nasa CR 65807

CHARRING ABLATION PERFORMANCE IN TURBULENT FLOW

Volume II - Computer Program

D2-114031-2

Prepared by

R. Colony, E. P. del Casal, R. S. Gaudette

THE BOEING COMPANY
Space Division
Seattle, Washington

November 1967

For
National Aeronautics and Space Administration
Manned Spacecraft Center
Houston, Texas

NASA Contract No. NAS9-6288

CHARRING ABLATION PERFORMANCE IN TURBULENT FLOW

Volume II - Computer Program

D2-114031-2

Prepared by

R. Colony, E. P. del Casal, R. S. Gaudette

THE BOEING COMPANY
Space Division
Seattle, Washington

November 1967

For
National Aeronautics and Space Administration
Manned Spacecraft Center
Houston, Texas

NASA Contract No. NAS9-6288

PREFACE

This report documents work completed for the National Aeronautics and Space Administration under Contract NAS9-6288, Charring Ablation Performance in Turbulent Flow, issued through the Manned Spacecraft Center, Houston, Texas 77058. The main body of the report is contained in Volume I. Volume II deals with numerical analysis and computer programming.

NASA technical monitor was Mr. D. M. Curry of the Thermal Technology Branch of the Structures and Mechanics Division. The Boeing Company program manager was Mr. V. Deriugin, Head of Structural Heating in the Spacecraft Mechanics and Materials Technology, Space Division.

The authors acknowledge the contribution of Mr. F. M. Knox in editing some portions of the text.

TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

CHARRING ABLATION PERFORMANCE IN TURBULENT FLOW

Volume II - Computer Program

By R. Colony, E. P. del Casal, and R. S. Gaudette
The Boeing Company

1.0 GENERAL INFORMATION

The two computer programs described in this document are in support of an integrated analytical and experimental investigation to predict the ablation performance of the Apollo heat shield. The principal objective was the determination of the ablation performance under turbulent flow conditions of AVCOAT 5026-39HC/G used on the Apollo vehicle. These programs which are written in FORTRAN IV, are machine independent and as much as possible system independent.

1.1 Purpose

The prediction of the performance of the charring ablator on the Apollo heat shield has obvious and immediate applications in the design of ablation thermal protection systems for reentry vehicles in general and the Apollo Command Module in particular. Ablator performance is generally dependent on the material chosen and environmental conditions. The boundary layer equations for heat, mass, and momentum transfer together with a suitable expression for eddy diffusivity applicable to the turbulent, transition, and laminar flow regimes provide the mathematical model of the environmental conditions. The thermal, mechanical, and chemical properties of the ablator are described by a number of correlations derived from experimental results obtained under this contract and from an extensive search of the literature.

Two computer programs were developed under the present investigation, one for the flow field and the other for the charring ablator. Both are coupled by a set of mutually consistent input parameters. The introduction of many simplifying assumptions makes the total solution economical in determining the performance of the ablator at all body positions and for any point in the trajectory.

1.2 Assumptions

The assumptions necessary to define complex flow and phase change mechanisms almost defy enumeration. Physical assumptions used in this program include:

(1) Molecular and transport properties based on air in thermodynamic equilibrium;

(2) the usual boundary layer assumptions such as $\frac{\partial p}{\partial \psi} = 0$, etc.;

(3) local similarity of the tangential velocity, enthalpy and species concentrations;

(4) suitability of mixing length theory to describe turbulent flow;

(5) existence and reliability of semi-empirical relations derived from experiment;

(6) a description of the inviscid flow field is available; and

(7) quasi-steady state conditions exist.

In addition, certain mathematical assumptions have been made. These assumptions and approximations are described in the course of the text.

It should be noted that the particular expressions used in this program for molecular and transport properties, eddy diffusivity and inviscid flow are inputs which may readily be changed by more recent and exact formulations if and when they are available.

1.3 Limitations

The limitations of a program generally depend on the assumptions which, for these programs, are manifold. The only statement that can be made here is that by suitably describing the surface geometry and inviscid flow field, a large number of cases of turbulent or laminar boundary layer flows may be considered. Many classical flat plate problems have been simulated with remarkable success.

2.0 PROCEDURE

The two programs, one describing the flow field and the other describing the ablator performance are used separately. The integrated flow and ablation problem is then solved by coupling the two programs by a set of mutually consistent input parameters. Both programs are described in the following sections 2.2 and 2.3.

2.1 Nomenclature

2.1.1 Nomenclature associated with the flow field program

A	parameter defined by equation (39)
a_o	constant used to determine transition
a	parameter determining transition
b_o	constant used to determine extent of transition
b	parameter used to determine extent of transition
c_f	local skin-friction coefficient
c_p	effective specific heat, Btu/$\mathrm{lb_m}$
F	parameter defined by equation (2)
F_o	ratio of wall mass flux to free-stream mass flux
f	function of
g	gravitational constant, 32.2 ft-$\mathrm{lb_f}$/$\mathrm{lb_m}$-$\mathrm{sec^2}$
G	parameter defined in equation (18)
h	local static enthalpy, Btu/$\mathrm{lb_m}$
H	local total enthalpy, Btu/$\mathrm{lb_m}$
H_{coeff}	heat transfer coefficient, $\mathrm{lb_m}$/$\mathrm{ft^2}$sec
J	Joule's constant, 778 ft-$\mathrm{lb_f}$/Btu
J	index for number of cards to describe y_m^+ array
j	index for y_m^+ value
k	universal constant, 0.36
k	integration index in equation (14)
K	maximum integration index in equation (14)
ℓ^+	dimensionless mixing length
LB	lower bound
M	local Mach number
M_{amb}	ambient molecular weight, $\mathrm{lb_m}$/$\mathrm{lb_m}$-mole
N	exponent in equation (5)

p	pressure, lb_f/ft^2; sometimes used as general function of y_m^+
P	empirical function used in expression for shear stress distribution
P	property
Pr	molecular Prandtl number
Pr^*	effective Prandtl number
q	heat flux at the wall, Btu/ft^2-sec
r_o	surface radius of revolution, ft
R	gas constant, 1545 ft-lb_f/lb_m-mole-°R
Re_D	Reynolds number based on displacement thickness
Re_m	Reynolds number based on momentum thickness
Re_x	Reynolds number based on distance along surface from stagnation point
Re_δ	Reynolds number based on boundary layer thickness
S	Re_D/Re_m
Sc	molecular Schmidt number
Sc^*	effective Schmidt number
St	Stanton number
T	temperature, °R
UB	upper bound
u	local tangential velocity, ft/sec
u_m^+	$u_e/\sqrt{\tau_w/\rho_e}$
$\tilde{u}$	local tangential velocity ratio, u/u_e
V_∞	free-stream velocity, ft/sec
w_c	mass fraction of combustible species
w_I	mass fraction of inert species
w_{IE}	mass fraction of inert species from the free stream
w_{IM}	mass fraction of inert species from ablator
w_{O_2}	mass fraction of oxygen species
w_p	mass fraction of products of combustion

x	coordinate along surface, ft
X	independent variable
y^+	$y\sqrt{\tau_w/\rho_e}/\nu_e$
y_m^+	maximum local shear thickness, treated as independent streamwise variable, $\delta\sqrt{\tau_w/\rho_e}/\nu_e$
Y	dependent variable
Z	compressibility factor
α	mass transfer parameter, $2m_w/(\rho_e u_c c_f)$
β_c	stoichiometric mass ratio for combustible species
β_{IM}	stoichiometric mass ratio for inert species from ablator
β_{O_2}	stoichiometric mass ratio for oxygen species
β_P	stoichiometric mass ratio for products of combustion
γ	ratio of specific heats
δ	boundary layer thickness, ft
ϵ	eddy diffusivity, ft^2/sec
ϵ	convergence criterion in equation (11)
η	local similarity parameter
μ	viscosity, $lb_m/ft\text{-}sec$
ν	kinematic viscosity, ft^2/sec
ρ	density, lb_m/ft^3
σ	variable of integration
τ	shear stress, lb_f/ft^2
ϕ	damping term in mixing length expression
ϕ_μ	viscosity ratio
ϕ_ρ	density ratio

Subscripts:

c	denotes point at which flow is assumed to be similar to that over a flat plate
e	edge of boundary layer

i	step index
j	index in y_m^+ array
k'	integration index in equation (14)
k	index of successive approximations , equation (9)
o	denotes origin conditions , equation (13)
s	denotes stagnation conditions
STOP	denotes point of termination of calculation
w	denotes wall or boundary layer-solid interface

Superscripts:

$*$	denotes dummy variable in integrations
$\wedge$	denotes interpolation routine defined in equation (15)
$\sim$	denotes computed value (of x_j)

2.1.2 Nomenclature associated with ablator program and program matching

A	matrix of partial derivatives used in the solution of the new X
A	stoichiometric coefficient for combustion of ablator surface material
A_{sub}	frequency factor for sublimation, $\text{lb}_m/\text{ft}^2\text{-sec}$
B_{sub}	activation temperature (activation energy/gas constant) for sublimation, °R
$\bar{c}_{p,c}$	specific heat of char, $\text{Btu}/\text{lb}_m\text{-}^\circ\text{R}$
$\bar{c}_{p,p}$	specific heat of pyrolyzed gas, $\text{Btu}/\text{lb}_m\text{-}^\circ\text{R}$
F_i	ith equation defining the ablation mechanism ($i = 1,2,3,4,5$)
f_p	maximum possible fraction of pyrolysis gas that undergoes combustion
G	matrix used in the solution of X
H_0	heat transfer coefficient (no blowing), $\text{lb}_m/\text{ft}^2\text{-sec}$
$\Delta H_{c,c}$	heat of combustion for char, Btu/lb_m
$\Delta H_{c,p}$	heat of combustion of pyrolysis gases, Btu/lb_m
ΔH_{pyr}	heat of pyrolysis, Btu/lb_m (based on ρ_p)
H	total enthalpy, Btu/lb_m
h_o	specific enthalpy, Btu/lb_m
i	tabular function index
j	equation index
k	approximation index
$K_{1,2,4}$	empirical constants (dimensional)
K_{O_2e}	mass fraction of oxygen at edge of boundary layer
$\dot{m}_c$	mass flux of char combustion
$\dot{m}_G$	total gas mass flux, $\text{lb}_m/\text{ft}^2\text{-sec}$
$\dot{m}_{sh}$	mass flux due to shear removal, $\text{lb}_m/\text{ft}^2\text{-sec}$
$\dot{m}_{surf}$	net mass flux of the surface, $\text{lb}_m/\text{ft}^2\text{-sec}$, $(\dot{m}_c + \dot{m}_{sh} + \dot{m}_{sub})$
$\dot{m}_{sub}$	mass flux due to sublimation ($= A_{sub}\, e^{-B_{sub}/T_w}$), $\text{lb}_m/\text{ft}^2\text{-sec}$
N	iteration index in program interface matching

p	pressure, atm
$\dot{q}_0$	convection heat transfer to smooth wall (no blowing), Btu/ft^2-sec
$\dot{q}_{rad}$	radiative heat flux, Btu/ft^2-sec
R_i	independent variable for ith tabular function
St	Stanton number
T_o	initial wall temperature, °R
T_w	wall temperature, °R
u_e	velocity at edge of boundary layer, ft/sec
X	vector containing independent variable
α	mass flux parameter, $2\dot{m}_G/(\rho_e u_e c_f)$
ϵ	error criterion
ϵ	emissivity
Λ	correction term for $\dot{m}_p$
λ	relaxation parameter
ρ_c	density of inert (to pyrolysis) fraction of virgin plastic, lb$_m$/ft^3
ρ_p	density of pyrolyzable fraction of virgin plastic, lb$_m$/ft^3
σ	Stephan-Boltzmann constant, 4.81×10^{-13} Btu/ft^2-sec-°R^4
φ_i	ith tabular function
Ψ	blocking function

Subscripts:

a	denotes value calculated by ablation program
f	denotes value calculated by flow field program
I	first iteration in program interface matching
H	based on enthalpy
j	equation index
k	approximation index
LB	lower bound
S	stagnation condition
UB	upper bound

w	wall condition
0	no blowing
ψ	based on blocking function

Superscript:

$\rightarrow$	Vector quantity

2.2 Mathematical Model - Flow Field Program

2.2.1 The asymptotic momentum equation

The local tangential velocity ratio $\tilde{u}$ at a given point (x, η) in the boundary layer is given by the integral equation:

$$\tilde{u}\,(x,\eta) = \frac{2y_m^+}{u_m^+}\int_0^{\eta} F(\tilde{u},\eta^*)\,d\eta^* \tag{1}$$

where

$$F(\tilde{u},\eta) = \frac{\frac{1}{\varphi_\nu}\left(\frac{\tau}{\tau_w}\right)}{1+\sqrt{1+\frac{4\varphi_e}{\varphi_\nu^2}(l^+)^2\left(\frac{\tau}{\tau_w}\right)}} \tag{2}$$

From the boundary condition $\tilde{u}(x,1)=1.0$, the relation between the skin friction coefficient and y_m^+ is obtained:

$$u_m^+ = 2y_m^+\int_0^1 F(\tilde{u},\eta^*)\,d\eta^* \tag{3}$$

$$\frac{C_f}{2} = \frac{1}{u_m^{+2}} = \frac{1}{\left[2y_m^+\int_0^1 F(\tilde{u},\eta^*)\,d\eta^*\right]^2} \tag{4}$$

In the present investigation, the dimensionless mixing length expression is assumed to be

$$l^+ = k y^+\left[1 - e^{-\varphi\eta}\right]$$

where

$$\varphi = \mathrm{Max}\left[\frac{y_m^+ - a}{b},\ 0\right]$$

$$a = a_0\left[1 + \left(\frac{\gamma-1}{2}\right)M_e^2\right]^{0.125} \tag{5}$$

$$b = b_0\left(\frac{T_w}{T_e}\right)^q\left[1 + Pr^{2/3}\left(\frac{\gamma-1}{2}\right)M_e^2\right]^N$$

and $$b_0 = 22\,, \quad q = N = 0 \tag{6}$$

The shear stress ratio, τ/τ_w, is assumed to be of the form:

$$\frac{\tau}{\tau_w} = 1 - P(\eta^*,x) + \alpha\left[\tilde{u} - P(\eta^*,x)\right] - \frac{2\delta}{C_f}\frac{d\ln u_e}{dx}\left[\eta^* - P(\eta^*,x)\right] \tag{7}$$

where $P(\eta^*,x)$ can be any convenient empirically or semi-empirically determined function. In the present case, a linear relation was used:

$$P(\eta^*,x) = \eta^* \tag{8}$$

The above equations are solved by successive approximations (Picard's method) with the iterative equations assuming the form

$$\tilde{u}_k(x,\eta) = \frac{2y_m^+}{u_{m_k}^+}\int_0^{\eta} F(\tilde{u}_{k-1},\eta^*)\,d\eta^* \tag{9}$$

$$u_{m_k}^+ = 2y_m^+\int_0^1 F(\tilde{u}_{k-1},\eta^*)\,d\eta^*$$

Initially, the local tangential velocity ratio profile is approximated by $u_0 = \eta^{0.1}$ and subsequently by

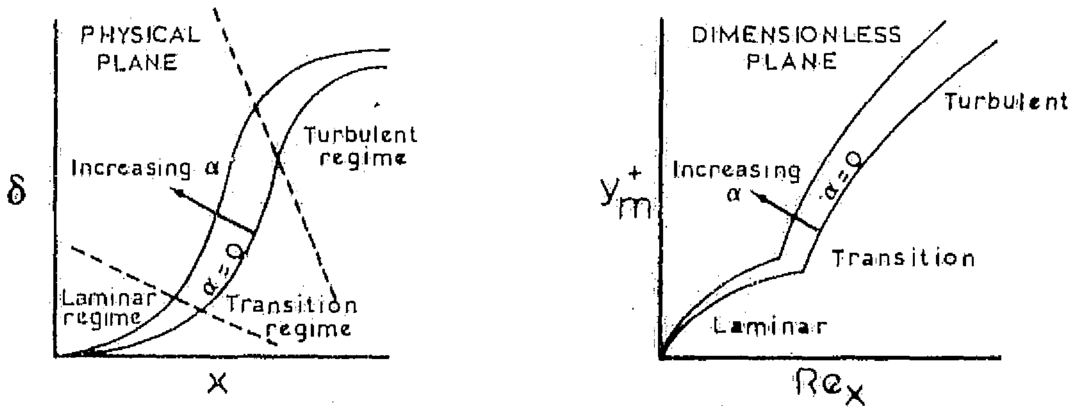
$$\tilde{u}_0(x,\eta) = \tilde{u}(x-\Delta x,\eta) \tag{10}$$

Convergence of $\tilde{u}$ is assumed when

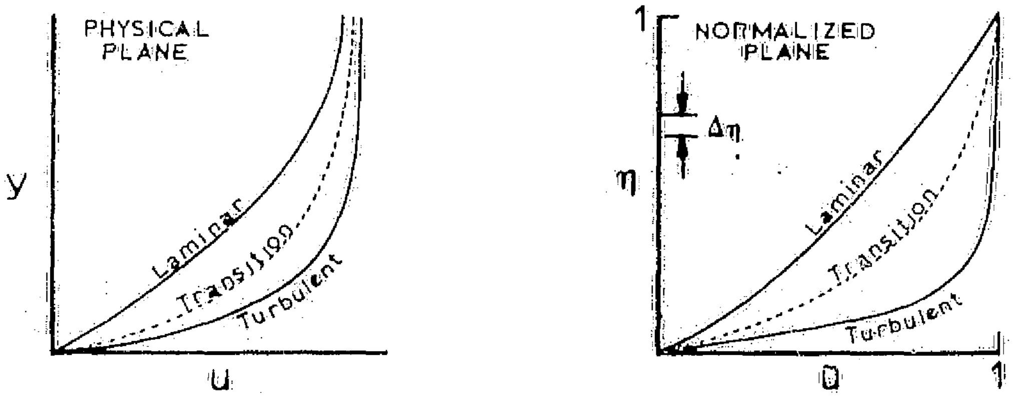
$$\left|\frac{\tilde{u}_k(x,\eta_i) - \tilde{u}_{k-1}(x,\eta_i)}{\tilde{u}_k(x,\eta_i)}\right| < \epsilon = 0.001 \tag{11}$$

for all integral steps, i.e., $i=1,\ldots,N$ provided $k \leq 20$. If the error criterion (11) is not satisfied in twenty iterations, then $\tilde{u}$ is set at $\tilde{u}(x,\eta) = \tilde{u}_{21}(x,\eta)$. This is done on the assumption that the nonconvergence of $\tilde{u}$ is not a fatal error and does not severely influence the total flow field.

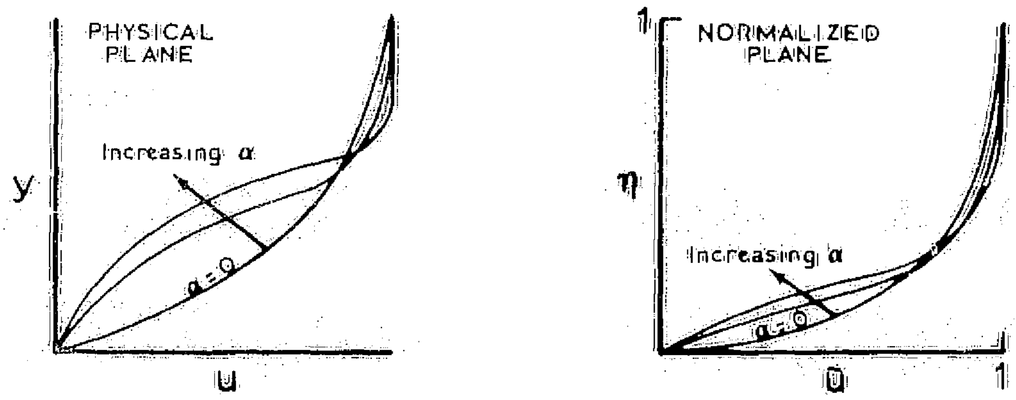
Typical flow field characteristics are shown on figure 1.



a) Boundary Layer Thickness



b) Velocity Profile – No Blowing (α = 0)



c) Velocity Profile – With Blowing

Figure 1: ILLUSTRATIVE TYPICAL RELATIONSHIPS OF BOUNDARY LAYER FLOW CHARACTERISTICS

2.2.2 The momentum integral equation

The momentum integral equation expressed in differential form is

$$\frac{d\,Re_m}{d\,Re_x} + Re_m\left[\frac{d\ln\mu_e}{d\,Re_x} + \frac{d\ln r_0^{\epsilon}}{d\,Re_x} + (1+S)\frac{d\ln u_e}{d\,Re_x}\right] = F_0 + \frac{C_f}{2} \tag{12}$$

In integral form, the above equation becomes

$$Re_x = Re_{x_0} + \int_{Re_{m_0}}^{Re_m} \frac{d\,Re_m^*}{\left\{F_0 + \frac{C_f}{2} - Re_m^*\left[\frac{d\ln u_e}{d\,Re_x} + \frac{d\ln r_0^{\epsilon}}{d\,Re_x} + (1+S)\frac{d\ln u_e}{d\,Re_x}\right]\right\}} \tag{13}$$

When stepping from the (j-1)th to the jth y_m^+, equation (13) is approximated by

$$Re_x(y^+_{m_j}) = Re_x(y^+_{m_{j-1}}) + \sum_{k=1}^{K} \Delta Re_x(\hat{y}^+_{m_k}) \tag{14}$$

where

$$\hat{y}^+_{m_k} = y^+_{m_{j-1}} + \frac{k}{K}\left(y^+_{m_j} - y^+_{m_{j-1}}\right) \tag{15}$$

In general, if p is a function of y_m^+, then

$$\hat{p}_k = p(y^+_{m_{j-1}}) + \frac{k}{K}\left[p(y^+_{m_j}) - p(y^+_{m_{j-1}})\right] \tag{16}$$

Thus,

$$\Delta Re_x(\hat{y}^+_{m_k}) = \int_{\hat{Re}_{k-1}}^{\hat{Re}_{m_k}} \hat{G}_{k-1}\, d\,Re_m^* \tag{17}$$

where

$$\frac{1}{G} = F_0 + \frac{C_f}{2} - Re_m^*\left\{\frac{d\ln\mu_e}{d\,Re_x} + \frac{d\ln r_0^{\epsilon}}{d\,Re_x} + (1+S)\frac{d\ln u_e}{d\,Re_x}\right\} \tag{18}$$

Once $Re_x(\hat{y}^+_{m_k})$ is evaluated, the corresponding value of x is approximated by

$$x(\hat{y}^+_{m_k}) = x(\hat{y}^+_{m_{k-1}}) + \Delta Re_x(\hat{y}^+_{m_k})\left[\frac{\mu_e}{\rho_e u_e}\right] \tag{19}$$

where ρ_e, u_e and μ_e are evaluated at $x(\hat{y}^{+}_{m_{k-1}})$. A case is terminated when

$$\left[x_{STOP} - x(y^{+}_{m_k})\right] \leqslant 0. \tag{20}$$

2.2.3 Functions of the asymptotic momentum equation

Once u has been determined for a given y^{+}_{m}, several parameters are immediately calculable. Those used in the program are:

$$Sc^{*} = Sc \frac{\left(1 + \frac{\epsilon}{D}\right)}{\left[1 + Sc\left(\frac{\epsilon}{D}\right)\right]} \tag{21}$$

$$Re_m = u^{+}_{m}\, y^{+}_{m} \int_0^1 \frac{\rho u}{\rho_e u_e}\left(1 - \frac{u}{u_e}\right) d\eta \tag{22}$$

$$Re_D = u^{+}_{m}\, y^{+}_{m} \int_0^1 \left(1 - \frac{\rho u}{\rho_e u_e}\right) d\eta \tag{23}$$

$$\frac{C_f}{2} = \frac{1}{u^{+2}_{m}} \tag{24}$$

$$Re_\delta = u^{+}_{m}\, y^{+}_{m} \tag{25}$$

$$\frac{2St}{C_f} = \left[\int_0^1 Pr^{*\,2/3}\, d\tilde{u}\right]^{-1}, \quad \alpha = 0 \tag{26}$$

$$\frac{2St}{C_f} = \left[\frac{\exp\left(\int_0^1 Pr^{*\,2/3}\left[\tilde{u} + \frac{1}{\alpha}\right]^{-1} d\tilde{u}\right) - 1}{\alpha}\right]^{-1}, \quad \alpha \neq 0 \tag{27}$$

$$Pr^{*} = Pr \frac{\left(1 + \frac{\epsilon}{D}\right)}{\left[1 + Pr\left(\frac{\epsilon}{D}\right)\right]} \tag{28}$$

$$H_{coeff} = \rho_e u_e (St) \tag{29}$$

$$q = H_{coeff}\left[H_e - H_w\right] \tag{30}$$

2.2.4 The asymptotic species equations

The chemical model of the flow field is given by a set of integral equations which represent the species continuity equations in the boundary layer. Included in the formulations is the simplified combustion model described in Volume I of this report. Five types of gases are considered: combustible species from the ablator, inert species from the ablator, oxygen species, inert species from the free stream and the products of combustion. This limitation is necessary due to computational difficulty. The word inert here refers to oxidation (combustion). Dissociation effects may be treated indirectly as thermodynamic effects.

The concept of a reaction plane, wherein all combustion takes place within a narrow region along the boundary layer is used. A suitable mass balance is prescribed by the equations below. Mass fluxes are not continuous in view of the reaction plane concept.

$$\beta_{IM} + \beta_c = 1.0 \qquad (31)$$
$$\beta_p = \beta_c + \beta_{O_2}$$

where β_{O_2} and β_{IM} are prescribed.

The position of the reaction plane is given by η_c and is obtained from the equation

$$\beta_{O_2} = \left[(w_{O_2})_e - \beta_{O_2}\right] \exp\left(\int_1^{\eta_c} A\, d\sigma\right) \qquad (32)$$

where

$$(w_{O_2})_e = 0.23 \qquad (33)$$
$$(w_I)_e = 1 - (w_{O_2})_e$$

With the above parameters defined, the species equations are easily evaluated:
Mass Fraction of inert species generated by the ablator:

$$w_{IM} = \beta_{IM}\left[1 - \exp\left(\int_0^{\eta} A\, d\sigma\right) \Big/ \exp\left(\int_0^{1} A\, d\sigma\right)\right] \qquad (34)$$

Mass fraction of inert species diffusing from the edge of the boundary layer:

$$w_{IE} = (w_{IE})_e \left[\exp\left(+\int_1^{\eta} A\, d\sigma\right)\right] \qquad (35)$$

Mass fraction of combustible species:

$$w_c = \beta_c\left[1 - \exp\int_{\eta_c}^{\eta} A\, d\sigma\right]; \quad 0 \leq \eta \leq \eta_c \qquad (36)$$
$$w_c = 0; \quad \eta_c < \eta \leq 1$$

Mass fraction of oxygen species:

$$W_{O_2} = (W_{O_2})_e \left[\exp \int_{\eta_c}^{\eta} A\,d\sigma - 1\right] \Big/ \left[\exp \int_{\eta_c}^{1} A\,d\sigma - 1\right]; \quad \eta_c < \eta \leq 1 \tag{37}$$

$$W_{O_2} = 0; \qquad 0 \leq \eta \leq \eta_c$$

Mass fraction of products of combustion:

$$W_P = \beta_P \left[1 - \exp\left(-\int_{\eta_c}^{1} A\,d\sigma\right)\right] \left[\exp\left(\int_{\eta_c}^{\eta} A\,d\sigma\right)\right]; \quad 0 \leq \eta \leq \eta_c$$

$$W_P = \beta_P \left[1 - \exp\left(\int_{1}^{\eta} A\,d\sigma\right)\right]; \qquad \eta_c < \eta \leq 1 \tag{38}$$

where for all cases σ is some variable of integration, in this case η, and

$$A = \frac{F_0 \, Re_\xi \, Sc}{\varphi_\mu \left(1 + \frac{\epsilon}{\nu}\right)} \tag{39}$$

2.2.5 Thermodynamic and transport properties

The gas properties used in this program were based on available equilibrium air data (Ref.). The simple expressions given below are used:

Viscosity: $\mu = \mu(T)$ (tabular) (40)

Viscosity ratio: $$\varphi_\mu = \frac{\mu}{\mu_e} = \left(\frac{T}{T_e}\right)^n \qquad n = 0.70 \tag{41}$$

Compressibility: $Z = 1 \qquad T \leq 6000\ ^\circ R$ (42)

$$Z \cong \frac{T}{6000} \qquad T > 6000\ ^\circ R \tag{43}$$

Density ratio:

$$\varphi_\rho = \frac{\rho}{\rho_e} \cong \frac{Z_e T}{Z\, T_e} \tag{44}$$

Eddy viscosity:

$$\frac{\epsilon}{\nu} = \ell^{+2} F(\tilde{\alpha}, \eta) \tag{45}$$

Wall enthalpy:

$$H_w = \frac{T_w}{C_{p_w}} \tag{46}$$

The temperature ratio is approximated by

$$\frac{T}{T_e} = (1-\tilde{u})\left[\frac{T_w}{T_e} + Pr^{*\,2/3}\left(\frac{\gamma-1}{2}\right)\tilde{u}\, M_e^2\right] + \tilde{u} \tag{47}$$

2.2.6 Flow conditions at the edge of the boundary layer

In view of the complex trajectory and geometry of the Apollo heat shield, no simple equations are presently available to describe the inviscid flow field in the fore and after body. Experimentally obtained static pressure distributions along the surface are, however, available. This section presents the relationships used in this program. Note that the inviscid flow field solutions are boundary conditions in this program and may be easily changed when more detailed knowledge of the flow is available. The relationships used here are:

Static enthalpy:

$$h_e = H_s - \frac{u_e^2}{2gJ} \tag{48}$$

Pressure:

$$p_e = p_e(x) \quad \text{(tabular)} \tag{49}$$

Temperature: $T_e = T_e(h_e, p_e)$ (tabular, bivariate) (50)

Viscosity: $\mu_e = \mu_e(T_e)$ (tabular) (51)

Density:

$$\rho_e = \frac{p_e M_{amb}}{R Z_e T_e}, \quad M_{amb} = 29.9 \tag{52}$$

Mach Number:

$$M_e \simeq u_e \Big/ \sqrt{\gamma_e Z_e g R T_e / M_{amb}} \tag{53}$$

The velocity profiles between the shock and the region near the stagnation point were obtained using Lighthill's incompressible flow approximation (Ref. 3). At a certain distance from the stagnation point (the shoulder location) the flow field is approximated by a flat plate solution. Thus

$$\begin{aligned} u_e &\cong \sqrt{\frac{1}{6}\left(2-\frac{1}{6}\right)}\;\frac{V_\infty x}{R} + \cdots ; && x \leqslant x_c \\ & && M_\infty >> 1 \\ u_e &= u_e(x_c) ; && x \geqslant x_c \end{aligned} \qquad (54)$$

2.2.7 Algorithm for approximating the boundary layer flow field

Given a set of input parameters, an effective algorithm for solving the integral momentum equation may now be defined. The origin ($x_o = 0.0$) is set at the stagnation point, in which case:

$$Re_m = Re_x = Re_D = y_m^+ = u_m^+ = 0.0 \qquad (55)$$

Choose the next y_m^+ which in this case will be the first y_m^+ of the input array. Assuming that the value of x corresponding to the new y_m^+ (call it x_1) is in the neighborhood of x_o, we assert that $P(x_o)$ is in the neighborhood of the property P at x_1. The solution of the asymptotic momentum equation is given as $\tilde{u}(x_1, \eta)$ where all properties, P, are evaluated at x_o. Associated with $\tilde{u}(x_1, \eta)$ are the parameters given in section 2.2.3. The value of Re_x at the new value of y_m^+ is then approximated using the iterative equation (14) from which the value of x, call it $\bar{x}$, is computed. Note that nowhere was the value of x used so that we define x as $\bar{x}$.

