NASA TECHNICAL NOTE **NASA TN D-5681**

NASA TN D-5681

FORTRAN PROGRAM FOR CALCULATING COMPRESSIBLE LAMINAR AND TURBULENT BOUNDARY LAYERS IN ARBITRARY PRESSURE GRADIENTS

by William D. McNally

Lewis Research Center
Cleveland, Ohio 44135

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MAY 1970

1. Report No.
NASA TN D-5681

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
FORTRAN PROGRAM FOR CALCULATING COMPRESSIBLE LAMINAR AND TURBULENT BOUNDARY LAYERS IN ARBITRARY PRESSURE GRADIENTS

5. Report Date
May 1970

6. Performing Organization Code

7. Author(s)
William D. McNally

8. Performing Organization Report No.
E-5256

9. Performing Organization Name and Address
Lewis Research Center
National Aeronautics and Space Administration
Cleveland, Ohio 44135

10. Work Unit No.
126-15

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D.C. 20546

13. Type of Report and Period Covered
Technical Note

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A FORTRAN IV program is presented which solves the two-dimensional, compressible laminar or turbulent boundary-layer equations in an arbitrary pressure gradient. Cohen and Reshotko's method is used for the laminar boundary layer, Sasman and Cresci's method for the turbulent boundary layer, and the Schlichting-Ulrich-Granville method to predict transition. Transition may also be forced at any point by the user. Separation, if it occurs, is predicted for both laminar and turbulent flow. The user may begin with a laminar boundary layer at a stagnation point, or may give initial values for displacement thickness and momentum thickness in either laminar or turbulent flow.

17. Key Words *(Suggested by Author(s))*
Compressible laminar boundary layer
Compressible turbulent boundary layer
Arbitrary pressure gradient
FORTRAN computer program
Turbomachinery - compressor, turbine

18. Distribution Statement
Unclassified - unlimited

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
111

22. Price*
$3.00

*For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151

CONTENTS

Page

SUMMARY 1

INTRODUCTION 2

INFORMATION FOR GENERAL USER 3
  DESCRIPTION OF METHOD 3
    Basic Characteristics of Program 3
    Input and Output Characteristics 4
    Limitations of Program 5
  BASIC EQUATIONS 5
    Laminar Solution 6
    Turbulent Solution 7
    Transition. 10
    Separation. 11
      Laminar case 11
      Turbulent case 11
  NUMERICAL EXAMPLES 11
    NACA 0012 Airfoil 14
    Converging-Diverging Channel 14
  INPUT 17
    Dictionary of Input Variables 17
    Special Instructions for Preparing Input 20
      Surface geometry. 20
      Leading-edge conditions 21
      Inlet conditions 22
      Surface distributions of pressure, velocity, and Mach number. 23
      Smoothing of surface distributions. 23
      Calculation of surface gradients 23
      Arbitrary selection of transition point. 23
      Use of initial values of δ^* and θ. 24
      Laminar separation and reattachment 24
      English or metric units 24
      Format for input data 24
  OUTPUT. 25
  ERROR CONDITIONS 33

INFORMATION FOR PROGRAMMER 36
  DESCRIPTION OF PROGRAM 36

- Conventions Used in Program 37
- Labeled COMMON Blocks 37

MAIN SUBROUTINES 37

- Subroutine INPUT 38
- Subroutine PRECAL 38
  - Reading of coefficients for curve fits 38
  - Initial calculations 38
  - Calculation and smoothing of surface flow distributions 38
  - Calculation of surface gradients 39
- Subroutine LAMNAR 39
  - Reading of coefficients for curve fits 39
  - Initial values 39
  - Solution of the laminar differential equation 41
  - Calculation of laminar-boundary-layer parameters 42
  - Check for instability and transition 44
  - Initial values for turbulent boundary layer 45
- Subroutine TURBLN 45
  - Solution of turbulent differential equations 45
  - Calculation of turbulent-boundary-layer parameters 45
- Subroutine PROFIL 47
  - Laminar profiles 47
  - Turbulent profiles 47
- Subroutine RUNKUT 48
- Subroutine FUNCT 49
- Function INT1 49
- Function INT2 50

DICTIONARY OF VARIABLES IN MAIN SUBROUTINES 50

AUXILIARY SUBROUTINES 57

- Subroutine SMOOTH 57
- Subroutine GRADNT 58
- Subroutine SPLINE 60
- Subroutine ROOT 60
- Subroutine LGRNGE 62
- Subroutine CURVFT 63
- Function SIMPS1 63

COMPLETE PROGRAM LISTING 64

APPENDIXES

- A - DERIVATION OF EQUATIONS 88

B - CURVE FITS USED IN PROGRAM 96
C - CHANGES TO PROGRAM FOR A GAS OTHER THAN AIR 100
D - SYMBOLS 102
REFERENCES 106

FORTRAN PROGRAM FOR CALCULATING COMPRESSIBLE LAMINAR AND TURBULENT BOUNDARY LAYERS IN ARBITRARY PRESSURE GRADIENTS

by William D. McNally

Lewis Research Center

SUMMARY

A computer program was written which gives the solution of the two-dimensional, compressible laminar and turbulent boundary-layer equations in an arbitrary pressure gradient. Cohen and Reshotko's method is used for the calculation of the laminar boundary layer, and Sasman and Cresci's method for the turbulent boundary layer. Both are "integral" methods. In the laminar regime, a single ordinary differential equation, the momentum integral equation, is solved numerically. For turbulent flow, coupled first-order ordinary differential equations, the momentum and moment-of-momentum integral equations, are solved using Runge-Kutta techniques.

Transition from laminar to turbulent boundary layer is predicted by the Schlichting-Ulrich-Granville method; or the user may specify a transition point, thus forcing transition. Separation is predicted in the laminar regime when negative skin friction occurs. Separation is predicted for turbulent flow when the level of incompressible form factor reaches a specified limit.

The program allows a variety of initial conditions: laminar boundary layer at a stagnation point, laminar boundary layer on a sharp leading edge, laminar boundary layer with initial displacement or momentum thickness given, or turbulent boundary layer with initial displacement and momentum thicknesses given.

The program input consists of surface geometry, a pressure or velocity distribution external to the boundary layer, wall temperatures, total flow conditions, and initial values of displacement and momentum thickness, if any. The output includes all the principal boundary-layer parameters, such as displacement thickness, momentum thickness, form factor, skin friction, heat transfer, and velocity profiles.

