.0.)GO TO 18
      IF(ZETA(I-1).GT..9)M=1
      CALL FTLUP(ETA(I),ZETA(I),M,NUMY,ETATAB,ZETATAB)
      M=2
      GO TO 19
   18 IF(ABS(RHOERO1(I-1)-1.).GE..1)M=1
      CALL FTLUP(ETA(I),RHOERO1(I),M,NUMY,ETATAB,RHOTAB)
   19 CONTINUE
   10 CONTINUE
C
C     I N I T I A L   F ( X I 1 ) S
C
      UES2HE1=UEDSTAB(1)
      ZETAW1=ZETWTAB(1)
      RERS1=RERSTAB(1)
      R1=RTAB(1)
      DUEDX1=DUDXTAB(1)
      DZWDX1=DZDXTAB(1)
      CAPU1=DUEDX1/UES2HE1
      CAPP1=-CAPU1
      CAPZ1=DZWDX1/(1.-ZETAW1)
```

```
C
C     C O M P U T E   I N I T I A L   T H E T A S
C
      IF (ITHETA.EQ.2)GO TO 11
      THETA1(1)=0.0
      DO 200 I=2,NUMETA
      THETA1(I)=(RHOERO1 (I)*(1.-UES2HE1**2-WEDS2HE**2)-ZETAW1+UES2HE1
     1**2*F1(I)**2+WEDS2HE**2*G1(I)**2)/(1.-ZETAW1)
  200 CONTINUE
      GO TO 12
   11 DO 201 I=1,NUMETA
      THETA1(I)=(ZETA(I)-ZETAW1)/(1.-ZETAW1)
  201 CONTINUE
   12 CONTINUE
C
C     I F   Z E T A S   A R E   G I V E N,  I N I T I A L   R H O E R O 1 S
C     M U S T   B E   C A L C U L A T E D
C
      IF(ZETATAB(2).EQ.0.)GO TO 28
      DO 29 I=1,NUMETA
      RHOERO1(I)=((1.-ZETAW1)*THETA1(I)+ZETAW1-UES2HE1**2*F1(I)**2
     1-WEDS2HE**2*G1(I)**2)/(1.-UES2HE1**2-WEDS2HE**2)
   29 CONTINUE
   28 CONTINUE
      DO 21 I=1,NUMETA
      RHORHOE(I)=1./RHOERO1(I)
   21 RHOERO2(I)=RHOERO1(I)
C
C     C O M P U T E   I N I T I A L   V S
C
      XIBAR1=(2.*XI0)**(2.*XNBAR )
      VA(1)=(2*XI1)**XNBAR /R1**J*RHORHOE(1)*RERS1*VWTAB(1)/RMUTAB(1)
      IF(GEE2.NE.0.)VA(1)=VA(1)*WEDS2HE/UES2HE1
      DO 20 I=2,NUMETA
      VA(I)=VA(I-1)-DELETA(I-1)*XIBAR1*(F1(I)+F1(I-1))*XNBAR /(2.*XI0)
   20 CONTINUE
      YDL(1)=0.0
      FNCRER1=RHOERO2(1)
      DO 42 I=2,NUMETA
      FNCRER2=RHOERO2(I)
      SUMRER(I)=SUMRER(I-1)+(FNCRER2+FNCRER1)/2.*DELETA(I-1)
      FNCRER1=FNCRER2
      YDL (I)=(2.*XI1)**XNBAR *1./RERS1*SUMRER(I)/(CAPRS*R1**J*UES2HE1)
   42 CONTINUE
C
C     C A L C U L A T E   I N I T I A L   M S
C
      VWA=VWTAB(1)
      IF(GEE2.NE.0.)VWA=VWA*WEDS2HE/UES2HE1
      DO 44 I=1,NUMETA
      HDCAPHE(I)=(1.-ZETAW1)*THETA1(I)+ZETAW1-UES2HE1**2*F1(I)**2
     1-WEDS2HE**2*G1(I)**2
   44 CONTINUE
      CALL CALCM(ZETAW1,UES2HE1,F1,WEDS2HE,ZETAE,
     1                     SHE,PR,     XMBAR1,XMCIRC1,XMSTAR1,XMPRIM1,PHIR,
     2HSHE,           RERS1,XI1,XNBAR ,R1,J,
     3DELETA,RURDRUS,     CAPRS,SUMRER,AP,BP,CP,       ABTAB,FCFTAB,NFCFAB,
     4NMAXG,G1)
   40 CONTINUE
```

```
C
C     B E G I N   N E W   I N T E R V A L
C
      IF(XI1.GE.XI0+30.*DELXI.AND.INIT.EQ.1)GO TO 46
      GO TO 48
   46 DELXI=DELXI0
      INIT=0
   48 CONTINUE
      IF(XI1.LT.XITEST)GO TO 35
      DELXI=DELXI*10.
      XITEST=XITEST*10.
   35 XI2=XI1+DELXI
      IF(ICOUNT.EQ.1)XI2=XI1
      XIA=(XI1+XI2)/2.
      ITERATE=1
      DO 30 I=1,NUMETA
      F2(I)=F1(I)
      FA(I)=F1(I)
      G2(I)=G1(I)
      THETA2(I)=THETA1(I)
      THETAA(I)=THETA1(I)
      XMBAR2(I)=XMBAR1(I)
   30 CONTINUE
   52 CONTINUE
C
C     I N T E R P O L A T E   F O R   F ( X I 2 ) S
C
      CALL FTLUP(XIA,RA,1,NUMX,XITAB,RTAB)
      CALL FTLUP(XIA,RERSA,1,NUMX,XITAB,RERSTAB)
      CALL FTLUP(XIA,RMRRMSA,1,NUMX,XITAB,RMUTAB)
      CALL FTLUP(XIA,VWA,1,NUMX,XITAB,VWTAB)
      CALL FTLUP(XI2,UES2HE2,1,NUMX,XITAB,UEDSTAB)
      CALL FTLUP(XI2,ZETAW2,1,NUMX,XITAB,ZETWTAB)
      CALL FTLUP(XI2,RERS2,1,NUMX,XITAB,RERSTAB)
      CALL FTLUP(XI2,R2,1,NUMX,XITAB,RTAB)
      CALL FTLUP(XI2,DUEDX2,1,NUMX,XITAB,DUDXTAB)
      CALL FTLUP(XI2,DZWDX2,1,NUMX,XITAB,DZDXTAB)
      UES2HEA=(UES2HE1+UES2HE2)/2.
      CAPU2=DUEDX2/UES2HE2
      CAPP2=-CAPU2
      CAPZ2=DZWDX2/(1.-ZETAW2)
      XIBAR2=(2.*XI2)**(2.*XNBAR)
      XIBARA=(XIBAR1+XIBAR2)/2.
      XIBCPUA=(XIBAR1*CAPU1+XIBAR2*CAPU2)/2.
      XIBCPPA=(XIBAR1*CAPP1+XIBAR2*CAPP2)/2.
      XIBCPZA=(XIBAR1*CAPZ1+XIBAR2*CAPZ2)/2.
      IF(ICOUNT.EQ.1)GO TO 110
      IF(GEE2.NE.0.)VWA=VWA*WEDS2HE/UES2HEA
C
C     I T E R A T E   W I T H   S A M E   X I 2
C
   37 DO 36 I=1,NUMETA
      RHOERO2(I)=((1.-ZETAW2)*THETA2(I)+ZETAW2-UES2HE2**2*F2(I)**2
     1-WEDS2HE**2*G2(I)**2)/(1.-UES2HE2**2-WEDS2HE**2)
      RHOEROA(I)=(RHOERO1(I)+RHOERO2(I))/2.
      RHORHOE(I)=1./RHOERO2(I)
   36 CONTINUE
      FNCRER1=RHOERO2(1)
      DO 38 I=2,NUMETA
```