The y_m^+ array is exhausted in a similar manner, stepping to the larger value each time assuming

$$\tilde{u}\,(x_j, \eta)\ \text{based on}\ P\,(x_{j-1}),$$

then defining x_j as $\bar{x}$.

2.2.8 Notes on quadrature

Wherever quadrature is required, such as $\int_a^b f(x)dx$, numerical approximations are made. The quadrature equation used throughout this program is a variable interval extension of Simpson's rule. An effective control of the global error can be achieved by a variable interval scheme. The exact description of the method is given in ref. 3 and will not be further discussed here. Note that the analysis presented here is not dependent on a particular quadrature formula. Any reasonably accurate approximation will suffice.

2.3 Mathematical Model - Ablator Program

2.3.1 Ablator equations

Five nonlinear equations discussed in Volume I provide the mathematical model of the ablation mechanism. These equations are summarized as:

Surface heat balance equation:

$$\begin{aligned} F_1 = {} & \Psi \dot{q}_0 + f \dot{m}_p \Delta H_{c,p} + \dot{m}_c \Delta H_{c,c} + \dot{q}_{rad} \\ & - \Big\{ \epsilon \sigma T_w^{\,4} + \dot{m}_p \left[\bar{C}_{p,p} (T_w - T_0) \right] + \dot{m}_c \left[\bar{C}_{p,c} (T_w - T_0) \right] \\ & + \dot{m}_p \Delta H_{pyr} + \dot{m}_{sh} \left[\bar{C}_{p,c} (T_w - T_0) \right] \\ & + \dot{m}_{sub} \left[\bar{C}_{p,c} (T_w - T_0) \right] + \dot{m}_{sub} \Delta H_{sub} \Big\} = 0 \end{aligned} \tag{56}$$

Correction term for $\dot{m}_c$ determination:

$$F_2 = \Lambda - \left\{ \varphi_1 \left[\frac{\dot{m}_p \sqrt{T_w}}{p} K_1^{\,10^4/T_w} \right] \right\} = 0 \tag{57}$$

Mass flux due to pyrolysis:

$$F_3 = \dot{m}_p - \left\{ \varphi_5 \left[T_w K_3^{\left[\dot{m}_p \left(\frac{\dot{m}_p}{\rho_p} - \frac{\dot{m}_{surf}}{\rho_c} \right) \right]} \right] \right\} = 0 \tag{58}$$

Mass flux due to combustion:

$$F_4 = \dot{m}_c - \left\{ A\, K_{O_2,e}\, \rho_e u_e (St)_0 - \Lambda \dot{m}_p \right\} = 0 \tag{59}$$

Mass flux due to shear stress:

$$F_5 = \dot{m}_{sh} - \left\{ \varphi_2 \left(\Psi \mathcal{H}_0 u_e K_2^{\,10^4/T_w} \right) \right\} = 0 \tag{60}$$

where $\varphi_i(R)$ means the ith tabular function of the relationship R. For i = 1, note that

$$R = \frac{\dot{m}_p \sqrt{T_w}}{p} K_1^{\,10^4/T_w} \tag{61}$$

The simultaneous solution of the above five equations determines the variables $\dot{m}_p$, Λ, T_w, $\dot{m}_{sh}$, and $\dot{m}_c$. The method of solution is based on the first order approximation of the functions F_j. Consider the kth approximation of the jth equation:

$$F_j + \nabla F_j \cdot (\vec{X}_k - \vec{X}_{k-1}) = 0 \tag{62}$$

where

$$\vec{X} = \begin{Bmatrix} \dot{m}_p \\ \Lambda \\ T_w \\ \dot{m}_{sh} \\ \dot{m}_c \end{Bmatrix} \tag{63}$$

$$\nabla F_j = \frac{\partial F_j}{\partial \dot{m}_p}(\quad) + \frac{\partial F_j}{\partial \Lambda}(\quad) + \frac{\partial F_j}{\partial T_w}(\quad) + \frac{\partial F_j}{\partial \dot{m}_c}(\quad) + \frac{\partial F_j}{\partial \dot{m}_{sh}}(\quad)$$

and both F_j and ∇F_j are evaluated at $\vec{X}_{k-1}$. The initial approximation to $\vec{X}$ is provided by the user.

Note that equation (62) is one of a set of five linear equations in the unknown $\vec{X}_k$. The set of linear equations can be written in matrix notation as

$$A \vec{X}_k = G \tag{64}$$

where

$$A = \begin{bmatrix} \frac{\partial F_1}{\partial \dot{m}_p} & \frac{\partial F_1}{\partial \Lambda} & \frac{\partial F_1}{\partial T_w} & \frac{\partial F_1}{\partial \dot{m}_{sh}} & \frac{\partial F_1}{\partial \dot{m}_c} \\ \frac{\partial F_2}{\partial \dot{m}_p} & \frac{\partial F_2}{\partial \Lambda} & \frac{\partial F_2}{\partial T_w} & \frac{\partial F_2}{\partial \dot{m}_{sh}} & \frac{\partial F_2}{\partial \dot{m}_c} \\ \frac{\partial F_3}{\partial \dot{m}_p} & \frac{\partial F_3}{\partial \Lambda} & \frac{\partial F_3}{\partial T_w} & \frac{\partial F_3}{\partial \dot{m}_{sh}} & \frac{\partial F_3}{\partial \dot{m}_c} \\ \frac{\partial F_4}{\partial \dot{m}_p} & \frac{\partial F_4}{\partial \Lambda} & \frac{\partial F_4}{\partial T_w} & \frac{\partial F_4}{\partial \dot{m}_{sh}} & \frac{\partial F_4}{\partial \dot{m}_c} \\ \frac{\partial F_5}{\partial \dot{m}_p} & \frac{\partial F_5}{\partial \Lambda} & \frac{\partial F_5}{\partial T_w} & \frac{\partial F_5}{\partial \dot{m}_{sh}} & \frac{\partial F_5}{\partial \dot{m}_c} \end{bmatrix} \tag{65}$$

and G is a column matrix having the element G_i of the ith row

$$G_i = \nabla F_i \cdot \vec{X}_{k-1} - F_i \tag{66}$$

All partial derivatives are evaluated at $\vec{X}_{k-1}$.

When attempting the iterative solution of nonlinear equations, experience dictates that care must be exercised to keep each successive approximation in the neighborhood of the last approximation. Also, certain bounds determined externally may not be exceeded due to mathematical or physical limitations on the domain. In order to effect this, we do not use the $\vec{X}_k$ predicted by equation (64) but rather, modify it by the algorithm below.

Taking the element in the kth row of the vector to be X

i) if $X > X_{UB}$ (X_{UB} is the upper bound of x), then let $\ln x = 1/2$ $(\ln X_{k-1} + \ln X_{UB})$ where X_{k-1} was the value of x on the previous iteration;

ii) if $X < X_{LB}$ (X_{LB} is the lower bound of x), then let $\ln x = 1/2$ $(\ln X_{k-1} + \ln X_{UB})$;

iii) if $X_{LB} < X < X_{UB}$ no change is made;

iv) $X_k = \lambda X_{k=1} + (1- \lambda) x$,

where typical values of λ are 1/2.

Convergence is defined as

$$\text{(A)} \qquad \left| \frac{X - X_{k-1}}{X} \right| < \epsilon$$

for all rows of the vector $\vec{X}_k$, where ϵ was chosen as 0.08, and

$$\text{(B)} \qquad \frac{|F_i|}{|\dot{q}_0|} < 0.05$$

The second inequality is required because steps i) and ii) may satisfy the inequality (A) but not satisfy equation (56). If the inequalities (A) and (B) are not satisfied before k = 100, the iteration is terminated and an error message is given (no convergence).

In some instances $\dot{m}_{sh}$ or $\dot{m}_c$ may be dropped from the vector $\vec{X}$. In these cases equation (59) and/or (60) are disregarded and the matrices adjusted. By definition, case 1 refers to the instance when $\dot{m}_{sh}$ and $\dot{m}_c$ are included in the vector $\vec{X}$; case 2 to $\dot{m}_c$ but not $\dot{m}_{sh}$; case 3 to $\dot{m}_{sh}$ but not $\dot{m}_c$; and case 4 to neither $\dot{m}_{sh}$ nor $\dot{m}_c$.

Due to the nonlinearity of equations (56) to (61), the existence of multiple roots is not surprising. Unfortunately, multiple roots often reside between the upper and lower bounds provided by the user. As of yet, no simple analytical means has been found to separate the desired solution from the spurious roots. An effective method of selecting the proper root is to underestimate the initial guess to $\overset{*}{X}$.

2.4 Mathematical Model - Coupling between Ablator and Flow-Field Programs

2.4.1 Ablator - flow field interface matching

Letting the iteration index N=I for the initial computations on both programs, and subscripts f and a refer to the flow field equations and ablation programs respectively (for T_w), the values of T_{sIf}, α_{If} and ψ_{If} must be assigned initial values, which are usually 535, 0 and 0.6, respectively. (The values of T_{wIf} and α_{If} should be chosen to give cold wall performance for no mass injection: the value of ψ_{Ia} may be any reasonable value based on a priori experience.) The values assigned to T_{wNf}, α_N, and ψ_N $(N>I)$ are:

$$T_{wN_f} = \left[T_{w(N-1)}\right]_a$$

and

$$\alpha_N = \left[\frac{\dot{m}_G}{H_{coeff}} \cdot \frac{St}{C_f/2}\right],$$

$$\psi_N = \frac{q_{wN}}{q_{wI}}$$

where

T_{wf} = wall temperature input into the flow field program

T_{wa} = wall temperature calculated by the ablation program

and $\dot{m}_G$ = total gas mass flux at the surface.

(For cases when $\frac{H_{wN}-H_{wI}}{H_s} > \epsilon_H$, where ϵ_H is arbitrarily assigned the value 0.1, the value of q_{wI} for use in calculating ψ_N should be recalculated by the flow field program at each iteration, letting $T_{wIf} = T_{wNf} = T_{w(N-I)a}$, $\alpha_I = 0$.) The iteration is terminated when $\left|\psi_N - \psi_{N-I}\right| / \psi_N < \epsilon_\psi$ where ϵ_ψ has been arbitrarily assigned the value 0.1. Since in most applications T_w and α will be functions of distance, iterations must be performed concomitantly at selected distances along the wetted path.

Input decks and program printouts are shown for the case of Apollo flight 202, body station 1, time 52 seconds and a=20 (first iteration only).

2.5 Results and Discussion

To test the overall performance, accuracy and reliability of the boundary layer computer program, the following points were considered: convergence performance, comparison with an exact solution, and comparison with experimental data.

(a) Convergence characteristics

(1) Asymptotic equations

In general, the convergence of the iteration scheme for the asymptotic equations is good at all external flow conditions and for small to moderately large values of α ($\leqslant 10$).

(2) Integral boundary layer equations

The convergence of the iterative scheme used to solve the integral boundary layer equations is dependent on the success of the iteration for the asymptotic equations and the step sizes of the y_m^+ array that is input. Sample cases #1 and 2 illustrate the latter point. For exactly the same external flow and surface conditions, two arrays of y_m^+ were used, one with a smaller step size than the other. The array with smaller step size converged whereas the other did not.

In general, the numerical results obtained using different converging arrays of y_m^+ are not significantly different. This is illustrated in sample cases #3 and 4. Two arrays were used which led to convergent solutions, one with smaller step sizes than the other. The numerical results obtained were within 1% of each other.

(b) Comparison with an exact solution

In order to test the accuracy of the computer program, it is necessary to compare it with an exact solution. An exact solution to the velocity profiles and skin friction distribution may be obtained when the physical properties are constant. For this case

$$\tilde{u} = \eta(2-\eta) \tag{69}$$

$$C_f/2 = \frac{0.8167}{\sqrt{Re_x}} \tag{70}$$

The exact solution and the numerical results are plotted in figure 2. It can be seen that agreement is very close.

(c) Comparison with experimental data

The overall reliability of a computer model can be tested ultimately by its ability to duplicate physically observed data. Unfortunately, the extent of experimental data available in the literature in the region of greatest interest in this program, i.e., hypersonic turbulent flow data,

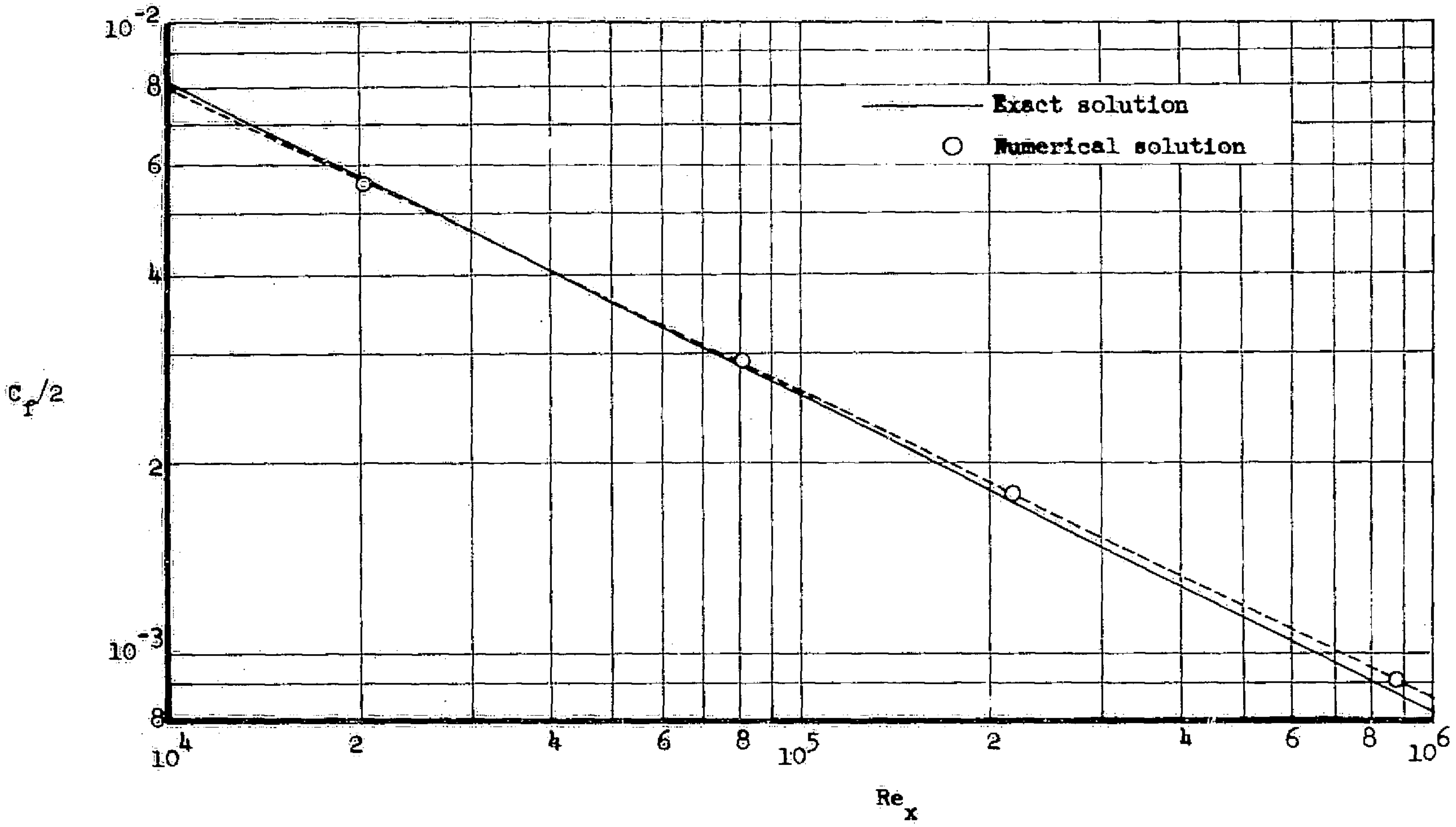


Figure 2.- Comparison of numerical solution with exact solution.

are scarce. The greatest amount of data is available for the flat plate geometry so that comparisons can only be made with data available for the flat plate (figure 3).

The solution described in the ablation analysis is best justified in only those regimes for which experimental data were correlated. The correlations developed were necessarily extrapolated, however, to help predict performance at all conditions encountered in the given Apollo trajectories. For the Apollo application involved, only two cases did not converge on $\dot{m}_p$, probably as a consequence of the correlation extrapolations. Few cases require more than fifty iterations.

As a test of the ability of the programmed ablation analysis to predict performance, the predicted surface mass flux is compared with experimental values in figure 4. Deviations are mainly due to the scatter in the experimental data about the line selected for correlation of shear-induced surface recession .

2.6 Conclusions and Recommendations

The following conclusions may be drawn:

(1) A computer program based on an integral solution of the boundary layer equations was developed and successfully applied for the calculation of smooth wall heat fluxes to a surface with mass transfer in laminar and turbulent flows. The program is essentially system independent and machine independent.

(2) A computer routine for simultaneously solving five transcendental algebraic equations describing ablation performance of AVCOAT 5026-39HC/G has been developed.

As has been stated in the text, the boundary layer computer program has been designed to be flexible and does not depend on the particular eddy diffusivity, thermodynamic and transport properties and potential flow field used. These are inputs into the program, which may be improved if better approximations are available. If very high Mach numbers ($M \sim 40$) are anticipated, the inclusion of shock and boundary layer gas radiation is necessary. This offers no major obstacle to the program as now formulated.

3.0 INPUT-OUTPUT

3.1 Input

This section describes the input format necessary to exercise the two programs described in the preceding sections. All user supplied input is via punched cards. In using this section note the distinction between CARD # and CARD SET #. A listing of the input cards used in the sample case is given in section 3.1.4.

3.1.1 Input of flow field program

CARD SET 1 Format (12A6)

Col 1-72 Any alphanumeric characters used for case identification

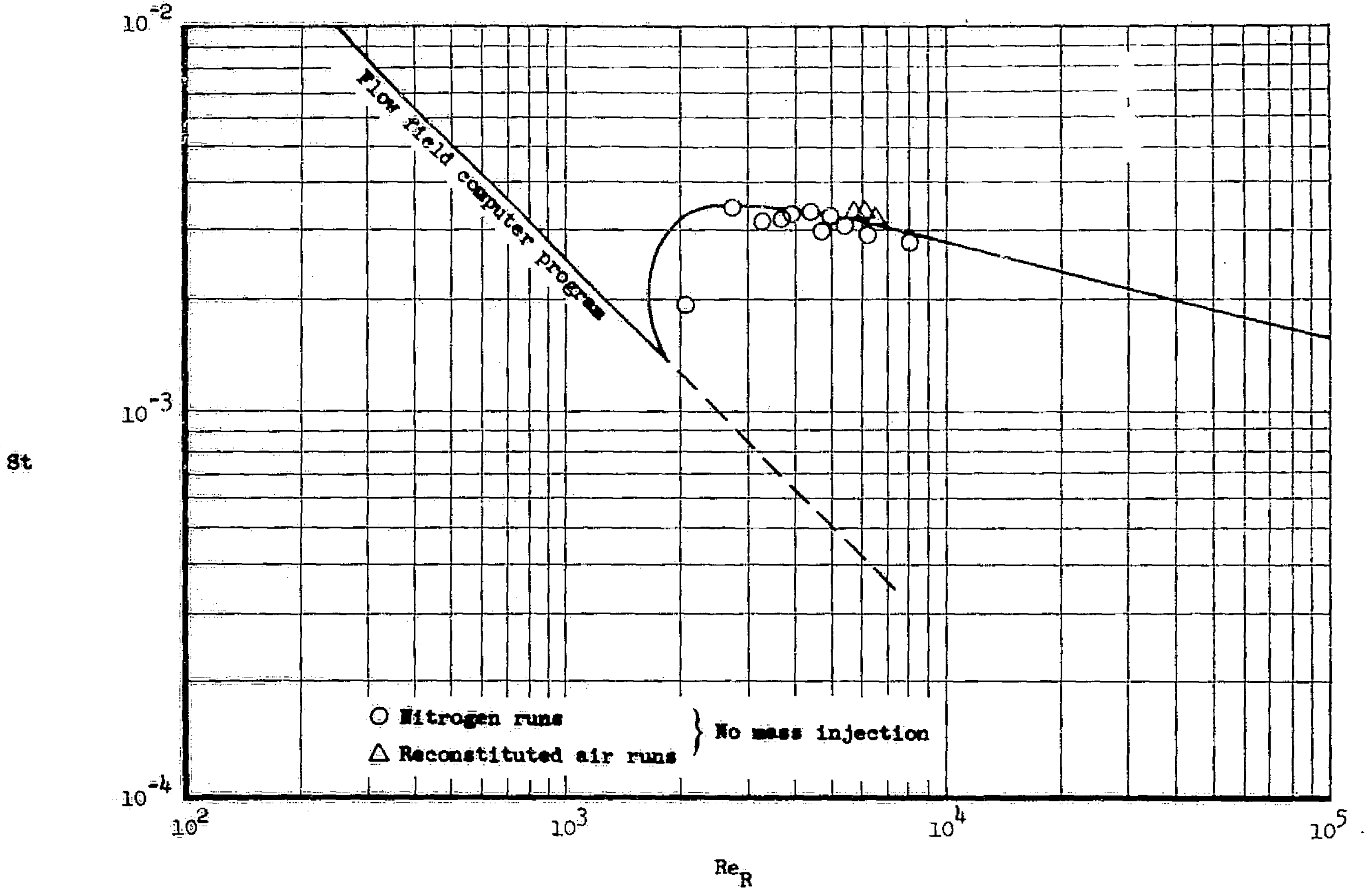


Figure 3.- Comparison of experimental heat transfer to computer prediction of heat transfer (non-blowing case).

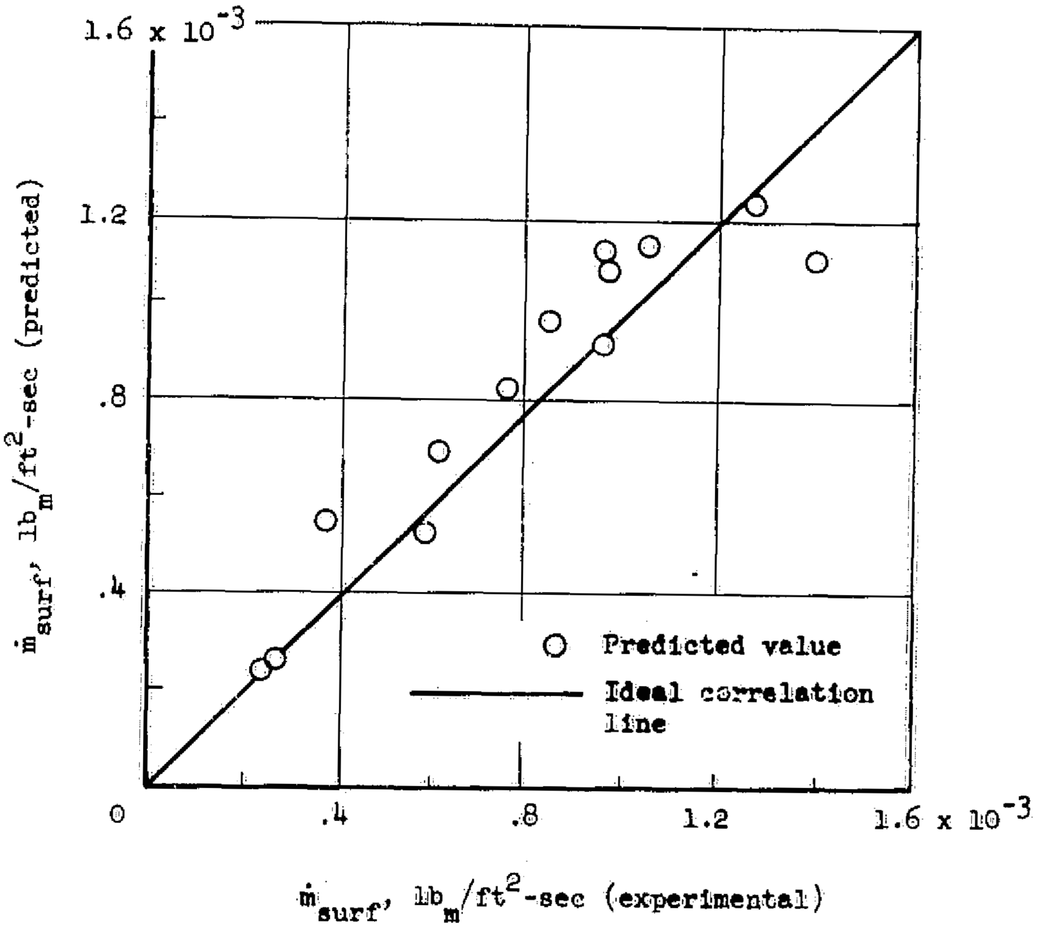


Figure 4.- Comparison of predicted and experimental ablator performance for turbulent flow.

CARD SET 2 FORMAT (2I5)

1-5	NYMP	Number of y_m^+ in following array
6-10	KPOUT	= 0 print solution of asymptotic variables 1 suppress print of asymptotic variables.

These two variables must be right adjusted in the field.

CARD SET 3 FORMAT (7F10.0)

CARD 1

Col 1-11	y_m^+ Number 1
Col 11-20	y_m^+ Number 2
Col 21-30	y_m^+ Number 3
Col 31-40	y_m^+ Number 4
Col 41-50	y_m^+ Number 5
Col 51-60	y_m^+ Number 6
Col 61-70	y_m^+ Number 7

If NYMP > 7 another card must be used.

CARD 2 FORMAT (7F10.0)

Col 1-10	y_m^+ Number 8
Col 11-20	y_m^+ Number 9
Col 21-30	y_m^+ Number 10
Col 31-40	y_m^+ Number 11
Col 41-50	y_m^+ Number 12
Col 51-60	y_m^+ Number 13
Col 61-70	y_m^+ Number 14

Similarly, if NYMP > 14, another card must be used. The number of cards used to prescribe the array is the integer J where

$$J-1 < \left[\frac{NYMP}{7}\right] \leqslant J$$

The restriction on the array is

$$0 < y_m^+ \text{ number } i-1 < y_m^+ \text{ number } i$$

$$i = 2,3,\ldots, NYMP$$

in order that the program will proceed in the downstream direction. Also an implied restriction is that y_m^+ number (i-1) is in the neighborhood of y_m^+ number i. The program will calculate the appropriate downstream distance, x_i, for each y_m^+ in the array.

CARD SET 4 FORMAT (8F10.0)

Col 1-10	$\Delta\eta_1$	Step size of η used for output and table definition between $\tilde{u} = 0$ and $\tilde{u} = \tilde{u}_x$
Col 11-20	$\Delta\eta_2$	Step size of η used for output and table definition between $\tilde{u} = \tilde{u}_x$ and $\tilde{u} = 1$
Col 21-30	$\tilde{u}_x$	Velocity at which to change from $\Delta\eta_1$ to $\Delta\eta_2$
Col 31-40	k	Constant in equation (5)
Col 41-50	a_o	Constant in equation (5)
Col 51-60	Pr	Molecular Prandtl number
Col 61-70	Sc	Molecular Schmidt number
Col 71-80	x_{STOP}	One criterion for termination of a case

CARD SET 5 FORMAT (4F10.0)

Col 1-10	P_s	Stagnation pressure
Col 11-20	V_∞	Free stream velocity
Col 21-30	H_s	Total stagnation enthalpy
Col 31-40	x_c	Value of x after which the flow is assumed to be similar to flow over a flat plate
Col 41-50	β_{O_2}	Input species mass ratios for combustion
Col 51-60	β_{IM}	

CARD SET 6 (Format described in Section 3.1.3)

Definition of pressure ratio as a function of x. The independent variable

$$X = x$$

and the dependent variable

$$Y = \frac{p}{p_s}$$

CARD SET 7 (Format described in Section 3.1.3)

Definition of wall temperature as a function of x. The independent variable

$$X = x$$

and the dependent variable

$$Y = T_w$$

CARD SET 8 (format described in Section 3.1.3)

Definition of α as a function of x. The independent variable

$$X = x$$

and the dependent variable

$$Y = \alpha$$

A case is terminated by one of three means:

1) $x > x_{STOP}$

2) y_m^+ array is exhausted

3) Some type of error is detected.

3.1.2 Input of ablator program

CARD SET 1 (Format described in Section 3.1.3)

Definition of the tabular function φ_1 where the independent variable

$$X = \frac{\dot m_p \sqrt{T_w}\, K_1^{10^4/T_w}}{p}$$

and the dependent variable $Y = \Lambda$.