This report includes a listing of the FORTRAN IV computer program with an explanation of the equations involved, the solution methods, and the preparation of input. Numerical examples are included to illustrate the input and show what type of results are obtained. Running times are less than 1 minute on IBM-7094 equipment.

INTRODUCTION

There are many analysis and design problems in the field of aeronautics which demand an understanding of the growth and properties of boundary layers. Flow through turbomachinery is one such problem area, since boundary layers influence the performance and losses of turbomachines to a great degree. A computer program which predicts the development of boundary layers under known conditions of pressure gradient allows the inclusion of real flow effects into flow models used in the analytical prediction of flow and performance in turbomachines.

Known conditions of pressure gradient can be obtained either analytically or experimentally. Programs already exist (refs. 1 to 3) for the analytical solution of ideal flow on the blade-to-blade surface of revolution of turbomachines. These programs compute ideal blade-surface velocity distributions, which can then be used to make an analysis of boundary-layer development. Experimental distributions of velocity or pressure may also serve as input to a boundary-layer analysis.

This report discusses a FORTRAN IV program for computing two-dimensional compressible laminar and turbulent boundary layers in arbitrary pressure gradients. Transition from laminar to turbulent flow is also predicted. Two of the most applicable integral methods available have been used in the laminar and turbulent sections of the program.

Cohen and Reshotko's method (ref. 4) is used for the solution of the laminar boundary layer. It is an approximate method of integral type for compressible laminar cases with heat transfer and arbitrary surface gradients. Sasman and Cresci's method (ref. 5) is used in the turbulent-boundary-layer section of the program. It is also an integral method, and involves coupled momentum and moment-of-momentum differential equations. It is an extension of Reshotko and Tucker's method (ref. 6) for compressible boundary layer with heat transfer and arbitrary gradients. The Schlichting-Ulrich-Granville method (refs. 7 to 9) is used in the program for predicting transition.

The program allows the arbitrary selection of initial values in both laminar and turbulent regions, as well as the selection of the point of transition if the user wishes. It also allows reattachment after laminar separation. The program will handle any two-dimensional case with subsonic or supersonic Mach numbers. It is not restricted to the compressor and turbine problems which motivated its existence.

This report includes the FORTRAN IV computer program called BLAYER, with an explanation of the equations involved and the method of solution. A detailed explanation is given of the input required and the output received. Numerical examples are included to illustrate the input and show what type of results are obtained.

The report is divided into two main parts: INFORMATION FOR GENERAL USER and INFORMATION FOR PROGRAMMER. Those wishing to use the program in its present form need only read the first part, INFORMATION FOR GENERAL USER. This contains descriptions of the general method, the preparation of input, and the output received. Information of interest to someone who may wish to change the program is contained in the second part, INFORMATION FOR PROGRAMMER. This part contains a complete program listing, as well as descriptions of the individual subroutines and variables.

A BLAYER source deck on tape is available from COSMIC (Computer Software Management and Information Center), Computer Center, University of Georgia, Athens, Georgia 30601. The COSMIC program number is LEW-11097.

INFORMATION FOR GENERAL USER

DESCRIPTION OF METHOD

The following sections outline the basic characteristics and limitations of this program and give short descriptions of input and output characteristics. This information should enable the reader to decide if the program is applicable to his needs. More detailed information about program procedure and operation is contained in the sections describing the individual subroutines.

Basic Characteristics of Program

The following statements give a general description of the characteristics of the overall program:

(1) The program is for two-dimensional, compressible or incompressible, laminar and/or turbulent boundary layers.

(2) It uses integral methods rather than differential methods. Ordinary differential equations, the momentum and moment-of-momentum integral boundary layer equations, are solved numerically.

(3) The program is applicable to all types of pressure gradients - favorable, zero, or unfavorable.

(4) Both subsonic and supersonic flows may be handled.

(5) The Cohen-Reshotko method (ref. 4) is used to solve the laminar boundary layer. It involves the momentum integral equation for compressible laminar cases with arbitrary pressure gradient and heat transfer. Cohen and Reshotko's method was chosen because it does not have the restrictions on compressibility, pressure gradient, heat transfer, Prandtl number, or type of free-stream velocity distribution which many of the other

laminar methods have. It is one of the most accurate, programmable, general methods available for the laminar case.

(6) The Sasman-Cresi method (ref. 5) is used for the solution of the turbulent boundary layer. It involves momentum and moment-of-momentum integral boundary-layer equations for compressible turbulent cases with arbitrary gradients and heat transfer. It extends Reshotko and Tucker's analysis (ref. 6) by using more recent empirical data to avoid some of the problems with strong adverse pressure gradients.

(7) The Schlichting-Ulrich-Granville method (refs. 7 to 9) is used to predict transition from laminar to turbulent boundary layer. The stability analysis of Schlichting and Ulrich (refs. 7 and 8) on Pohlhausen velocity profiles leads to a method for predicting where the laminar boundary layer becomes unstable. The experimental curve of Granville (ref. 9) is then used to predict the distance between the point of instability and the point of transition. Both analyses are for incompressible flow, but can be applied to compressible flow through the use of suitable transformations.

(8) Transition may be accomplished in one of three ways: (a) naturally, using the Schlichting-Granville criteria, (b) at a point specified by the user, or (c) through laminar separation and reattachment.

(9) Separation is predicted for laminar flow when negative skin friction occurs. Separation is predicted in the turbulent regime when the level of incompressible form factor passes a specified limit.

(10) Typical program runs on IBM-7094 equipment take less than 1/2 minute.

(11) The program is written in FORTRAN IV, and is available from COSMIC (see INTRODUCTION).

Input and Output Characteristics

The following statements briefly describe some of the input and output features of the program:

(1) Input to the program consists principally of four arrays: X- and Y-coordinates of surface points, a surface flow distribution at the X-Y points, and wall temperature at the X-Y points. Besides these four basic arrays, gas constants, stagnation conditions, upstream Mach number, initial values (if any), and some integer constants are required. Surface flow distribution can be given as any one of five quantities: static pressure, relative free-stream velocity, relative free-stream Mach number, ratio of static pressure to total pressure, or ratio of relative velocity to critical velocity.

(2) Initial values may be given for displacement and momentum thicknesses in either the laminar or turbulent regions of flow, or in both. The laminar region can be completely suppressed if initial values are given for the turbulent layer at the initial calculation station.

(3) Surface distributions of velocity may be smoothed by the program prior to the computation of surface gradients.