```
      FNCRER2=RHOERO2(I)
      SUMRER(I)=SUMRER(I-1)+(FNCRER2+FNCRER1)/2.*DELETA(I-1)
      FNCRER1=FNCRER2
      YDL  (I)=(2.*XI2)**XNBAR *1./RERS2*SUMRER(I)/(CAPRS*R2**J*UES2HE2)
   38 CONTINUE
C
C     C A L C U L A T E   M S
C
      DO 39  I=1,NUMETA
      HDCAPHE(I)=(1.-ZETAW2)*THETA2(I)+ZETAW2-UES2HE2**2*F2(I)**2
     1-WEDS2HE**2*G2(I)**2
   39 CONTINUE
      CALL CALCM(ZETAW2,UES2HE2,F2,WEDS2HE,ZETAE,
     1              SHE,PR,     XMBAR2,XMCIRC2,XMSTAR2,XMPRIM2,PHIR,
     2HSHE,         RERS2,XI2,XNBAR ,R2,J,
     3DELETA,RURDRUS,     CAPRS,SUMRER,AP,BP,CP,      ABTAB,FCFTAB,NFCFAB,
     4NMAXG,G2)
      DO 54  I=1,NUMETA
      XMBARA(I)=(XMBAR1(I)+XMBAR2(I))/2.
      IF(GEE2.NE.0.)XMCIRCA(I)=(XMCIRC1(I)+XMCIRC2(I))/2.
      XMSTARA(I)=(XMSTAR1(I)+XMSTAR2(I))/2.
      XMPRIMA(I)=(XMPRIM1(I)+XMPRIM2(I))/2.
   54 CONTINUE
      NUMDELE=NUMETA-1
      DO 50  I=1,NUMDELE
      XMBARA2(I)=(XMBARA(I+1)+XMBARA(I))/2.
      IF(GEE2.NE.0.)XMCIRA2(I)=(XMCIRCA(I+1)+XMCIRCA(I))/2.
      XMSTRA2(I)=(XMSTARA(I+1)+XMSTARA(I))/2.
      XMPRMA2(I)=(XMPRIMA(I+1)+XMPRIMA(I))/2.
   50 CONTINUE
C
C     C A L C U L A T E   F S
C
      ICHD=1
      FPREV=F2(2)
   55 CALL ABCDGS(NUMDELE,DELETA,XIBARA,DELXI,FA,VA,XMBARA2,ICHD,F1,
     1XIBCPUA,XIBCPPA,RHOEROA,G1,THETA1,XIBCPZA,THETAA,UES2HEA,XMPRMA2,
     2F2,WEDS2HE,G2,CAPG,SMLG,XK)
      CALL COMPUTE(NMAXF,FTEST,NUMDELE,   DELETA,EPSLONE,CAPG,SMLG,F2     )
C
C     U P D A T E   FA S   A N D   VA S
C
      DO 70  I=2,NUMETA
      FA(I)=(F1(I)+F2(I))/2.
   70 CONTINUE
      DO 73  I=1,NUMDELE
      FA2(I)=(FA(I+1)+FA(I))/2.
   73 CONTINUE
      TIMES=XNBAR/XIA
      VA(1)=(2*XIA)**XNBAR /RA**J/RHOEROA(1)*RERSA*VWA/RMRRMSA
      DO 74  I=2,NUMETA
      VA(I)=VA(I-1)-DELETA(I-1)*XIBARA/(2.*DELXI)*(F2(I)+F2(I-1)-F1(I)
     1-F1(I-1))-DELETA(I-1)*XIBARA*FA2(I-1)*TIMES
   74 CONTINUE
C
C     C A L C U L A T E   G S
C
      IF(GTAB(2).EQ.0.)GO TO 75
      ICHD=2
```

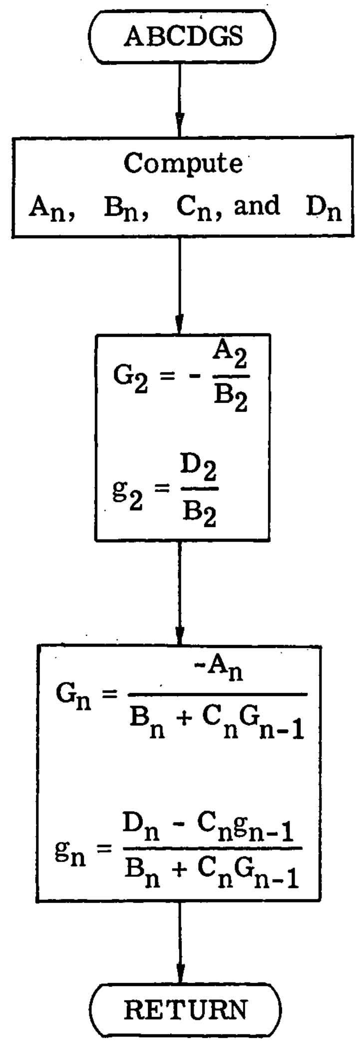
```
      GPREV=G2(2)
      CALL ABCDGS(NUMDELE,DELETA,XIBARA,DELXI,FA,VA,XMCIRA2,ICHD,F1,
     1XIBCPUA,XIBCPPA,RHOEROA,G1,THETA1,XIBCPZA,THETAA,UES2HEA,XMPRMA2,
     2F2,WEDS2HE,G2,CAPG,SMLG,XK)
      CALL COMPUTE(NMAXG,GTEST,NUMDELE,  DELETA,EPSLONE,CAPG,SMLG,G2    )
C
C     C A L C U L A T E   T H E T A S
C
   75 CONTINUE
      ICHD=3
      THETAPR=THETA2(2)
      CALL ABCDGS(NUMDELE,DELETA,XIBARA,DELXI,FA,VA,XMSTRA2,ICHD,F1,
     1XIBCPUA,XIBCPPA,RHOEROA,G1,THETA1,XIBCPZA,THETAA,UES2HEA,XMPRMA2,
     2F2,WEDS2HE,G2,CAPG,SMLG,XK)
      IF(XI1.EQ.XI0.AND.ITFRATE.EQ.1)NMAXTH=1.1*NMAXF
      CALL COMPUTE(NMAXTH,THTEST,NUMDELE,  DELETA,EPSLONE,CAPG,SMLG,
     1THETA2    )
C
C     U P D A T E   T H E T A S
C
      DO 76 I=1,NUMETA
   76 THETAA(I)=(THETA1(I)+THETA2(I))/2.
      ITERATE=ITERATE+1
      IF(ITERATE.EQ.2)GO TO 37
C
C     C O N V E R G E N C F   C R I T E R I A
C
C
      IF(ABS((F2(2)-FPREV)/FPREV).LE.EPSLONW)GO TO 90
      GO TO 37
   90 IF(G1(2).EQ.0.)GO TO 100
      IF(ABS((G2(2)-GPREV)/GPREV).LE.EPSLONW)GO TO 100
      GO TO 37
  100 IF(ABS((THETA2(2)-THETAPR)/THETAPR).LE.EPSLONW)GO TO 110
      GO TO 37
  110 CONTINUE
C     O U T P U T
C
C
C     C O M P U T E   RHOERHO S
C
      DO 80 I=1,NUMETA
      RHOERO2(I)=((1.-ZETAW2)*THETA2(I)+ZETAW2-UES2HE2**2*F2(I)**2-
     1WEDS2HE**2*G2(I)**2)/(1.-UES2HE2**2-WEDS2HE**2)
      RHORHOE(I)=1./RHOERO2(I)
      IF(GEE2.NE.0.)XMMEG(I)=G2(I)*SQRT(RHORHOE(I))
      XMMEF(I)=F2(I)*SQRT(RHORHOE(I))
   80 CONTINUE
      FNCRER1=RHOERO2(1)
      DO 102 I=2,NUMETA
      FNCRER2=RHOERO2(I)
      SUMRER(I)=SUMRER(I-1)+(FNCRER2+FNCRER1)/2.*DELETA(I-1)
      FNCRER1=FNCRER2
      YDL  (I)=(2.*XI2)**XNBAR *1./RERS2*SUMRER(I)/(CAPRS*R2**J*UES2HE2)
  102 CONTINUE
      CALL FTLUP(XI2,XXL,1,NUMX,XITAB,XL)
      IF(XLPR(2).EQ.0.)GO TO 116
      IF(XXL  .GE.XLPR(NPRINT))GO TO 103
```

```
      IF(ICOUNT.EQ.1)GO TO 103
      GO TO 115
  116 IF(NPRINT.EQ.NSTEPS)GO TO 101
      NPRINT=NPRINT+1
      IF(ICOUNT.EQ.1)GO TO 103
      GO TO 115
  101 NPRINT=1
  103 CONTINUE
      DO 91  I=1,NUMETA
      ZETA(I)=THETA2(I)*(1.-ZETAW2)+ZETAW2
   91 CONTINUE
      WRITE(6,111)XI2,XXL
  111 FORMAT(1H0,3HXI=,E20.8,4X,4HX/L=,E20.8)
      MAX=NMAXF+10
      GO TO (117,118,119)IVEG
  117 WRITE(6,112)
  112 FORMAT(1H0,7X,3HETA,11X,3HY/L,12X,1HF,13X,1HV,11X,5HTHETA,9X,
     14HZETA,8X,8HRHO/RHOE,8X,5HM/MEF)
      DO 113  I=1,MAX
      WRITE(6,114)I,ETA(I),YDL(I),F2(I),VA(I),THETA2(I),ZETA(I),
     1RHORHOE(I),XMMEF(I)
  113 CONTINUE
      GO TO 115
  114 FORMAT(I4,9E14.4)
  118 WRITE(6,141)
  141 FORMAT(1H0,7X,3HETA,11X,3HY/L,12X,1HF,11X,6HEPSDMU,8X,5HTHETA,
     19X,4HZETA,8X,8HRHO/RHOE,8X,5HM/MEF)
      DO 142  I=1,MAX
      WRITE(6,114)I,ETA(I),YDL  (I),F2(I),EPSDMU(I),THETA2(I),ZETA(I),
     1RHORHOE(I),XMMEF(I)
  142 CONTINUE
      GO TO 115
  119 WRITE(6,143)
  143 FORMAT(1H0,7X,3HETA,11X,3HY/L,12X,1HF,13X,1HG,11X,5HTHETA,9X,
     14HZETA,8X,8HRHO/RHOE,8X,5HXMBAR,11X,5HXMMEF)
      DO 144  I=1,MAX
      WRITE(6,114)I,ETA(I),YDL  (I),F2(I),G2(I),THETA2(I),ZETA(I),
     1RHORHOE(I),XMBAR2(I),XMMEF(I)
  144 CONTINUE
  115 CONTINUE
C
C     U P D A T E   V A L U E S
C
      XIBAR2=(2.*XI2)**(2.*XNBAR)
      DO 120  I=1,NUMETA
      F1(I)=F2(I)
      G1(I)=G2(I)
      THETA1(I)=THETA2(I)
      RHOERO1(I)=RHOERO2(I)
  120 CONTINUE
      UES2HE1=UES2HE2
      DO 106  I=1,NUMETA
      HDCAPHE(I)=(1.-ZETAW2)*THETA1(I)+ZETAW2-UES2HE1**2*F1(I)**2
     1-WEDS2HE**2*G1(I)**2
  106 CONTINUE
      CALL  CALCM(ZETAW2,          UES2HE1,F1,WEDS2HE,          ZETAE,
     1                    SHE,PR,     XMBAR1,XMCIRC1,XMSTAR1,XMPRIM1,PHIR,
     2HSHE,           RERS2,XI2,XNBAR ,R2,J,
     3DELETA,RURDRUS,      CAPRS,SUMRER,AP,BP,CP,       ABTAB,FCFTAB,NFCFAB,
```