CARD SET 2 (Format described in Section 3.1.3)

Definition of the tabular function φ_5 where the independent variable

$$X = T_w K_3^{\dot m_p \left(\frac{\dot m_p}{\rho_p} - \frac{\dot m_{surf}}{\rho_c}\right)}$$

and the dependent variable

$$Y = \dot m_p .$$

CARD SET 3 (Format described in Section 3.1.3)

Definition of the tabular function φ_2 where the independent variable

$$X = \psi H_0 u_e K_2^{10^4/T_w}$$

and the dependent variable

$$Y = \dot m_{sh} .$$

CARD SET 4 FORMAT (8F10.0)

Col 1-10	$\mathcal{H}_o$
Col 11-20	ψ
Col 21-30	$\dot{q}_o$
Col 31-40	p
Col 41-50	T_o
Col 51-60	u_e
Col 61-70	ρ_p
Col 71-80	ρ_c

NOTE: For the Apollo material, ρ_p and ρ_c have been "dummied in" as 10^{-6}, in which case K_3 = 1.10.

CARD SET 5 FORMAT (8F10.0)

Col 1-10	$\bar{c}_{p,c}$
Col 11-20	$\bar{c}_{p,p}$
Col 21-30	$\Delta H_{c,c}$
Col 31-40	$\Delta H_{c,p}$
Col 41-50	ΔH_{pyr}
Col 51-60	ϵ
Col 61-70	σ
Col 71-80	A

CARD SET 6 FORMAT (8F10.0)

Col 1-10	$K_{O_2,e}$
Col 11-20	K_1
Col 21-30	K_3
Col 31-40	K_2

CARD SET 7 FORMAT (5F10.0)

Col 1-10	A_{sub}
Col 11-20	B_{sub}
Col 21-30	$\dot{q}_{rad}$
Col 31-40	f_p
Col 41-50	ΔH_{sub}

CARD SET 8 FORMAT (5F10.0)

Col 1-10	$\dot{m}_p$(UB)	Upper bound of $\dot{m}_p$
Col 11-20	Λ (UB)	" Λ
Col 21-30	T_w(UB)	" T_w
Col 31-40	$\dot{m}_{sh}$(UB)	" $\dot{m}_{sh}$
Col 41-50	$\dot{m}_c$(UB)	" $\dot{m}_c$

CARD SET 9 FORMAT (5F10.0)

Col 1-10	$\dot{m}_p$(LB)	Lower bound of $\dot{m}_p$
Col 11-20	Λ(LB)	" Λ
Col 21-30	T_w(LB)	" T_w
Col 31-40	$\dot{m}_{sh}$(LB)	" $\dot{m}_{sh}$
Col 41-50	$\dot{m}_c$(LB)	" $\dot{m}_c$

CARD SET 10 FORMAT (5F10.0)

Col 1-10	$\dot{m}_p$(IG)	Initial guess to $\dot{m}_s$
Col 11-20	Λ(IG)	" Λ
Col 21-30	T_w(IG)	" T_w
Col 31-40	$\dot{m}_{sh}$(IG)	" $\dot{m}_{sh}$
Col 41-50	$\dot{m}_c$ (IG)	" $\dot{m}_c$

CARD SET 11 FORMAT (I5)

Col 1-5	KASE	Control integer (must be right adjusted in field)
KASE = 1		Begin next case at CARD SET 1
KASE = 2		Begin next case at CARD SET 4
KASE = 3		Begin next case at CARD SET 10

This last assignment to KASE is provided to make several different guesses to X. This may be useful if convergence is not easily attainable.

While the user should be careful to input a meaningful set of inputs, the program may change some of the inputs to insure consistency. Such examples would be

1) $\dot{m}_p$ (IG) < $\dot{m}_p$ (LB)

2) Λ (IG) < (from table) min (Λ_1)

3.1.3 Tabular input format

This section gives the input format for all tabular functions defined via input. The general tabular function will have the form $f(X_i) = Y_i$ $i = 1,2,\ldots n$. The format scheme is

CARD 1 FORMAT (I5)

Col 1-5	NPT	n, the number of ordered pairs constituting the tabular function (this number must be right adjusted in the field)

CARD 2 FORMAT (8F10.0)

Col 1-10	X_1	first independent variable
Col 11-20	Y_1	first dependent variable
Col 21-30	X_2	second independent variable
Col 31-40	Y_2	second dependent variable
Col 41-50	X_3	third independent variable
Col 51-60	Y_3	third dependent variable
Col 61-70	X_4	fourth independent variable
Col 71-80	Y_4	fourth dependent variable

If NPT > 4, another card must be used.

CARD 3 FORMAT (8F10.0)

Col 1-10	X_5
Col 11-20	Y_5
Col 21-30	X_6
Col 31-40	Y_6
Col 41-50	X_7
Col 51-60	Y_7
Col 61-70	X_8
Col 71-80	Y_8

Similarly, if NPT > 8, another card must be used. The number of cards used for a table will be the integer J where

$$J-1 < I + \frac{NPT}{4} \leqslant J$$

Restrictions on the tabular function are

1) $NPT \geqslant 2$

2) $X_{i-1} < X_i \qquad i = 2,3,\ldots, n$

3.1.4 Listing of input cards

3.1.4.1 Listing of flow field input for first iteration.

[SEE
SAMPLE #5]

3.1.4.2 Listing of ablator input based on first iteration of flow field.

[SEE
SAMPLE #6]

3.2 Output

The output is annoted such that further description here is minimal. The output of the sample case, given in Section 3.2.1, best illustrates the output. All output is of printed form.

3.2.1 Output of sample case

The case presented is the 202 trajectory at 60 sec.

3.2.1.1 Output of flow field program for first iteration (summary only as KPØUT = 1).

[SEE
SAMPLE #7]

3.2.1.2 Output of ablator program based on first iteration of flow field.

[SEE
SAMPLE #8]

In this sample, the first row under each iteration statement represents the F_i's, $i = 1,2,3,4,5$; the case statement identifies each column in the remaining 3 rows. The necessary concordance of nonobvious relationships between the program language and nomenclature, respectively, follows:

PHI 1 = φ_1
PHI 2 = φ_5
PHI 3 = φ_2
M DOT S = $\dot{m}_{sh}$

The values for tables 1,2, and 3 were obtained from Figures 34, 37, and 36, respectively, of Volume 1 of this report; the extrapolations used should not be considered final.

4.0 OPERATING INSTRUCTION

No special operating instructions are necessary for execution of either program. All computer input (both programs and data) may consist of punched cards. All output is of printed form.

Typical execution times for the Univac 1108 are 1 minute for the flow field program and 10 seconds for the ablation program. Output in pages from the flow field program is about NYMP+6 where NYMP is the number of y_m^+'s used.

Output from the ablation program is about 4 pages.

5.0 PROGRAMMING INFORMATION

5.1 Flow Diagrams

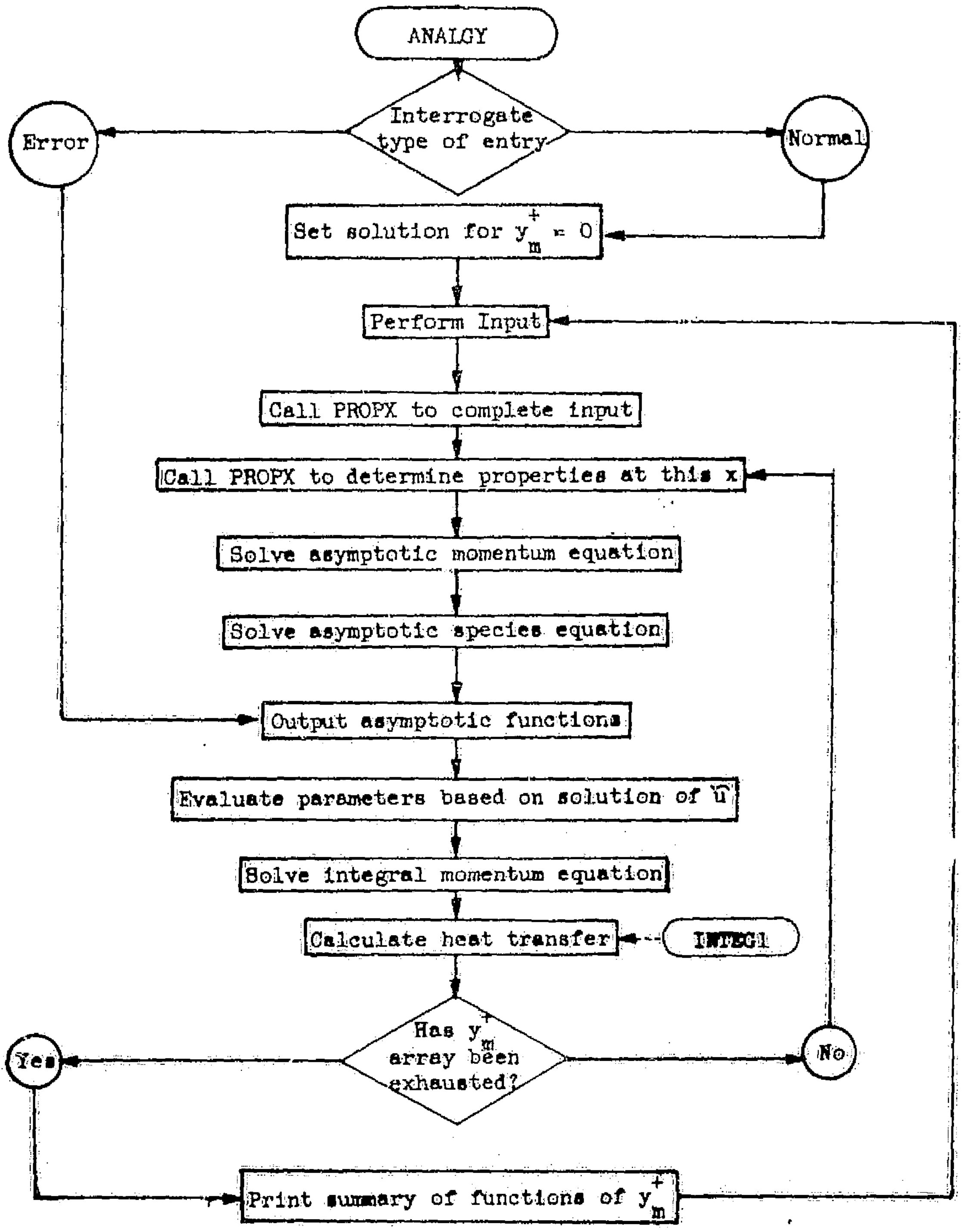
5.1.1 Flow field program

Flow diagrams follow for subroutines developed for the flow field program -- ANALGY, PROPX, SPECIE, VONKAR, FUNCT, and MOMENT. In using these, several pre-existing packages are further necessary -- INTEG1, LAGIT, DLAGIT, DTAB, and TAB -- which are described in References 3 and 4. (DLAGIT is a Fortran IV rewrite of subroutine NUMDER of Reference 4.)

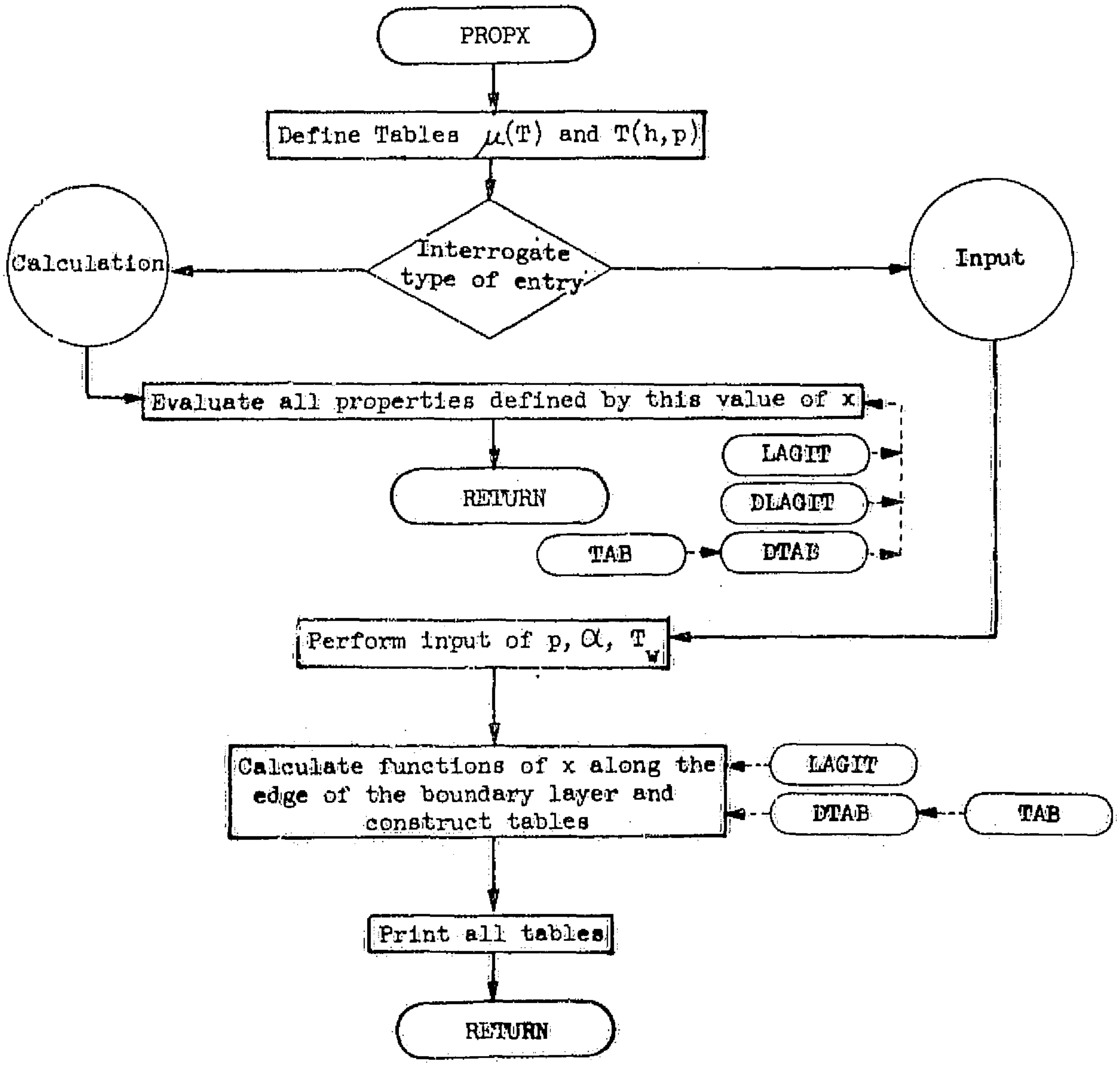
In addition to routines developed under this contract, subroutine ANALGY calls INTEG1 (a quadrature routine); subroutine PROPX calls LAGIT (a Lagrangian interpolation routine, DLAGIT (the derivative of the Lagrangian interpolation formula), and DTAB (a double table lookup routine); subroutine SPECIE calls INTEG1 and LAGIT; subroutine VONKAR calls INTEG1; subroutine FUNCT calls LAGIT; and subroutine MOMENT calls INTEG1 and LAGIT.

The flow field program is listed in Section 7.1.

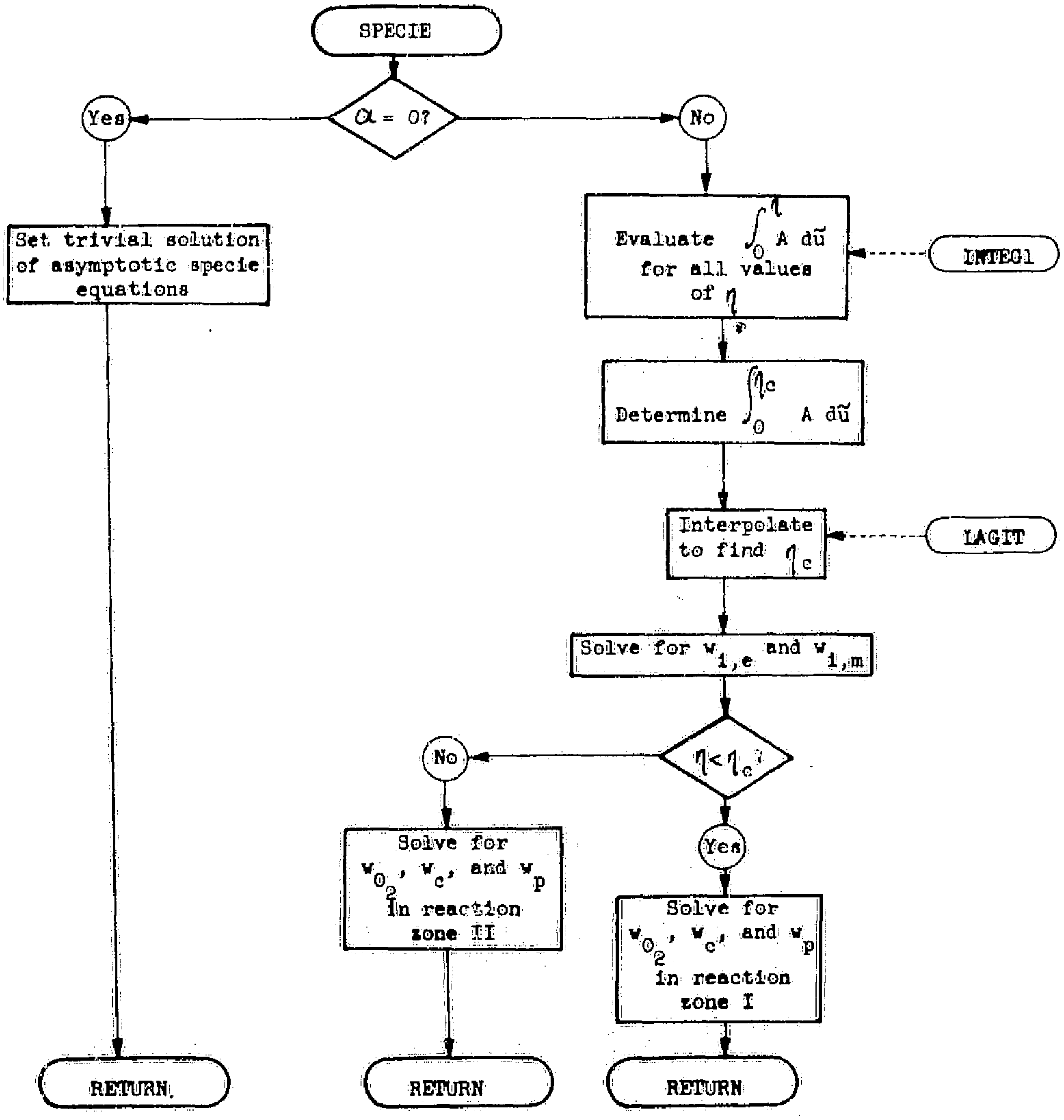
5.1.1.1 Subroutine ANALGY



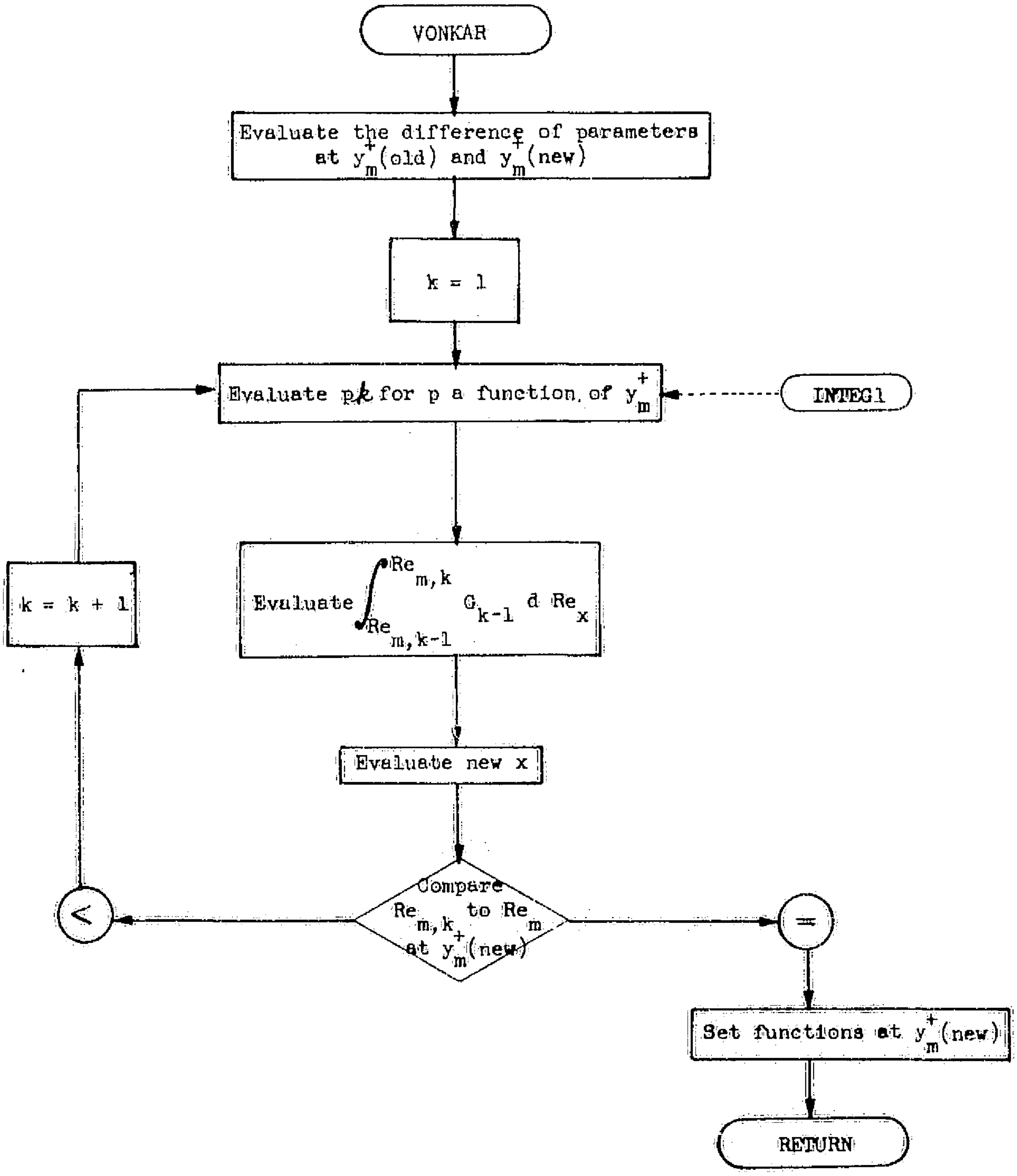
5.1.1.2 Subroutine PROPX



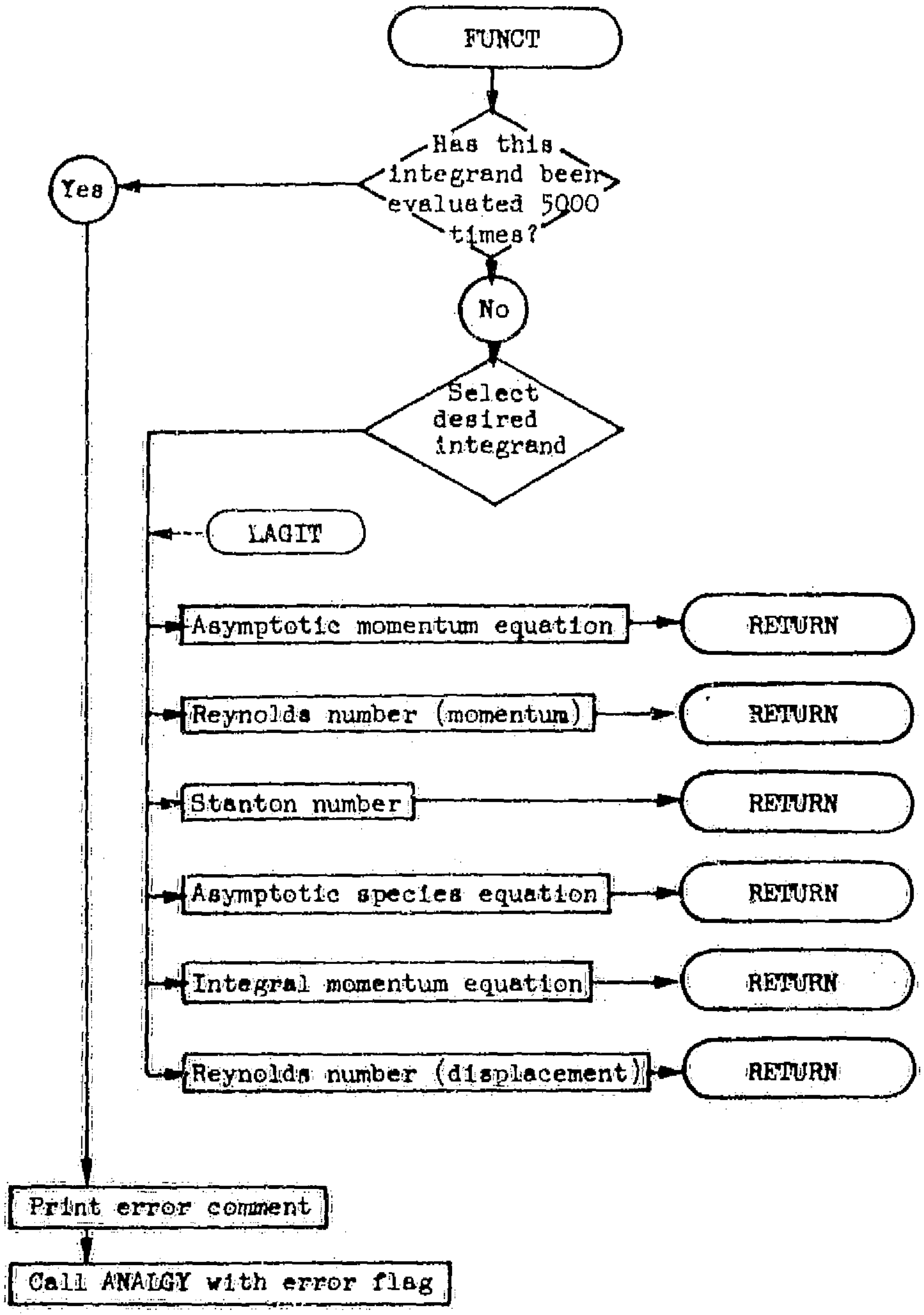
5.1.1.3 Subroutine SPECIE



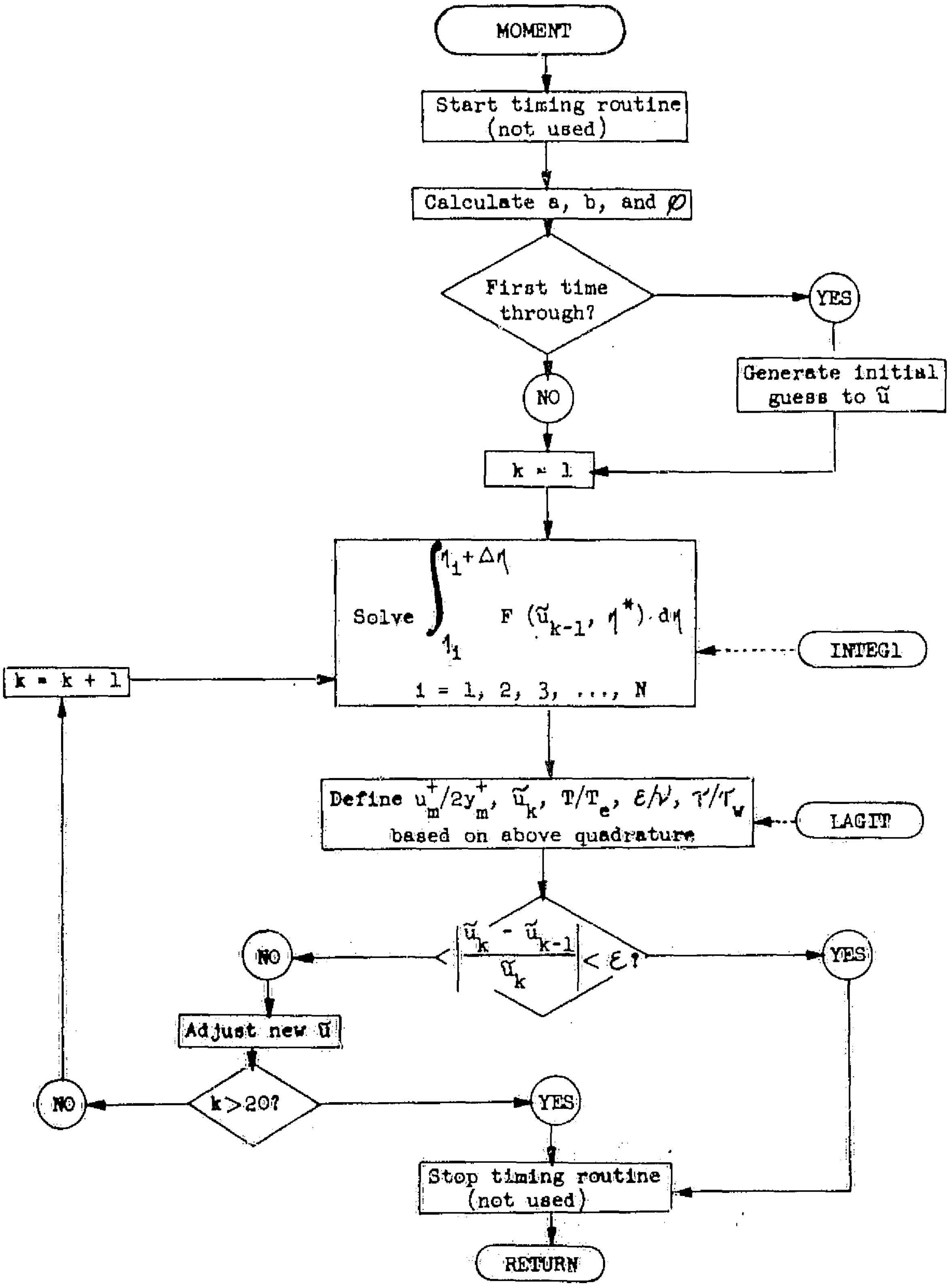
5.1.1.4 Subroutine VONKAR



5.1.1.5 Function FUNCT



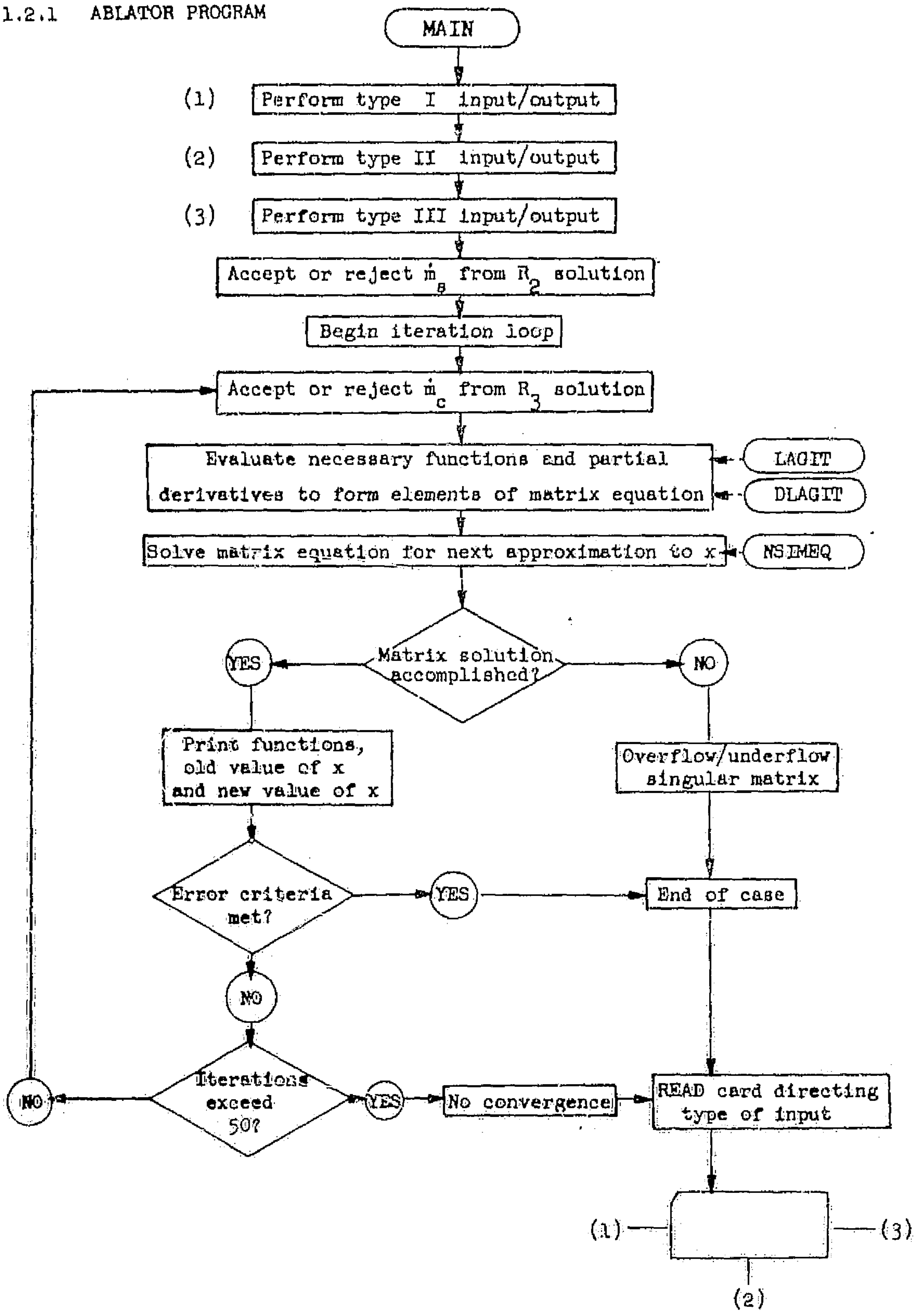
5.1.1.6 Subroutine MOMENT



5.1.2 Ablator program

A flow diagram of the ablator program MAIN follows; a listing is given in Section 7.2. This routine requires several pre-existing packages (documented in References 3 and 4). These are LAGIT, DLAGIT, and NSIMEQ (a routine for solving N simultaneous linear equations).