(4) English units (pounds force, slugs, feet, degrees Rankine, and foot-pounds) or metric units (Newtons, kilograms, meters, degrees Kelvin, and Joules) may be used for input and output.

(5) Output from the program includes all the principal boundary-layer parameters: displacement thickness, momentum thickness, form factor, skin friction, heat transfer, etc. Velocity profiles are calculated by using Pohlhausen's expression in the laminar regime, and the power law in the turbulent regime.

Limitations of Program

The following are the principal limitations of the program:

(1) Surface curvature, surface roughness, initial turbulence level of the flow, and shock - boundary-layer interactions are not taken into account by the program.

(2) The program cannot be used along surfaces where relative total pressure is changing from point to point, such as a turbomachine rotor with change in radius along streamlines. However, the program could be easily extended to handle this case.

(3) The program presented herein is valid only for air. However, it can be easily altered for use with other gases. These alterations are described in appendix C. It could also be altered for use with liquids; but alterations would be so numerous that it is suggested a separate program be written for this case.

(4) In strong favorable gradients, the laminar solution procedure has to be extrapolated, so that results are questionable under these circumstances (see ref. 4). Very strong adverse gradients, such as those coming off the velocity peak at the leading edge of a turbomachine blade, can also give erroneous results in both the laminar and turbulent regimes. Such a condition is indicated by separation after only three or four data points.

(5) The program does not continue computing past a point of separation, unless the user requests reattachment after the laminar separation.

BASIC EQUATIONS

The following sections discuss the differential equations solved in the laminar and turbulent sections of the program. The methods used in predicting transition and separation are also outlined.

Laminar Solution

After the program reads input data and does preliminary calculations, the laminar boundary-layer equation is solved. This momentum integral equation is derived as follows in the Cohen-Reshotko reference (ref. 4): Prandtl's boundary-layer equations are transformed for compressible flow by Stewartson's transformations (ref. 10). The resulting first-order differential equations are then expressed in terms of dimensionless parameters related to wall shear, surface heat transfer, and free-stream velocity gradient. This gives two equations with three unknowns. Thwaite's concept (ref. 11) that these three quantities are related in a unique way without specifying a type of velocity profile is then assumed. The relations are obtained by examining exact solutions for the incompressible laminar boundary layer. A unique correlation relating the variables is chosen, reducing the problem to the solution of one first-order, ordinary, nonhomogeneous differential equation in terms of a free-stream velocity gradient parameter. This is equation (28) of reference 4,

$$-U_e \frac{d\left(\dfrac{n}{\dfrac{dU_e}{dX_{tr}}}\right)}{dX_{tr}} = N(n, S_w)$$

where

U_e transformed free-stream velocity, $U_e = \frac{a_0'}{a_e} u_e = a_0' M_e$

X_{tr} transformed x-coordinate, $X_{tr} = \int_0^x k_{su} \frac{a_e}{a_0'} \frac{P_e}{P_0'} dx$

N momentum parameter, a function of n and S_w

and

$$\frac{dU_e}{dX_{tr}} = a_0' \frac{dM_e}{dx} \frac{1}{\dfrac{dX_{tr}}{dx}} = a_0' \frac{dM_e}{dx} \frac{a_0' P_0'}{k_{su} a_e P_e}$$

For isothermal, or nearly isothermal, surfaces, the solution of equation (1) is simplified since N can be expressed as a linear function of the pressure gradient parameter (correlation number) n as follows:

$$N = A + Bn \tag{2}$$

The solution of equation (1) is then equation (32) of reference 4,

$$n = -AU_e^{-B} \frac{dU_e}{dX_{tr}} \int_0^{X_{tr}} U_e^{B-1}\, dX_{tr} \tag{3}$$

When transformed back to physical quantities by using Stewartson's transformation, this becomes

$$n = -AM_e^{-B} \frac{dM_e}{d\left(\frac{x}{L}\right)} \left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{(3\gamma-1)/(2\gamma-2)} \int_0^{x/L} \frac{M_e^{B-1}}{\left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{(3\gamma-1)/(2\gamma-2)}}\, d\left(\frac{x}{L}\right) \tag{4}$$

When appropriate initial values of n are used, equation (4) is solved numerically along the boundary-layer surface, giving the distribution of pressure gradient parameter (or correlation number) n.

Once n is obtained at each station, the other boundary-layer and heat-transfer parameters are easily obtained. The shear-stress parameter l and the heat-transfer parameter, C_fR_w/Nu_x, are obtained through correlation with n. The momentum thickness is then obtained from n, the shear stress and skin friction from l, and the heat transfer from C_fR_w/Nu_x. Form factor, finally, is obtained from an experimental correlation with n, S_w, and M_e described in reference 12; δ^* follows from form factor and momentum thickness.

Laminar velocity profiles are computed by the familiar Pohlhausen technique described in reference 7.

Turbulent Solution

The equations derived in the Sasman-Cresci reference (ref. 5) are the momentum and moment-of-momentum integral boundary-layer equations. The momentum integral equation is obtained by applying a Mager-type transformation (ref. 13) to Prandtl's equations in which flow variables appear as time-averaged quantities. The momentum equation is then integrated across the boundary layer to give the momentum integral equation. The moment-of-momentum integral equation is obtained in somewhat the same way, after the momentum equation is multiplied by the transformed y-coordinate. The general procedure of reference 14 is used to reduce the resulting equation to manageable form. The two derived integral equations are the following:

$$\frac{d\theta_{tr}}{dX_{tr}} + \frac{\theta_{tr}}{U_e}\frac{dU_e}{dX_{tr}}\left[2 + H_i + \frac{\int_0^{\delta_{tr}}\left(\frac{h'}{h'_0} - 1\right) dY_{tr}}{\theta_{tr}}\right] = \left(\frac{T'_0}{T_e}\right)\left(\frac{\overline{T}}{T'_0}\right)\frac{\tau_w}{\rho_e U_e^2} \tag{5}$$