```
     4NMAXG,G1)
      IF(ICOUNT.EQ.1)GO TO 104
      IF(XLPR(2).EQ.0.)GO TO 145
      IF(XXL  .GE.XLPR(NPRINT))GO TO 105
      GO TO 123
  105 NPRINT=NPRINT+1
      GO TO 104
  145 IF(NPRINT.NE.1)GO TO 123
  104 CONTINUE
      XINB=(2.*XI2)**XNBAR
      CFF=PHIR(1)/RERS2*2.*R2**J/XINB*(F2(2)-F2(1))/DELETA(1)*RURDRUS
      IF(GEE2.NE.0.)CFG=PHIR(1)/RERS2*2.*R2**J/XINB*(G2(2)-G2(1))/DELETA
     1(1)*UES2HE1/WEDS2HE*RURDRUS
      IF(GEE2.NE.0.)TAFDTAG=CFF/CFG*(UES2HE1/WEDS2HE)**2
      SUMF=0.0
      FUNCF1=F2(1)*(1.-F2(1))
      DO 92  I=2,NMAXF
      FUNCF2=F2(I)*(1.-F2(I))
      SUMF=SUMF+(FUNCF2+FUNCF1)/2.*(ETA(I)-ETA(I-1))
      FUNCF1=FUNCF2
   92 CONTINUE
      RCURJ=RERS2*CAPRS*UES2HE2*R2**J
      TSDLF=XINB/RCURJ*SUMF
      IF(GEE2.EQ.0.)GO TO 150
      SUMG=0.0
      FUNCG1=G2(1)*(1.-G2(1))
      DO 94  I=2,NMAXG
      FUNCG2=G2(I)*(1.-G2(I))
      SUMG=SUMG+(FUNCG2+FUNCG1)/2.*(ETA(I)-ETA(I-1))
      FUNCG1=FUNCG2
   94 CONTINUE
      TSDLG=XINB/RCURJ*SUMG
  150 CONTINUE
      SMDEL=0.0
      SMTHE=0.0
      FNCDEL1=1.-F2(1)
      FNCTHE1=F2(1)*(1.-F2(1))
      DO 56  I=2,NMAXF
      FNCDEL2=1.-F2(I)
      FNCTHE2=F2(I)*(1.-F2(I))
      SMDEL=SMDEL+(FNCDEL2+FNCDEL1)/2.*(YDL(I)-YDL(I-1))
      SMTHE=SMTHE+(FNCTHE2+FNCTHE1)/2.*(YDL(I)-YDL(I-1))
      FNCDEL1=FNCDEL2
      FNCTHE1=FNCTHE2
   56 CONTINUE
      HISF=SMDEL/SMTHE
      IF(GEE2.EQ.0.)GO TO 151
      SMDEL=0.0
      SMTHE=0.0
      FNCDEL1=1.-G2(1)
      FNCTHE1=G2(1)*(1.-G2(1))
      DO 57  I=2,NMAXG
      FNCDEL2=1.-G2(I)
      FNCTHE2=G2(I)*(1.-G2(I))
      SMDEL=SMDEL+(FNCDEL2+FNCDEL1)/2.*(YDL  (I)-YDL  (I-1))
      SMTHE=SMTHE+(FNCTHE2+FNCTHE1)/2.*(YDL  (I)-YDL  (I-1))
      FNCDEL1=FNCDEL2
      FNCTHE1=FNCTHE2
```

```
 57 CONTINUE
    HISG=SMDEL/SMTHE
151 CONTINUE
    SUMFRR=0.0
    FNCFRR1=1.-F2(1)*RHORHOE(1)
    DO 93 I=2,NMAXF
    FNCFRR2=1.-F2(I)*RHORHOE(I)
    SUMFRR=SUMFRR+(FNCFRR2+FNCFRR1)/2.*(YDL(I)-YDL(I-1))
    FNCFRR1=FNCFRR2
 93 CONTINUE
    DSDLF=SUMFRR
    IF(GEE2.EQ.0.)GO TO 96
    SUMGRR=0.0
    FNCGRR1=1.-G2(1)*RHORHOE(1)
    DO 97 I=2,NMAXG
    FNCGRR2=1.-G2(I)*RHORHOE(I)
    SUMGRR=SUMGRR+(FNCGRR2+FNCGRR1)/2.*(YDL  (I)-YDL  (I-1))
    FNCGRR1=FNCGRR2
 97 CONTINUE
    DSDLG=SUMGRR
 96 CONTINUE
    REX=CAPRS*XXL*UES2HE2*RERS2**2/RURDRUS
    RDXL=REX/XXL
    RETSF=RDXL*TSDLF
    IF(GEE2.NE.0.)RETSG=RDXL*TSDLG
   1*WEDS2HE/UES2HE2
    REDSF=RDXL*DSDLF
    IF(GEE2.NE.0.)REDSG=RDXL*DSDLG
   1*WEDS2HE/UES2HE2
    HEDCPHE=1.-UES2HE2**2-WEDS2HE**2
    QBAR=PHIR(1)*RURDRUS*R2**J*UES2HE2*CAPRS/(PR*XINB)*(ZETA(2)
   1        -ZETAW2)/DELETA(1)
    ST=QBAR/(RERS2*CAPRS*UES2HE2*(FRO*(1.-HEDCPHE)+HEDCPHE-ZETAW2))
    WRITE(6,121)
121 FORMAT(1H0,7X,3HCFF,10X,9HTHETA*/LF,7X,9HDELTA*/LF,9X,3HREX,11X,
   19HRETHETA*F,7X,9HREDELTA*F,9X,4HQBAR,13X,2HST)
    WRITE(6,122)CFF,TSDLF,DSDLF,REX,RETSF,REDSF,QBAR,ST
122 FORMAT(8E16.4)
    WRITE(6,124)
124 FORMAT(1H0,7X,4HHI*F)
    WRITE(6,122)HISF
    IF(GEE2.EQ.0.)GO TO 98
    WRITE(6,131)
131 FORMAT(1H0,7X,3HCFG,10X,9HTAUF/TAUG,6X,9HTHETA*/LG,7X,
   19HDELTA*/LG,7X,9HRETHETA*G,7X,9HREDELTA*G,9X,4HHI*G)
    WRITE(6,122)CFG,TAFDTAG,TSDLG,DSDLG,RETSG,REDSG,HISG
 98 CONTINUE
    ICOUNT=ICOUNT+1
123 CONTINUE
    IF(XISTOP.LE.XI2)GO TO 130
    XI1=XI2
    XIBAR1=XIBAR2
    CAPU1=CAPU2
    CAPP1=CAPP2
    CAPZ1=CAPZ2
    GO TO 40
130 CONTINUE
    STOP
    END
```

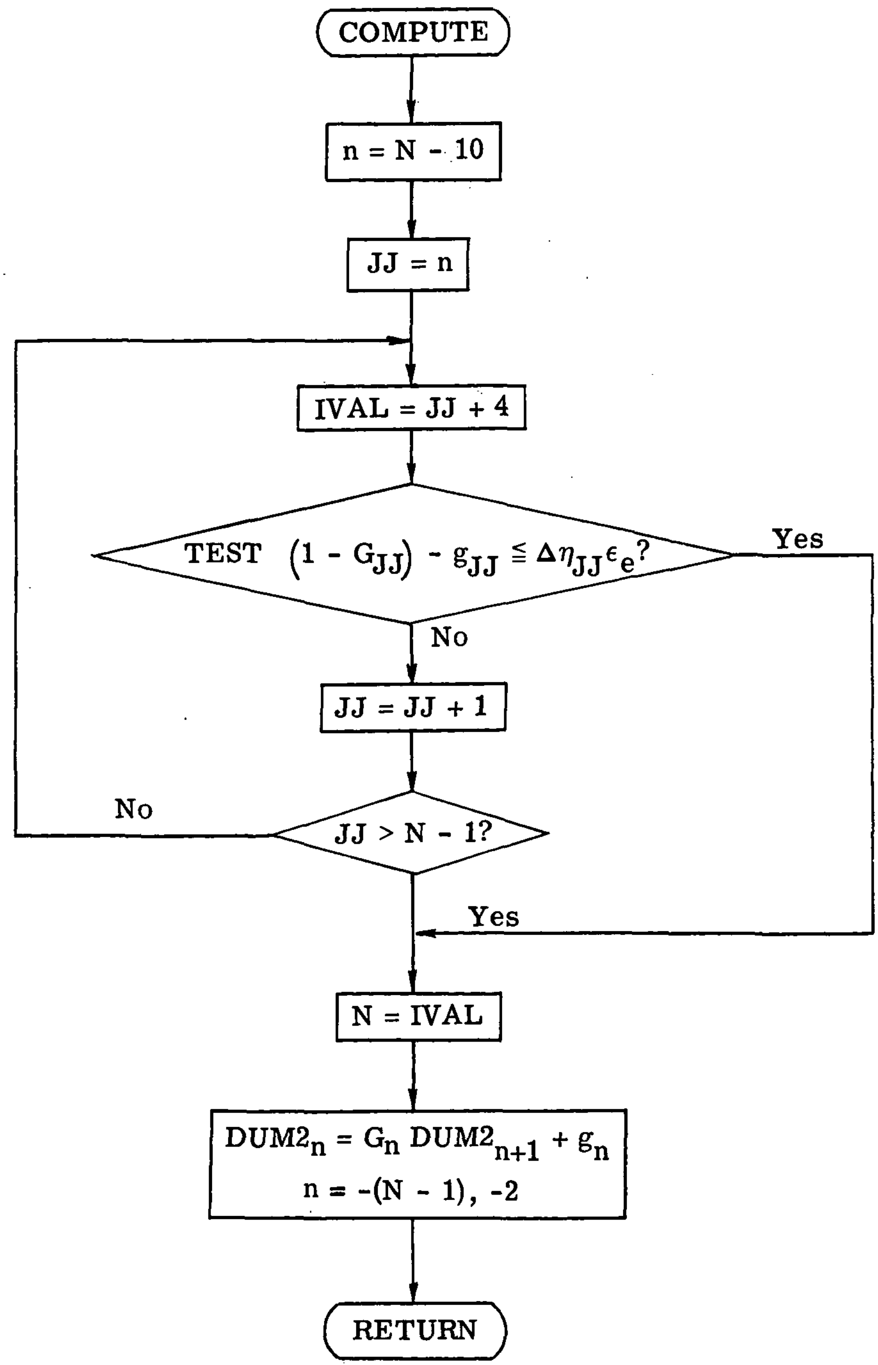
ABCDGS.- Subroutine ABCDGS computes coefficients G_n and g_n for the boundary layer equations by using recursion formulas (32). The flow diagram for subroutine ABCDGS is as follows: (In this flow diagram, A, B, C, D, G, and g are used as general notation for such coefficients as $\bar{A}$, A^*, $\hat{A}$, etc.)



The program listing for subroutine ABCDGS is as follows:

```
      SUBROUTINE ABCDGS(NUMDELE,DELETA,XIBARA,DELXI,FA,VA,CAPMA2,ICHD,
     1F1,XIBCPUA,XIBCPPA,RHOEROA,G1,THETA1,XIBCPZA,THETAA,UES2HEA,
     2XMPRMA2,F2,WEDS2HE,G2,CAPG,SMLG,XK)
      DIMENSION DELETA(350),F1(350),F2(350),FA(350),G1(350),G2(350),
     1THETA1(350),THETA2(350),THETAA(350),VA(350),CAPMA2(350),CAPA(350),
     2CAPB(350),CAPC(350),CAPD(350),RHOEROA(350),XMPRMA2(350),CAPG(350),
     3SMLG(350)
      FCT1=2.*XK/(1.+XK)
      FCT2=2./(1.+XK)
      DO 10 I=2,NUMDELE
      DELDEL2=2.*DELETA(I)*DELETA(I-1)
      XIDELFA=XIBARA/DELXI*FA(I)
      VADLDL2=VA(I)/(2.*(DELETA(I)+DELETA(I-1)))
      TMDLDL2=(CAPMA2(I)*FCT2+CAPMA2(I-1)*FCT1)/DELDEL2
      CAPA(I)=VADLDL2-CAPMA2(I)*FCT2/DELDEL2
      CAPB(I)=XIDELFA+TMDLDL2
      CAPC(I)=-(VADLDL2+CAPMA2(I-1)*FCT1/DELDEL2)
      IF(ICHD.EQ.1)GO TO 20
      IF(ICHD.EQ.2)GO TO 30
      IF(ICHD.EQ.3)GO TO 40
   20 CAPD(I)=-CAPA(I)*F1(I+1)+(XIDELFA-TMDLDL2)*F1(I)-CAPC(I)*F1(I-1)
     1-XIBCPUA*FA(I)**2-XIBCPPA*RHOEROA(I)
      GO TO 10
   30 CAPD(I)=-CAPA(I)*G1(I+1)+(XIDELFA-TMDLDL2)*G1(I)-CAPC(I)*G1(I-1)
      GO TO 10
   40 CAPD(I)=-CAPA(I)*THETA1(I+1)+(XIDELFA-TMDLDL2)*THETA1(I)-CAPC(I)*
     1THETA1(I-1)+XIBCPZA*FA(I)*(THETAA(I)-1.)-UES2HEA**2/DELDEL2*
     2(XMPRMA2(I)*FCT2*(F1(I+1)**2+F2(I+1)**2-F1(I)**2-F2(I)**2)
     3-XMPRMA2(I-1)*FCT1*(F1(I)**2+F2(I)**2-F1(I-1)**2-F2(I-1)**2))
     4-WEDS2HE**2/DELDEL2*(XMPRMA2(I)*FCT2*(G1(I+1)**2+G2(I+1)**2
     5-G1(I)**2-G2(I)**2)-XMPRMA2(I-1)*FCT1*(G1(I)**2+G2(I)**2-G1(I-1)
     6**2-G2(I-1)**2))
   10 CONTINUE
      CAPG(2)=-CAPA(2)/CAPB(2)
      SMLG(2)=CAPD(2)/CAPB(2)
      DO 50 I=3,NUMDELE
      BCG=CAPB(I)+CAPC(I)*CAPG(I-1)
      CAPG(I)=-CAPA(I)/BCG
      SMLG(I)=(CAPD(I)-CAPC(I)*SMLG(I-1))/BCG
   50 CONTINUE
      RETURN
      END
```

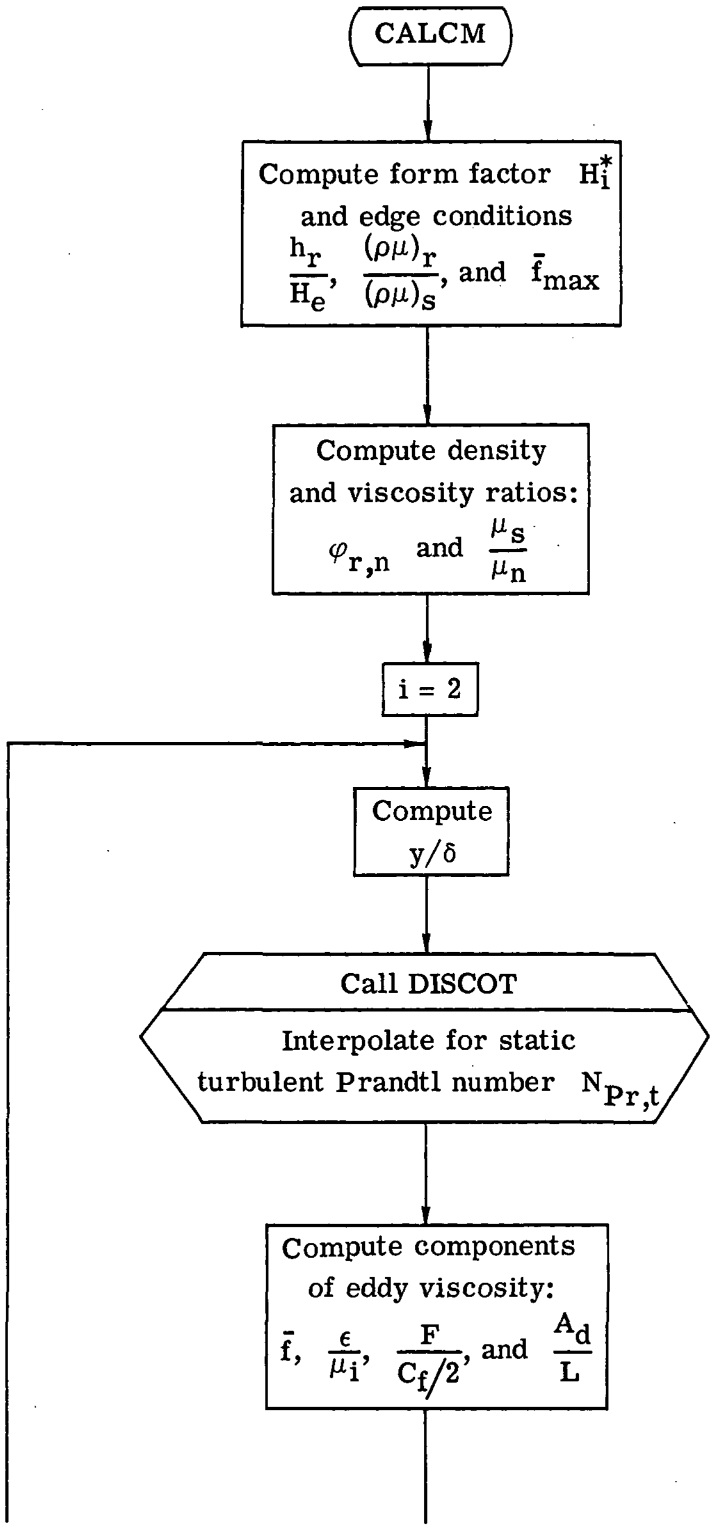
COMPUTE.- Subroutine COMPUTE determines the number of equations (or $\Delta\eta$ steps) to be used and computes new values for the dependent variables by using formula (31), (42), or (43). The flow diagram for subroutine COMPUTE is as follows: (In this flow diagram, G and g are used as general notation for such coefficients as $\bar{G}$, $\bar{g}$, G^*, etc.)

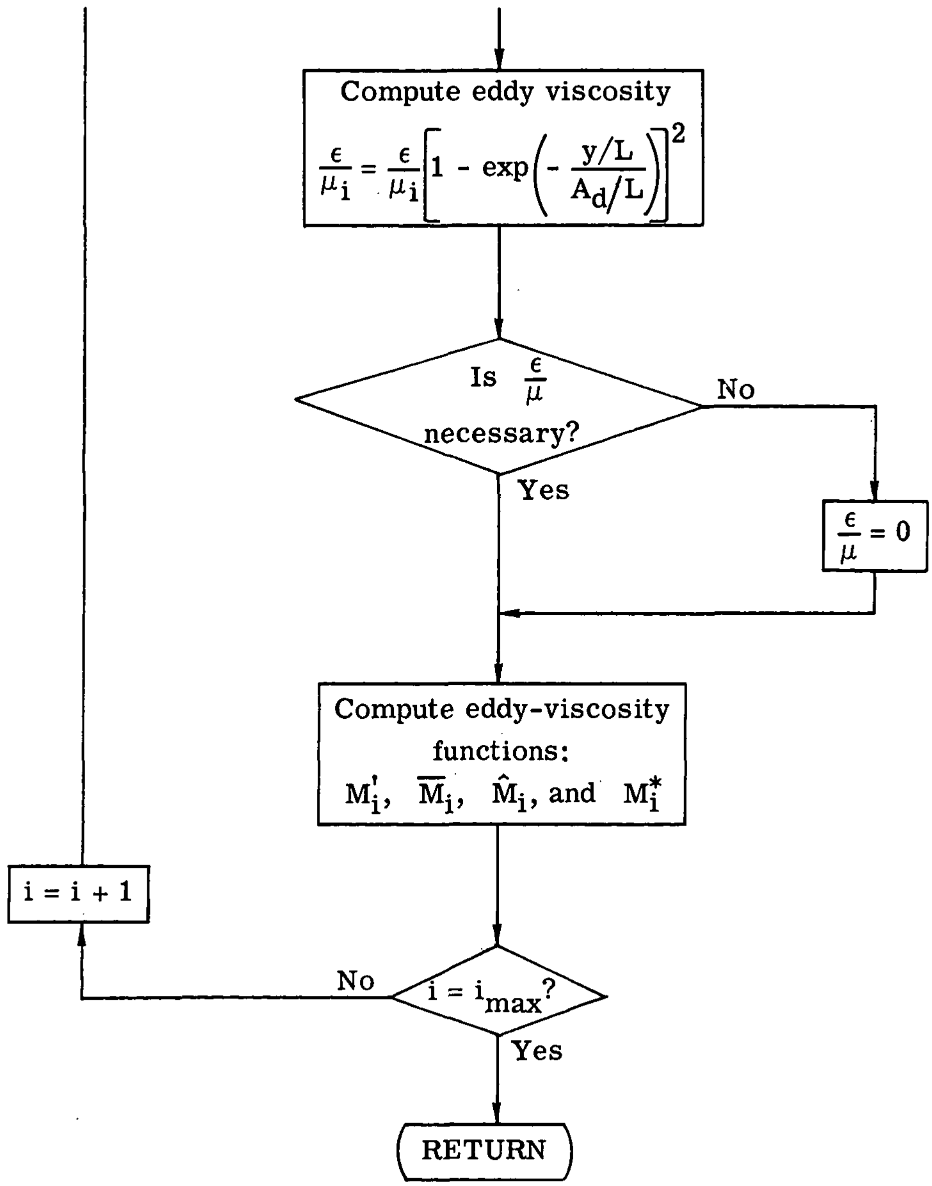


The program listing for subroutine COMPUTE is as follows:

```
      SUBROUTINE COMPUTE(NMAX,TEST,NUMDELE,   DELETA,EPSLONE,CAPG,SMLG,
     1DUM2    )
      DIMENSION CAPG(350),SMLG(350),DUM2(350),DELETA(350)
C
C     T E S T - F I N D I N G   E D G E
C
      KMAX=NMAX-10
      DO 10 JJ=KMAX,NUMDELE
      TSTVAL=EPSLONE*DELETA(JJ)
      CKVAL=TEST*(1.0-CAPG(JJ))-SMLG(JJ)
      IVAL=JJ+4
      IF(ABS(CKVAL).LE.ABS(TSTVAL))GO TO 20
   10 CONTINUE
   20 NMAX=IVAL
C
C     C O M P U T E   D U M 2 S
C
   21 CONTINUE
      NBACK=-(NMAX-1)
      M2=-2
      DO 30 NF=NBACK,M2
      KF=IABS(NF)
      DUM2(KF)=CAPG(KF)*DUM2(KF+1)+SMLG(KF)
   30 CONTINUE
      RETURN
      END
```

CALCM.- Subroutine CALCM computes the viscosity functions. The flow diagram for this subroutine is as follows:





Compute eddy viscosity
$\frac{\epsilon}{\mu_i} = \frac{\epsilon}{\mu_i}\left[1 - \exp\left(-\frac{y/L}{A_d/L}\right)\right]^2$
Is $\frac{\epsilon}{\mu}$ necessary?
No
Yes
$\frac{\epsilon}{\mu} = 0$
Compute eddy-viscosity functions:
M_i', $\overline{M}_i$, $\hat{M}_i$, and M_i^*
$i = i + 1$
No
$i = i_{max}$?
Yes
RETURN

The program listing for subroutine CALCM is as follows:

```
      SUBROUTINE CALCM(ZETAW,UEDS2HE,F2,WEDS2HE,ZETAE,
     1                SHE,PR,     XMBAR,XMCIRC,XMSTAR,XMPRIM,PHIR,HSHE,
     2         RERS,XI,XNBAR,R,J,                               DELETA,
     3RURDRUS,     CAPRS,SUMRER,AP,BP,CP,     ABTAB,FCFTAB,NFCFAB,NMAXG,
     4G2)
      DIMENSION F2(350),G2(350),XMBAR(350),XMCIRC(350),
     1XMSTAR(350),XMPRIM(350),PHIR(350),DELETA(350),SUMRER(350),
     2ABTAB(20),FCFTAB(20),EMUSDMU(350)
      COMMON/EPSDMU/EPSDMU(350)
      COMMON/TABLE1/RHOERO2(350)
      COMMON/TABLE2/YDL(350),RHORHOE(350)
      COMMON/THREE/NUMETA,NMAXF
      COMMON/FEB12/VWA
      COMMON/FEB11/PRTTAB(20),YDDPRT(20),NYP
      COMMON/HDCAPHE/HDCAPHE(350),GEE2
      COMMON/MUUSE/IUSEEMU,MPWEMU
      COMMON/FBAR/FBARTAB(20),IFBLU,YDDFB(20),NFBY
      COMMON/IWLDMP/IWLDMP
      HRDCPHE=ZETAE-UEDS2HE**2-WEDS2HE**2
      RURDRUS=SQRT(HRDCPHE/HSHE)*(HSHE+SHE)/(HRDCPHE+SHE)
     1*RERS*HRDCPHE
     2/HSHE
      RUSDRUR=1./RURDRUS
      SMRER=0.0
      FNCRER1=RHOERO2(1)
      IF(GEE2.NE.0.)GO TO 24
      DO 25 I=2,NMAXF
      FNCRER2=RHOERO2(I)
      SMRER=SMRER+(FNCRER2+FNCRER1)/2.*DELETA(I-1)
      IF(F2(I).GE..995)GO TO 23
      FNCRER1=FNCRER2
   25 CONTINUE
      GO TO 23
   24 CONTINUE
      DO 22 I=2,NMAXG
      FNCRER2=RHOERO2(I)
      SMRER=SMRER+(FNCRER2+FNCRER1)/2.*DELETA(I-1)
      IF(G2(I).GE..995)GO TO 23
      FNCRER1=FNCRER2
   22 CONTINUE
   23 CONTINUE
      SMDEL=0.0
      SMTHE=0.0
      IF(GEE2.NE.0.)GO TO 26
      FNCDEL1=1.-F2(1)
      FNCTHE1=F2(1)*(1.-F2(1))
      DO 27 I=2,NMAXF
      FNCDEL2=1.-F2(I)
      FNCTHE2=F2(I)*(1.-F2(I))
      SMDEL=SMDEL+(FNCDEL2+FNCDEL1)/2.*(YDL(I)-YDL(I-1))
      SMTHE=SMTHE+(FNCTHE2+FNCTHE1)/2.*(YDL(I)-YDL(I-1))
      FNCDEL1=FNCDEL2
      FNCTHE1=FNCTHE2
   27 CONTINUE
      GO TO 28
```

```
26 CONTINUE
   FNCDEL1=1.-G2(1)
   FNCTHE1=G2(1)*(1.-G2(1))
   DO 11  I=2,NMAXG
   FNCDEL2=1.-G2(I)
   FNCTHE2=G2(I)*(1.-G2(I))
   SMDEL=SMDEL+(FNCDEL2+FNCDEL1)/2.*(YDL  (I)-YDL  (I-1))
   SMTHE=SMTHE+(FNCTHE2+FNCTHE1)/2.*(YDL  (I)-YDL  (I-1))
   FNCDEL1=FNCDEL2
   FNCTHE1=FNCTHE2
11 CONTINUE
28 CONTINUE
   HIS=SMDEL/SMTHE
   FBARMAX=AP+BP*HIS+CP*HIS*HIS
   DO 12  I=1,NUMETA
   PHIR(I)=SQRT(HDCAPHE(I)/HRDCPHE)*(HRDCPHE+SHE)/(HDCAPHE(I)+SHE)
   EMUSDMU(I)=RUSDRUR*RHORHOE(I)*RERS/PHIR(I)
12 CONTINUE
   DO 10  I=1,NUMETA
   YDD=SUMRER(I)/SMRER
   CALL  DISCOT(YDD,YDD,YDDPRT,PRTTAB,PRTTAB,-11,NYP,0,PRT)
   IF(I.EQ.1)GO TO 32
   IF(IFBLU.EQ.1)GO TO 13
   IF(YDD.LE..1)FBAR=.4*YDD
   IF(YDD.GT..1.AND.YDD.LE..3)FBAR=.04+((YDD-.1)/.2)*(FBARMAX-.04)
   IF(YDD.GT..3)FBAR=FBARMAX
   GO TO 14
13 CALL  DISCOT(YDD,YDD,YDDFB,FBARTAB,FBARTAB,-11,NFBY,0,FBAR)
14 CONTINUE
   IF(I.EQ.NUMETA)GO TO 32
   IF(GEE2.NE.0.)GO TO 29
   EPSDMU(I)=(2.*XI)**XNBAR/R**J* FBAR**2*EMUSDMU(I)*RHORHOE(I)**2*
  1SMRER**2
  1*ABS((F2(I+1)-F2(I-1))/(DELETA(I)+DELETA(I-1)))
   GO TO 34
29 CONTINUE
   EPSDMU(I)=(2.*XI)**XNBAR/R**J*FBAR**2*EMUSDMU(I)*RHORHOE(I)**2
  1*SMRER**2
  1*SQRT(ABS(((F2(I+1)-F2(I-1))/(DELETA(I)+DELETA(I-1)))**2
  1+((G2(I+1)-G2(I-1))/(DELETA(I)+DELETA(I-1)))**2
  1*(WEDS2HE/UEDS2HE)**2))
34 CONTINUE
   IF(IWLDMP.EQ.1.AND.I.NE.2)GO TO 15
   IF(I.NE.2)GO TO 33
   IF(GEE2.NE.0.)GO TO 40
   FDCFD2=((RHORHOE(1)*VWA)/(PHIR(1)/RERS*(2.*R**J)/((2.*XI)**XNBAR)*
  1F2(2)/DELETA(1)*RURDRUS))*2.
   GO TO 41
40 FDCFD2=((RHORHOE(1)*VWA)/(PHIR(1)/RERS*(2.*R**J)/((2.*XI)**XNBAR)*
  1SQRT((F2(2)/DELETA(1))**2+(G2(2)/DELETA(1))**2*(WEDS2HE/UEDS2HE)**
  22)*RURDRUS))*2.
  2*UEDS2HE/SQRT(WEDS2HE**2+UEDS2HE**2)
41 CONTINUE
   CALL  FTLUP(FDCFD2,AB,1,NFCFAB,FCFTAB,ABTAB)
   IF(GEE2.NE.0.)GO TO 35
15 CONTINUE
   IF(IWLDMP.EQ.1)GO TO 16
   ADL= AB*SQRT((2.*XI)**XNBAR)/(CAPRS*SQRT(EMUSDMU(1))*RHORHOE(1)
```