5.1.2.1 ABLATOR PROGRAM



6.0 SAMPLE CASES

```
****** ****** ****** ****** ****** ****** ****** ******

ERROR IN QUADRATURE

THIS QUADRATURE EXCEEDED 5000 EVALUATIONS OF THE INTEGRAND

N =  5            VARIABLE OF INTEGRATION =    1.29309+01


   ETA        U/UE          T/TE          W(1)        TAU/TAUW        EP/NU

   .0000    0.00000       2.68325-01    1.00000+00    1.00000+00    0.00000
   .2000    4.16340-02    3.1066' -01   1.06919+00    1.01744+00    0.00000
   .4000    3.15036-02    3.46624-01    1.13994+00    1.04299+00    0.00000
   .6000    1.19843-01    3.74037-01    1.21238+00    1.05969+00    0.00000
   .8000    1.56828-01    4.00474-01    1.28662+00    1.07368+00    0.00000
  1.0000    1.92590-01    4.19641-01    1.36273+00    1.08234+00    0.00000
  1.2000    2.27235-01    4.44590-01    1.44075+00    1.09222+00    0.00000
  1.4000    2.60845-01    4.74776-01    1.52071+00    1.10180+00    0.00000
  1.6000    2.93498-01    4.98077-01    1.60261+00    1.10712+00    0.00000
  1.8000    3.25219-01    5.20720-01    1.68645+00    1.11062+00    0.00000
  2.0000    3.56084-01    5.42737-01    1.77220+00    1.11238+00    0.00000
  2.2000    3.86121-01    5.64156-01    1.85954+00    1.11249+00    0.00000
  2.4000    4.15362-01    5.85000-01    1.94932+00    1.11100+00    0.00000
  2.6000    4.43834-01    6.05289-01    2.04059+00    1.10797+00    0.00000
  2.8000    4.71560-01    6.25039-01    2.13356+00    1.10345+00    0.00000
  3.0000    4.98557-01    6.44264-01    2.22816+00    1.09747+00    0.00000
  3.2000    5.24643-01    6.62977-01    2.32430+00    1.09006+00    0.00000
  3.4000    5.50431-01    6.81185-01    2.42186+00    1.08125+00    0.00000
  3.6000    5.75330-01    6.98899-01    2.52073+00    1.07107+00    0.00000
  3.8000    5.99551-01    7.16124-01    2.62078+00    1.05952+00    0.00000
  4.0000    6.23099-01    7.32865-01    2.72186+00    1.04663+00    0.00000
  4.2000    6.45980-01    7.49123-01    2.82380+00    1.03239+00    0.00000
  4.4000    6.68198-01    7.64901-01    2.92645+00    1.01681+00    0.00000
  4.6000    6.89755-01    7.80211-01    3.02960+00    9.99920-01    0.00000
  5.1000    7.40760-01    8.16348-01    3.28838+00    9.51724-01    0.00000
  5.6000    7.87623-01    8.49643-01    3.54558+00    8.95572-01    0.00000
  6.1000    8.30285-01    8.79818-01    3.79719+00    8.30667-01    0.00000
  6.6000    8.68652-01    9.07055-01    4.03867+00    7.57514-01    0.00000
  7.1000    9.02589-01    9.31011-01    4.26504+00    6.75131-01    0.00000
  7.6000    9.31896-01    9.50553-01    4.47072+00    5.87806-01    0.00000
  8.1000    9.56377-01    9.68039-01    4.65010+00    4.87195-01    0.00000
  8.6000    9.75766-01    9.81924-01    4.79727+00    3.76422-01    0.00000
  9.1000    9.89744-01    9.91985-01    4.90625+00    2.54851-01    0.00000
  9.6000    9.97921-01    9.97953-01    4.97115+00    1.21714-01    0.00000
 10.0000    1.00000+00    1.00000+00    4.98778+00    1.00000-05    0.00000
```

Sample case #1. Nonconvergent case (Flight 202, Station 4, 612 sec) too large a step size in y_m^+ array (y_m^+: 0, 1, 4, 8, 10, 12, 14, 16, 18, 20, 25, 30, 35, 40, 45, 50).

```
TURBULENT BOUNDARY LAYER ANALYSIS WITH COMBUSTION  AS 2591

INTEGRAL SOLUTION OF MOMENTUM EQUATION
ENERGY AND SPECIES EQUATIONS APPROXIMATED BY REYNOLDS ANALOGY


FLIGHT 202  STATION  4       TIME 612 SEC

    X          RE(X)        RE(M)       P(ATM)       U(EDGE)     CF/2         ST/CF/2     G(W)        Y(M)+        U(M)+        H(COEF)


0.0000       0.0000       0.0000      0.0000       0.0000      1.0000+39    0.0000      0.0000      0.0000       0.0000       0.0000
0.0000       1.6191-39    1.6191-01   1.1689-01    3.1565+01   1.5709+00    1.2752+00   1.0754+02   1.0000+00    7.9760-01    5.1877-02
8.0332-04    3.9179-01    6.4510-01   1.1689-01    3.2045+01   3.9280-01    1.2752+00   2.7301+01   2.0000+00    1.5956+00    1.3170-02
5.5155-03    2.8209+00    1.4571+00   1.1686-01    3.5036+01   1.7457-01    1.2752+00   1.3262+01   3.0000+00    2.3934+00    6.3976-03
2.0363-02    1.1982+01    2.5917+00   1.1679-01    4.4725+01   9.8179-02    1.2752+00   9.5155+00   4.0000+00    3.1915+00    4.5902-03
5.5783-02    4.3040+01    4.0530+00   1.1661-01    6.2310+01   6.2795-02    1.2753+00   9.2824+00   5.0000+00    3.9906+00    4.4777-03
1.1962-01    1.3095+02    5.8401+00   1.1628-01    1.1142+02   4.3546-02    1.2756+00   1.0472+01   6.0000+00    4.7921+00    5.0518-03
2.0517-01    3.1955+02    7.9582+00   1.1581-01    1.7156+02   3.1903-02    1.2760+00   1.1772+01   7.0000+00    5.5987+00    5.6766-03
3.0484-01    6.3534+02    1.0417+01   1.1527-01    2.4182+02   2.4338-02    1.2765+00   1.2607+01   8.0000+00    6.4100+00    6.0814-03
4.0501-01    1.0633+03    1.3217+01   1.1472-01    3.1379+02   1.9152-02    1.2772+00   1.2819+01   9.0000+00    7.2260+00    6.1840-03
4.5548-01    1.2963+03    1.6358+01   1.1444-01    3.3461+02   1.5450-02    1.2776+00   1.1608+01   1.0000+01    8.0453+00    5.3103-03
5.3123-01    1.7051+03    1.9810+01   1.1409-01    3.6394+02   1.2746-02    1.2781+00   9.8475+00   1.1000+01    8.8574+00    4.7504-03
6.5515-01    2.4259+03    2.3628+01   1.1338-01    4.1104+02   1.0675-02    1.2786+00   9.2630+00   1.2000+01    9.6757+00    4.4708-03
8.2571-01    3.5498+03    2.7813+01   1.0997-01    4.7829+02   9.0509-03    1.2794+00   8.8839+00   1.3000+01    1.0511+01    4.2855-03
1.0150+00    4.9468+03    3.2372+01   1.0629-01    5.5071+02   7.7550-03    1.2805+00   8.5690+00   1.4000+01    1.1356+01    4.1331-03
1.2363+00    6.8051+03    3.7196+01   1.0184-01    6.3216+02   6.7286-03    1.2816+00   8.3705+00   1.5000+01    1.2191+01    4.0379-03
1.5224+00    9.5658+03    4.2354+01   8.2700-02    7.4289+02   5.8891-03    1.2830+00   7.0658+00   1.6000+01    1.3031+01    3.4865-03
2.0616+00    1.3280+04    4.7788+01   3.5552-02    9.0751+02   5.1961-03    1.2848+00   3.3628+00   1.7000+01    1.3573+01    1.6222-03
2.2802+00    1.4105+04    5.3795+01   2.2587-02    9.0751+02   4.5783-03    1.2881+00   1.8873+00   1.8000+01    1.4779+01    9.1043-04
3.3314+00    1.4890+04    6.0267+01   2.3721-03    9.0751+02   4.0736-03    1.2894+00   1.7653-01   1.9000+01    1.5686+01    3.5159-05
5.7849+00    1.5597+04    6.6513+01   2.0175-03    9.0751+02   3.5303-03    1.2957+00   1.3075-01   2.0000+01    1.6836+01    6.3073-05
1.1020+01    1.6752+04    8.5458+01   1.3367-03    9.0751+02   2.7495-03    1.3079+00   6.8104-02   2.2000+01    1.9071+01    3.2853-05
1.7259+01    1.7480+04    1.0431+02   4.9449-04    9.0751+02   2.1420-03    1.3269+00   1.9944-02   2.4000+01    2.1567+01    3.0207-06
2.0216+01    1.7568+04    1.1295+02   4.1058-05    9.0751+02   1.9948-03    1.3265+00   1.5393-03   2.6000+01    2.2390+01    7.4253-07
2.0216+01    1.7571+04    1.1621+02  -2.4309-05    9.0751+02   1.9452-03    1.3195+00  -8.8497-04   2.8000+01    2.2673+01   -4.2647-07
2.0216+01    1.7573+04    1.2293+02  -2.0563-05    9.0751+02   1.9141-03    1.3143+00  -7.3295-04   3.0000+01    2.2857+01   -3.5357-07
2.0216+01    1.7575+04    1.2731+02  -1.8396-05    9.0751+02   1.8921-03    1.3089+00  -6.4552-04   3.2000+01    2.2950+01   -3.1139-07
2.0217+01    1.7577+04    1.3145+02  -1.7025-05    9.0751+02   1.8748-03    1.3005+00  -5.8611-04   3.4000+01    2.3056+01   -2.8370-07
2.0217+01    1.7578+04    1.3558+02  -1.6251-05    9.0751+02   1.8601-03    1.2971+00  -5.5555-04   3.6000+01    2.3136+01   -2.6799-07
2.0217+01    1.7579+04    1.3964+02  -1.5680-05    9.0751+02   1.8470-03    1.2845+00  -5.2799-04   3.8000+01    2.3206+01   -2.5426-07
2.0218+01    1.7580+04    1.4367+02  -1.5154-05    9.0751+02   1.8350-03    1.2806+00  -5.0455-04   4.0000+01    2.3344+01   -2.4339-07
2.0218+01    1.8237+04    1.4757+02   0.0000      9.0751+02   1.8237-03    1.2771+00   0.0000      4.2000+01    2.3415+01    0.0000
2.0218+01    1.9340+04    1.5169+02   0.0000      9.0751+02   1.8130-03    1.2740+00   0.0000      4.4000+01    2.3486+01    0.0000
2.0218+01    2.0443+04    1.5568+02   0.0000      9.0751+02   1.8026-03    1.2712+00   0.0000      4.6000+01    2.3553+01    0.0000
2.0218+01    2.1554+04    1.5965+02   0.0000      9.0751+02   1.7926-03    1.2691+00   0.0000      4.8000+01    2.3619+01    0.0000
2.0218+01    2.2669+04    1.6367+02   0.0000      9.0751+02   1.7829-03    1.2671+00   0.0000      5.0000+01    2.3683+01    0.0000
2.0218+01    2.5474+04    1.7364+02   0.0000      9.0751+02   1.7598-03    1.2621+00   0.0000      5.5000+01    2.3835+01    0.0000
2.0218+01    2.9323+04    1.8363+02   0.0000      9.0751+02   1.7382-03    1.2542+00   0.0000      6.0000+01    2.3956+01    0.0000
2.0218+01    3.1215+04    1.9365+02   0.0000      9.0751+02   1.7178-03    1.2510+00   0.0000      6.5000+01    2.4128+01    0.0000
2.0218+01    3.4143+04    2.0368+02   0.0000      9.0751+02   1.6985-03    1.2509+00   0.0000      7.0000+01    2.4264+01    0.0000
2.0218+01    3.7105+04    2.1371+02   0.0000      9.0751+02   1.6804-03    1.2476+00   0.0000      7.5000+01    2.4395+01    0.0000
2.0218+01    4.0125+04    2.2383+02   0.0000      9.0751+02   1.6632-03    1.2448+00   0.0000      8.0000+01    2.4521+01    0.0000
2.0218+01    4.3176+04    2.3396+02   0.0000      9.0751+02   1.6469-03    1.2422+00   0.0000      8.5000+01    2.4642+01    0.0000
2.0218+01    4.6270+04    2.4412+02   0.0000      9.0751+02   1.6314-03    1.2391+00   0.0000      9.0000+01    2.4759+01    0.0000
2.0218+01    5.2561+04    2.6454+02   0.0000      9.0751+02   1.6025-03    1.2355+00   0.0000      1.0000+02    2.4980+01    0.0000
2.0218+01    5.9000+04    2.8509+02   0.0000      9.0751+02   1.5763-03    1.2324+00   0.0000      1.1000+02    2.5167+01    0.0000
2.0218+01    6.5593+04    3.0580+02   0.0000      9.0751+02   1.5523-03    1.2292+00   0.0000      1.2000+02    2.5321+01    0.0000
```

Sample case #2. Convergent case for same conditions as sample case #1., except smaller step size in y_m^+ array.

```
TURBULENT BOUNDARY LAYER ANALYSIS WITH COMPUSTION  AS 2591

INTEGRAL SOLUTION OF MOMENTUM EQUATION
ENERGY AND SPECIES EQUATIONS APPROXIMATED BY REYNOLDS ANALOGY


FLIGHT 501  STATION  4       TIME 120 SEC

    X          RE(X)        RE(M)       RE(D)        RE(DEL)      CF/2         ST/CF/2      Q(N)         Y(M)+        U(M)+        H(COEF)


0.0000       0.0000       0.0000       0.0000      0.0000       1.0000+38    0.0000       0.0000       0.0000       0.0000       0.0000
0.0000       5.4925-39    3.6617-01  -2.1128-01    9.8054-01    1.0401+00    1.3076+00    3.3894+02    1.0000+00    9.8054-01    3.1677-02
3.7022-03    9.3895-01    1.4660+00  -8.4635-01    3.9223+00    2.6000-01    1.3076+00    9.0851+01    2.0000+00    1.9612+00    8.4906-03
2.3766-02    7.2559+00    3.2955+00  -1.9015+00    8.8252+00    1.1556-01    1.3076+00    5.5908+01    3.0000+00    2.9417+00    5.2250-03
7.1773-02    3.0961+01    5.8614+00  -3.3827+00    1.5689+01    6.5002-02    1.3076+00    5.3096+01    4.0000+00    3.9223+00    4.9622-03
1.5220-01    9.8343+01    9.1537+00  -5.2833+00    2.4513+01    4.1604-02    1.3076+00    5.7630+01    5.0000+00    4.9027+00    5.3860-03
2.6466-01    2.5112+02    1.3177+01  -7.6013+00    3.5297+01    2.8896-02    1.3076+00    6.3254+01    6.0000+00    5.8823+00    5.9116-03
3.9955-01    5.2498+02    1.7933+01  -1.0329+01    4.8035+01    2.1236-02    1.3076+00    6.7168+01    7.0000+00    6.8622+00    6.277[illegible]-03
5.1943-01    8.3692+02    2.3401+01  -1.3457+01    6.2724+01    1.6267-02    1.3076+00    5.9945+01    8.0000+00    7.8405+00    5.6523-03
6.9520-01    1.3718+03    2.9599+01  -1.6999+01    7.9371+01    1.2858-02    1.3076+00    5.6141+01    9.0000+00    8.8190+00    5.2468-03
9.2225-01    2.1953+03    3.6511+01  -2.0919+01    9.7960+01    1.0421-02    1.3076+00    5.4454+01    1.0000+01    9.7960+00    5.0592-03
1.1927+00    3.3679+03    4.4104+01  -2.5177+01    1.1847+02    8.6206-03    1.3076+00    5.3706+01    1.1000+01    1.0770+01    5.0[illegible]-03
1.4977+00    4.9270+03    5.2327+01  -2.9717+01    1.4069+02    7.2539-03    1.3076+00    5.2237+01    1.2000+01    1.1741+01    4.8520-03
1.8038+00    6.7199+03    6.1152+01  -3.4425+01    1.6518±02    6.1944-03    1.3076+00    5.1010+01    1.3000+01    1.2706+01    4.7673-03
2.0866+00    8.6050+03    6.9647+01  -3.7915+01    1.9101+02    5.3719-03    1.3075+00    4.3760+01    1.4000+01    1.3644+01    4.0597-03
2.4249+00    1.0484+04    7.7802+01  -4.0127+01    2.1826+02    4.7222-03    1.3076+00    2.5201+01    1.5000+01    1.4552+01    2.3552-03
2.6640+00    1.1264+04    8.6216+01  -4.1955+01    2.4729+02    4.1862-03    1.3076+00    1.2940+01    1.6000+01    1.5456+01    1.2094-03
3.1717+00    1.1968+04    9.7302+01  -4.7338+01    2.7917+02    3.7081-03    1.3076+00    3.0817+00    1.7000+01    1.6422+01    2.8501-04
3.8741+00    1.2261+04    1.0909+02  -5.3071+01    3.1298+02    3.3076-03    1.3076+00    1.5632+00    1.8000+01    1.7388+01    1.4609-04
4.5680+00    1.2516+04    1.2154+02  -5.9131+01    3.4872+02    2.9666-03    1.3076+00    1.2752+00    1.9000+01    1.8354+01    1.1918-04
5.2443+00    1.2745+04    1.3467+02  -6.5519+01    3.8639+02    2.6792-03    1.3076+00    1.0770+00    2.0000+01    1.9320+01    1.0066-04
6.5346+00    1.3142+04    1.6300+02  -7.9313+01    4.6754+02    2.2142-03    1.3076+00    7.7908-01    2.2000+01    2.1252+01    7.2811-05
7.7384+00    1.3462+04    1.9399+02  -9.4388+01    5.5641+02    1.8605-03    1.3076+00    5.6316-01    2.4000+01    2.3184+01    5.2632-05
8.8612+00    1.3717+04    2.2766+02  -1.1077+02    6.5300+02    1.5853-03    1.3076+00    4.0715-01    2.6000+01    2.5116+01    3.8051-05
9.9105+00    1.3918+04    2.6407+02  -1.2851+02    7.5733+02    1.3669-03    1.3076+00    2.9249-01    2.8000+01    2.7047+01    2.7335-05
1.0891+01    1.4072+04    3.0314+02  -1.4753+02    8.6939+02    1.1907-03    1.3076+00    2.0709-01    3.0000+01    2.8980+01    1.9354-05
1.1812+01    1.4187+04    3.4491+02  -1.6786+02    9.8917+02    1.[illegible]66-03  1.3076+00    1.4268-01    3.2000+01    3.0911+01    1.3335-05
1.2675+01    1.4270+04    3.8920+02  -1.8935+02    1.1167+03    9.2705-04    1.3076+00    9.3708-02    3.4000+01    3.2843+01    8.7578-06
1.3493+01    1.4326+04    4.3634+02  -2.1228+02    1.2519+03    8.2690-04    1.3076+00    5.5992-02    3.6000+01    3.4775+01    5.2329-06
1.4268+01    1.4357+04    4.8617+02  -2.3653+02    1.3949+03    7.4215-04    1.3076+00    2.6805-02    3.8000+01    3.6707+01    2.5052-06
1.5003+01    1.4369+04    5.3869+02  -2.6208+02    1.5456+03    6.6979-04    1.3076+00    4.1093-03    4.0000+01    3.8639+01    3.2405-07
1.5146+01    1.4370+04    5.9391+02  -2.8894+02    1.7040+03    6.0752-04    1.3076+00    1.2620-04    4.2000+01    4.0571+01    1.1794-08
1.5143+01    1.4371+04    6.5182+02  -3.1711+02    1.8701+03    5.5355-04    1.3076+00    1.1778-04    4.4000+01    4.2503+01    1.1008-08
1.5140+01    1.4372+04    7.1271+02  -3.4681+02    2.0440+03    5.0646-04    1.3076+00    1.1607-04    4.6000+01    4.4435+01    1.0848-08
1.5137+01    1.4372+04    7.7603+02  -3.7762+02    2.2256+03    4.6513-04    1.3076+00    1.0326-04    4.8000+01    4.6367+01    9.6504-09
1.5135+01    1.4373+04    8.4204+02  -4.0975+02    2.4150+03    4.2867-04    1.3076+00    9.6703-05    5.0000+01    4.8299+01    9.0376-09
```

Sample case #3. Convergent case for input identical to case #4 except for step size in y_m^+ array.

```
TURBULENT BOUNDARY LAYER ANALYSIS WITH COMBUSTION  AS 2591

INTEGRAL SOLUTION OF MOMENTUM EQUATION
ENERGY AND SPECIES EQUATIONS APPROXIMATED BY REYNOLDS ANALOGY

FLIGHT 501  STATION  4       TIME 120 SEC

     X          RE(X)         RE(M)       RE(D)        RE(DEL)     CF/2        ST/CF/2     Q(W)         Y(M)+       U(M)+       H(COEF)


0.0000      0.0000      0.0000      0.0000      0.0000      1.0000+38   0.0000      0.0000      0.0000      0.0000      0.0000
0.0000      5.4925-39   3.6617-01  -2.1128-01   9.8054-01   1.0401+00   1.3076+00   3.3894+02   1.0000+00   9.8054-01   3.1677-02
3.7022-03   9.3895-01   1.4660+00  -2.4635-01   3.9223+00   2.6000-01   1.3076+00   9.0851+01   2.0000+00   1.9612+00   8.4908-03
5.0816-02   1.8907+01   5.8638+00  -3.3853+00   1.5690+01   6.4997-02   1.3076+00   4.2789+01   4.0000+00   3.9224+00   3.9990-03
2.3239-01   1.9474+02   1.3181+01  -7.6040+00   3.5300+01   2.8890-02   1.3076+00   5.5245+01   6.0000+00   5.8833+00   5.1631-03
4.9422-01   7.4778+02   2.3430+01  -1.3503+01   6.2744+01   1.6257-02   1.3076+00   5.7801+01   8.0000+00   7.8430+00   5.4020-03
8.7976-01   1.9991+03   3.6543+01  -2.0998+01   9.7992+01   1.0414-02   1.3076+00   5.2134+01   1.0000+01   9.7992+00   4.5724-03
1.4542+00   4.6377+03   5.2457+01  -2.9997+01   1.4101+02   7.2424-03   1.3076+00   5.0807+01   1.2000+01   1.1751+01   4.7483-03
2.0831+00   8.4968+03   7.0969+01  -3.9990+01   1.9161+02   5.3383-03   1.3076+00   4.4356+01   1.4000+01   1.3687+01   4.1455-03
2.6867+00   1.1312+04   8.5590+01  -4.5746+01   2.4834+02   4.1496-03   1.3076+00   1.2970+01   1.6000+01   1.5524+01   1.2121-03
3.7337+00   1.2241+04   1.0912+02  -5.7069+01   3.1298+02   3.707 -03   1.3076+00   1.6141+00   1.6000+01   1.7388+01   1.5285-04
5.0885+00   1.2731+04   1.3472+02  -6.5557+01   3.6640+02   2.6791-03   1.3076+00   1.1006+00   2.0000+01   1.9320+01   1.0286-04
6.3639+00   1.3137+04   1.6200+02  -7.9313+01   4.6754+02   2.2142-03   1.3076+00   7.9430-01   2.2000+01   2.1252+01   7.4233-05
7.5566+00   1.3454+04   1.9399+02  -9.4396+01   5.5641+02   1.9605-03   1.3076+00   5.7651-01   2.4000+01   2.3164+01   5.3906-05
8.6690+00   1.3714+04   2.2765+02  -1.1077+02   6.5300+02   1.5853-03   1.3076+00   4.1945-01   2.6000+01   2.5116+01   3.9201-05
9.7087+00   1.3910+04   2.6407+02  -1.2851+02   7.5733+02   1.3669-03   1.3076+00   3.0363-01   2.8000+01   2.7047+01   2.8377-05
1.0681+01   1.4078+04   3.0314+02  -1.4753+02   8.6939+02   1.1907-03   1.3076+00   2.1723-01   3.0000+01   2.8980+01   2.0302-05
1.1593+01   1.4199+04   3.4491+02  -1.6786+02   9.8917+02   1.0466-03   1.3076+00   1.5195-01   3.2000+01   3.0911+01   1.4201-05
1.2448+01   1.4282+04   3.8920+02  -1.8935+02   1.1167+03   9.2705-04   1.3076+00   1.0222-01   3.4000+01   3.2843+01   9.5530-06
1.3259+01   1.4349+04   4.3634+02  -2.1226+02   1.2519+03   8.2590-04   1.3076+00   6.3330-02   3.6000+01   3.4775+01   5.9655-06
1.4027+01   1.4395+04   4.8617+02  -2.3653+02   1.3949+03   7.4215-04   1.3076+00   3.4051-02   3.8000+01   3.6707+01   3.1624-06
1.4755+01   1.4404+04   5.3869+02  -2.6206+02   1.5456+03   6.6979-04   1.3076+00   1.0525-02   4.0000+01   3.8639+01   1.0120-06
1.5449+01   1.4404+04   5.9391+02  -2.8894+02   1.7049+03   6.0752-04   1.3076+00  -7.3521-03   4.2000+01   4.0571+01  -6.6711-07
1.6110+01   1.4399+04   6.5182+02  -3.1711+02   1.8701+03   5.5355-04   1.3076+00  -2.1620-02   4.4000+01   4.2503+01  -2.0206-06
1.6745+01   1.4362+04   7.1271+02  -3.4681+02   2.0440+03   5.0646-04   1.3076+00  -3.2879-02   4.6000+01   4.4435+01  -3.0728-06
1.7349+01   1.4323+04   7.7602+02  -3.7762+02   2.2256+03   4.6513-04   1.3076+00  -4.1070-02   4.8000+01   4.6367+01  -3.8944-06
1.7930+01   1.4275+04   8.4204+02  -4.0975+02   2.4150+03   4.2867-04   1.3076+00  -4.8540-02   5.0000+01   4.8299+01  -4.5365-06
```

Sample case #4. Convergent case for input identical to case #3 except for step size in y_m^+ array.