and

$$\begin{aligned}
\frac{dH_i}{dX_{tr}} = \frac{-1}{U_e}\frac{dU_e}{dX_{tr}}\left[\frac{H_i(H_i + 1)^2(H_i - 1)}{2}\right]\Bigg[1 + \frac{2}{(H_i + 1)\theta_{tr}}\int_0^{\delta_{tr}}\left(\frac{h'}{h'_0} - 1\right) dY_{tr} \\
- \frac{2(H_i - 1)}{H_i^2(H_i + 1)\theta_{tr}^2}\int_0^{\delta_{tr}}\left(\frac{h'}{h'_0} - 1\right) Y_{tr}\, dY_{tr}\Bigg] + \frac{H_i(H_i^2 - 1)}{\theta_{tr}}\left(\frac{T'_0}{T_e}\right)\left(\frac{\overline{T}}{T'_0}\right)\frac{\tau_w}{\rho_e U_e^2} \\
- \frac{(H_i^2 - 1)(H_i + 1)}{\theta_{tr}}\left(\frac{T'_0}{T_e}\right)\left(\frac{\overline{T}}{T'_0}\right)\frac{\tau_w}{\rho_e U_e^2}\int_0^1 \frac{\tau}{\tau_w}\, d\left(\frac{Y_{tr}}{\delta_{tr}}\right)
\end{aligned} \tag{6}$$

where

U_e transformed free-stream velocity, $U_e = \frac{a'_0}{a_e} u_e = a'_0 M_e$

X_{tr} transformed x-coordinate, $X_{tr} = \int_0^x \frac{T'_0}{\overline{T}} \left(\frac{T_e}{T'_0}\right)^{(\gamma+1)/(2\gamma-2)} dx$

Y_{tr} transformed y-coordinate, $Y_{tr} = \left(\frac{T_e}{T'_0}\right)^{1/2} \int_0^y \frac{\rho}{\rho_0} dy$

θ_{tr} transformed momentum thickness, $\theta_{tr} = \int_0^{\delta_{tr}} \frac{U}{U_e}\left(1 - \frac{U}{U_e}\right) dY_{tr}$

H_i transformed form factor for adiabatic flow, $H_i = \frac{\int_0^{\delta_{tr}} \left(1 - \frac{U}{U_e}\right) dY_{tr}}{\theta_{tr}}$

Equations (5) and (6) are not in solvable form, however. Reference 5 makes use of Crocco's relation and a power-law assumption for velocity profiles to evaluate the enthalpy integrals in both equations (5) and (6). The Ludwieg-Tillmann skin friction relation (ref. 15), transformed for compressible flow, is used for the shear-stress terms in both equations. And, finally, the normalized shear distribution integral of equation (6) is evaluated by using the results of equilibrium turbulent boundary-layer analysis. With these substitutions and the relation

$$f = \left(\frac{U_e \theta_{tr}}{\nu'_0}\right)^{1.268} = \left(\frac{M_e a'_0 \theta_{tr}}{\nu'_0}\right)^{1.268} \tag{7}$$

equations (5) and (6) can be put into the following form:

$$\frac{df}{dx} = 1.268 \left\{ \frac{-f}{M_e} \frac{dM_e}{dx} \left[1 + (1 + S_w) H_i\right] + A \right\} \tag{8}$$

$$\frac{dH_i}{dx} = \frac{-1}{2M_e}\frac{dM_e}{dx}\left[H_i(H_i+1)^2(H_i-1)\right]\left[1 + S_w \frac{H_i^2 + 4H_i - 1}{(H_i+1)(H_i+3)}\right]$$

$$+ \frac{H_i^2 - 1}{f} A\left[H_i - \frac{0.011(H_i+1)(H_i-1)^2}{H_i^2}\frac{2}{C_f}\frac{T_0'}{\overline{T}}\right] \tag{9}$$

where

$$A = 0.123\, e^{-1.561\, H_i}\left(\frac{M_e a_0'}{\nu_0'}\right)\frac{T_e}{\overline{T}}\left(\frac{T_e}{T_0'}\right)^{(\gamma+1)/(2\gamma-2)}\left(\frac{\overline{\mu}}{\mu_0'}\right)^{0.268}$$

and

$$\frac{C_f}{2} = \frac{\tau_w}{\rho_e u_e^2} = 0.123\, e^{-1.561 H_i} f^{-0.2113}\left(\frac{T_e}{\overline{T}}\right)\left(\frac{\overline{\mu}}{\mu_0'}\right)^{0.268}$$

Equations (8) and (9) are the coupled, first-order, ordinary differential equations which govern the development of the turbulent boundary layer. These equations are not uncoupled in the Sasman-Cresci method as they were in Reshotko and Tucker's analysis.

Using initial values calculated in the laminar routine or given by the user, equations (8) and (9) are solved by a Runge-Kutta scheme, giving the distribution of f and H_i along the surface. The usual boundary-layer parameters are then obtained from these two distributions. Turbulent velocity profiles are calculated by means of the power law (see p. 47).

Transition

The Schlichting-Ulrich-Granville method (refs. 8 and 9) is used for the theoretical prediction of transition from laminar to turbulent flow. Details of the method are summarized in reference 7.

Schlichting and Ulrich used sixth-degree Pohlhausen velocity profiles to calculate curves of neutral stability (see ref. 7) for laminar boundary layers in various pressure gradients. From these curves, a single curve of critical momentum-thickness Reynolds number against shape factor K was obtained. This curve is used by the program for predicting the point of instability of the laminar boundary layer. The distance between the point of instability and the point of transition is predicted by means of an experimental curve by Granville. This curve represents the difference in momentum-thickness Reynolds numbers at the instability and transition points plotted against a mean Pohlhausen parameter $\overline{K}$. Once an instability point is located, $\overline{K}$ can be calculated and the location of transition determined.

Separation

Laminar case. - In the laminar boundary layer, separation is assumed to occur at the station where skin friction coefficient C_f or wall shear stress τ_W passes from positive to negative, indicating backflow. The user should check the values of C_f at the separation station and the previous station in order to more exactly determine the point of separation.

Turbulent case. - In the turbulent boundary layer, separation is predicted based on the magnitude of H_i. (The turbulent equations do not allow a negative C_f.) Separation is predicted by the program at the station where H_i achieves a value greater than 2.8. This is a relatively high value for H_i, and H_i grows rapidly near separation. The values of H_i are printed at each output station and at each point where the turbulent differential equations are solved. If the user feels that a lower value of H_i is more appropriate for separation (2.0 to 2.5 is typical), a different separation point can be chosen from these printed values.

Another way of predicting turbulent separation is to plot the output distribution of C_f as a function of X. The separation point is then predicted by extrapolating C_f to zero. The extrapolation should be started where C_f is falling rapidly and continued to zero without a point of inflection.

NUMERICAL EXAMPLES

Two numerical examples are given to illustrate the use of the program. The first is an NACA airfoil which has both laminar and turbulent boundary layers, and the second is a converging-diverging channel with turbulent boundary layer throughout. The airfoil illustrates the prediction of transition by the program, and the channel illustrates the use of initial values of displacement and momentum thickness at the beginning of the boundary layer. In both cases the computer input is included.