```
      1*RERS
      1*UEDS2HE*SQRT(R**J)*SQRT(ABS(F2(2))/DELETA(1)))
       GO TO 36
16     ADL=AB*SQRT((2.*XI)**XNBAR)*SQRT(EMUSDMU(1))/(CAPRS*EMUSDMU(I)
      1*SQRT(RHORHOE(1))*SQRT(RHORHOE(I))*RERS
      1*UEDS2HE*SQRT(R**J)*SQRT(ABS(F2(2))/DELETA(1)))
       GO TO 36
35     CONTINUE
       ADL= AB*SQRT((2.*XI)**XNBAR)/(CAPRS*SQRT(EMUSDMU(1))*RHORHOE(1)
      1*RERS
      1*UEDS2HE*SQRT(R**J)
      1*((F2(2)/DELETA(1))**2+(G2(2)/DELETA(1))**2*(WEDS2HE/UEDS2HE)**2)
      1**.25)
36     CONTINUE
33     EPSDMU(I)=EPSDMU(I)*(1.-EXP(-YDL  (I)/ADL))**2
51     IF(EPSDMU(I).LT.0.)EPSDMU(I)=0.
       GO TO 31
32     EPSDMU(I)=0.0
31     CONTINUE
       IF(IUSEEMU.EQ.0)EPSDMU(I)=0.
       XMBAR(I)=PHIR(I)*(1.+EPSDMU(I))
       XMCIRC(I)=XMBAR(I)
       XMSTAR(I)=PHIR(I)/PR*(1.+EPSDMU(I)*PR/PRT)
       IF( MPWEMU.EQ.1)GO TO 37
       XMPRIM(I)=PHIR(I)/PR*(1.-PR)/(1.-ZETAW)
       GO TO 10
37     XMPRIM(I)=PHIR(I)/PR*(1.-PR)/(1.-ZETAW)*(1.+EPSDMU(I)*PR/PRT*(1.-P
      1RT)
      1/(1.-PR))
10     CONTINUE
       RETURN
       END
```

USAGE

The program is run on the Control Data 6000 series computer under the SCOPE 3.0 operating system. Minimum machine requirements are 75 000 octal locations of core storage. The time required to calculate a grid point is approximately 0.002 second per iteration. Each x-step typically uses three iterations. The restrictions and usual values for input quantities are given in their description.

Input Description

The FORTRAN NAMELIST capability is used for data input with NAM1 as the NAMELIST name. The maximum allowable dimension appears following the variable name.

NUMETA — maximum number of steps in η direction, 350 maximum

NMAXF — initial guess at number of η steps to outer edge of F profile, generally out to $F = 0.999$

Note: The procedure to obtain NMAXF is to (a) estimate $\Delta\eta$, making sure that there are at least four steps in linear portion of profile near wall, (b) choose a K value, (c) estimate an η_e value, and (d) solve equation (24) for $N = \text{NMAXF}$. Generally, $\text{NUMETA} = \text{NMAXF} + 50$ has been used successfully.

NMAXG — initial guess at number of η steps to outer edge of g profile, generally out to $g = 0.999$

DELETA — value of $\Delta\eta$ nearest wall $\left(\text{that is, } \Delta\eta_1\right)$; there must be at least four steps in linear portion of profile near wall

XK — $\Delta\eta_n / \Delta\eta_{n-1}$ ratio, generally taken as 1.02

XI0 — value of ξ at input station

DELXI0 — initial $\Delta\xi$, generally $\left(10^{-3}\right)\left(\xi_o\right)$

XITEST — value of ξ where $\Delta\xi$ is increased by a factor of 10

XISTØP — value of ξ where solution is to be terminated

FTAB	(100)	input u/u_e profile (values must correspond to YL table); there must be at least four points in linear region near wall
ETATAB	(100)	input η table, used when input profiles are known as $f(\eta)$ rather than $f(y/L)$ (it can be omitted as input if not required)
VWTAB	(75)	v_w/u_e for axisymmetric or two-dimensional flow, v_w/w_e for swept-cylinder problem (values must correspond to XL table)
EPSLØNE		accuracy criteria for $\partial(F,g,\theta)/\partial\eta$ at outer edge of profile, generally set equal to 0.05 for turbulent flows and 0.0001 for laminar flows
EPSLØNW		convergence criteria for iterations on F, g, and θ profiles; allowable percent change in wall slope between iterations, 0.01 generally used
GTAB	(100)	input w/w_e profile (values must correspond to YL table and must be zero if not used)
UEDSTAB	(75)	$u_e/\sqrt{2H_e}$ table (values must correspond to XL table)
WEDS2HE		$w_e/\sqrt{2H_e}$
PR		molecular Prandtl number
ZETWTAB	(75)	ζ_w table (values must correspond to XL table)
XNBAR		$\bar{n}$, generally taken as 0.5 for laminar flows and 0.8 for turbulent flows
RERSTAB	(75)	ρ_e/ρ_s table (values must correspond to XL table)
CAPRS		reference Reynolds number, $\rho_s\sqrt{2H_e}\,L/\mu_s$; the subscript "s" must be taken as isentropic stagnation conditions except for swept-cylinder problems in which "s" is taken as stagnation-line values
RTAB	(75)	r/L table (values must correspond to XL table)

J		body shape index ($j = 0$ for two-dimensional flows; $j = 1$ for axisymmetric flows)
RHØTAB	(100)	input ρ_e/ρ table, can be used instead of input ζ profile, values must correspond to YL table (can be omitted as input if not required)
SHE		Sutherland's constant, $S/H_e = 202/T_S$, where T_S is in ^{O}R for air
ZETATAB	(100)	input ζ profile, values must correspond to YL table (can be omitted as input if not required)
ITHETA		code for input temperature profile (ITHETA = 1 if RHØTAB used; ITHETA = 2 if ZETATAB used)
HSHE		yaw parameter, $1 - \frac{w_e^2}{2H_e}$
XL	(75)	table of x/L values, first entry must equal x_o value
NUMX		number of values in XL table, 75 maximum
YL	(100)	table of y/L values for initial profiles, first entry must equal zero
NUMY		number of values in YL table, 100 maximum
X0		initial value of x/L, must not equal zero
ØL		reference length, given in feet
DUDXTAB	(75)	$d\left(u_e/\sqrt{2H_e}\right)/d(x/L)$ table, values must correspond to XL table
DZDXTAB	(75)	$d\zeta_w/d(x/L)$ table, values must correspond to XL table
FR0		temperature recovery factor, generally 0.85 for laminar flows and 0.89 for turbulent flows; used to compute Stanton number
NSTEPS		number of ξ steps between profile printouts (can be omitted as input if XLPR table is used to designate printout)

AP BP CP		constants A, B, and C for $\bar{f}_{max}$ (see eq. (56a))	
ABTAB	(20)	table of A_b values in Van Driest damping function (eq. (59)), input as f(FCFTAB)	Generally use values from sample case shown
FCFTAB	(20)	$\frac{\rho_w v_w}{\rho_e u_e} \frac{2}{C_f}$ values corresponding to ABTAB values	
NFCFAB		number of ABTAB values in table, 20 maximum	
PRTTAB	(20)	turbulent Prandtl number table, input as $f(y/\delta)$	
YDDPRT	(20)	y/δ values corresponding to PRTTAB, values must start at zero	
NYP		number of values in PRTTAB table, 20 maximum	
XLPR	(30)	x/L values where profile printout is required (can be omitted as input if NSTEPS is used)	
IVEG		= 1 for printout of V profile = 2 for ϵ/μ profile = 3 for g profile	
INIT		= 0 for $\Delta\xi_{initial}$ = DELXI0 = 1 for small $\Delta\xi_{initial}$; after 30 ξ steps, $\Delta\xi$ is set back to DELXI0 (used to smooth input profile)	
IUSEEMU		= 0 for laminar ($\epsilon/\mu \equiv 0$) solution = 1 for turbulent solution	
MPWEMU		= 0 for PRTTAB = $N_{Pr,T}$ values = 1 for PRTTAB = $N_{Pr,t}$ values	
FBARTAB	(20)	$\bar{f}$ values corresponding to YDDFB values	
IFBLU		= 0 for computing $\bar{f}$ = 1 for table lookup of $\bar{f}$	

YDDFB	(20)	y/δ values corresponding to FBARTAB
NFBY		number of FBARTAB values in table
IWLDMP		= 0 for using wall properties in wall damping function = 1 for using local properties in wall damping function (should be set equal to 0 for swept-leading-edge situation)

Output Description

The output of program D2630 consists of printing only. The main program prints the NAMELIST input and the output described below. If IVEG = 2 in the input, the ϵ/μ profile will be printed as in the sample output. For IVEG = 1, the V profile is printed in the place of EPSDMU, whereas IVEG = 3 prints the g profile. The frequency of the output is controlled by one of two input quantities. If a table of x/L values is given for XLPR, printout will appear for the computation step nearest each value. Otherwise, NSTEPS is used and denotes the number of ξ steps between profile printouts.

XI	ξ
X/L	x/L
ETA	η
Y/L	y/L
F	F
EPSDMU	ϵ/μ (for IVEG = 2); V (for IVEG = 1); G (for IVEG = 3)
THETA	θ
ZETA	ζ
RHO/RHOE	ρ/ρ_e
M/MEF	$(M/M_e)_F$
CFF	wall-skin-friction coefficient for chordwise profile

THETA*/LF	$(\theta^*/L)_F$
DELTA*/LF	$(\delta^*/L)_F$
REX	$\frac{\rho_e u_e x}{\mu_e}$
RETHETA*F	$\frac{\rho_e u_e \theta^*_F}{\mu_e}$
REDELTA*F	$\frac{\rho_e u_e \delta^*_F}{\mu_e}$
QBAR	heat-transfer parameter $\dot{q}_w L/(\mu_s H_e)$
ST	Stanton number, computed using input FR0
HI*F	$(H^*_i)_F$

If a spanwise velocity profile "g" is used, the following output also appears:

CFG	wall-skin-friction coefficient for spanwise profile
TAUF/TAUG	ratio of chordwise to spanwise shear at wall
THETA*/LG	$(\theta^*/L)_g$
DELTA*/LG	$(\delta^*/L)_g$
RETHETA*G	$\frac{\rho_e u_e \theta^*_g}{\mu_e}$
REDELTA*G	$\frac{\rho_e u_e \delta^*_g}{\mu_e}$
HI*G	$(H^*_i)_g$

Sample Case

The listing of the input data for the McLafferty-Barber Mach 3 adverse pressure gradient flow sample case, as described in reference 16, is as follows:

```
$NAM1

NUMETA=230,

NMAXF=100,

NMAXG=1,

DELETA=2.E-2,

XK=1.02,

XI0=1.6E6,

DELXI0=.15E3,

XITEST=1.E15,

XISTOP=1.63705E6,

ETAB=0.,.0858,.1716,.2574,.3432,.41,.5,.58,.627,.664,.683,.718,.743,

.789,.826,.872,.912,.945,.964,.978,.988,.994,.999,.9995,.9997,.9999,1.,

VWTAB=8*0.,

EPSLONE=.5E-3,

EPSLONW=.01,

GTAB=27*0.,

UEDSTAB=.8015,.790,.775,.754,.74,.701,.663,.625,

WEDS2HE=0.,

PR=.7,

ZETWTAB=8*.93,

XNBAR=.5,

RERSTAB=.0762,.0872,.101,.1218,.1377,.1841,.2352,.290,

CAPRS=1.08E6,
```

```
RTAB=8*1.,
J=0,
SHF=.331,
ZETATAB=.93,.936,.942,.948,.954,.9587,.965,.9706,.9739,.9765,.9778,
.9802,.9820,.9852,.9878,.9910,.9938,.9962,.9975,.9985,.9992,.9996,.99993,
.99996,.99998,.99999,1.,
ITHETA=2,
HSHF=1.,
XL=.001,.3,.63,1.,1.25,1.88,2.51,3.14,
NUMX=8,
YL=0.,.001,.002,.003,.004,.006,.01,.015,.020,.025,.030,.040,.050,.07,
.09,.12,.15,.18,.2,.22,.24,.26,.28,.3,.32,.34,1.,
NUMY=27,
X0=.001,
OL=.0834,
DUDXTAB=-.0295,-.0415,-.05,-.0575,-.06,-.0615,-.06,-.058,
DZDXTAB=8*0.,
FR0=.89,
NSTEPS=1,
AP=.1,
BP=0.,
CP=0.,
ABTAB(1)=60.,26.,18.5,14.8,12.5,11.2,10.,9.1,6.,3.8,3.,
FCFTAB(1)=-1.,0.,1.,2.,3.,4.,5.,6.,12.,24.,100.,
NFCFAB=11,
PRTTAB=4*.9,
YDDPRT=0.,.5,1.,15.,
```

```
NYP=4,

XLPR=.3,.63,1.,1.25,1.88,2.51,3.14,

IVEG=2,

INIT=0,

IUSEEMU=1,

MPWEMU=1,

IFBLU=0,

IWLDMP=0,

$
```

The sample output for this case is as follows:

```
XI=        1.60900000E+06     X/L=        3.02740161E-01
```

	ETA	Y/L	F	EPSDMU	THETA	ZETA	RHO/RHOE	M/MEF
1	0.	0.	0.	0.	0.	9.3000E-01	4.0441E-01	0.
2	2.0000E-02	1.1904E-03	5.6497E-02	1.0750E-03	1.2946E-02	9.3091E-01	4.0488E-01	3.5949E-02
3	4.0400E-02	2.4009E-03	1.1945E-01	1.8160E-02	4.6906E-02	9.3328E-01	4.0686E-01	7.6192E-02
4	6.1208E-02	3.6268E-03	1.8632E-01	9.1657E-02	1.0450E-01	9.3732E-01	4.1074E-01	1.1941E-01
5	8.2432E-02	4.8626E-03	2.5176E-01	2.6789E-01	1.7880E-01	9.4252E-01	4.1651E-01	1.6248E-01
6	1.0408E-01	6.1038E-03	3.1059E-01	5.6683E-01	2.5572E-01	9.4790E-01	4.2367E-01	2.0216E-01
7	1.2616E-01	7.3477E-03	3.6076E-01	9.8223E-01	3.2488E-01	9.5274E-01	4.3153E-01	2.3699E-01
8	1.4869E-01	8.5932E-03	4.0276E-01	1.5030E+00	3.8302E-01	9.5681E-01	4.3957E-01	2.6703E-01
9	1.7166E-01	9.8408E-03	4.3796E-01	2.1211E+00	4.3073E-01	9.6015E-01	4.4748E-01	2.9297E-01
10	1.9509E-01	1.1091E-02	4.6771E-01	2.8311E+00	4.6980E-01	9.6289E-01	4.5510E-01	3.1552E-01
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
115	8.5592E+00	2.7093E-01	9.9940E-01	3.0451E+00	9.9948E-01	9.9996E-01	9.9811E-01	9.9846E-01
116	8.7503E+00	2.7554E-01	9.9950E-01	1.7780E+00	9.9951E-01	9.9997E-01	9.9842E-01	9.9871E-01
117	8.9453E+00	2.8025E-01	9.9956E-01	1.3587E+00	9.9957E-01	9.9997E-01	9.9863E-01	9.9888E-01
118	9.1443E+00	2.8504E-01	9.9962E-01	1.2634E+00	9.9964E-01	9.9997E-01	9.9881E-01	9.9903E-01
119	9.3471E+00	2.8993E-01	9.9968E-01	1.2481E+00	9.9971E-01	9.9998E-01	9.9900E-01	9.9918E-01
120	9.5541E+00	2.9492E-01	9.9974E-01	1.2394E+00	9.9975E-01	9.9998E-01	9.9919E-01	9.9934E-01
121	9.7652E+00	3.0001E-01	9.9980E-01	1.2665E+00	9.9980E-01	9.9999E-01	9.9938E-01	9.9949E-01
122	9.9805E+00	3.0520E-01	9.9987E-01	8.5618E-01	9.9987E-01	9.9999E-01	9.9959E-01	9.9966E-01
123	1.0200E+01	3.1049E-01	9.9989E-01	2.4930E-01	9.9986E-01	9.9999E-01	9.9966E-01	9.9972E-01
124	1.0424E+01	3.1589E-01	9.9989E-01	4.8238E-02	9.9986E-01	9.9999E-01	9.9967E-01	9.9973E-01
125	1.0653E+01	3.2139E-01	9.9989E-01	5.9639E-02	9.9986E-01	9.9999E-01	9.9968E-01	9.9973E-01
126	1.0886E+01	3.2701E-01	9.9990E-01	8.9812E-02	9.9986E-01	9.9999E-01	9.9970E-01	9.9975E-01
127	1.1123E+01	3.3273E-01	9.9990E-01	4.6930E-02	9.9986E-01	9.9999E-01	9.9971E-01	9.9976E-01
128	1.1366E+01	3.3857E-01	9.9991E-01	1.7555E-02	9.9986E-01	9.9999E-01	9.9971E-01	9.9976E-01
129	1.1613E+01	3.4453E-01	9.9991E-01	1.7555E-02	9.9987E-01	9.9999E-01	9.9972E-01	9.9976E-01
130	1.1865E+01	3.5061E-01	9.9991E-01	1.7555E-02	9.9987E-01	9.9999E-01	9.9972E-01	9.9977E-01
131	1.2123E+01	3.5681E-01	9.9991E-01	1.7555E-02	9.9987E-01	9.9999E-01	9.9972E-01	9.9977E-01
132	1.2385E+01	3.6313E-01	9.9991E-01	1.7555E-02	9.9987E-01	9.9999E-01	9.9973E-01	9.9977E-01
133	1.2653E+01	3.6958E-01	9.9991E-01	1.7556E-02	9.9987E-01	9.9999E-01	9.9973E-01	9.9978E-01
134	1.2926E+01	3.7616E-01	9.9991E-01	1.7556E-02	9.9987E-01	9.9999E-01	9.9973E-01	9.9978E-01
135	1.3204E+01	3.8287E-01	9.9991E-01	1.7556E-02	9.9988E-01	9.9999E-01	9.9974E-01	9.9978E-01
136	1.3488E+01	3.8971E-01	9.9991E-01	1.7556E-02	9.9988E-01	9.9999E-01	9.9974E-01	9.9978E-01
137	1.3778E+01	3.9669E-01	9.9992E-01	1.7556E-02	9.9988E-01	9.9999E-01	9.9974E-01	9.9979E-01

CFF	THETA*/LF	DELTA*/LF	REX	RETHETA*F	REDELTA*F	QBAR	ST
1.2057E-03	1.7192E-02	8.9293E-02	5.1939E+04	2.9495E+03	1.5319E+04	1.0289E+00	1.0078E-02

HI*F
1.5292E+00

Langley Research Center,

National Aeronautics and Space Administration,

Hampton, Va., February 4, 1971.

APPENDIX

LANGLEY LIBRARY SUBROUTINES

Subroutine FTLUP

Language: FORTRAN

Purpose: Computes y = F(x) from a table of values using first- or second-order interpolation. An option to give y a constant value for any x is also provided.

Use: CALL FTLUP(X, Y, M, N, VARI, VARD)

X The name of the independent variable x.

Y The name of the dependent variable y = F(x).

M The order of interpolation (an integer)
M = 0 for y a constant. VARD(I) corresponds to VARI(I) for I = 1, 2, . . ., N. For M = 0 or N ≦ 1, y = F(VARI(1)) for any value of x. The program extrapolates.
M = 1 or 2. First or second order if VARI is strictly increasing (not equal).
M = -1 or -2. First or second order if VARI is strictly decreasing (not equal).

N The number of points in the table (an integer).

VARI The name of a one-dimensional array which contains the N values of the independent variable.

VARD The name of a one-dimensional array which contains the N values of the dependent variable.

Restrictions: All the numbers must be floating point. The values of the independent variable x in the table must be strictly increasing or strictly decreasing. The following arrays must be dimensioned by the calling program as indicated: VARI(N), VARD(N).

Accuracy: A function of the order of interpolation used.

References: (a) Nielsen, Kaj L.: Methods in Numerical Analysis. The Macmillan Co., c.1956, pp. 87-91.
(b) Milne, William Edmund: Numerical Calculus. Princeton Univ. Press, c.1949, pp. 69-73.

Storage: 430_8 locations.

Error condition: If the VARI values are not in order, the subroutine will print TABLE BELOW OUT OF ORDER FOR FTLUP AT POSITION xxx TABLE IS STORED IN LOCATION xxxxxx (absolute). It then prints the contents of VARI and VARD, and STOPS the program.

Subroutine date: September 12, 1969.

Subroutine DISCOT

Language: FORTRAN

Purpose: DISCOT performs single or double interpolation for continuous or discontinuous functions. Given a table of some function y with two independent variables, x and z, this subroutine performs K_xth- and K_zth-order interpolation to calculate the dependent variable. In this subroutine all single-line functions are read in as two separate arrays and all multi-line functions are read in as three separate arrays; that is,

$$x_i \qquad (i = 1, 2, \ldots, L)$$

$$y_j \qquad (j = 1, 2, \ldots, M)$$

$$z_k \qquad (k = 1, 2, \ldots, N)$$

Use: CALL DISCOT (XA, ZA, TABX, TABY, TABZ, NC, NY, NZ, ANS)

XA The x argument

ZA The z argument (may be the same name as x on single lines)

TABX A one-dimensional array of x values

TABY A one-dimensional array of y values

TABZ A one-dimensional array of z values

NC A control word that consists of a sign (+ or -) and three digits. The control word is formed as follows:

(1) If NX = NY, the sign is negative. If NX ≠ NY, then NX is computed by DISCOT as NX = NY/Nz, and the sign is positive and may be omitted if desired.

(2) A one in the hundreds position of the word indicates that no extrapolation occurs above z_{max}. With a zero in this position, extrapolation occurs when $z > z_{max}$. The zero may be omitted if desired.

(3) A digit (1 to 7) in the tens position of the word indicates the order of interpolation in the x-direction.

(4) A digit (1 to 7) in the units position of the word indicates the order of interpolation in the z-direction.

NY The number of points in y array

NZ The number of points in z array

ANS The dependent variable y

The following programs will illustrate various ways to use DISCOT:

CASE I:

Given $y = f(x)$
NY = 50
NX (number of points in x array) = NY
Extrapolation when $z > z_{max}$
Second-order interpolation in x-direction
No interpolation in z-direction
Control word = -020