SAMPLE CASE # 5: Input of first iteration for flow field program

(Combustion routine not used)

```
FLIGHT 202  STATIONS 1,2,5  TIME 52  SEC
  26     1
1.0       2.0       4.0       6.0       8.0       10.0      12.0
14.0      16.0      18.0      20.0      22.0      24.0      26.0
28.0      30.0      32.0      34.0      36.0      38.0      40.0
42.0      44.0      46.0      48.0      50.0
0.02      0.05      0.9       0.36      20.0      0.72      0.72      24.  ◄
0.00363   15100.    27287.    10.0
  20
0.0       1.0       0.642     0.998     1.283     0.993     1.925     0.986
2.567     0.974     3.208     0.96      3.850     0.943     4.492     0.92
4.524000  0.918000  5.133     0.89      6.417     0.832     7.70      0.767
8.983     0.685     10.267    0.565     10.4590000.539000 10.908    0.460
11.229    0.292     11.55     0.1       12.192    0.02      16.36300  0.010000
   2
0.00      535.      22.0      535.
   2
0.0       0.0       20.0      0.0
```

SAMPLE CASE # 6: Input of first iteration for ablator program

```
0.00099022 0.7      14.8      0.00341   535.      4365.     0.000001  0.000001
0.43      0.62      2130.     6154.     250.      0.75      4.8060E-131.866
0.23      0.12      1.10      0.50
0.00      103600.   0.00      1.00      126800.
0.0500    1000000.  7500.     0.05      0.05
1.0000E-13-7.00     200.      1.0000E-171.0000E-17
1.0000E-0910.0      600.      1.0000E-111.0000E-11
```

```
TURBULENT BOUNDARY LAYER ANALYSIS WITH COMBUSTION  AS 2591

INTEGRAL SOLUTION OF MOMENTUM EQUATION
ENERGY AND SPECIES EQUATIONS APPROXIMATED BY REYNOLDS ANALOGY


FLIGHT 202  STATIONS 1,2,5  TIME 52  SEC .

    X          RE(X)       RE(M)       P(ATM)      U(EDGE)     CF/2        ST/CF/2     G(M)        Y(M)+       U(M)+       H(COEF)


0.0000      0.0000      0.0000      3.6266-03   8.0445+01   1.0000+38   0.0000      0.0000      0.0000      0.0000      0.0000
0.0000      2.8629-39   2.8629-01   3.6266-03   8.0445+01   1.5097+00   1.2446+00   1.8511+01   1.0000+00   8.1367-01   1.2369-03
1.1106-01   1.7551+00   1.1463+00   3.6254-03   2.8147+02   3.7747-01   1.2446+00   1.4037+01   2.0000+00   1.0276+00   9.3791-04
7.0462-01   4.0990+01   4.5850+00   3.6176-03   1.1107+03   9.4344-02   1.2446+00   1.4206+01   4.0000+00   3.2557+00   9.4918-04
2.0594+00   2.5634+02   1.0258+01   3.5667-03   2.4730+03   4.2060-02   1.2446+00   1.4696+01   6.0000+00   4.6750+00   9.5195-04
3.9465+00   8.3369+02   1.7892+01   3.4081-03   4.3650+03   2.3931-02   1.2446+00   1.4619+01   8.0000+00   6.4642+00   9.9318-04
6.2014+00   1.9441+03   2.6555+01   3.0526-03   6.6381+03   1.5796-02   1.2446+00   1.4187+01   1.0000+01   7.9561+00   9.4794-04
8.6757+00   3.5921+03   3.4491+01   2.5554-03   9.1261+03   1.1679-02   1.2446+00   1.2952+01   1.2000+01   9.2534+00   8.6541-04
1.0456+01   4.9977+03   4.0064+01   1.9562-03   1.0457+04   9.4578-03   1.2446+00   9.8166+00   1.4000+01   1.0263+01   6.5591-04
1.2555+01   5.5571+03   4.7019+01   6.9380-03   1.0458+04   7.7257-03   1.2446+00   7.1695-01   1.6000+01   1.1377+01   4.7934-05
```

Sample #7. Output for sample case #5.

Sample #8. Output for sample case #6

```
TABLE 1             (PHI1,LAMDA)
   0.00000          1.00000+06     1.00000-05     1.00000+03     1.00000-04     1.60000+01     4.00000-04     1.20000+01
   1.00000-03       8.00000+00     4.00000-03     5.00000+00     1.00000-02     2.15000+00     2.00000-02     1.30000+00
   3.00000-02       1.00000+00     4.00000-02     8.00000-01     7.00000-02     6.50000-01     1.00000-01     5.00000-01
   2.00000-01       2.70000-01     5.00000-01    -5.00000-02     1.00000+00    *2.00000-01     1.00000+01    -7.00000-01

TABLE 2             (PHI2,M DOT P)
   0.00000          1.00000-12     2.00000+02     1.00000-12     5.00000+02     1.00000-11     8.00000+02     1.00000-10
   1.06000+03       1.00000-07     2.85700+03     1.40000-03     3.90300+03     2.05000-03     4.34800+03     2.30000-03
   5.37600+03       3.72000-03     7.54700+03     5.30000-03     1.21210+04     6.60000-03     2.16220+04     7.40000-03
   4.07640+04       7.85000-03     7.95030+04     8.05000-03     1.56096+05     8.20000-03     1.00000+06     1.00000-02

TABLE 3             (PHI3,M DOT S)
   0.00000          1.00000-17     1.00000-05     1.00000-16     1.00000-04     1.00000-15     1.00000-03     1.00000-14
   1.00000-02       1.00000-12     1.00000-01     1.00000-10     4.00000-01     1.00000-07     1.00000+00     2.00000-04
   1.35000+00       2.50000-04     2.60000+00     3.00000-04     4.00000+00     4.20000-04     6.00000+00     6.30000-04
   8.00000+00       9.00000-04     1.00000+01     1.30000-03     3.00000+01     6.50000-03     1.00000+02     6.70000-02

PARAMETER INPUT CARDS
  9.902000-04     7.000000-01    1.480000+01    3.410000-03    5.350000+02    4.365000+03    1.000000-06    1.000000-06
 -4.300000-01     6.200000-01    2.130000+03    6.154000+03    2.500000+02    7.500000-01    4.866000-13    1.800000+00
  2.300000-01     1.200000-01    1.100000+00    5.000000-01
  0.000000        1.036000+05    0.000000       1.000000+00    1.268000+05
UPPER BOUND                  1.000000-02       1.000000+06       7.580000+03       5.000000-02       5.000000-02
LOWER BOUND                  1.000000-13      -7.000000-01       2.000000+02       1.000000-17       1.000000-17

ITERATION   1        CASE 1          (X) = ( M DOT P , LAMDA , T * , M DOT S , M DOT C )
F               1.0313291+01    -9.9999001+05    9.6000001-10     9.9997091-12     -4.2496403-04
   X            9.9999999-10     1.0000000+01    6.0000000+02     9.9999999-12      9.9999999-12
DELX           -2.0031194-06     9.9999001+05    2.9332229+04     0.0000000        -5.5499475-04
X1              5.0004999-10     1.0000000+06    4.0499999+03     9.9999999-12      5.0000049-12

ITERATION   2        CASE 1          (X) = ( M DOT P , LAMDA , T * , M DOT S , M DOT C )
F              -1.7262693-01    -4.9987897+05   -9.8556027-04    -1.7896929-08     -4.9957760-05
   X            7.5002499-10     5.0000500+05    2.3249999+03     9.9999999-12      7.5000024-12
DELX            3.4700751-08     4.9467865+05   -1.3347065+03     1.8688655-05     -1.7071630-02
X1              3.5450777-08     9.9468365+05    9.9029343+02     1.8688665-05      3.7500062-12

ITERATION   3        CASE 1          (X) = ( M DOT P , LAMDA , T * , M DOT S , M DOT C )
F               7.6539581+00    -2.5259551+05   -4.6566097-04     9.3442965-06      1.3162256-02
   X            1.6100400-08     7.4734432+05    1.6576457+03     9.3443374-06      5.6250042-12
DELX           -1.4392312-08     2.5295226+05   -5.7324550+02    -9.0599127-06     -6.9241524-03
X1              3.7080891-09     8.7367216+05    1.0844012+03     2.8442468-07      2.8125071-12

ITERATION   4        CASE 1          (X) = ( M DOT P , LAMDA , T * , M DOT S , M DOT C )
F               9.0950056+00    -1.8946948+05   -2.4236315-04     4.8143698-06      8.4130063-03
   X            1.0904245-08     8.1050824+05    1.3710240+03     4.8143810-06      4.2187557-12
DELX           -5.8257151-09     1.8949918+05   -2.8714595+02     1.2681025-05     -5.7525635-03
X1              5.0785299-09     9.0525413+05    1.0838780+03     1.7495406-05      2.1093826-12

ITERATION   5        CASE 3          (X) = ( M DOT P , LAMDA , T * , M DOT S )
F               9.5622780+00    -1.4211857+05   -1.3053976-04     1.1154892-05
   X̄            7.9913874-09     8.5788118+05    1.2274510+03     1.1154893-05
   X           -4.2052005-03     2.9162868+05   -5.0466045+03    -1.1136243-05
                3.9957436-09     9.2894059+05    7.1372548+02     1.8649757-08

       ON.  6        CASE 3          (X) = ( M DOT P , LAMDA , T * , M DOT S )
                1.0051009+01    -1.0658913+05   -5.9651679-08     5.5867714-06
```

```
   X          5.9935655-09     8.9341068+05      9.7056823+02      5.5867715-06
DELX          2.7277876-04     1.0649954+05      8.7658116+03     -5.5845036-06
X1            2.7278475-04     9.9991042+05      4.2352940+03      2.2676819-09

ITERATION   7      CASE 3         (X) = ( M DOT P , LAMDA , T W , M DOT S )
F            -5.5524874+00     9.4664993+05     -1.0692123-03      2.7574529-05
   X          1.3639537-04     9.4666065+05      2.6029411+03      2.7945197-06
DELX          1.1098188-03    -9.4668344+05     -1.6553652+01     -0.0000000
X1            1.2462142-03     4.7332998+05      2.5863875+03      2.7945197-06

ITERATION   8      CASE 3         (X) = ( M DOT P , LAMDA , T W , M DOT S )
F            -5.0498496+00     7.0998923+05     -5.9813770-04      2.7580473-06
   X          6.9130478-04     7.0999531+05      2.5946642+03      2.7945197-06
DELX          5.3315396-04    -7.0998893+05     -2.6415862+02     -0.0000000
X1            1.2244587-03     6.3828125+00      2.3305057+03      2.7945197-06

ITERATION   9      CASE 3         (X) = ( M DOT P , LAMDA , T W , M DOT S )
F            -3.1973307+00     3.5499439+05     -3.0964658-04      2.7673478-06
   X          9.5786176-04     3.5500085+05      2.4625850+03      2.7945197-06
DELX          2.8112192-04    -3.5499360+05     -1.6287767+02     -0.0000000
X1            1.2390037-03     7.2392812+00      2.2997071+03      2.7945197-06

ITERATION  10      CASE 3         (X) = ( M DOT P , LAMDA , T W , M DOT S )
F            -1.6764576+00     1.7749717+05     -1.5631691-04      2.7726887-06
   X          1.0984427-03     1.7750404+05      2.3811460+03      2.7945197-06
DELX          1.3688876-04    -1.7749663+05     -9.2846604+01     -1.4617479-06
X1            1.2353315-03     7.4023438+00      2.2882994+03      1.3327718-06

ITERATION  11      CASE 3         (X) = ( M DOT P , LAMDA , T W , M DOT S )
F            -8.6169728-01     8.8746604+04     -7.7811186-05      2.0456983-06
   X          1.1666871-03     8.8755722+04      2.3347227+03      2.0636457-06
DELX          6.4413029-05    -8.8748302+04     -4.9921842+01     -1.4462310-06
X1            1.2313001-03     7.4199219+00      2.2848009+03      6.2341475-07

ITERATION  12      CASE 3         (X) = ( M DOT P , LAMDA , T W , M DOT S )
F            -4.3737794-01     4.4374314+04     -3.8556492-05      1.3266167-06
   X          1.1990936-03     4.4381571+04      2.3097619+03      1.3435302-06
DELX          3.0868445-05    -4.4374157+04     -2.5957023+01     -1.0716912-06
X1            1.2299620-03     7.4140625+00      2.2838048+03      2.7193901-07

ITERATION  13      CASE 3         (X) = ( M DOT P , LAMDA , T W , M DOT S )
F            -2.2052537-01     2.2187164+04     -1.8949838-05      7.9161375-07
   X          1.2145278-03     2.2194492+04      2.2967853+03      8.0766462-07
DELX          1.4236455-05    -2.2187082+04     -1.3206538+01     -7.1155556-07
X1            1.2287643-03     7.4096680+00      2.2835768+03      9.6129049-08

ITERATION  14      CASE 3         (X) = ( M DOT P , LAMDA , T W , M DOT S )
F            -1.1071023-01     1.1093586+04     -9.3919952-06      4.3626262-07
   X          1.2216460-03     1.1100951+04      2.2901809+03      4.5196684-07
DELX          7.0436869-06    -1.1093545+04     -6.6766007+00     -4.4396466-07
X1            1.2286897-03     7.4061279+00      2.2635033+03      8.1027771-09

ITERATION  15      CASE 3         (X) = ( M DOT P , LAMDA , T W , M DOT S )
F            -5.5501927-02     5.5467947+03     -4.6124709-06      2.1457613-07
   X          1.2251679-03     5.5541786+03      2.2862416+03      2.3090480-07
DELX          3.3137158-06    -5.5467740+03     -3.3446324+00     -1.7735651-07
X1            1.2284816-03     7.4045410+00      2.2834970+03      5.2060287-08
```

```
ITERATION  16       CASE 3            (X) = ( M DOT P , LAMDA , T W , M DOT S )
F               -2.7761684-02       2.7733963+03       -2.2692256-06       1.2601561-07
   X             1.2268247-03       2.7807915+03        2.2851693+03       1.4133655-07
DELX             1.6248953-06      -2.7733880+03       -1.6755226+00      -1.1077746-07
X1               1.2284496-03       7.4035950+00        2.2834938+03       3.0559084-08

ITERATION  17       CASE 3            (X) = ( M DOT P , LAMDA , T W , M DOT S )
F               -1.3878832-02       1.3866996+03       -1.1071973-06       7.6080829-08
   X             1.2276371-03       1.3940976+03        2.2843315+03       8.5947815-08
DELX             7.7262686-07      -1.3866944+03       -8.3590862-01      -6.6449098-08
X1               1.2284098-03       7.4031677+00        2.2834957+03       1.9498717-08

ITERATION  18       CASE 3            (X) = ( M DOT P , LAMDA , T W , M DOT S )
F               -6.9406753-03       6.9335006+02       -5.3868280-07       3.7483186-08
   X             1.2280235-03       7.0075037+02        2.2839136+03       5.2723266-08
DELX             3.6040845-07      -6.9334742+02       -4.1638093-01      -3.3220216-08
X1               1.2283839-03       7.4029465+00        2.2834972+03       1.9503050-08

ITERATION  19       CASE 3            (X) = ( M DOT P , LAMDA , T W , M DOT S )
F               -3.4698001-03       3.4667515+02       -2.6189081-07       2.0886477-08
   X             1.2282037-03       3.5407666+02        2.2837054+03       3.6113159-08
DELX             1.6442975-07      -3.4667382+02       -2.0713189-01      -1.6609029-08
X1               1.2283681-03       7.4028359+00        2.2834983+03       1.9504129-08

ITERATION  20       CASE 3            (X) = ( M DOT P , LAMDA , T W , M DOT S )
F               -1.7335295-03       1.7333763+02       -1.2805685-07       1.2586631-08
   X             1.2282859-03       1.8073975+02        2.2836018+03       2.7806644-08
DELX             8.0919603-08      -1.7333697+02       -1.0346649-01      -9.6882870-09
X1               1.2283668-03       7.4027767+00        2.2834983+03       1.8121357-08

ITERATION  21       CASE 3            (X) = ( M DOT P , LAMDA , T W , M DOT S )
F               -8.6672205-04       8.6668850+01       -6.1700121-08       7.7478155-09
   X             1.2283263-03       9.4071262+01        2.2835501+03       2.2964501-08
DELX             3.7150474-08      -8.6665513+01       -5.1625458-02      -6.9200934-09
X1               1.2283635-03       7.4027491+00        2.2834985+03       1.6044407-08

ITERATION  22       CASE 3            (X) = ( M DOT P , LAMDA , T W , M DOT S )
F               -4.3352146-04       4.3334440+01       -2.9642251-08       3.2894292-09
   X             1.2283449-03       5.0737005+01        2.2835243+03       1.9504454-08
DELX             1.7250064-08      -4.3334270+01       -2.5785761-02      -3.8060206-09
X1               1.2283621-03       7.4027352+00        2.2834985+03       1.5698433-08

ITERATION  23       CASE 3            (X) = ( M DOT P , LAMDA , T W , M DOT S )
F               -2.1657506-04       2.1667228+01       -1.4971702-08       2.3872466-09
   X             1.2283535-03       2.9069870+01        2.2835114+03       1.7601443-08
DELX             6.9667005-09      -2.1667141+01       -1.2782492-02      -2.2490032-09
X1               1.2283605-03       7.4027286+00        2.2834986+03       1.5352446-08

ITERATION  24       CASE 3            (X) = ( M DOT P , LAMDA , T W , M DOT S )
F               -1.0820277-04       1.0833618+01       -6.7811925-09       1.2631571-09
   X             1.2283570-03       1.8236299+01        2.2835050+03       1.6476942-08
DELX             3.1516755-09      -1.0833575+01       -6.3636456-03      -1.2109993-09
X1               1.2283601-03       7.4027248+00        2.2834986+03       1.5265942-08

ITERATION  25       CASE 3            (X) = ( M DOT P , LAMDA , T W , M DOT S )
F               -5.4505653-05       5.4168113+00       -3.2596290-09       6.5786220-10
   X             1.2283586-03       1.2819512+01        2.2835018+03       1.5871442-08
DELX             1.4099763-09      -5.4167892+00       -3.1921959-03      -6.4874897-10
```

```
X1              1.2283599-03      7.4027230+00      2.2834986+03      1.5222693-08

ITERATION  26      CASE 3          (X) = ( M DOT P , LAMDA , T W , M DOT S )
F              -2.7529076-05      2.7084068+00     -1.5576549-09      3.3359004-10
   X            1.2283592-03      1.0111118+01      2.2835002+03      1.5547068-08
DELX            5.5984673-10     -2.7083955+00     -1.6030097-03     -3.6274937-10
X1              1.2283598-03      7.4027220+00      2.2834986+03      1.5244319-08

ITERATION  27      CASE 3          (X) = ( M DOT P , LAMDA , T W , M DOT S )
F              -1.3706191-05      1.3542037+00     -7.4214768-10      1.8226731-10
   X            1.2283595-03      8.7569197+00      2.2834995+03      1.5395693-08
DELX            2.2808637-10     -1.3541960+00     -7.9459582-04     -1.7299960-10
X1              1.2283597-03      7.4027218+00      2.2834986+03      1.5222693-08

ITERATION  28      CASE 3          (X) = ( M DOT P , LAMDA , T W , M DOT S )
F              -7.0399392-06      6.7710233-01     -3.4924597-10      9.5790486-11
   X            1.2283596-03      8.0798208+00      2.2834990+03      1.5309193-08
DELX            6.2205462-11     -6.7709936-01     -4.0377571-04     -9.1906027-11
X1              1.2283597-03      7.4027214+00      2.2834986+03      1.5217287-08

ITERATION  29      CASE 3          (X) = ( M DOT P , LAMDA , T W , M DOT S )
F              -3.0714812-06      3.3855122-01     -1.3096724-10      4.9851900-11
   X            1.2283597-03      7.7412711+00      2.2834989+03      1.5263240-08
DELX           -5.1837924-12     -3.3854990-01     -1.7468160-04     -4.5953009-11
X1              1.2283597-03      7.4027212+00      2.2834987+03      1.5217287-08

ITERATION  30      CASE 3          (X) = ( M DOT P , LAMDA , T W , M DOT S )
F              -1.4299791-06      1.6927570-01     -5.8207661-11      2.6881608-11
   X            1.2283597-03      7.5719962+00      2.2834987+03      1.5240264-08
DELX           -6.4797423-12     -1.6927506-01     -8.1294965-05     -2.7031182-11
X1              1.2283597-03      7.4027211+00      2.2834987+03      1.5213232-08

ITERATION  31      CASE 3          (X) = ( M DOT P , LAMDA , T W , M DOT S )
F              -9.1317758-07      8.4637940-02     -2.9103830-11      1.3367085-11
   X            1.2283597-03      7.4873586+00      2.2834987+03      1.5226748-08
DELX           -1.8143281-11     -8.4637531-02     -5.0687898-05     -1.3515590-11
X1              1.2283596-03      7.4027211+00      2.2834987+03      1.5213232-08

ITERATION  32      CASE 3          (X) = ( M DOT P , LAMDA , T W , M DOT S )
F              -6.3987567-07      4.2319059-02     -2.9103830-11      6.6116222-12
   X            1.2283596-03      7.4450399+00      2.2834987+03      1.5219990-08
DELX            1.2959488-12     -4.2318787-02     -3.6275046-05     -6.0820157-12
X1              1.2283596-03      7.4027211+00      2.2834986+03      1.5213908-08

CASE COMPLETE
```