TABLE I. - INPUT FOR NACA 0012 AIRFOIL

```
BECKER  (WRL-682)  NACA 0012  C=5.0  CL=0.0

GAM       R          PTZ        TTZ       UFMACH
1.400     1716.48    2645.00    600.00    0.2840

NST       NVP        NTURB      KPVM      KEM       KSMTH     KSPLN     KLE       KATCH     CTHET
 33        20          0          2        0          0         0         0         0        0.

DLAM      TLAM       DTURB      TTURB
0.        0.         0.         0.

KPRE      KGRAD      KSDE       KLAM      KMAIN     KPROF
 1         1          1          1         1         1

   X          Y           UF          TWAL
0.         0.          0.          600.0000
0.02500    0.05765     270.64000   600.0000
0.06250    0.09470     339.99200   600.0000
0.12500    0.13075     376.86600   600.0000
0.25000    0.17775     397.16400   600.0000
0.37500    0.21000     400.54700   600.0000
0.50000    0.23415     401.90000   600.0000
0.75000    0.26725     401.90000   600.0000
1.00000    0.28685     400.20900   600.0000
1.25000    0.29705     397.16400   600.0000
1.30000    0.29810     396.38600   600.0000
1.35000    0.29890     395.57400   600.0000
1.40000    0.29960     394.79600   600.0000
1.45000    0.29995     393.95000   600.0000
1.50000    0.30010     393.10500   600.0000
1.55000    0.29995     392.22500   600.0000
1.60000    0.29960     391.31200   600.0000
1.65000    0.29900     390.39800   600.0000
1.70000    0.29825     389.48500   600.0000
1.75000    0.29735     388.53800   600.0000
2.00000    0.29015     383.97100   600.0000
2.25000    0.27905     379.43700   600.0000
2.50000    0.26470     374.83600   600.0000
2.75000    0.24760     370.10000   600.0000
3.00000    0.22815     365.36400   600.0000
3.25000    0.20685     360.89800   600.0000
3.50000    0.18320     356.23000   600.0000
3.75000    0.15805     351.15500   600.0000
4.00000    0.13115     345.74300   600.0000
4.25000    0.10275     338.63800   600.0000
4.50000    0.07240     330.85700   600.0000
4.75000    0.04035     322.06200   600.0000
5.00000    0.00630     311.91300   600.0000
```

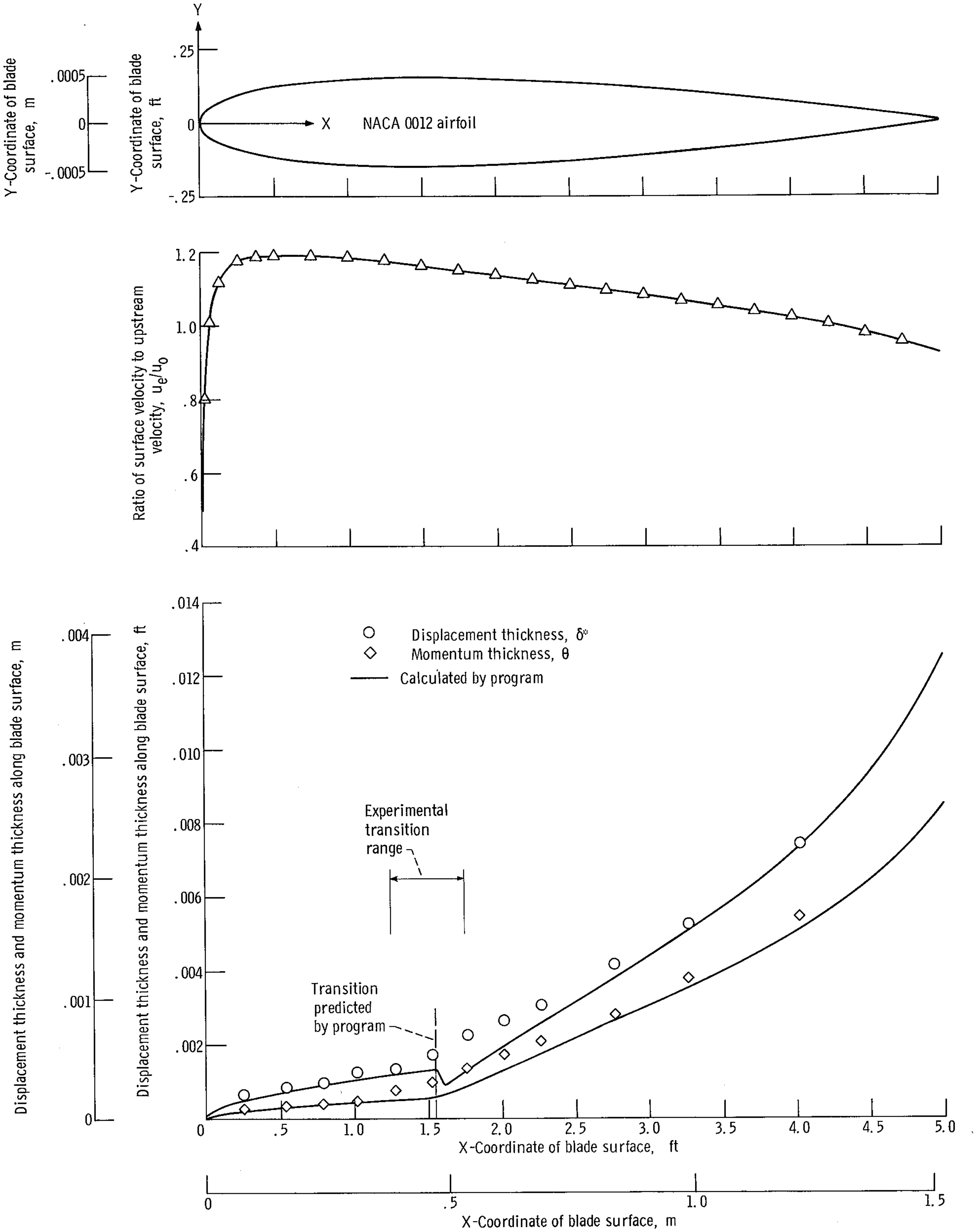

Figure 1. - Comparison of analytical results with experimental data on NACA 0012 airfoil.