```
  DIMENSION TABX (50), TABY (50)
1 FORMAT (8E 9.5)
  READ (5,1) TABX, TABY
  READ (5,1) XA
  CALL DISCOT (XA, XA, TABX, TABY, TABY, -020, 50, 0, ANS)
```

CASE II:

Given $y = f(x,z)$
NY = 800
NZ = 10
NX = NY/NZ (computed by DISCOT)
Extrapolation when $z > z_{max}$
Linear interpolation in x-direction
Linear interpolation in z-direction
Control word = 11

```
  DIMENSION TABX (800), TABY (800), TABZ (10)
1 FORMAT (8E 9.5)
  READ (5,1) TABX, TABY, TABZ
  READ (5,1) XA, ZA
  CALL DISCOT (XA, ZA, TABX, TABY, TABZ, 11, 800, 10, ANS)
```

CASE III:

Given $y = f(x,z)$
NY = 800
NZ = 10
NX = NY
Extrapolation when $z > z_{max}$
Seventh-order interpolation in x-direction
Third-order interpolation in z-direction
Control word = -73

```
  DIMENSION TABX (800), TABY (800), TABZ (10)
1 FORMAT (8E 9.5)
  READ (5,1) TABX, TABY, TABZ
  READ (5,1) XA, ZA
  CALL DISCOT (XA, ZA, TABX, TABY, TABZ, -73, 800, 10, ANS)
```

CASE IV:

Same as Case III with no extrapolation above z_{max}. Control word = -173

```
  CALL DISCOT (XA, ZA, TABX, TABY, TABZ, -173, 800, 10, ANS)
```

Restrictions: See rule (5c) of section "Method" for restrictions on tabulating arrays and discontinuous functions. The order of interpolation in the x- and z-directions may be from 1 to 7. The following subprograms are used by DISCOT: UNS, DISSER, LAGRAN.

Method: Lagrange's interpolation formula is used in both the x- and z-directions for interpolation. This method is explained in detail in reference (a) of this subroutine. For a search in either the x- or z-direction, the following rules are observed:

(1) If $x < x_1$, the routine chooses the following points for extrapolation:

$$x_1, x_2, \ldots, x_{k+1} \text{ and } y_1, y_2, \ldots, y_{k+1}$$

(2) If $x > x_n$, the routine chooses the following points for extrapolation:

$$x_{n-k}, x_{n-k+1}, \ldots, x_n \text{ and } y_{n-k}, y_{n-k+1}, \ldots, y_n$$

(3) If $x \leqq x_n$, the routine chooses the following points for interpolation:

When k is odd,

$$x_{i-\frac{k+1}{2}}, x_{i-\frac{k+1}{2}+1}, \ldots, x_{i-\frac{k+1}{2}+k} \text{ and } y_{i-\frac{k+1}{2}}, y_{i-\frac{k+1}{2}+1}, \ldots, y_{i-\frac{k+1}{2}+k}$$

When k is even,

$$x_{i-\frac{k}{2}}, x_{i-\frac{k}{2}+1}, \ldots, x_{i-\frac{k}{2}+k} \text{ and } y_{i-\frac{k}{2}}, y_{i-\frac{k}{2}+1}, \ldots, y_{i-\frac{k}{2}+k}$$

(4) If any of the subscripts in rule (3) become negative or greater than n (number of points), rules (1) and (2) apply. When discontinuous functions are tabulated, the independent variable at the point of discontinuity is repeated.

(5) The subroutine will automatically examine the points selected before interpolation and if there is a discontinuity, the following rules apply. Let x_d and x_{d+1} be the point of discontinuity.

(a) If $x \leqq x_d$, points previously chosen are modified for interpolation as shown:

$$x_{d-k}, x_{d-k+1}, \ldots, x_d \text{ and } y_{d-k}, y_{d-k+1}, \ldots, y_d$$

(b) If $x > x_d$, points previously chosen are modified for interpolation as shown:

$$x_{d+1}, x_{d+2}, \ldots, x_{d+k} \text{ and } y_{d+1}, y_{d+2}, \ldots, y_{d+k}$$

(c) When tabulating discontinuous functions, there must always be $k + 1$ points above and below the discontinuity in order to get proper interpolation.

(6) When tabulating arrays for this subroutine, both independent variables must be in ascending order.

(7) In some engineering programs with many tables, it is quite desirable to read in one array of x values that could be used for all lines of a multi-line function or different functions. Even though this situation is not always applicable, the subroutine has been written to handle it. This procedure not only saves much time in preparing tabular data, but also can save many locations previously used when every y coordinate had to have a corresponding x coordinate. Another additional feature that may be useful is the possibility of a multi-line function with no extrapolation above the top line.

Accuracy: A function of the order of interpolation used.

Reference: (a) Nielsen, Kaj L.: Methods in Numerical Analysis. The Macmillan Co., c.1956.

Storage: 555_8 locations.

Subprograms used: UNS	40_8	locations.
DISSER	110_8	locations.
LAGRAN	55_8	locations.

Subroutine date: August 1, 1968.

REFERENCES

1. Cebeci, Tuncer; Smith, A. M. O.; and Mosinskis, G.: Calculation of Compressible Adiabatic Turbulent Boundary Layers. AIAA Paper No. 69-687, June 1969.

2. Patankar, S. V.; and Spalding, D. B.: Heat and Mass Transfer in Boundary Layers. Morgan-Grampian (London), 1967.

3. Sontowski, J. F.: An Eddy Viscosity Model for Compressible Turbulent Boundary Layers. Tech. Inform. Ser. No. 69SD241, Re-Entry Systems, Gen. Elec. Co., July 1969.

4. Herring, H. James; and Mellor, George L.: A Method of Calculating Compressible Turbulent Boundary Layers. NASA CR-1144, 1968.

5. Anderson, Larry W.; and Kendal, Robert M.: A Nonsimilar Solution for Multicomponent Reacting Laminar and Turbulent Boundary Layer Flows Including Transverse Curvature. AFWL-TR-69-106, U.S. Air Force, Mar. 1970. (Available from DDC as AD 867 904.)

6. Harris, Julius Elmore: Numerical Solution of the Compressible Laminar, Transitional, and Turbulent Boundary Layer Equations With Comparisons to Experimental Data. Ph. D. Thesis, Virginia Polytech. Inst., May 1970.

7. Bushnell, Dennis M.; and Beckwith, Ivan E.: Calculation of Nonequilibrium Hypersonic Turbulent Boundary Layers and Comparisons With Experimental Data. AIAA J., vol. 8, no. 8, Aug. 1970, pp. 1462-1469.

8. Hunt, James L.; Bushnell, Dennis M.; and Beckwith, Ivan E.: Finite-Difference Analysis of the Compressible Turbulent Boundary Layer on a Blunt Swept Slab With Leading-Edge Blowing. Analytic Methods in Aircraft Aerodynamics, NASA SP-228, 1970, pp. 417-472.

9. Blottner, F. G.: Nonequilibrium Laminar Boundary-Layer Flow of Ionized Air. AIAA J., vol. 2, no. 11, Nov. 1964, pp. 1921-1927.

10. Beckwith, Ivan E.; and Bushnell, Dennis M. (With appendix C by Carolyn C. Thomas): Detailed Description and Results of a Method for Computing Mean and Fluctuating Quantities in Turbulent Boundary Layers. NASA TN D-4815, 1968.

11. Maise, George; and McDonald, Henry: Mixing Length and Kinematic Eddy Viscosity in a Compressible Boundary Layer. AIAA J., vol. 6, no. 1, Jan. 1968, pp. 73-80.

12. Schlichting, Hermann (J. Kestin, transl.): Boundary-Layer Theory. Sixth ed., McGraw-Hill Book Co., 1968.

13. Lin, C. C., ed.: Turbulent Flows and Heat Transfer. Princeton Univ. Press, 1959.

14. Stainback, P. Calvin (With appendix by P. Calvin Stainback and Kathleen C. Wicker): Effect of Unit Reynolds Number, Nose Bluntness, Angle of Attack, and Roughness on Transition on a 5° Half-Angle Cone at Mach 8. NASA TN D-4961, 1969.

15. Richtmyer, Robert D.: Difference Methods for Initial-Value Problems. Interscience Publ., Inc., 1957.

16. Beckwith, Ivan E.: Recent Advances in Research on Compressible Turbulent Boundary Layers. Analytic Methods in Aircraft Aerodynamics, NASA SP-228, 1970, pp. 355-416.

NASA-Langley, 1971 —— 12 L-7399

National Aeronautics and Space Administration
Washington, D.C. 20546

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

FIRST CLASS MAIL

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AN
SPACE ADMINISTRATION

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

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

— National Aeronautics and Space Act of 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546