7.0 PROGRAM LISTINGS

7.1 Flow Field Program

```
=     FOR BLAYER                                                        BLAYER01
C                                                                       BLAYER02
      CALL ANALGY(1)                                                    BLAYER03
C                                                                       BLAYER04
      END                                                               BLAYER05
```

```
=     FOR ANALGY
      SUBROUTINE ANALGY(JENTER)
C     INTEGRAL SOLUTION OF BOUNDARY LAYER BY ANALOGY
C     APPLICABLE FOR FLAT PLATE WITH MASS TRANSPORT AT WALL
C     CONTRACT NAS 9-6288    ED DEL CASAL PRINCIPAL INV. (ANALYSIS)
C
C     REF. == CHARRING ABLATION PERFORMANCE IN TURBULENT FLOW
C             D2-113078=1
C     REF. == AN ANALYSIS OF THE TURBULENT BOUNDARY LAYER USING A
C             MODIFIED MISING LENGTH EXPRESSION D2-23990-1 DEL CASAL,
C             KOH  1965
C
      DIMENSION HEAD(12),HCOEF(100),PT(100),UET(100)
      COMMON  GAM,   RGAS,   G,   XJ,   CP,   SAA,   NLOOPU,   DETA1,
     1          DETA2,   UX,   MXLPU,   HE,   XNMU,   TAUTWX,   PR,
     2          EPNUX,   SC,   TTEX,   TIMEU(2),   AK,   ERR,   AMOL
      COMMON  /PROFIL /  ETA(1000),  U(1000),  ETAZ(1000),  UZ(1000),
     1          TTE(1000),  EPNU(1000),  TAUTW(1000),  WI(1000),
     2          NETA,  NETAZ,  NY,  NYMP
      COMMON  /FUNCTX/  YMP(100),  REM(100),  REX(100),  CF2(100),
     1          STCF2(100),  TETW,  XME,  ALPHA,  PHI,  YMPX,
     2UMP(100),  XD(100),  X,  UMPX,  TE,  UE,
     3RHOE,  XMUE,  P,  TW,  DLNUE,  DLNMUE,  FZ,  VINF,
     4RED(100),  DLNR,  S,  REDEL(100),  CF2XX,  REDXX,  ZE
     5  ,  HW,  XSTOP,  RHOW,  QZ(100),  XMUW
      COMMON/MASSD/ETAC,A(1000),WO2(1000),WIE(1000),
     1WC(1000),WP(1000),WIM(1000),BETAO2,BETAIM
C
      DATA GAM, RGAS, G, XJ, XNMU, CP  /1.4, 1545., 32., 778., .7, .25 /
      DATA AMOL/29.9/
C
      GO TO (5,55,56,57,58),JENTER
C
    5 CONTINUE
      YMP(1)=0.0
      UMP(1)=0.0
      REM(1)=0.0
      REX(1)=0.0
      CF2(1)=1.E+38
```

```
      TAUTW(1)=1.0                                                      ANALG040
      XD(1)=0.0                                                         ANALG041
      RED(1)=0.0                                                        ANALG042
      REDEL(1)=0.0                                                      ANALG043
      HCOEFF(1)=0.                                                      ANALG044
      QZ(1)=0.0                                                         ANALG045
      MXLPU=20                                                          ANALG046
      ERR=.01                                                           ANALG047
C                                                                       ANALG048
   10 CONTINUE                                                          ANALG049
      READ(5,101)HEAD                                                   ANALG050
  101 FORMAT(12A6)                                                      ANALG051
      READ(5,105)NYMP,KPOUT                                             ANALG052
 105  FORMAT(3I5)                                                       ANALG053
      NYMP=NYMP+1                                                       ANALG054
      READ(5,110)(YMP(I),I=2,NYMP)                                      ANALG055
 110  FORMAT(7F10.0)                                                    ANALG056
C                                                                       ANALG057
   20 CONTINUE                                                          ANALG058
      READ(5,102)DETA1,DETA2,UX,AK,SAA,PR,SC,XSTOP                      ANALG059
  102 FORMAT(8F10.0)                                                    ANALG060
      IF(DETA1)10,10,21                                                 ANALG061
 21   CONTINUE                                                          ANALG062
C                                                                       ANALG063
      CALL PROPX(3)                                                     ANALG064
C                                                                       ANALG065
      WRITE(6,201)                                                      ANALG066
  201 FORMAT(60H1 TURBULENT BOUNDARY LAYER ANALYSIS WITH COMBUSTION  AS ANALG067
     12591/40H0 INTEGRAL SOLUTION OF MOMENTUM EQUATION/63H  ENERGY AND SANALG068
     2PECIES EQUATIONS APPROXIMATED BY REYNOLDS ANALOGY//)              ANALG069
 202  FORMAT(2H0 12A6)                                                  ANALG070
      WRITE(6,202)HEAD                                                  ANALG071
C                                                                       ANALG072
      KSTOPY=0                                                          ANALG073
      NY=1                                                              ANALG074
      X=0.0                                                             ANALG075
   30 CONTINUE                                                          ANALG076
      NY=NY+1                                                           ANALG077
C                                                                       ANALG078
      CALL PROPX(1)                                                     ANALG079
```

```
C                                                                       ANALG080
      YMPX=YMP(NY)                                                      ANALG081
      TTE(1)=1.0/TETW                                                   ANALG082
C                                                                       ANALG083
      CALL MOMENT                                                       ANALG084
      CALL SPFCIE                                                       ANALG085
C                                                                       ANALG086
      IF(KPOUT)40,40,50                                                 ANALG087
   40 CONTINUE                                                          ANALG088
      REXA = REX(NY=1)                                                  ANALG089
      WRITE(6,205)YMPX,REXA,X                                           ANALG090
  205 FORMAT(46H1 DEPENDENT VARIABLE DISTRIBUTION FOR Y(M)+ = 1PE12.5/  ANALG091
     128H  ASSUMED VALUE OF RE(X) IS E12.5,10X ,5H X = E12.5)           ANALG092
      WRITE(6,206)                                                      ANALG093
  206 FORMAT(36H0 ASSUMED PROPERTIES AT THIS STATION)                   ANALG094
      WRITE(6,207)P  ,TE,RHOE,XME,UE,HE,ZE,XMUE                         ANALG095
  207  FORMAT(17H0 EDGE CONDITIONS/5X,2H P,13X,2H T,13X,4H RHO,11X,5H MACANALG096
     1H,10X,2H U,13X,2H H,13X,2H Z,13X,3H MU/1P8E15.5)                  ANALG097
      WRITE(6,208)ALPHA,FZ,TW,HW,RHOW,XMUW                              ANALG098
  208  FORMAT(17H0 WALL CONDITIONS/5X,6H ALPHA,9X,6H F (0),9X,2H T,13X,  ANALG099
     12H H,13X,4H RHO,11X,3H MU/1P8E15.5)                               ANALG100
      WRITE(6,209)                                                      ANALG101
  209 FORMAT(1H0/7H0    ETA ,5X,5H U/UE,7X,5H T/TE,                     ANALG102
     16X,9H TAU/TAUW,3X,6H EP/NU,7X,5H W O2,7X,5H W IE,                 ANALG103
     27X,4H W C,8X,4H W P,8X,5H W IM, 8X,2H A//)                        ANALG104
      DO 45 I=1,NETA                                                    ANALG105
      WRITE(6,210)ETA(I),U(I),TTE(I),TAUTW(I),EPNU(I),                  ANALG106
     1WO2(I),WIE(I),WC(I),WP(I),WIM(I),A(I)                             ANALG107
  210 FORMAT(F7.3,1P10E12.3)                                            ANALG108
   45 CONTINUE                                                          ANALG109
      WRITE(6,211)TIMEU,NLOOPU                                          ANALG110
  211 FORMAT(36H0TIME FOR THIS INTEGRATION OF U --= 2A6,5X,I3,11H ITERATANALG111
     1IONS)                                                             ANALG112
   50 CONTINUE                                                          ANALG113
C                                                                       ANALG114
      AMAGRM=.0001                                                      ANALG115
      AMAGST=.0001                                                      ANALG116
      AMAGRD=.0001                                                      ANALG117
C                                                                       ANALG118
C     SOLUTION OF REYNOLDS NUMBER BASED ON MOMENTUM THICKNESS           ANALG119
```

```
C     SOLUTION OF RATIO OF STANTON NO. TO SKIN FRICTION COEFFICIENT     ANALG120
C                                                                       ANALG121
      CALL INTEG1(0.0,1.0,H,AMAGRM,2,ERR,REMX,HX)                       ANALG122
      CALL INTEG1(0.0,1.0,H,AMAGST,3,ERR,STCFU,HX)                      ANALG123
      CALL INTEG1(0.0,1.0,H,AMAGRD,6,ERR,REDX,HX)                       ANALG124
C                                                                       ANALG125
      UMP(NY)=UMPX                                                      ANALG126
      REM(NY)=UMPX*YMPX*REMX                                            ANALG127
      CF2(NY)=1.0/UMPX**2                                               ANALG128
      REDEL(NY)=UMP(NY)*YMP(NY)                                         ANALG129
      RED(NY)=REDEL(NY)*REDX                                            ANALG130
      IF(ALPHA)51,52,51                                                 ANALG131
   51 CONTINUE                                                          ANALG132
      STCF2(NY)=ALPHA/(EXP(STCFU)-1.0)                                  ANALG133
      GO TO 53                                                          ANALG134
   52 CONTINUE                                                          ANALG135
      STCF2(NY)=1.0/STCFU                                               ANALG136
   53 CONTINUE                                                          ANALG137
      GO TO 60                                                          ANALG138
C                                                                       ANALG139
   55 CONTINUE                                                          ANALG140
      NYMP=NY-1                                                         ANALG141
      GO TO 59                                                          ANALG142
   56 CONTINUE                                                          ANALG143
      NYMP=NY-1                                                         ANALG144
      GO TO 59                                                          ANALG145
   57 CONTINUE                                                          ANALG146
      GO TO 59                                                          ANALG147
   58 CONTINUE                                                          ANALG148
   59 CONTINUE                                                          ANALG149
   60 CONTINUE                                                          ANALG150
C                                                                       ANALG151
      CALL VONKAR                                                       ANALG152
C                                                                       ANALG153
      HCOEF(NY)=STCF2(NY)*CF2(NY)*RHOE*UE                               ANALG154
      QZ(NY)=HCOEF(NY)*(HE-HW+UE**2/2.0/G/XJ)                           ANALG155
      UET(NY)=UE                                                        ANALG156
      PT(NY)=P/2116.                                                    ANALG157
C                                                                       ANALG158
      IF(NY-NYMP)30,80,80                                               ANALG159
```

```
   80 CONTINUE                                                          ANALG160
      WRITE(6,201)                                                      ANALG161
C                                                                       ANALG162
      WRITE(6,202)HEAD                                                  ANALG163
      WRITE(6,220)                                                      ANALG164
  220 FORMAT(1H0,3X,2H X,8X,6H RE(X),5X,6H RE(M),5X,6HP(ATM),5X,8H U(EDGANALG165
     1E),3X,5H CF/2,6X,8H ST/CF/2,3X,5H Q(W),6X,6H Y(M)+,5X,6H U(M)+,5X ANALG166
     2,8H H(COEF)///)                                                   ANALG167
      WRITE(6,225)(XD(I),REX(I),REM(I), PT(I),UET  (I),                 ANALG168
     1CF2(I),STCF2(I),QZ(I),YMP(I),UMP(I),HCOEF(I),I=1,NYMP)            ANALG169
  225 FORMAT(1P11E11.4)                                                 ANALG170
      GO TO 20                                                          ANALG171
      END                                                               ANALG172
```

```
=     FOR PROPX                                                         PROPX001
      SUBROUTINE PROPX(KGO)                                             PROPX002
C     DETERMINE CONDITION AS A FUNCTION OF X                            PROPX003
C                                                                       PROPX004
      DIMENSION XXP(50),PXT(50),XXTW(50),TWXT(50),                      PROPX005
     1XXALP(50),ALPXT(50),XXTE(50),TEXT(50),XXUE(50),UEXT(50),          PROPX006
     2TMUT(50),XMUT(50),XXMUE(50),XMUET(50),XMUEL(50),XLNUET(50),       PROPX007
     3XXR(50),XLNRT(50)                                                 PROPX008
      DIMENSION LT(13),HTT(16),PHT(6),THPT(6,16)                        PROPX009
      COMMON  GAM,   RGAS,   G,   XJ,   CP,   SAA,   NLOOPU,   DETA1,   PROPX010
     1         DETA2,  UX,    MXLPU,   HE,   XNMU,   TAUTWX,   PR,      PROPX011
     2         EPNUX,   SC,   TTEX,   TIMEU(2),   AK,   ERR,   AMOL     PROPX012
      COMMON  /PROFIL /  ETA(1000),   U(1000),   ETAZ(1000),   UZ(1000),PROPX013
     1         TTE(1000),   EPNU(1000),   TAUTW(1000),   W1(1000),      PROPX014
     2         NETA,   NETAZ,   NY,   NYMP                              PROPX015
      COMMON  /FUNCTX/  YMP(100),   REM(100),   REX(100),   CF2(100),   PROPX016
     1         STCF2(100),   TETW,   XME,   ALPHA,   PHI,   YMPX,       PROPX017
     2UMP(100),   XD(100),   X,   UMPX,   TE,   UE,                     PROPX018
     3RHOE,   XMUE,   P,   TW,   DLNUE,   DLNMUE,   FZ,   VINF,         PROPX019
     4RED(100),   DLNR,   S,   REDEL(100),   CF2XX,   REDXX,   ZE       PROPX020
     5   ,   HW,   XSTOP,   RHOW,   QZ(100),   XMUW                     PROPX021
      COMMON/MASSD/ETAC,A(1000),WO2(1000),WIE(1000),                    PROPX022
     1WC(1000),WP(1000),WIM(1000),BETAO2,BETAIM                         PROPX023
C                                                                       PROPX024
      DATA(TMUT(I),XMUT(I),I=1,11)/0.0,0.23E-6,2000.,0.92E-6,           PROPX025
     14000.,1.37E-6,6000.,1.70E-6,8000.,2.07E-6,10000.,2.63E-6,         PROPX026
     212000.,3.17E-6,14000.,3.58E-6,16000.,3.76E-6,18000.,3.47E-6,      PROPX027
     3200000.,2.35E-6/                                                  PROPX028
C                                                                       PROPX029
      DATA (HTT(I),I=1,16) / 0.0 , 2000. , 4000. , 6000. , 8000. ,      PROPX030
     1 10000. , 12000. , 14000. , 16000. , 18000. , 20000. , 22000. ,   PROPX031
     2 24000. , 26000. , 28000. , 30000. /                              PROPX032
      DATA (PHT(I),I=1,6) / .00001 , .0001 , .001 , .01 , .1 , 1.0 /    PROPX033
      DATA (THPT(1,I),I=1,16)/  0.0 , 3933. , 6345. , 7020. , 7362. ,   PROPX034
     1 7650. , 7866. , 8172. , 9450. , 11250. , 11988. , 12510. , 12870.PROPX035
     2 , 13140. , 13374. , 13500. /                                     PROPX036
      DATA (THPT(2,I),I=1,16)/  0.0 , 4230. , 6696. , 7506. , 7897. ,   PROPX037
     1 8226. , 8496. , 8865. , 9900. , 11952. , 12852. , 13410. , 13842.PROPX038
     2 , 14130. , 14364. , 14580. /                                     PROPX039
```

```
      DATA (THPT(3,I),I=1,16)/  0.0 , 4554. , 7092. , 8100. , 8595. ,   PROPX040
     1 8955. , 9288. , 9756. , 10530. , 12690. , 13896. , 14580. ,      PROPX041
     2 15120. , 15480. , 15840. , 16074. /                              PROPX042
      DATA (THPT(4,I),I=1,16)/  0.0 , 4905. , 7452. , 8820. , 9450. ,   PROPX043
     1 9828. , 10206. , 10485. , 11520. , 13410. , 15030. , 15930. ,    PROPX044
     2 16650. , 17100. , 17640. , 17910. /                              PROPX045
      DATA (THPT(5,I),I=1,16)/  0.0 , 5328. , 7830. , 9594. , 10386. ,  PROPX046
     1 10926. , 11430. , 11944. , 12708. , 14130. , 16056. , 17370. ,   PROPX047
     2 18270. , 19080. , 19620. , 19980. /                              PROPX048
      DATA (THPT(6,I),I=1,16)/  0.0 , 5725. , 8271. , 10494. , 11538. , PROPX049
     1 12186. , 12780. , 13446. , 14220. , 15372. , 17280. , 18900. ,   PROPX050
     2 20070. , 21150. , 21870. , 22680. /                              PROPX051
C                                                                       PROPX052
      GO TO(20,60,80),KGO                                               PROPX053
   20 CONTINUE                                                          PROPX054
      IF(X-XC)30,32,32                                                  PROPX055
   30 CONTINUE                                                          PROPX056
C                                                                       PROPX057
      CALL LAGIT(X,XXTE,TEXT,40,2,TE,IERR)                              PROPX058
      CALL LAGIT(X,XXUE,UEXT,40,2,UE,IERR)                              PROPX059
      CALL LAGIT(X,XXMUE,XMUET,40,2,XMUE,IERR)                          PROPX060
      CALL DLAGIT(X,XXUE,XLNUET,40,3,DLNUE,IERR)                        PROPX061
      CALL DLAGIT(X,XXMUE,XMUEL,40,3,DLMUE,IERR)                        PROPX062
      CALL DLAGIT(X,XXR,XLNRT,40,3,DLNR,IERR)                           PROPX063
C                                                                       PROPX064
      GO TO 40                                                          PROPX065
   32 CONTINUE                                                          PROPX066
      IF(NXC)34,34,36                                                   PROPX067
   34 CONTINUE                                                          PROPX068
      NXC=1                                                             PROPX069
C                                                                       PROPX070
      CALL LAGIT(XC,XXTE,TEXT,40,2,TE,IERR)                             PROPX071
      CALL LAGIT(XC,XXUE,UEXT,40,2,UE,IERR)                             PROPX072
      CALL LAGIT(XC,XXMUE,XMUET,40,2,XMUE,IERR)                         PROPX073
C                                                                       PROPX074
      DLNUE=0.0                                                         PROPX075
      DLNMUF=0.0                                                        PROPX076
      DLNR=0.0                                                          PROPX077
   36 CONTINUE                                                          PROPX078
 40   CONTINUE                                                          PROPX079
```

```
C                                                                       PROPX080
      CALL LAGIT(X,XXP,PXT,NPT,2,P,IERR)                                PROPX081
      CALL LAGIT(X,XXTW,TWXT,NTWT,2,TW,IERR)                            PROPX082
      CALL LAGIT(X,XXALP,ALPXT,NALP,2,ALPHA,IERR)                       PROPX083
      CALL LAGIT(TW,TMUT,XMUT,11,2,XMUW,IERR)                           PROPX084
C                                                                       PROPX085
      XMUW=XMUW*G                                                       PROPX086
      IF(TE-6000.0)44,44,46                                             PROPX087
   44 ZE=1.0                                                            PROPX088
      GO TO 48                                                          PROPX089
   46 ZE=(1.0E-3)/6.0*TE                                                PROPX090
   48 CONTINUE                                                          PROPX091
      P=PES*P*2114.0                                                    PROPX092
      HE=HS -UE**2/G2J                                                  PROPX093
          RHOE=P/RGAS/TE/ZE*AMOL                                        PROPX094
      DLNUE=DLNUE*XMUE/RHOE/UE                                          PROPX095
      DLNMUE=DLMUE*XMUE/RHOE/UE                                         PROPX096
      DLNR=DLNR*XMUE/RHOE/UE                                            PROPX097
      TETW=TE/TW                                                        PROPX098
      XME=UE/SQRT(ZE*GAM*G*RGAS*TE/AMOL)                                PROPX099
      FZ=ALPHA/UMP(NY-1)**2                                             PROPX100
      RHOW=P/RGAS/TW*AMOL                                               PROPX101
      HW = CP*TW                                                        PROPX102
C                                                                       PROPX103
      RETURN                                                            PROPX104
 60   CONTINUE                                                          PROPX105
      RETURN                                                            PROPX106
 80   CONTINUE                                                          PROPX107
      NXC=0                                                             PROPX108
      CAPR=15.0                                                         PROPX109
      G2J=2.0*G*XJ                                                      PROPX110
      READ(5,105)PES, HS,VINF,XC,BETAO2,BETAIM                          PROPX111
 105  FORMAT(8E10.0)                                                    PROPX112
      WRITE(6,150)PES, HS,VINF                                          PROPX113
 150  FORMAT(8H0 PES = 1PE20.7,10X,7H HES = E20.7,                      PROPX114
     18H VINF = E20.7///)                                               PROPX115
      READ(5,110)NPT,(XXP(I),PXT(I),I=1,NPT)                            PROPX116
      READ(5,110)NTWT,(XXTW(I),TWXT(I),I=1,NTWT)                        PROPX117
      READ(5,110)NALP,(XXALP(I),ALPXT(I),I=1,NALP)                      PROPX118
 110  FORMAT(I5/(8E10.0))                                               PROPX119
```

```
C                                                                       PROPX120
      LT(1)=LOC(LT(1))                                                  PROPX121
      LT(2)=LOC(PHT(1))                                                 PROPX122
      LT(3)=1                                                           PROPX123
      LT(4)=6                                                           PROPX124
      LT(5)=1                                                           PROPX125
      LT(6)=1                                                           PROPX126
      LT(7)=0                                                           PROPX127
      LT(8)=0                                                           PROPX128
      LT(9)=LOC(HTT(1))                                                 PROPX129
      LT(10)=LOC(THPT(1,1))                                             PROPX130
      LT(11)=1                                                          PROPX131
      LT(12)=6                                                          PROPX132
      LT(13)=16                                                         PROPX133
C                                                                       PROPX134
      DELXX=.4                                                          PROPX135
      XX=0.0                                                            PROPX136
      DO 85 I=1,40                                                      PROPX137
      XXTE(I)=XX                                                        PROPX138
      XXUE(I)=XX                                                        PROPX139
      XXMUE(I)=XX                                                       PROPX140
      XXR(I)=XX                                                         PROPX141
      XX=XX+DELXX                                                       PROPX142
      XLNRT(I)=ALOG(XX)                                                 PROPX143
C                                                                       PROPX144
      CALL LAGIT(XX,XXP,PXT,NPT,2,PATM,IERR)                            PROPX145
C                                                                       PROPX146
      UEXT(I)=XX*SQRT(1.0/6.0*(2.0-1.0/6.0))*VINF/CAPR                  PROPX147
      HE=HS-UEXT(I)**2/G2J                                              PROPX148
      PATM = PATM*PES                                                   PROPX149
      PATM=AMAX1(.00001,PATM)                                           PROPX150
      PATM=AMIN1(1.0,PATM)                                              PROPX151
      HE=AMAX1(200.,HE)                                                 PROPX152
      HE=AMIN1(30000.,HE)                                               PROPX153
C                                                                       PROPX154
      TE=DTAB(HE,PATM,LT(1))                                            PROPX155
C                                                                       PROPX156
      TEXT(I)=TE                                                        PROPX157
   85 CONTINUE                                                          PROPX158
      UEXT(1)=UEXT(2)*.1                                                PROPX159
```

```
      DO 90 I=1,40                                                      PROPX160
C                                                                       PROPX161
      CALL LAGIT(TEXT(I),TMUT,XMUT,11,2,XMUET(I),IERR)                  PROPX162
C                                                                       PROPX163
      XMUET(I)=XMUET(I)*G                                               PROPX164
      XMUEL(I)=ALOG(XMUET(I))                                           PROPX165
      UE=UEXT(I)                                                        PROPX166
      XLNUET(I)=ALOG(UE)                                                PROPX167
   90 CONTINUE                                                          PROPX168
C                                                                       PROPX169
      WRITE(6,200)                                                      PROPX170
  200 FORMAT(13H0 TABLES USED///)                                       PROPX171
      WRITE(6,210)(XXP(I),PXT(I),I=1,NPT)                               PROPX172
  210 FORMAT(7H0 (X,P)/1P6E20.7/(6E20.7))                               PROPX173
      WRITE(6,220)(XXTW(I),TWXT(I),I=1,NTWT)                            PROPX174
  220 FORMAT(8H0 (X,TW)/1P6E20.7/(6E20.7))                              PROPX175
      WRITE(6,230)(XXALP(I),ALPXT(I),I=1,NALP)                          PROPX176
  230 FORMAT(11H0 (X,ALPHA)/1P6E20.7/(6E20.7))                          PROPX177
      WRITE(6,240)(XXTE(I),TEXT(I),I=1,40)                              PROPX178
  240 FORMAT(8H0 (X,TE)/1P6E20.7/(6E20.7))                              PROPX179
      WRITE(6,250)(XXUE(I),UEXT(I),I=1,40)                              PROPX180
  250 FORMAT(8H0 (X,UE)/1P6E20.7/(6E20.7))                              PROPX181
      WRITE(6,260)(XXMUE(I),XMUET(I),I=1,40)                            PROPX182
  260 FORMAT(9H0 (X,MUE)/1P6E20.7/(6E20.7))                             PROPX183
      WRITE(6,280)(XXR(I),XLNRT(I),I=1,40)                              PROPX184
  280 FORMAT(12H0 (X,LOG(R))/1P6E20.7/(6E20.7))                         PROPX185
C                                                                       PROPX186
      RETURN                                                            PROPX187
      END                                                               PROPX188
```

```
8    =     FOR SPECIE                                                               SPCIE001
           SUBROUTINE SPECIE                                                        SPCIE002
     C     SOLUTION OF SPECIE EQUATION BY ANALOGY                                   SPCIE003
     C     MODEL OF COMPLETE COMBUSTION                                             SPCIE004
     C                                                                              SPCIE005
           COMMON  GAM,   RGAS,   G,   XJ,   CP,   SAA,   NLOOPU,   DETA1,         SPCIE006
          1        DETA2,   UX,   MXLPU,   HE,   XNMU,   TAUTWX,   PR,             SPCIE007
          2        EPNUX,   SC,   TTEX,   TIMEU(2),   AK,   ERR,   AMOL            SPCIE008
           COMMON  /PROFIL /  ETA(1000),   U(1000),   ETAZ(1000),   UZ(1000),SPCIE009
          1        TTE(1000),   EPNU(1000),   TAUTW(1000),   W1(1000),           SPCIE010
          2        NETA,   NETAZ,   NY,   NYMP                                   SPCIE011
           COMMON  /FUNCTX/  YMP(100),   REM(100),   REX(100),   CF2(100),       SPCIE012
          1        STCF2(100),   TETW,   XME,   ALPHA,   PHI,   YMPX,            SPCIE013
          2UMP(100),   XD(100),   X,   UMPX,   TE,   UE,                         SPCIE014
          3RHOE,   XMUE,   P,   TW,   DLNUE,   DLNMUE,   FZ,   VINF,             SPCIE015
          4RED(100),   DLNR,   S,   REDEL(100),   CF2XX,   REDXX,   ZE           SPCIE016
          5   ,   HW,   XSTOP,   RHOW,   QZ(100),   XMUW                         SPCIE017
           COMMON/MASSD/ETAC,A(1000),WO2(1000),WIE(1000),                       SPCIE018
          1WC(1000),WP(1000),WIM(1000),BETAO2,BETAIM                            SPCIE019
     C                                                                              SPCIE020
           BETAC=1.0-BETAIM                                                         SPCIE021
           BETAP=BETAC+BETAO2                                                       SPCIE022
           WO2E=0.23                                                                SPCIE023
           WIEE=1.0-WO2E                                                            SPCIE024
     C                                                                              SPCIE025
           IF(FZ)10,10,15                                                           SPCIE026
        10 CONTINUE                                                                 SPCIE027
           DO 12 I=1,NETA                                                           SPCIE028
           WO2(I)=WO2E                                                              SPCIE029
           WIE(I)=WIEE                                                              SPCIE030
           WP(I)=0.0                                                                SPCIE031
           WC(I)=0.0                                                                SPCIE032
           WIM(I)=0.0                                                               SPCIE033
        12 CONTINUE                                                                 SPCIE034
           RETURN                                                                   SPCIE035
     C                                                                              SPCIE036
        15 CONTINUE                                                                 SPCIE037
     C                                                                              SPCIE038
     C     DETERMINE POSITION OF REACTION PLANE                                     SPCIE039
```

```
C                                                                       SPCIE040
      AMAG4=1.0                                                         SPCIE041
      A(1)=0.0                                                          SPCIE042
      DO 20 I=2,NETA                                                    SPCIE043
C                                                                       SPCIE044
      CALL INTEG1(U(I-1),U(I),H,AMAG4,4,ERR,DA,HX)                      SPCIE045
C                                                                       SPCIE046
      A(I)=A(I-1)+DA                                                    SPCIE047
   20 CONTINUE                                                          SPCIE048
      DO 25 I=1,NETA                                                    SPCIE049
      A(I)=A(I)*FZ*UMPX*YMPX*SC                                         SPCIE050
   25 CONTINUE                                                          SPCIE051
      AXC=A(NETA)+ALOG(BETAO2/(WO2E+BETAO2))                            SPCIE052
      IF(AXC)10,10,26                                                   SPCIE053
   26 CONTINUE                                                          SPCIE054
C                                                                       SPCIE055
      CALL LAGIT( AXC ,A,ETA,NETA,2,ETAC,IERR)                          SPCIE056
C                                                                       SPCIE057
C     EVALUATE MASS FRACTIONS                                           SPCIE058
      DO 40 I=1,NETA                                                    SPCIE059
C                                                                       SPCIE060
      AX=A(I)                                                           SPCIE061
C                                                                       SPCIE062
      WIE(I)=WIEE*EXP(AX-A(NETA))                                       SPCIE063
      WIM(I)=BETAIM*(1.0-EXP(AX-A(NETA)))                               SPCIE064
C                                                                       SPCIE065
      IF(ETA(I)-ETAC)30,30,32                                           SPCIE066
   30 CONTINUE                                                          SPCIE067
      WO2(I)=0.0                                                        SPCIE068
      WC(I)=BETAC*(1.0-EXP(AX-AXC))                                     SPCIE069
      WP(I)=BETAP*(1.0-EXP(-(A(NETA)-AXC)))*EXP(AX-AXC)                 SPCIE070
      GO TO 35                                                          SPCIE071
   32 CONTINUE                                                          SPCIE072
      WO2(I)=WO2E*(EXP(AX-AXC)-1.0)/(EXP(A(NETA)-AXC)-1.0)              SPCIE073
      WC(I)=0.0                                                         SPCIE074
      WP(I)=BETAP*(1.0-EXP(AX-A(NETA)))                                 SPCIE075
   35 CONTINUE                                                          SPCIE076
   40 CONTINUE                                                          SPCIE077
      RETURN                                                            SPCIE078
      END                                                               SPCIE079
```

```
-      FOR VONKAR                                                       VNKAR001
       SUBROUTINE VONKAR                                                VNKAR002
C      SOLUTION OF VON KARMAN MOMENTUM EQ                               VNKAR003
C                                                                       VNKAR004
       COMMON  GAM,   RGAS,   G,   XJ,   CP,   SAA,   NLOOPU,   DETA1,  VNKAR005
      1         DETA2,   UX,   MXLPU,   HE,   XNMU,   TAUTWX,   PR,     VNKAR006
      2         EPNUX,   SC,   TTEX,   TIMEU(2),   AK,   ERR,   AMOL    VNKAR007
       COMMON  /PROFIL /  ETA(1000),   U(1000),   ETAZ(1000),   UZ(1000),VNKAR008
      1         TTE(1000),   EPNU(1000),   TAUTW(1000),   WI(1000),     VNKAR009
      2         NETA,   NETAZ,   NY,   NYMP                             VNKAR010
       COMMON  /FUNCTX/  YMP(100),   REM(100),   REX(100),   CF2(100),  VNKAR011
      1         STCF2(100),   TETW,   XME,   ALPHA,   PHI,   YMPX,      VNKAR012
      2UMP(100),   XD(100),   X,   UMPX,   TE,   UE,                    VNKAR013
      3RHOE,   XMUE,   P,   TW,   DLNUE,   DLNMUE,   FZ,   VINF,        VNKAR014
      4RED(100),   DLNR,   S,   REDEL(100),   CF2XX,   REDXX,   ZE      VNKAR015
      5   ,   HW,   XSTOP,   RHOW,   QZ(100),   XMUW                    VNKAR016
C                                                                       VNKAR017
       NREXXX=50                                                        VNKAR018
       NREXXX=10                                                        VNKAR019
       DREDXX=(RED(NY)-RED(NY-1))/NREXXX                                VNKAR020
       DCF2XX=(CF2(NY)-CF2(NY-1))/NREXXX                                VNKAR021
       REDXX=RED(NY-1)                                                  VNKAR022
       CF2XX=CF2(NY-1)                                                  VNKAR023
C                                                                       VNKAR024
       DREMX=(REM(NY)-REM(NY-1))/NREXXX                                 VNKAR025
       REM1=REM(NY-1)                                                   VNKAR026
       X=XD(NY-1)                                                       VNKAR027
       REXX2=REX(NY-1)                                                  VNKAR028
       AMAGRX=.0001                                                     VNKAR029
C                                                                       VNKAR030
       DO 40 I=1,NREXXX                                                 VNKAR031
       REDXX=REDXX+DREDXX                                               VNKAR032
       CF2XX=CF2XX+DCF2XX                                               VNKAR033
       REM2=REM1+DREMX                                                  VNKAR034
C                                                                       VNKAR035
       CALL PROPX(1)                                                    VNKAR036
C                                                                       VNKAR037
```

```
      CALL INTEG1(REM1,REM2,H,AMAGRX,5,FRR,REXXX,HX)                    VNKAR038
C                                                                       VNKAR039
      REM1=REM2                                                         VNKAR040
      REXX2=REXX2+REXXX                                                 VNKAR041
      X=XMUE/RHOE/UE*REXXX+X                                            VNKAR042
      IF(XSTOP-X)30,30,35                                               VNKAR043
   30 CONTINUE                                                          VNKAR044
      NYMP=NY-1                                                         VNKAR045
      GO TO 50                                                          VNKAR046
   35 CONTINUE                                                          VNKAR047
   40 CONTINUE                                                          VNKAR048
      REX(NY)=REXX2                                                     VNKAR049
      XD(NY)=X                                                          VNKAR050
   50 CONTINUE                                                          VNKAR051
      RETURN                                                            VNKAR052
      END                                                               VNKAR053
```

```
= FOR FUNCT                                                             FUNCT001
      FUNCTION FUNCT(VIN,N)                                             FUNCT002
C                                                                       FUNCT003
      COMMON  GAM,   RGAS,   G,   XJ,   CP,   SAA,   NLOOPU,   DETA1,   FUNCT004
     1          DETA2,   UX,   MXLPU,   HE,   XNMU,   TAUTWX,   PR,      FUNCT005
     2          EPNUX,   SC,   TTEX,   TIMEU(2),   AK,   ERR,   AMOL    FUNCT006
      COMMON  /PROFIL /  ETA(1000),   U(1000),   ETAZ(1000),   UZ(1000),FUNCT007
     1          TTE(1000),   EPNU(1000),   TAUTW(1000),   W1(1000),     FUNCT008
     2          NETA,   NETAZ,   NY,   NYMP                             FUNCT009
      COMMON  /FUNCTX/  YMP(100),   REM(100),   REX(100),   CF2(100),   FUNCT010
     1          STCF2(100),   TETW,   XME,   ALPHA,   PHI,   YMPX,        FUNCT011
     2UMP(100),   XD(100),   X,   UMPX,   TE,   UE,                     FUNCT012
     3RHOE,   XMUE,   P,   TW,   DLNUE,   DLNMUE,   FZ,   VINF,          FUNCT013
     4RED(100),   DLNR,   S,   REDEL(100),   CF2XX,   REDXX,   ZE        FUNCT014
     5    ,   HW,   XSTOP,   RHOW,   QZ(100),   XMUW                     FUNCT015
C                                                                       FUNCT016
      IF(N)90,90,1                                                      FUNCT017
    1 CONTINUE                                                          FUNCT018
      IF(NSAVE-N)2,3,2                                                  FUNCT019
    2 CONTINUE                                                          FUNCT020
      NSAVE=N                                                           FUNCT021
      NSTEP=1                                                           FUNCT022
      NUSAV=0                                                           FUNCT023
      GO TO 4                                                           FUNCT024
    3 CONTINUE                                                          FUNCT025
      NSTEP=NSTEP+1                                                     FUNCT026
      IF(NSTEP-5000)4,4,95                                              FUNCT027
    4 CONTINUE                                                          FUNCT028
C                                                                       FUNCT029
      GO TO(10,20,30,40,50,60),N                                        FUNCT030
C                                                                       FUNCT031
   10 CONTINUE                                                          FUNCT032
C     INTEGRAND OF DEGENERATE MOMENTUM EQ.                              FUNCT033
C                                                                       FUNCT034
      CALL LAGIT(VIN,ETAZ,UZ,NETAZ,2,U1,IERR)                           FUNCT035
C                                                                       FUNCT036
      TTEX=(1.0-U1)*(1.0/TETW+U1*(GAM-1.0)/2.0*XME**2)+U1               FUNCT037
      IF(TTEX*TE-6000.0)15,15,16                                        FUNCT038
   15 Z=1.0                                                             FUNCT039
      GO TO 17                                                          FUNCT040
```

```
   16 Z=(1.0E-3)/6.0*TTEX*TE                                           FUNCT041
 17   CONTINUE                                                         FUNCT042
      PHIRHO=ZE/Z/TTEX                                                 FUNCT043
      PHIMU=TTEX**XNMU                                                 FUNCT044
      TAUTWX=1.0-VIN+(ALPHA)*(U1-VIN)                                  FUNCT045
      TAUTWX=AMAX1(.00001,TAUTWX)                                      FUNCT046
      F1=TAUTWX/PHIMU                                                  FUNCT047
      F1=AMAX1(F1,0.0)                                                 FUNCT048
      F2=(2.0*VIN*YMPX*AK*(1.0-EXP(-PHI*VIN)))**2                      FUNCT049
      F3=1.0+F1*F2*PHIRHO/PHIMU                                        FUNCT050
      F4=F1/(1.0+SQRT(F3))                                             FUNCT051
      FUNCT=F4                                                         FUNCT052
      EPNUX=.5*F2*F4                                                   FUNCT053
      RETURN                                                           FUNCT054
C                                                                      FUNCT055
   20 CONTINUE                                                         FUNCT056
C     INTEGRAND OF REYNOLDS NO. BASED ON MOMENTUM THICKNESS            FUNCT057
C                                                                      FUNCT058
      CALL LAGIT(VIN,ETA ,U ,NETA ,2,U1,IERR)                          FUNCT059
C                                                                      FUNCT060
      TTEX=(1.0-U1)*(1.0/TETW+U1*(GAM-1.0)/2.0*XME**2)+U1              FUNCT061
      IF(TTEX*TE-6000.0)25,25,26                                       FUNCT062
 25   Z=1.0                                                            FUNCT063
      GO TO 27                                                         FUNCT064
   26 Z=(1.0E-3)/6.0*TTEX*TE                                           FUNCT065
 27   CONTINUE                                                         FUNCT066
      PHIRHO=ZE/Z/TTEX                                                 FUNCT067
      FUNCT=PHIRHO*U1*(1.0-U1)                                         FUNCT068
      RETURN                                                           FUNCT069
C                                                                      FUNCT070
   30 CONTINUE                                                         FUNCT071
C     INTEGRAND OF STANTON NO. / SKIN FRICTION COEFFICIENT             FUNCT072
C                                                                      FUNCT073
      CALL LAGIT(VIN,U,EPNU,NETA,2,EPNUX,IERR)                         FUNCT074
C                                                                      FUNCT075
      PRSTR=PR*(1.0+EPNUX)/(1.0+PR*EPNUX)                              FUNCT076
      IF(ALPHA)32,34,32                                                FUNCT077
   32 CONTINUE                                                         FUNCT078
      FUNCT=PRSTR**.666/(VIN+1.0/ALPHA)                                FUNCT079
      RETURN                                                           FUNCT080
```

```
   34 CONTINUE                                                          FUNCT081
      FUNCT=PRSTR**.666                                                 FUNCT082
      RETURN                                                            FUNCT083
C                                                                       FUNCT084
 40   CONTINUE                                                          FUNCT085
C     INTEGRAND OF SPECIES EQUATION                                     FUNCT086
C                                                                       FUNCT087
      CALL LAGIT(VIN,U,EPNU,NETA,2,EPNUX,IERR)                          FUNCT088
      CALL LAGIT(VIN,U,ETA,NETA,2,ETAUX,IERR)                           FUNCT089
      CALL LAGIT(ETAUX,ETA,TTE,NETA,2,TTEX,IERR)                        FUNCT090
C                                                                       FUNCT091
      PHIMU=TTEX**XNMU                                                  FUNCT092
      FUNCT=1.0/(PHIMU*(1.0+EPNUX))                                     FUNCT093
      RETURN                                                            FUNCT094
 50   CONTINUE                                                          FUNCT095
      S=REDXX/VIN                                                       FUNCT096
      F1111=1.0/(FZ+CF2XX-VIN*((1.0+S)*DLNUE+DLNMUE+DLNR))              FUNCT097
      FUNCT=ABS(F1111)                                                  FUNCT098
      RETURN                                                            FUNCT099
 60   CONTINUE                                                          FUNCT100
C                                                                       FUNCT101
      CALL LAGIT(VIN,ETA ,U ,NETA ,2,U1,IERR)                           FUNCT102
C                                                                       FUNCT103
      TTEX=(1.0-U1)*(1.0/TETW+U1*(GAM-1.0)/2.0*XME**2)+U1               FUNCT104
      IF(TTEX*TE-6000.0)65,65,66                                        FUNCT105