NACA 0012 Airfoil

The first example is an NACA 0012 airfoil at zero angle of attack (ref. 16). The free-stream inlet Mach number is 0.284. The configuration listed in reference 16 had a chord length of 5.0 feet. The input for this example is given in table I. The blade geometry, surface velocity distribution, and comparison of output with experimental results is given in figure 1.

The boundary layer was assumed to begin at a stagnation point at the leading edge of the blade, and no initial values were used. Transition was predicted naturally by the program, and occurred within the range in which it was measured experimentally. A stronger concentration of input points was given in the transition range so that the point of transition could be located more precisely (see table I). Output values of δ^* and θ are compared with experimental values in figure 1. The agreement is good except for a region around the transition point. This is understandable since transition occurs gradually in the actual flow, but is forced to occur at a single point in the program.

Execution time for this program was 0.4 minute.

Converging-Diverging Channel

The second example is a two-dimensional channel in which the flow is turbulent. The upper wall of the channel converges and diverges and the bottom wall is flat (see fig. 2). Boundary layers were computed on both walls. Experimental data on this configuration were taken at Lewis but have not previously been reported. The input for the curved wall of this example is given in table II. The wall pressure distributions, and comparison of output for both walls with experimental data, are given in figure 2.

On both surfaces, initial values were given for δ^* and θ, and the turbulent boundary layer was started at the initial station. Agreement between measured and predicted values on the flat wall is extremely good. On the curved wall, agreement is also good. There is some doubt about the experimental values of δ^* and θ on the converging-diverging portion of the curved wall. Because of the curvature of streamlines in the free-stream flow at these points, it was difficult to locate the edge of the boundary layer on the measured velocity profiles. The values of δ used in the integrations to calculate δ^* and θ may have been conservative.

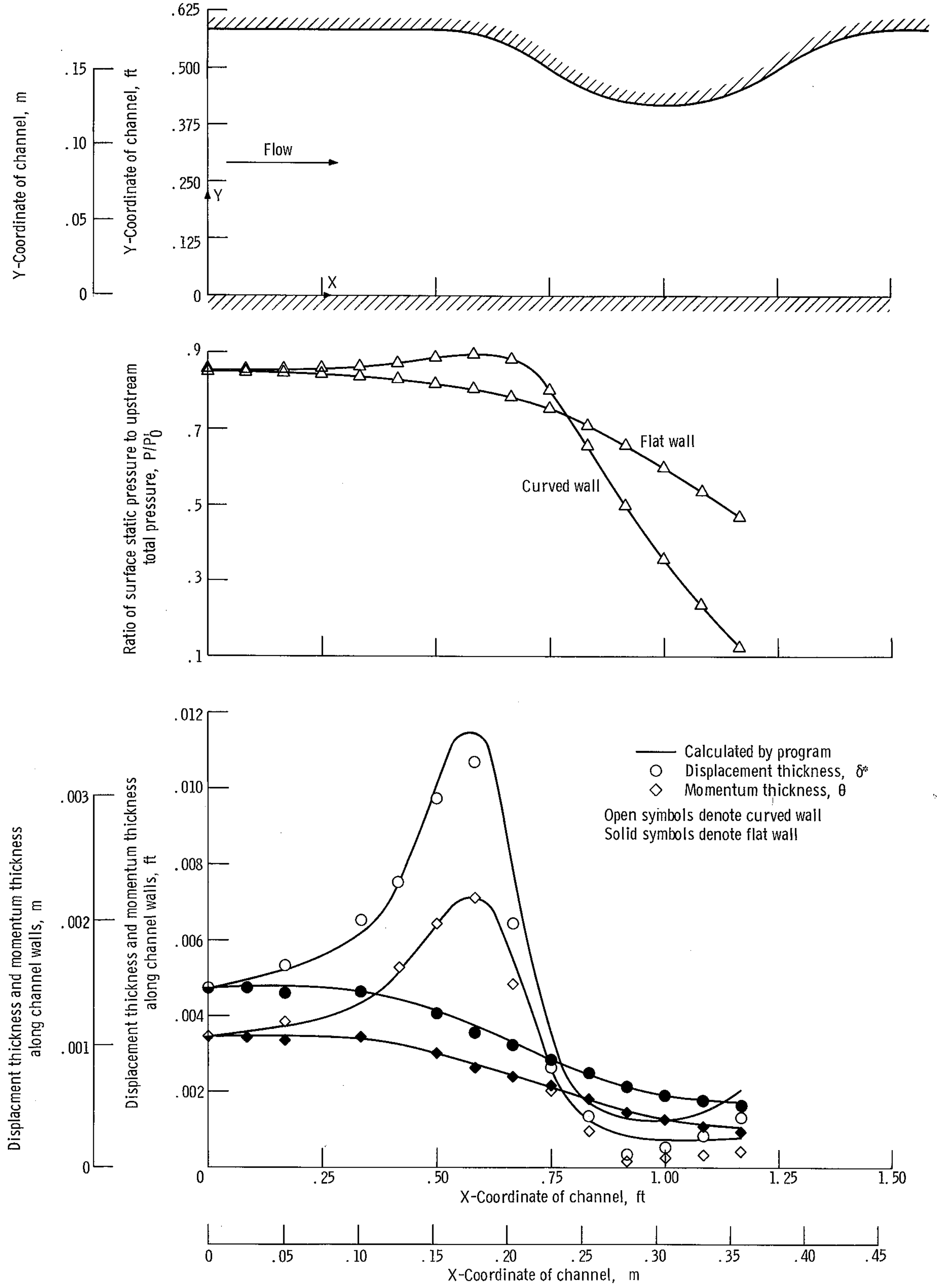

Figure 2. - Comparison of analytical results with experimental data on converging-diverging channel.

TABLE II. - INPUT FOR CURVED-WALL CHANNEL

```
CURVED WALL - DRY AIR

GAM        R           PTZ        TTZ       JPMACH
1.400    1716.48     2004.77     532.00     0.4821

NST       NVP        NTURB       KPVM       KEM       KSMTH      KSPLN      KLE       KATCH      DTHET
 15        10           1          4          0          0          0          0          0         0.

DLAM      TLAM        DTURB       TTURB
0.        0.          0.004710    0.003420

KPRE      KGRAD       KSDE        KLAM      KMAIN      KPROF
 1          1           1           1         1          1

   X          Y          POPTZ        TWAL
0.          0.58300     0.853000    532.0000
0.08330     0.58300     0.854300    532.0000
0.16700     0.58300     0.855100    532.0000
0.25000     0.58300     0.858300    532.0000
0.33300     0.58300     0.862300    532.0000
0.41670     0.58300     0.871400    532.0000
0.50000     0.58300     0.889600    532.0000
0.58330     0.57300     0.896800    532.0000
0.66700     0.54300     0.882000    532.0000
0.75000     0.49700     0.800000    532.0000
0.83300     0.45100     0.653000    532.0000
0.91670     0.42500     0.498000    532.0000
1.00000     0.41670     0.355000    532.0000
1.08330     0.42500     0.235000    532.0000
1.16700     0.45100     0.124000    532.0000
```

INPUT

Figure 3 shows the input variables as they are punched on the data cards. The first input data card is for a title. In any of the columns from 1 to 80 of this card, the user may put whatever information he wishes in order to identify the data deck. The remaining cards are for input variables. These are defined in the "dictionary" below. Further explanation of the proper preparation of input is given in the section Special Instructions for Preparing Input (p. 20).

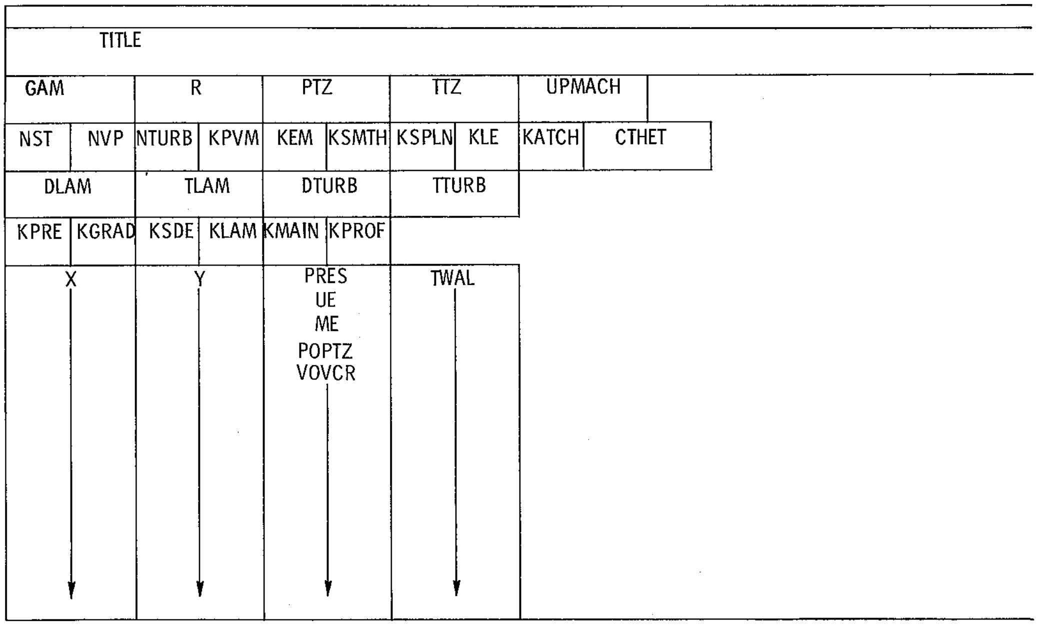

Figure 3. - Input form.

Dictionary of Input Variables

After a title card, the following input variables are given:

GAM	specific-heat ratio, γ
R	universal gas constant, $\mathscr{R}$, (ft)(lbf)/(slug)($^{\circ}$R); J/(kg)(K)
PTZ	inlet or upstream relative total pressure (station 0), P_0', lbf/ft^2; N/m^2
TTZ	inlet or upstream relative total temperature (station 0), T_0', $^{\circ}$R; K
UPMACH	inlet or upstream Mach number relative to surface, M_0

NST	integer number of input stations ($\leq$100) along boundary-layer surface
NVP	integer number of points desired in velocity profile at each station
NTURB	integer number of station, if any, at which user wishes turbulent boundary layer to begin (NTURB is usually zero, allowing program to calculate position of transition to turbulent boundary layer. NTURB may also be given any value from 1 to NST. If NTURB = 1, initial values <u>must</u> be given for DTURB and TTURB. If NTURB > 1, initial values may or may not be given.)
KPVM	integer from 1 to 5 indicating which form of surface flow distribution is given as input: Pressure . . . . 1 Free-stream velocity . . . . 2 Free-stream Mach number . . . . 3 Ratio of pressure to total pressure . . . . 4 Ratio of free-stream velocity to free-stream critical velocity 5
KEM	integer (0 to 1) indicating which of the two allowable sets of units are used in input: English (pounds force, slugs, feet, seconds, degrees Rankine, and foot-pounds) . . . . 0 Metric (Newtons, kilograms, meters, seconds, degrees Kelvin, and Joules) 1
KSMTH	integer (0, 1, 2, . . .) indicating number of times distribution of free-stream velocity is to be smoothed prior to computation of surface gradients
KSPLN	integer (0 or 1) indicating manner in which surface gradients are to be calculated: Weighted-difference technique . . . . 0 Spline curve-fit technique 1
KLE	integer (0 to 1) indicating type of initial condition existing at station 1: Stagnation point or initial values given . . . . 0 Sharp leading edge 1
KATCH	integer (0 or 1) indicating whether laminar-boundary-layer separation (if encountered) should reattach as a turbulent boundary layer: Separation and stop . . . . 0 Reattach 1

CTHET	real variable used when KATCH = 1, indicating ratio of momentum thickness after reattachment to momentum thickness at laminar separation
DLAM	initial displacement thickness, if any, of laminar boundary layer at station 1, ft; m (DLAM may be zero or have some finite value.)
TLAM	initial momentum thickness, if any, of laminar boundary layer at station 1, ft; m (TLAM may be zero or have some finite value.)
DTURB	initial displacement thickness, if any, of turbulent boundary layer, ft; m (DTURB may be given for station designated by NTURB, or for station at which transition is calculated by program.)
TTURB	initial momentum thickness, if any, of turbulent boundary layer, ft; m (see DTURB)
KPRE	integer (0 or 1) indicating whether printing of output from PRECAL is desired (see OUTPUT): Output suppressed. . . . . 0 Output printed 1
KGRAD	integer (0 or 1, see KPRE) indicating whether printing of surface gradients of velocity and Mach number is desired (see OUTPUT)
KSDE	integer (0 or 1, see KPRE) indicating whether printing of solutions of laminar and turbulent differential equations is desired (see OUTPUT)
KLAM	integer (0 or 1, see KPRE) indicating whether printing of laminar calculations for location of instability and transition is desired (see OUTPUT)
KMAIN	integer (0 or 1, see KPRE) indicating whether printing of principal calculated boundary-layer parameters is desired (see OUTPUT)
KPROF	integer (0 or 1, see KPRE) indicating whether printing of velocity profiles is desired (see OUTPUT)
X	array of X-coordinates of input stations, ft; m (see figs. 4 and 5)
Y	array of Y-coordinates of input stations, ft; m (see figs. 4 and 5)
PRES	array of static pressure P at X-Y input stations, lbf/ft^2; N/m^2
UE	array of free-stream velocities u_e relative to surface at X-Y input stations, ft/sec; m/sec