 65   Z=1                                                               FUNCT106
      GO TO 67                                                          FUNCT107
   66 Z=(1.0E-3)/6.0*TTEX*TE                                            FUNCT108
 67   CONTINUE                                                          FUNCT109
      PHIRHO=ZE/Z/TTEX                                                  FUNCT110
      FUNCT=1.0-U1*PHIRHO                                               FUNCT111
      RETURN                                                            FUNCT112
C                                                                       FUNCT113
   90 CONTINUE                                                          FUNCT114
      WRITE(6,200)                                                      FUNCT115
  200 FORMAT(1H1,8(6H ****** )/21H0 ERROR IN QUADRATURE)                FUNCT116
      WRITE(6,201)NSAVE,VIN                                             FUNCT117
  201 FORMAT(91H0 SUSPECTED VALUE OF QUADRATURE BETWEEN PREVIOUS VALUES FUNCT118
     1OF THE INTEGRATION VARIABLE IS ZERO/6H0 N = I2,10X,27H VARIABLE OFFUNCT119
     2 INTEGRATION = 1PE12.5)                                           FUNCT120
```

```
      WRITE(6,209)                                                      FUNCT121
      WRITE(6,210)(ETAZ(I),UZ(I),TTE(I),W1(I),TAUTW(I),EPNU(I),         FUNCT122
     1I=1,NETAZ)                                                        FUNCT123
  209 FORMAT(1H0/1H0,3X,4H ETA,9X,5H U/UE,9X,5H T/TE,9X,5H W(1),5X,     FUNCT124
     19H TAU/TAUW, 8X,6H EP/NU//)                                       FUNCT125
  210 FORMAT(F10.4,1P5E14.5)                                            FUNCT126
C                                                                       FUNCT127
      CALL ANALGY(2)                                                    FUNCT128
C                                                                       FUNCT129
   95 CONTINUE                                                          FUNCT130
      NSTEP=0                                                           FUNCT131
      IF(N-1)96,97,96                                                   FUNCT132
   96 CONTINUE                                                          FUNCT133
      WRITE(6,200)                                                      FUNCT134
      WRITE(6,205)N,VIN                                                 FUNCT135
  205 FORMAT(60H0 THIS QUADRATURE EXCEEDED 5000 EVALUATIONS OF THE INTEGFUNCT136
     1RAND/6H0 N = I2,10X,27H VARIABLE OF INTEGRATION = 1PE12.5)        FUNCT137
      WRITE(6,209)                                                      FUNCT138
      WRITE(6,210)(ETAZ(I),UZ(I),TTE(I),W1(I),TAUTW(I),EPNU(I),         FUNCT139
     1I=1,NETAZ)                                                        FUNCT140
C                                                                       FUNCT141
      CALL ANALGY(3)                                                    FUNCT142
C                                                                       FUNCT143
   97 CONTINUE                                                          FUNCT144
      IF(NUSAV-NLOOPU)98,96,98                                          FUNCT145
   98 CONTINUE                                                          FUNCT146
      NUSAV=NLOOPU                                                      FUNCT147
      GO TO 10                                                          FUNCT148
      END                                                               FUNCT149
```

```
-     FOR MOMENT                                                        MOMNT001
      SUBROUTINE MOMENT                                                 MOMNT002
C     INTEGRAL SOLUTION OF MOMENTUM EQUATION                            MOMNT003
C     EQ. 51 OF D2-113078-1        EQ. 31 OF D2-23990-1                 MOMNT004
C     EQ. 7 OF GILL AND SCHER -- MODIFICATION OF MOMENTUM TRANSPORT     MOMNT005
C     REYNOLDS ANALOGY USED FOR ENERGY DISTRIBUTION                     MOMNT006
C                                                                       MOMNT007
      COMMON  GAM,   RGAS,   G,   XJ,   CP,   SAA,   NLOOPU,   DETA1,   MOMNT008
     1         DETA2,   UX,   MXLPU,   HE,   XNMU,   TAUTWX,   PR,      MOMNT009
     2         EPNUX,   SC,   TTEX,   TIMEU(2),   AK,   ERR,   AMOL     MOMNT010
      COMMON  /PROFIL /  ETA(1000),   U(1000),   ETAZ(1000),   UZ(1000),MOMNT011
     1         TTE(1000),   EPNU(1000),   TAUTW(1000),   W1(1000),      MOMNT012
     2         NETA,   NETAZ,   NY,   NYMP                              MOMNT013
      COMMON  /FUNCTX/  YMP(100),   REM(100),   RFX(100),   CF2(100),   MOMNT014
     1         STCF2(100),   TETW,   XME,   ALPHA,   PHI,   YMPX,       MOMNT015
     2UMP(100),   XD(100),   X,   UMPX,   TE,   UE,                     MOMNT016
     3RHOE,   XMUE,   P,   TW,   DLNUE,   DLNMUE,   FZ,   VINF,         MOMNT017
     4RED(100),   DLNR,   S,   REDEL(100),   CF2XX,   REDXX,   ZE       MOMNT018
     5   ,   HW,   XSTOP,   RHOW,   OZ(100),   XMUW                     MOMNT019
      DATA SBB,SQQ,NBB/22.0,0.,0/                                       MOMNT020
C                                                                       MOMNT021
C     CALL FLTI                                                         MOMNT022
C                                                                       MOMNT023
      U(1)=0.0                                                          MOMNT024
      ETA(1)=0.0                                                        MOMNT025
      ALPUMP=ALPHA/UMP(NY-1)                                            MOMNT026
      IF(ALPUMP)10,10,12                                                MOMNT027
 10   CONTINUE                                                          MOMNT028
      SBB1=SBB*((1.0+PR**.66666*XME**2*(GAM-1.0)/2.0)/TETW**SQQ)**NBB   MOMNT029
      GO TO 14                                                          MOMNT030
 12   CONTINUE                                                          MOMNT031
      SBB1=SBB                                                          MOMNT032
 14   CONTINUE                                                          MOMNT033
      SAA1=SAA*(1.0+(GAM-1.0)/2.0*XME**2)**0.125                        MOMNT034
      PHI=(YMPX-SAA1)/SBB1                                              MOMNT035
      PHI=AMAX1(0.0,PHI)                                                MOMNT036
      IF(NY-2) 15,15,22                                                 MOMNT037
```

```
   15 CONTINUE                                                          MOMNT038
C                                                                       MOMNT039
C     GENERATE INITIAL GUESS OF U                                       MOMNT040
C                                                                       MOMNT041
      DO 20 I=2,20                                                      MOMNT042
      ETAZ(I)=.05*FLOAT(I-1)                                            MOMNT043
      UZ(I)=ETAZ(I)**.1                                                 MOMNT044
 20   CONTINUE                                                          MOMNT045
      NETAZ=21                                                          MOMNT046
   22 CONTINUE                                                          MOMNT047
C                                                                       MOMNT048
      NLOOPU=0                                                          MOMNT049
   30 CONTINUE                                                          MOMNT050
      NLOOPU=NLOOPU+1                                                   MOMNT051
      AMAGU=.01                                                         MOMNT052
      KSTOPU=0                                                          MOMNT053
      DETA=DETAI                                                        MOMNT054
      UXX=UX*UMPX/2.0/YMPX                                              MOMNT055
C                                                                       MOMNT056
      DO 60 I=2,1000                                                    MOMNT057
      I1=I-1                                                            MOMNT058
      ETA(I)=ETA(I-1)+DETA                                              MOMNT059
      IF(1.0-ETA(I)-.2*DETA)40,40,45                                    MOMNT060
 40   CONTINUE                                                          MOMNT061
      ETA(I)=1.0                                                        MOMNT062
      NETA=I                                                            MOMNT063
      KSTOPU=1                                                          MOMNT064
 45   CONTINUE                                                          MOMNT065
C                                                                       MOMNT066
      CALL INTEG1(ETA(I1),ETA(I),H,AMAGU,1,ERR,DU,HX)                   MOMNT067
C                                                                       MOMNT068
      U(I)=U(I-1)+DU                                                    MOMNT069
      TTE(I)=TTEX                                                       MOMNT070
      EPNU(I)=EPNUX                                                     MOMNT071
      TAUTW(I)=TAUTWX                                                   MOMNT072
C                                                                       MOMNT073
      AMAGU=U(I)                                                        MOMNT074
```

```
      IF(UXX-U(I))50,50,55                                              MOMNT075
 50   CONTINUE                                                          MOMNT076
      DETA=DETA2                                                        MOMNT077
 55   CONTINUE                                                          MOMNT078
      IF(KSTOPU)60,60,70                                                MOMNT079
   60 CONTINUE                                                          MOMNT080
      WRITE(6,250)                                                      MOMNT081
  250 FORMAT(6H ERROR)                                                  MOMNT082
C                                                                       MOMNT083
   70 CONTINUE                                                          MOMNT084
C                                                                       MOMNT085
      UMPX=2.0*YMPX*U(NETA)                                             MOMNT086
      KERRU=0                                                           MOMNT087
C                                                                       MOMNT088
      DO 74 I=2,NETA                                                    MOMNT089
C                                                                       MOMNT090
      CALL LAGIT(ETA(I),ETAZ,UZ,NETAZ,2,UZX,IERR)                       MOMNT091
C                                                                       MOMNT092
      U(I)=2.0*YMPX*U(I)/UMPX                                           MOMNT093
      ARERR=ABS((U(I)-UZX)/U(I))                                        MOMNT094
      IF(ARERR-.001)74,72,72                                            MOMNT095
   72 CONTINUE                                                          MOMNT096
      KERRU=KERRU+1                                                     MOMNT097
   74 CONTINUE                                                          MOMNT098
C                                                                       MOMNT099
      DO 76 I=2,NETA                                                    MOMNT100
      UZ(I)=AMAX1(U(I),0.0)                                             MOMNT101
      UZ(I)=AMIN1(UZ(I),1.0)                                            MOMNT102
      ETAZ(I)=ETA(I)                                                    MOMNT103
   76 CONTINUE                                                          MOMNT104
      NETAZ=NETA                                                        MOMNT105
      IF(KERRU)80,80,78                                                 MOMNT106
   78 CONTINUE                                                          MOMNT107
      IF(NLOOPU-MXLPU)30,30,79                                          MOMNT108
   79 CONTINUE                                                          MOMNT109
   80 CONTINUE                                                          MOMNT110
C                                                                       MOMNT111
```

```
C     CALL FLT3(TIMEU)                                                  MOMNT112
C                                                                       MOMNT113
      RETURN                                                            MOMNT114
      END                                                               MOMNT115
```

```
      SUBROUTINE INTEG1 (XL,XU,H,AMAG,N,ERR,ANS,HX )                    INTEG001
C     FORTRAN IV ROUTINE TO EVALUATE DOUBLE INTEGRAL                    INTEG002
C     REQUIRES FUNCTION SUBPROGRAM WRITTEN BY USER                      INTEG003
C     SPECIFICATION STMNTS                                              INTEG004
      LOGICAL SWCHX                                                     INTEG005
C     RESET OVERFLOW AND DIVIDE CHECK INDICATORS                        INTEG006
      CALL OVERFL(I)                                                    INTEG007
      CALL DVCHK(I)                                                     INTEG008
C     ENSURE CONSISTENCY OF INPUT DATA                                  INTEG009
      IF( XU .EQ. XL )  GO TO 102                                       INTEG010
      XLA = AMIN1( XU,XL )                                              INTEG011
      XUA = AMAX1( XU,XL )                                              INTEG012
      IF( H .LE. 0.0 )  H = 0.25 * ( XUA - XLA )                        INTEG013
      IF( AMAG .EQ. 0.0 )  GO TO 101                                    INTEG014
      IF( ERR .LT. 0.000001 )  ERR = 0.000001                           INTEG015
      GO TO 200                                                         INTEG016
  102 ANS = 0.0                                                         INTEG017
      RETURN                                                            INTEG018
C     WHEN AMAG = 0 SET N = -1 AS A SIGNAL, RETURN DIRECTLY             INTEG019
C     TO CALLING PROGRAM.                                               INTEG020
  101 N = - 1                                                           INTEG021
      RETURN                                                            INTEG022
C     COMPUTE CONSTANTS FOR LATER USE                                   INTEG023
  200 CMAG = 0.0001 * ABS( AMAG )                                       INTEG024
      RERR = 1. / ERR                                                   INTEG025
      DX = H                                                            INTEG026
C     X LINE                                                            INTEG027
  400 ANS = 0.0                                                         INTEG028
      XA = XLA                                                          INTEG029
      TA = FUNCT( XA, N )                                               INTEG030
      SWCHX = .FALSE.                                                   INTEG031
      GO TO 401                                                         INTEG032
C     RETURN FROM RATIO, CYCLE REJECTED.                                INTEG033
  402 DX = HT                                                           INTEG034
      SWCHX = .FALSE.                                                   INTEG035
      GO TO 401                                                         INTEG036
C     STEP FOR NEW PASS, FIFTH ORDINATE IS NOW FIRST.                   INTEG037
  407 XA = XB                                                           INTEG038
      TA = TB                                                           INTEG039
  401 X1 = XA + DX                                                      INTEG040
```

```
      X2 = X1 + DX                                                      INTEG041
      X3 = X2 + DX                                                      INTEG042
      XB = X3 + DX                                                      INTEG043
C     TEST FOR END OF X INTERVAL                                        INTEG044
      IF( XB - XUA )405,404,403                                         INTEG045
C     IF OVERSTEPPING END, ADJUST                                       INTEG046
403   DX = 0.25 * ( XUA - XA )                                          INTEG047
      X1 = XA + DX                                                      INTEG048
      X2 = X1 + DX                                                      INTEG049
      X3 = X2 + DX                                                      INTEG050
      XB = XUA                                                          INTEG051
404   SWCHX = .TRUE.                                                    INTEG052
C     USE TEMPORARY STORAGE TO SAVE RECOMPUTING FUNCT                   INTEG053
405   T2 = FUNCT( X2, N )                                               INTEG054
      TB = FUNCT( XB, N )                                               INTEG055
C     COARSE AND FINE APPROXIMATIONS                                    INTEG056
      S1 = 0.66666667 * DX * (TA + 4.0*T2 + TB )                        INTEG057
      S2 = 0.33333333 * DX*( TA + 4.0*FUNCT(X1, N) + 2.0*T2             INTEG058
     1    + 4.0*FUNCT(X3, N ) + TB )                                    INTEG059
      DEL S = S2 - S1                                                   INTEG060
C     FORM TEST RATIO                                                   INTEG061
450   RATIO = (RERR * ABS(DEL S) ) / AMAX1(ABS(S2), CMAG )              INTEG062
      HT = DX                                                           INTEG063
C     RATIO TEST                                                        INTEG064
500   IF( RATIO - 1.0 )502,501,501                                      INTEG065
502   IF( RATIO - 0.5 )504,503,503                                      INTEG066
504   IF( RATIO - .01 )506,406,406                                      INTEG067
C     REJECT CYCLE, BRANCH TO X LINE OR Y STRIP.                        INTEG068
501   HT = 0.66666667 * HT                                              INTEG069
      GO TO 402                                                         INTEG070
C     ACCEPT CYCLE, BRANCH TO X LINE OR Y STRIP                         INTEG071
503   HT = 0.66666667 * HT                                              INTEG072
      GO TO 406                                                         INTEG073
506   HT = 1.5 * HT                                                     INTEG074
C     RESUME X LINE                                                     INTEG075
406   HX = AMIN1( HT,DX )                                               INTEG076
      DX = HT                                                           INTEG077
C     ADD EXTRAPOLATED VALUE TO PARTIAL SUM.                            INTEG078
423   ANS = ANS + S2 + 0.0666666667 * DEL S                             INTEG079
C     CHECK SWCHX IF DONE, IF NOT MAKE ANOTHER PASS                     INTEG080
```

```
      IF ( SWCHX ) RETURN
      GO TO 407
      END
```

```
      SUBROUTINE LAGIT(XBAR,X,Y,N,NP,YBAR,IO)                           LAGIT001
C     LAGRANGE INTERPOLATION ROUTINE BY A.PASTER, UNIVAC SYSTEMS PROGRAMMLAGIT002
C   ING.                                                                LAGIT003
      DIMENSION X(N),Y(N)                                               LAGIT004
      LOGICAL SW                                                        LAGIT005
      IF(NP.LT.2)  GOTO 9020                                            LAGIT006
C   TEST FOR XBAR IN RANGE                                              LAGIT007
      IO=0                                                              LAGIT008
      IF(XBAR-X(N)) 40, 45,9000                                         LAGIT009
45    YBAR=Y(N)                                                         LAGIT010
      RETURN                                                            LAGIT011
40    IF(XBAR-X(1)) 9010,50,50                                          LAGIT012
9010  IO=-NP                                                            LAGIT013
      GOTO 302                                                          LAGIT014
9000  IO=NP                                                             LAGIT015
      GOTO 305                                                          LAGIT016
C   FIND I=2**J .GE. N                                                  LAGIT017
50    I=2                                                               LAGIT018
      J=1                                                               LAGIT019
30    IF(N-I) 55,55,60                                                  LAGIT020
60    I=I+I                                                             LAGIT021
      J=J+1                                                             LAGIT022
      GOTO 30                                                           LAGIT023
C   SET UP FOR BINARY SEARCH                                            LAGIT024
55    I=I/2                                                             LAGIT025
      K=I                                                               LAGIT026
C   BINARY SEARCH FOR X(I) NEAR XBAR                                    LAGIT027
      J1=2                                                              LAGIT028
90    K=K/2                                                             LAGIT029
      IF(XBAR-X(I)) 120,100,110                                         LAGIT030
100   YBAR = Y(I)                                                       LAGIT031
      RETURN                                                            LAGIT032
110   I=I+K                                                             LAGIT033
115   IF(I.LE.N)   GOTO150                                              LAGIT034
      K=K/2                                                             LAGIT035
      I=I-K                                                             LAGIT036
      J1=J1+1                                                           LAGIT037
      GOTO 115                                                          LAGIT038
120   I=I-K                                                             LAGIT039
150   J1=J1+1                                                           LAGIT040
```

```
      IF(J1-J) 90,90,200                                                LAGIT041
200   XINC= XBAR-X(I)                                                   LAGIT042
      IP1=I+1                                                           LAGIT043
      IF(XINC) 220,100,210                                              LAGIT044
C  X(I) SHALL BE LESS THAN XBAR                                         LAGIT045
220   I=I-1                                                             LAGIT046
      GOTO 200                                                          LAGIT047
210   IEV= NP/2                                                         LAGIT048
      IF(NP-IEV*2) 9999,240,230                                         LAGIT049
230    IF(X(IP1)-XBAR.LT.XINC) GOTO240                                  LAGIT050
      IP1=I                                                             LAGIT051
240   J=IP1-IEV                                                         LAGIT052
      IF(J.LE.0) GOTO302                                                LAGIT053
      SW=.FALSE.                                                        LAGIT054
      GOTO310                                                           LAGIT055
C  SET FOR EXTRAPOLATE                                                  LAGIT056
302   J=1                                                               LAGIT057
      SW=.TRUE.                                                         LAGIT058
310   J1=J+NP-1                                                         LAGIT059
      IF(J1.LE.N)GOTO340                                                LAGIT060
      IF(SW) GOTO9020                                                   LAGIT061
C  EXTRAPOLATE OR ADJUST POINTS                                         LAGIT062
305   J=N-NP+1                                                          LAGIT063
      IF(J.LE.0)GOTO9020                                                LAGIT064
      J1=N                                                              LAGIT065
340   YBAR=0.0                                                          LAGIT066
      DO 400 K=J,J1                                                     LAGIT067
      PROD = 1.0                                                        LAGIT068
      DO 390 L= J,J1                                                    LAGIT069
      IF(K .EQ.L) GOTO 390                                              LAGIT070
      PROD =(XBAR-X(L))/(X(K)-X(L))*PROD                                LAGIT071
390   CONTINUE                                                          LAGIT072
400   YBAR= YBAR+ Y(K)*PROD                                             LAGIT073
      RETURN                                                            LAGIT074
9020  WRITE(6,9030)                                                     LAGIT075
      WRITE(6,9040)   NP,N,I,X(I),XBAR                                  LAGIT076
      STOP                                                              LAGIT077
9030  FORMAT(      1H121HINPUT ERROR IN LAGIT )                         LAGIT078
9040  FORMAT(1H03HNP=I6,4X2HN=I6,4X9HMID-POINTI5,4X4HX(I)F14.7,4X4HXBARFLAGIT079
     114.7)                                                             LAGIT080
```

```
9999  WRITE(6,9990)                                                          LAGIT081
      STOP                                                                   LAGIT082
9990  FORMAT(24H1MACHINE ERROR IN LAGIT )                                    LAGIT083
      END                                                                    LAGIT084
```

```
=      FOR DLAGIT                                                          DLAGI001
       SUBROUTINE DLAGIT(XBAR,X,FX,N,K,FXBAR,IERR)                         DLAGI002
C      LAGRANGIAN NUMERICAL DIFFERENTIATION                                DLAGI003
C      K=2 INDICATES THE ORDER OF POLYNOMINAL FIT BEING USED               DLAGI004
C      POINT TO BE INTERPOLATED MUST FALL WITHIN GIVEN                     DLAGI005
C      RANGE OF X POINTS (USER BEWARE OF INDICATOR )                       DLAGI006
C                                                                          DLAGI007
C      IF NEAR END POINTS THEN IT WILL REDUCE TO LOWEST POLY.              DLAGI008
C      TO FIT DATA POINTS(K-1,ETC. TO 3)                                   DLAGI009
C      THIS ROUTINE RETURNS ONLY ONE VALUE AT A TIME                       DLAGI010
C                                                                          DLAGI011
C      N=THE NUMBER OF POINT IN ARRAY                                      DLAGI012
C      K=THE NUMBER OF POINTS TO BE USED IN FORMULA                        DLAGI013
C      X=INDEPENDENT VARIABLE ARRAY                                        DLAGI014
C      FX=DEPENDENT VARIABLE ARRAY                                         DLAGI015
C      XBAR= TO POINT WHERE Y = F(XBAR) IS BEING SOUGHT                    DLAGI016
C      FXBAR= RESULT OF CORRESPONDING XBAR                                 DLAGI017
C      IND= ERROR INDICATOR ,-1 THEN XBAR.GT.X(N),                         DLAGI018
C      0 THEN XBAR OR, 1 THEN XBAR.LT.X(1)                                 DLAGI019
       DIMENSION X(N),FX(N)                                                DLAGI020
       LOGICAL ODD                                                         DLAGI021
       IND=0                                                               DLAGI022
       KAT=MOD(K,2)                                                        DLAGI023
       IF(KAT.EQ.1)ODD=.TRUE.                                              DLAGI024
       DO 20 I=1,N                                                         DLAGI025
       L=I                                                                 DLAGI026
       IF(XBAR.LT.X(I))GO TO 25                                            DLAGI027
  20   CONTINUE                                                            DLAGI028
       IND=-1                                                              DLAGI029
       RETURN                                                              DLAGI030
  25   CONTINUE                                                            DLAGI031
       IF(L.NE.1)GO TO 30                                                  DLAGI032
       IND=1                                                               DLAGI033
       RETURN                                                              DLAGI034
  30   CONTINUE                                                            DLAGI035
       IF(ODD)GO TO 40                                                     DLAGI036
  35   IF(L.LE.K/2 ) GO TO 66                                              DLAGI037
```

```
      IF(L.GT.(N-K/2+1)) GO TO 46                                       DLAGI038
   36 CONTINUE                                                          DLAGI039
      IHIGH=L+K/2-1                                                     DLAGI040
      ILOW=L-K/2                                                        DLAGI041
      GO TO 60                                                          DLAGI042
 40   CONTINUE                                                          DLAGI043
      IF(L.LE.K/2) GO TO 67                                             DLAGI044
      IF(L.GT.(N-(K+1)/2+1)) GO TO 46                                   DLAGI045
   37 CONTINUE                                                          DLAGI046
      IHIGH=L+(K+1)/2-1                                                 DLAGI047
      ILOW=L-(K-1)/2                                                    DLAGI048
      GO TO 60                                                          DLAGI049
  66   CONTINUE                                                         DLAGI050
      K=K-1                                                             DLAGI051
      GO TO 35                                                          DLAGI052
  67  CONTINUE                                                          DLAGI053
      K=K-1                                                             DLAGI054
       GOTO 40                                                          DLAGI055
 46   CONTINUE                                                          DLAGI056
      K=2+N-(L-1)                                                       DLAGI057
      IF(K.EQ.3)GO TO 50                                                DLAGI058
      IF(MOD(K,2).EQ.1)GO TO 37                                         DLAGI059
      GO TO 36                                                          DLAGI060
C     STRAIGHT LINE FIT AT FAR END POINT TWO BEHIND AND ONE AHEAD       DLAGI061
   50 CONTINUE                                                          DLAGI062
      IHIGH=L                                                           DLAGI063
      ILOW=L-2                                                          DLAGI064
 60   CONTINUE                                                          DLAGI065
      FXBAR=0.0                                                         DLAGI066
      DO 90 IQ=ILOW,IHIGH,1                                             DLAGI067
      SUMUJ=0.0                                                         DLAGI068
      DO 80 JQ=ILOW,IHIGH,1                                             DLAGI069
      IF(JQ.EQ.IQ)GO TO 80                                              DLAGI070
      PROD=1.                                                           DLAGI071
      DO 70 LQ=ILOW,IHIGH,1                                             DLAGI072
      IF(LQ.EQ.IQ)GO TO 70                                              DLAGI073
      IF(LQ.EQ.JQ)GO TO 70                                              DLAGI074
```

```
      PROD=PROD*(XBAR-X(LQ))                                            DLAGI075
   70 CONTINUE                                                          DLAGI076
      SUMUJ=SUMUJ+PROD                                                  DLAGI077
   80 CONTINUE                                                          DLAGI078
      PROD1=1.                                                          DLAGI079
      DO 85 MQ=ILOW,IHIGH,1                                             DLAGI080
      IF(MQ.EQ.IQ)GO TO 85                                              DLAGI081
      PROD1=PROD1*(X(IQ)-X(MQ))                                         DLAGI082
   85 CONTINUE                                                          DLAGI083
      FXBAR=FXBAR+FX(IQ)*(SUMUJ/PROD1)                                  DLAGI084
   90 CONTINUE                                                          DLAGI085
      RETURN                                                            DLAGI086
      END                                                               DLAGI087
```

```
      FUNCTION DTAB(X,Z,L)                                              DTAB001
      DIMENSION TEMP(1), LIST(8), L(13), AUXY(6), AUXZ(6)               DTAB002
      INTEGER DELY,DELX,DELXT,DELYT,SWITCH                              DTAB003
      LIST(1) = LOC(LIST(1))                                            DTAB004
      LIST(2) = L(2)                                                    DTAB005
      LIST(4) = L(3)                                                    DTAB006
      NDIF = LIST(4)                                                    DTAB007
      IF(NDIF-1) 10, 9, 10                                              DTAB008
C     THREE ARRAYS                                                      DTAB009
    9 DELY = L(12)                                                      DTAB010
      DELXT = 1                                                         DTAB011
      DELYT = 1                                                         DTAB012
      SWITCH = 1                                                        DTAB013
      DELX = L(11)                                                      DTAB014
      LAMDA = DELX-1                                                    DTAB015
      IF(LAMDA)12,11,12                                                 DTAB016
C     X ARRAY SINGLY SUBSCRIPTED                                        DTAB017
   11 DELXT = LAMDA                                                     DTAB018
C     X ARRAY DOUBLY SUBSCRIPTED                                        DTAB019
   12 LOCX = L(9)                                                       DTAB020
      N = L(13)                                                         DTAB021
      LOCY = L(10)                                                      DTAB022
      GO TO 126                                                         DTAB023
C     ONE ARRAY                                                         DTAB024
   10 N = (LIST(4)+1)/2                                                 DTAB025
      DELXT = L(3)                                                      DTAB026
      DELYT = L(3)                                                      DTAB027
      SWITCH = 0                                                        DTAB028
      DELX = 2                                                          DTAB029
      DELY = 2                                                          DTAB030
  126 KX = L(6)                                                         DTAB031
      KZ = L(5)                                                         DTAB032
      KZP1 = KZ+1                                                       DTAB033
C     SET UP AUXILLARY Z ARRAY                                          DTAB034
      M2 = L(4)                                                         DTAB035
      NZS = L(4)-1                                                      DTAB036
      INITZT = L(2)-LOC(TEMP(1))+1                                      DTAB037
      IF (INITZT.LT.0)  INITZT=65536+INITZT                             DTAB038
      IF (NZS-KZ) 38, 52, 49                                            DTAB039
   49 IZ = INITZT+NDIF                                                  DTAB040
```

```
      KZD2 = KZ/2                                                       DTAB041
      KZP1D2 = KZP1/2                                                   DTAB042
      IK = INITZT+(KZP1D2+1)*NDIF                                       DTAB043
      IHI = INITZT+NDIF*(NZS-KZP1D2)                                    DTAB044
      IF (TEMP(INITZT).GT.TEMP(IZ))  GO TO 50                           DTAB045
      IF (TEMP(IK-NDIF)-Z)  51,52,52                                    DTAB046
   51 DO 53 I=IK, IHI, NDIF                                             DTAB047
      IF (TEMP(I)-Z)  53, 54, 54                                        DTAB048
   53 CONTINUE                                                          DTAB049
      I = INITZT+NZS *NDIF                                              DTAB050
      GO TO 62                                                          DTAB051
   52 I = INITZT+KZ*NDIF                                                DTAB052
      GO TO 62                                                          DTAB053
   50 IF (Z-TEMP(IK-NDIF)) 56, 52, 52                                   DTAB054
   56 DO 57 I=IK, IHI, NDIF                                             DTAB055
      IF (Z-TEMP(I)) 57, 54, 54                                         DTAB056
   57 CONTINUE                                                          DTAB057
      I = INITZT+(NZS)*NDIF                                             DTAB058
      GO TO 62                                                          DTAB059
   54 IF ((KZD2 + KZD2).NE.KZ)  GO TO 55                                DTAB060
      IZ = I-NDIF                                                       DTAB061
      IF (ABS(TEMP(IZ)-Z)-ABS(TEMP(I)-Z)) 67, 55, 55                    DTAB062
   67 I = IZ                                                            DTAB063
      GO TO 55                                                          DTAB064
   55 I = I+KZD2*NDIF                                                   DTAB065
   62 JK = I-INITZT                                                     DTAB066
      DO 65 N=1, KZP1                                                   DTAB067
      AUXZ(N) = TEMP(I)                                                 DTAB068
   65 I = I-NDIF                                                        DTAB069
      IF(SWITCH)17,18,17                                                DTAB070
   17 IF(LAMDA)19,20,19                                                 DTAB071
   19 LIST(2) = LOCX + JK                                               DTAB072
      GO TO 21                                                          DTAB073
   20 LIST(2) = LOCX                                                    DTAB074
   21 LIST(3) = LOCY + JK                                               DTAB075
      LIST(4) = DELX                                                    DTAB076
      LIST(5) = DELY                                                    DTAB077
      LIST(6) = KX                                                      DTAB078
      LIST(7) = N                                                       DTAB079
      LIST(8) = 0                                                       DTAB080
```

```
      GO TO 30                                                          DTAB081
   18 LIST(2) = LIST(2)+JK+1                                            DTAB082
      LIST(3) = LIST(2)+1                                               DTAB083
      LIST(4) = 2                                                       DTAB084
      LIST(5) = 2                                                       DTAB085
      LIST(6) = KX                                                      DTAB086
      LIST(7) = N                                                       DTAB087
      LIST(8) = 0                                                       DTAB088
C     SET UP AUXILLARY Y ARRAY                                          DTAB089
   30 DO 35 I=1,KZP1                                                    DTAB090
      AUXY(I) = TAB(X,LIST)                                             DTAB091
      IF(LIST(8)-2)31,37,38                                             DTAB092
   31 LIST(2) = LIST(2)-DELXT                                           DTAB093
      LIST(3) = LIST(3)-DELYT                                           DTAB094
   35 CONTINUE                                                          DTAB095
      LIST(2) = LOC(AUXZ(1))                                            DTAB096
      LIST(3) = LOC(AUXY(1))                                            DTAB097
      LIST(4) = 1                                                       DTAB098
      LIST(5) = 1                                                       DTAB099
      LIST(6) = KZ                                                      DTAB100
      LIST(7) = KZP1                                                    DTAB101
      LIST(8) = 0                                                       DTAB102
      YDEP = TAB(Z,LIST)                                                DTAB103
      IF(LIST(8)-2)36,37,38                                             DTAB104
   36 L(8) = 1                                                          DTAB105
      DTAB = YDEP                                                       DTAB106
      RETURN                                                            DTAB107
   37 L(8) = 2                                                          DTAB108
      RETURN                                                            DTAB109
   38 L(8) = 3                                                          DTAB110
      RETURN                                                            DTAB111
      END                                                               DTAB112
```

```
      FUNCTION TAB (X,L)                                                TAB001
      DIMENSION L(8),T(1) ,XX(6),YY(6)                                  TAB002
      EQUIVALENCE (I,II)                                                TAB003
      EQUIVALENCE (T(1),INITXT)                                         TAB004
      IVLX=L(4)                                                         TAB005
      IVLY=L(5)                                                         TAB006
      K=L(6)                                                            TAB007
      NTAB=L(7) -1                                                      TAB008
 24   INITXT=L(2)-LOC (T(1)) +1                                         TAB009
      INITYT=L(3)-LOC (T(1)) +1                                         TAB010
      IF (INITXT.LT.0)       INITXT=65536+INITXT                        TAB011
      IF (INITYT.LT.0)       INITYT=65536+INITYT                        TAB012
      KP1=K+1                                                           TAB013
      IF (NTAB-K)       106,1000,23                                     TAB014
 23   KD2=K/2                                                           TAB015
      KP1D2=(KP1)/2                                                     TAB016
      IX=INITXT+IVLX                                                    TAB017
 26   J=INITYT+(KP1D2+1)*IVLY                                           TAB018
      II=INITXT+(KP1D2+1)*IVLX                                          TAB019
      IHIGH=INITXT+IVLX*(NTAB-KP1D2)                                    TAB020
      IF (T(INITXT).GT.T(IX) )    GO TO 105                             TAB021
      IF (T(II-IVLX)-X) 101,1000,1000                                   TAB022
 101  DO 27 I=II,IHIGH,IVLX                                             TAB023
      IF (T(I)-X)           27,6,6                                      TAB024
 27   J=J+IVLY                                                          TAB025
 105  J=INITYT+(NTAB)*IVLY                                              TAB026
      I=INITXT+(NTAB)*IVLX                                              TAB027
      GO TO 102                                                         TAB028
 6    IF ((KD2+KD2).NE.K)        GO TO 5                                TAB029
      IX=I-IVLX                                                         TAB030
      IF (ABS(T(IX)-X)-ABS(T(I)-X))   7,5,5                             TAB031
 5    I=I+KD2*IVLX                                                      TAB032
      J=J+KD2*IVLY                                                      TAB033
 102  DO 8 N=1,KP1                                                      TAB034
      M1 = KP1-N+1                                                      TAB035
      XX(M1) = X-T(I)                                                   TAB036
      YY(M1) = T(J)                                                     TAB037
      I=I-IVLX                                                          TAB038
 8    J=J-IVLY                                                          TAB039
      CALL OVERFL(M)                                                    TAB040
```

```
      DO 9 N=1,K                                                        TAB041
      NP1=N+1                                                           TAB042
      DO 9 NN=NP1,KP1                                                   TAB043
      YY(NN)=(YY(N)*XX(NN)-YY(NN)*XX(N))/(XX(NN)-XX(N))                 TAB044
      CALL OVERFL(M)                                                    TAB045
      IF (M.EQ.1)      GO TO 10                                         TAB046
9     CONTINUE                                                          TAB047
      L(8)=1                                                            TAB048
      TAB =YY(KP1)                                                      TAB049
      RETURN                                                            TAB050
106   L(8)=3                                                            TAB051
      GO TO 107                                                         TAB052
10    L(8)=2                                                            TAB053
107   TAB =X                                                            TAB054
      RETURN                                                            TAB055
7     I=IX                                                              TAB056
      J=J-IVLY                                                          TAB057
      GO TO 5                                                           TAB058
100   IF (X-T(II-IVLX))    103,1000,1000                                TAB059
103   DO 104 I=II,IHIGH,IVLX                                            TAB060
      IF (X-T(I)) 104,6,6                                               TAB061
104   J=J+IVLY                                                          TAB062
      GO TO 105                                                         TAB063
1000  I=INITXT+K*IVLX                                                   TAB064
      J=INITYT+K*IVLY                                                   TAB065
      GO TO 102                                                         TAB066
      END                                                               TAB067
```

7.2 Ablator Program

```
= FOR MAIN                                                              MAIN 001
C       SOLUTION OF 5 ALGEBRAIC EQUATIONS WITH THE                      MAIN 002
C       UNKNOWNS X(I)=(MDP,LAM,TW,MDS,MDC)     CASE 1                   MAIN 003
C       UNKNOWNS X(I)=(MDP,LAM,TW,MDC)         CASE 2                   MAIN 004
C       UNKNOWNS X(I)=(MDP,LAM,TW,MDS)         CASE 3                   MAIN 005
C       UNKNOWNS X(I)=(MDP,LAM,TW)             CASE 4                   MAIN 006
C                                                                       MAIN 007
        REAL MDP,MDC,MDS,LAM,KO2E,K1,K2,K4,MDEX                         MAIN 008
        REAL LAM1,MDP1,MDS1                                             MAIN 009
        DIMENSION X(5),F(5),DF(5,5),X1(5),XL(5),XUB(5),XLB(5),E(5)      MAIN 010
        DIMENSION LAM1(100),TPHI1(100),MDP1(100),TPHI2(100),MDS1(100),  MAIN 011
       1TPHI3(100)                                                      MAIN 012
        EQUIVALENCE(X(1),MDP),(X(2),LAM),(X(3),TW),(DF(1,1),X1(1))      MAIN 013
        DATA(XL(I),I=1,5)/.5,.5,.5,.5,.5/                               MAIN 014
C                                                                       MAIN 015
      1 CONTINUE                                                        MAIN 016
        READ(5,100)NPHI1,(TPHI1(I),LAM1(I),I=1,NPHI1)                   MAIN 017
        WRITE(6,201)(TPHI1(I),LAM1(I),I=1,NPHI1)                        MAIN 018
  201   FORMAT(9H0 TABLE 1,10X,13H (PHI1,LAMDA)/                        MAIN 019
       11P8E15.5/(8E15.5))                                              MAIN 020
        READ(5,100)NPHI2,(TPHI2(I),MDP1(I),I=1,NPHI2)                   MAIN 021
        WRITE(6,202)(TPHI2(I),MDP1(I),I=1,NPHI2)                        MAIN 022
  202   FORMAT(9H0 TABLE 2,10X,15H (PHI2,M DOT P)/                      MAIN 023
       11P8E15.5/(8E15.5))                                              MAIN 024
        READ(5,100)NPHI3,(TPHI3(I),MDS1(I),I=1,NPHI3)                   MAIN 025
        WRITE(6,203)(TPHI3(I),MDS1(I),I=1,NPHI3)                        MAIN 026
  203   FORMAT(9H0 TABLE 3,10X,15H (PHI3,M DOT S)/                      MAIN 027
       11P8E15.5/(8E15.5))                                              MAIN 028
C                                                                       MAIN 029
      2 CONTINUE                                                        MAIN 030
        WRITE(6,205)                                                    MAIN 031
  205   FORMAT(23H0 PARAMETER INPUT CARDS)                              MAIN 032
        READ(5,110)H0,PSI,Q0,P,T0,U,RHOP,RHOC                           MAIN 033
        WRITE(6,210)H0,PSI,Q0,P,T0,U,RHOP,RHOC                          MAIN 034
        READ(5,110)CPC,CPP,DHCC,DHCP,DHPYR,FP,SIG,A                     MAIN 035
        WRITE(6,210)CPC,CPP,DHCC,DHCP,DHPYR,FP,SIG,A                    MAIN 036
        READ(5,110)KO2E,K1,K2,K4                                        MAIN 037
```

```
      WRITE(6,210)KO2E,K1,K2,K4                                         MAIN 038
      READ(5,110)AS,BS,QRAD,FP,DHS                                      MAIN 039
      WRITE(6,210)AS,BS,QRAD,FP,DHS                                     MAIN 040
      READ(5,110)(XUB(I),I=1,5)                                         MAIN 041
      READ(5,110)(XLB(I),I=1,5)                                         MAIN 042
C                                                                       MAIN 043
      XUB(1)=AMIN1(MDP1(NPHI2),XUB(1))                                  MAIN 044
      XUB(2)=AMIN1(LAM1(1),XUB(2))                                      MAIN 045
      XUB(4)=AMIN1(MDS1(NPHI3),XUB(4))                                  MAIN 046
      XLB(1)=AMAX1(MDP1(1),XLB(1))                                      MAIN 047
      XLB(2)=AMAX1(LAM1(NPHI1),XLB(2))                                  MAIN 048
      XLB(4)=AMAX1(MDS1(1),XLB(4))                                      MAIN 049
C                                                                       MAIN 050
      KDONE=0                                                           MAIN 051
    3 CONTINUE                                                          MAIN 052
      READ(5,110)MDP,LAM,TW,MDS,MDC                                     MAIN 053
  100 FORMAT(I5,5X/(8F10.0))                                            MAIN 054
  110 FORMAT(8F10.0)                                                    MAIN 055
  210 FORMAT(1P8E15.6)                                                  MAIN 056
C                                                                       MAIN 057
C     SHOULD MDS BE INCLUDED IN VECTOR                                  MAIN 058
C                                                                       MAIN 059
      IF(U)10,10,11                                                     MAIN 060
   10 CONTINUE                                                          MAIN 061
C     CASE 2 OR CASE 4 (DISCARD MDS FROM VECTOR)                        MAIN 062
      NEQ=4                                                             MAIN 063
      LS=5                                                              MAIN 064
      LC=4                                                              MAIN 065
      KLS=1                                                             MAIN 066
      MDS=0.0                                                           MAIN 067
      X(5)=MDS                                                          MAIN 068
      XUB(4)=XUB(5)                                                     MAIN 069
      XLB(4)=XLB(5)                                                     MAIN 070
      GO TO 12                                                          MAIN 071
   11 CONTINUE                                                          MAIN 072
C     CASE 1 OR CASE 3 (MDS IS IN VECTOR)                               MAIN 073
      NEQ=5                                                             MAIN 074
```

```
      LS=4                                                              MAIN 075
      LC=5                                                              MAIN 076
      KLS=0                                                             MAIN 077
   12 CONTINUE                                                          MAIN 078
      X(LS)=MDS                                                         MAIN 079
      X(LC)=MDC                                                         MAIN 080
C                                                                       MAIN 081
      WRITE(6,220)XUB,XLB                                               MAIN 082
  220 FORMAT(12H UPPER BOUND,8X,1P5E20.6/12H LOWER BOUND,8X,5E20.6)     MAIN 083
C                                                                       MAIN 084
      DO 13 I=1,5                                                       MAIN 085
      X(I)=AMAX1(X(I),XLB(I))                                           MAIN 086
      X(I)=AMIN1(X(I),XUB(I))                                           MAIN 087
   13 CONTINUE                                                          MAIN 088
      IF(KDONE)14,14,90                                                 MAIN 089
   14 CONTINUE                                                          MAIN 090
      KMDC=0                                                            MAIN 091
      KLC=0                                                             MAIN 092
C                                                                       MAIN 093
      DO 50 JJJ=1,50                                                    MAIN 094
C                                                                       MAIN 095
      XMDC=A*H0*KO2E-MDP*LAM                                            MAIN 096
      IF(XMDC)15,15,17                                                  MAIN 097
   15 CONTINUE                                                          MAIN 098
      KMDC=KMDC+1                                                       MAIN 099
      IF(3-KMDC)16,16,18                                                MAIN 100
   16 CONTINUE                                                          MAIN 101
C     CASE 3 OR CASE 4 (DISCARD MDC FROM VECTOR)                        MAIN 102
      KLC=1                                                             MAIN 103
      NEQ=4-KLS                                                         MAIN 104
      MDC=0.0                                                           MAIN 105
      X(LC)=XLB(LC)                                                     MAIN 106
      GO TO 18                                                          MAIN 107
   17 CONTINUE                                                          MAIN 108
C     CASE 1 OR CASE 2 (MDC IS IN VECTOR)                               MAIN 109
      KLC=0                                                             MAIN 110
      KMDC=0                                                            MAIN 111
```

```
      NEQ=5-KLS                                                         MAIN 112
      MDC=AMAX1(MDC,XLB(LC))                                            MAIN 113
   18 CONTINUE                                                          MAIN 114
C                                                                       MAIN 115
      MDEX=MDP*((MDP/RHOP)-(MDC+MDS+AS*EXP(-BS/TW))/RHOC)               MAIN 116
      MDEX=AMAX1(0.0,MDEX)                                              MAIN 117
      TEMP=TW*ALOG(K2)*K2**MDEX                                         MAIN 118
      TEMP1=TEMP*(2.0*MDP/RHOP-(MDC+MDS+AS*EXP(-BS/TW))/RHOC)           MAIN 119
      DIFM=PSI*H0*0.23-MDP*0.90909]                                     MAIN 120
      IF(DIFM)52,52,53                                                  MAIN 121
   52 FPI=(PSI*H0*0.23)/(MDP*0.90909])                                  MAIN 122
      DFPI= 0.0                                                         MAIN 123
      GO TO 54                                                          MAIN 124
   53 FPI=1.0                                                           MAIN 125
      DFPI=FP*DHCP                                                      MAIN 126
   54 CONTINUE                                                          MAIN 127
C                                                                       MAIN 128
      PHI1=MDP*SQRT(TW)*K1**(1.0E+4/TW)/P                               MAIN 129
      PHI2=          (TW)*K2**MDEX                                      MAIN 130
C                                                                       MAIN 131
      PHI1=AMAX1(PHI1,TPHI1(1))                                         MAIN 132
      PHI2=AMAX1(PHI2,TPHI2(1))                                         MAIN 133
      PHI1=AMIN1(PHI1,TPHI1(NPHI1))                                     MAIN 134
      PHI2=AMIN1(PHI2,TPHI2(NPHI2))                                     MAIN 135
C                                                                       MAIN 136
      CALL LAGIT(PHI1,TPHI1,LAM1,NPHI1,2,FPHI1,IERR)                    MAIN 137
      CALL LAGIT(PHI2,TPHI2,MDP1,NPHI2,2,FPHI2,IERR)                    MAIN 138
      CALL DLAGIT(PHI1,TPHI1,LAM1,NPHI1,3,DPHI1,IERR)                   MAIN 139
      CALL DLAGIT(PHI2,TPHI2,MDP1,NPHI2,3,DPHI2,IERR)                   MAIN 140
C                                                                       MAIN 141
      F(1)=PSI*Q0+QRAD+FP*FPI*DHCP*MDP+(MDC+MDS)*DHCC-EP*SIG            MAIN 142
     1*TW**4-MDP*(CPP*(TW-T0)+DHPYR)-(MDC+MDS+AS*EXP(-BS/TW))           MAIN 143
     2*CPC*(TW-T0)-AS*EXP(-BS/TW)*DHS                                   MAIN 144
C                                                                       MAIN 145
      DF(1,1)=DFPI    -CPP*(TW-T0)-DHPYR                                MAIN 146
      DF(1,2)=0.0                                                       MAIN 147
      DF(1,3)=-4.0*EP*SIG*TW**3-MDP*CPP-(MDC+MDS+AS*EXP(-BS             MAIN 148
```

```
     1/TW))*CPC-AS*CPC*(TW-T0)*EXP(-BS/TW)*BS/TW**2-AS*DHS
     2*EXP(-BS/TW)*BS/TW**2
      DF(1,LS)=DHCC-CPC*(TW-T0)
      DF(1,LC)=DF(1,LS)
C
      F(2)=LAM-FPHI1
C
      DF(2,1)=-DPHI1*SQRT(TW)*K1**(1.0E+4/TW)/P
      DF(2,2)=1.0
      DF(2,3)=-DPHI1*MDP/P*K1**(1.0E+4/TW)*(SQRT(TW)
     1*ALOG(K1)*(-1.0E+4)/TW**2+0.5/SQRT(TW))
      DF(2,LS)=0.0
      DF(2,LC)=0.0
C
      F(3)=MDP-FPHI2
C
      DF(3,1)=1.0-DPHI2*TEMP1
      DF(3,2)=0.0
      DF(3,3)=-DPHI2*(K2**MDEX-TEMP*MDP* AS*EXP(-BS/TW)
     1*(BS/TW**2)/RHOC)
      DF(3,LC)=-DPHI2     *TEMP*(-MDP/RHOC)
      DF(3,LS)=DF(3,LC)
C
      F(LC)=MDC-XMDC
C
      DF(LC,1)=LAM
      DF(LC,2)=MDP
      DF(LC,3)=0.0
      DF(LC,LC)=1.0
      DF(LC,LS)=0.0
C
      IF(KLS)21,21,22
   21 CONTINUE
      PHI3=PSI*H0*U*K4**(1.0E+4/TW)
      PHI3=AMAX1(PHI3,TPHI3(1))
      PHI3=AMIN1(PHI3,TPHI3(NPHI3))
C
```

```
      CALL LAGIT(PHI3,TPHI3,MDS1,NPHI3,2,FPHI3,IERR)
      CALL DLAGIT(PHI3,TPHI3,MDS1,NPHI3,3,DPHI3,IERR)
C
      F(LS)=MDS-FPHI3
C
      DF(LS,1)=0.0
      DF(LS,2)=0.0
      DF(LS,3)=-DPHI3*PSI*HO*U*ALOG(K4)*((K4)**(1.0F+4/TW))
     1*(-1.0E+4/(TW**2))
      DF(LS,LS)=1.0
      DF(LS,LC)=0.0
   22 CONTINUE
C
      JKASE=2*KLC+KLS+1
      GO TO(141,142,143,144),JKASE
  141 CONTINUE
      WRITE(6,241)JJJ
  241 FORMAT(12H0 ITERATION I3,10X,7H CASE 1,10X,
     153H  (X) = ( M DOT P , LAMDA , T W , M DOT S , M DOT C ))
      GO TO 145
  142 CONTINUE
      WRITE(6,242)JJJ
  242 FORMAT(12H0 ITERATION I3,10X,7H CASE 2,10X,
     143H  (X) = ( M DOT P , LAMDA , T W , M DOT C ))
      GO TO 145
  143 CONTINUE
      WRITE(6,243)JJJ
  243 FORMAT(12H0 ITERATION I3,10X,7H CASE 3,10X,
     143H  (X) = ( M DOT P , LAMDA , T W , M DOT S ))
      GO TO 145
  144 CONTINUE
      WRITE(6,244)JJJ
  244 FORMAT(12H0 ITERATION I3,10X,7H CASE 4,10X,
     133H  (X) = ( M DOT P , LAMDA , T W ))
  145 CONTINUE
C
      WRITE(6,250)(F(I),I=1,NFQ)
```

```
  250 FORMAT(3H   F,10X,1P5E20.7)
      WRITE(6,251)(X(I),I=1,NEQ)
  251 FORMAT(6H      X,7X,1P5E20.7)
C
      DO 24 I=1,NEQ
      F(I)=-F(I)
   24 CONTINUE
C
      D=1.0
      M=NSIMEQ(5,NEQ,1,DF,F,D,E)
C
      GO TO(25,85,85),M
 25   CONTINUE
      WRITE(6,252)(X1(I),I=1,NEQ)
 252  FORMAT(6H   DFLX,7X,1P5E20.7)
C
      DO 35 I=1,NEQ
      X1(I)=X(I)+X1(I)
      IF(XUB(I)-X1(I))28,30,30
 28   CONTINUE
      X1(I)=.5*(XUB(I)+X(I))
      GO TO 35
 30   CONTINUE
      IF(X1(I)-XLB(I))32,35,35
 32   CONTINUE
      X1(I)=.5*(XLB(I)+X(I))
 35   CONTINUE
C
      WRITE(6,255)(X1(I),I=1,NEQ)
 255  FORMAT(4H   X1,9X,1P5E20.7)
C
      SUMF=0.0
      DO 40 I=1,NEQ
      SUMF=SUMF+ABS(F(I))
      ERR=ABS((X(I)-X1(I))/X(I))
      IF(ERR-.040)40,42,42
   40 CONTINUE
```

```
      IF(SUMF-.05)60,42,42
   42 CONTINUE
      DO 45 I=1,NEQ
      X(I)=XL(I)*X1(I)+(1.0-XL(I))*X(I)
   45 CONTINUE
C
      IF(KLS)46,46,47
 46   MDS=X(LS)
 47   IF(KLC)48,48,49
 48   MDC=X(LC)
 49   CONTINUE
   50 CONTINUE
      WRITE(6,260)
  260 FORMAT(15H0 NO COVERGENCE)
      GO TO 90
C
   60 CONTINUE
      WRITE(6,270)
 270  FORMAT(15H0 CASE COMPLETE)
      GO TO 90
 85   CONTINUE
      WRITE(6,280)M
 280  FORMAT(29H ERROR IN MATRIX SOLUTION M =I2)
   90 CONTINUE
      READ(5,115)KASE
  115 FORMAT(I5)
      GO TO (1,2,3),KASE
      END
```

```
      FUNCTION NSIMEQ(N,LN,LM,A,B,D,E)                                  NSIME001
      M=1                                                               NSIME002
      CALL SIMEQ(N,LN,LM,A,B,D,E,M)                                     NSIME003
      NSIMEQ=M                                                          NSIME004
      RETURN                                                            NSIME005
      END                                                               NSIME006
```

```
      SUBROUTINE SIMEQ(N,LN,LM,A,B,D,E,M)
      INTEGER E
      EQUIVALENCE (SAVE,ISAVE)
      DIMENSION E(LN),A(N,N),B( N,LM)
      IF (M.EQ.0)      GO TO 2
      DO 1 I=1,LN
1     E(I)=I
2     LNM1=LN-1
      CALL OVERFL  (IBIG)
      DO 139 K=1,LNM1
      SAVE=-1.0
      K1=K+1
      DO 4 J=K,LN
      DO 4 I=K,LN
      IF (SAVE -  ABS(A(I,J)))   3,4,4
3     SAVE=ABS(A(I,J))
      IBIG=I
      JBIG=J
4     CONTINUE
      IF (K.EQ.IBIG)      GO TO 61
      D=-D
      DO 6 J=K,LN
      SAVE=A(K,J)
      A(K,J)=A(IBIG,J)
6     A(IBIG,J)=SAVE
      IF (M.EQ.0)      GO TO 61
      DO 7 J=1,LM
      SAVE=B(K,J)
      B(K,J)=B(IBIG,J)
7     B(IBIG,J)=SAVE
61    IF (K-JBIG) 8,89,8
8     D=-D
      DO 9 I=1,LN
      SAVE=A(I,K)
      A(I,K)=A(I,JBIG)
9     A(I,JBIG)=SAVE
      IF (M.EQ.0)      GO TO 89
      ISAVE=E(K)
      E(K)=E(JBIG)
      E(JBIG)=ISAVE
```

```
89    IF (A(K,K))  600,10,600                                         SIMEQ041
 600  IF (D) 602, 603, 602                                            SIMEQ042
 602  D = D*A(K,K)                                                    SIMEQ043
 603  DO 139 I=K1, LN                                                 SIMEQ044
      SAVE=A(I,K)/A(K,K)                                              SIMEQ045
      DO 15 J=K1,LN                                                   SIMEQ046
15    A(I,J)=A(I,J)-SAVE*A(K,J)                                       SIMEQ047
      CALL OVERFL(IBIG)                                               SIMEQ048
      IF (IBIG-1)     710,12,710                                      SIMEQ049
710   IF (M.EQ.0)       GO TO 139                                     SIMEQ050
      DO 138 J=1,LM                                                   SIMEQ051
138   B(I,J)=B(I,J)-SAVE*B(K,J)                                       SIMEQ052
      CALL OVERFL(IBIG)                                               SIMEQ053
      IF (IBIG-1)     139,12,139                                      SIMEQ054
139   CONTINUE                                                        SIMEQ055
      IF (A(LN,LN))     601,10,601                                    SIMEQ056
 601  IF (D) 604, 150, 604                                            SIMEQ057
 604  IF (LN.NE.1) D=D*A(LN,LN)                                      SIMEQ058
      CALL OVERFL(IBIG)                                               SIMEQ059
      IF (IBIG-1)       150,12,150                                    SIMEQ060
150   IF (M) 118,250,118                                              SIMEQ061
118   DO 20 J=1,LM                                                    SIMEQ062
        LN,J)=B(LN,J)/A(LN,LN)                                        SIMEQ063
      CALL OVERFL(IBIG)                                               SIMEQ064
      IF (IBIG-1)       18,12,18                                      SIMEQ065
18    DO 20 JBIG=1,LNM1                                               SIMEQ066
      I=LN-JBIG                                                       SIMEQ067
      SAVE=0.                                                         SIMEQ068
      IP1=I+1                                                         SIMEQ069
      DO 19 K=IP1,LN                                                  SIMEQ070
19    SAVE=SAVE+A(I,K)*B(K,J)                                         SIMEQ071
      B(I,J)=(B(I,J)-SAVE)/A(I,I)                                     SIMEQ072
      CALL OVERFL(IBIG)                                               SIMEQ073
      IF (IBIG-1)       20,12,20                                      SIMEQ074
20    CONTINUE                                                        SIMEQ075
      DO 21 K=1,LN                                                    SIMEQ076
      I=E(K)                                                          SIMEQ077
      DO 21 J=1,LM                                                    SIMEQ078
21    A(I,J)=B(K,J)                                                   SIMEQ079
250   M=1                                                             SIMEQ080
```

```
      RETURN                                                            SIMEQ081
   12 M=2                                                               SIMEQ082
      RETURN                                                            SIMEQ083
   10 M=3                                                               SIMEQ084
      RETURN                                                            SIMEQ085
      END                                                               SIMEQ086
```

8.0 REFERENCES

1. Fluid Mechanics Group: Thermodynamic and Transport Properties of Air in Dissociation and Ionization Equilibrium. Boeing Document D2-5129, December 30, 1959.

2. Hayes, W. D.; and Probstein, R. F.: Hypersonic Flow Theory. New York Academic Press, New York, 1959.

3. Sperry Rand Corporation: UNIVAC 1107 BEEF Math Routines. UP-3984, March 23, 1965.

4. Price, J. F.; and Simonsen, R. H.: Various Methods and Computer Routines for Approximation, Curve Fitting, and Interpolation. Boeing Document D1-82-0151 (BSRL Math Note 249), February 1962.