ME	array of free-stream Mach numbers M_e relative to surface at X-Y input stations
POPTZ	array of ratios of static pressure to relative total pressure P/P_0' at X-Y input stations
VOVCR	array of ratios of relative free-stream velocities to relative critical velocities u_e/u_{cr} at X-Y input stations (u_{cr} is the speed of sound at Mach 1, and is only a function of relative total temperature.)

$$u_{cr} = \sqrt{\frac{2\gamma R}{\gamma + 1} T_0'} \tag{10}$$

TWAL	array of static wall temperatures at X-Y input stations, $^{\circ}R$; K (If TWAL is unknown and surface is nearly isothermal, the value of TTZ may be used for TWAL.)

Special Instructions for Preparing Input

Surface geometry. - The X and Y input arrays give coordinates of points which describe the surface geometry (see figs. 4 and 5). These are the points at which pressure (velocity, etc.) and wall temperature are given, and at which boundary-layer parameters

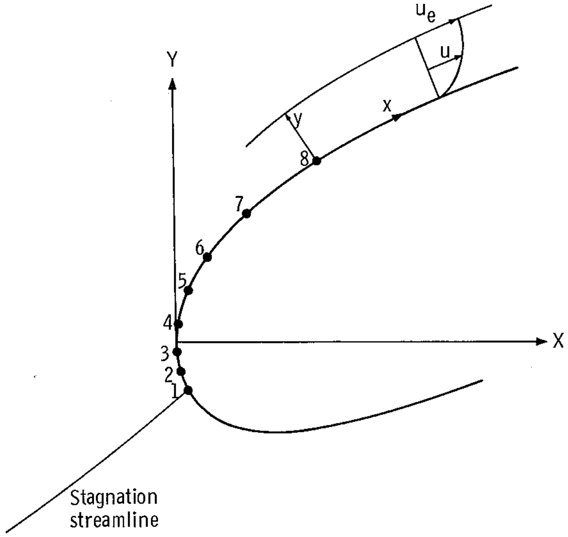

Figure 4. - Coordinate systems for geometry input and boundary-layer output on a blade surface.

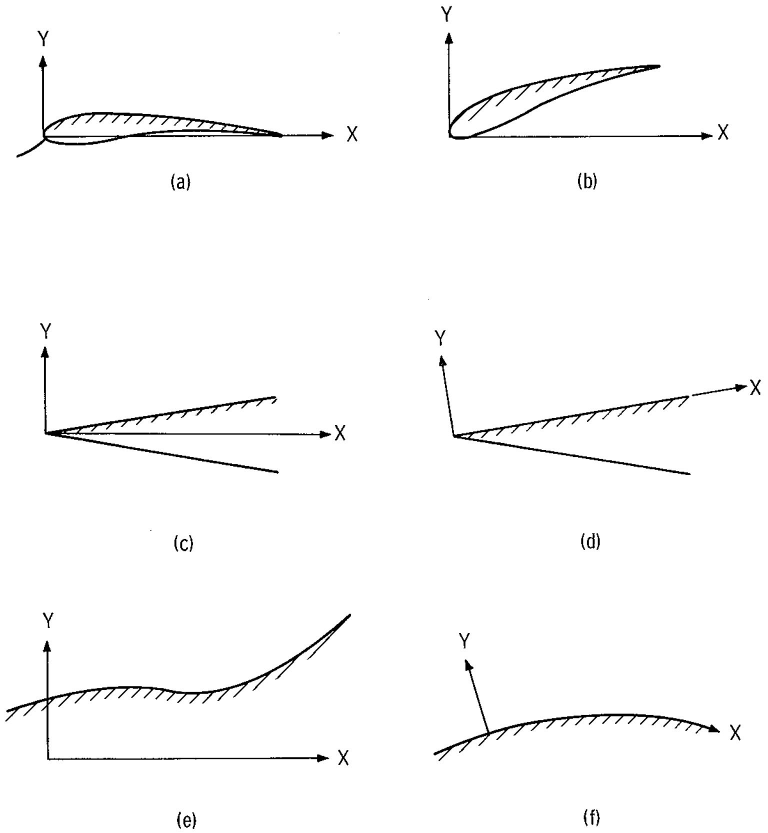

Figure 5. - Variation in placement of X-Y coordinate system.

will be given as output. Enough points (20 or 30 is typical) should be given so that surface length (used in the calculation of surface velocity gradients) can be calculated with reasonable accuracy from the formula

$$x_i = x_{i-1} + \sqrt{(X_i - X_{i-1})^2 + (Y_i - Y_{i-1})^2} \tag{11}$$

More points should be concentrated in areas where surface curvature is high; not only so that surface length is accurate, but because it is in these regions where velocity gradients will change most rapidly.

Figure 5 shows some ways in which the X-Y axes can be related to the boundary-layer surface. The origin and orientation of these axes are completely arbitrary, since the X-Y coordinates are only used to identify the surface points and compute surface length by formula (11). These axes can also be curvilinear (with $Y = 0$ everywhere), agreeing with the boundary-layer axes x and y (see fig. 5(f)).

Leading-edge conditions. - There are two principal leading-edge conditions: a stagnation point at station 1 (KLE = 0), or a sharp or pointed leading edge at station 1 (KLE = 1).

The program will also accept data which begin at some point within the boundary layer other than the starting point of the boundary layer. In these cases, KLE = 0 is used, and initial values where they are known are given for DLAM, or TLAM, or DTURB and TTURB.

Inlet conditions. - PTZ, TTZ, and UPMACH (P_0', T_0', M_0) should be given at a reference point outside the boundary layer or in the free stream opposite station 1 (see fig. 6) where the flow is not affected by the boundary layer. Free-stream Mach number M_0 is not a critical input; it is used to obtain static temperature, from which Prandtl number and thermal conductivity are obtained by means of curve fits.

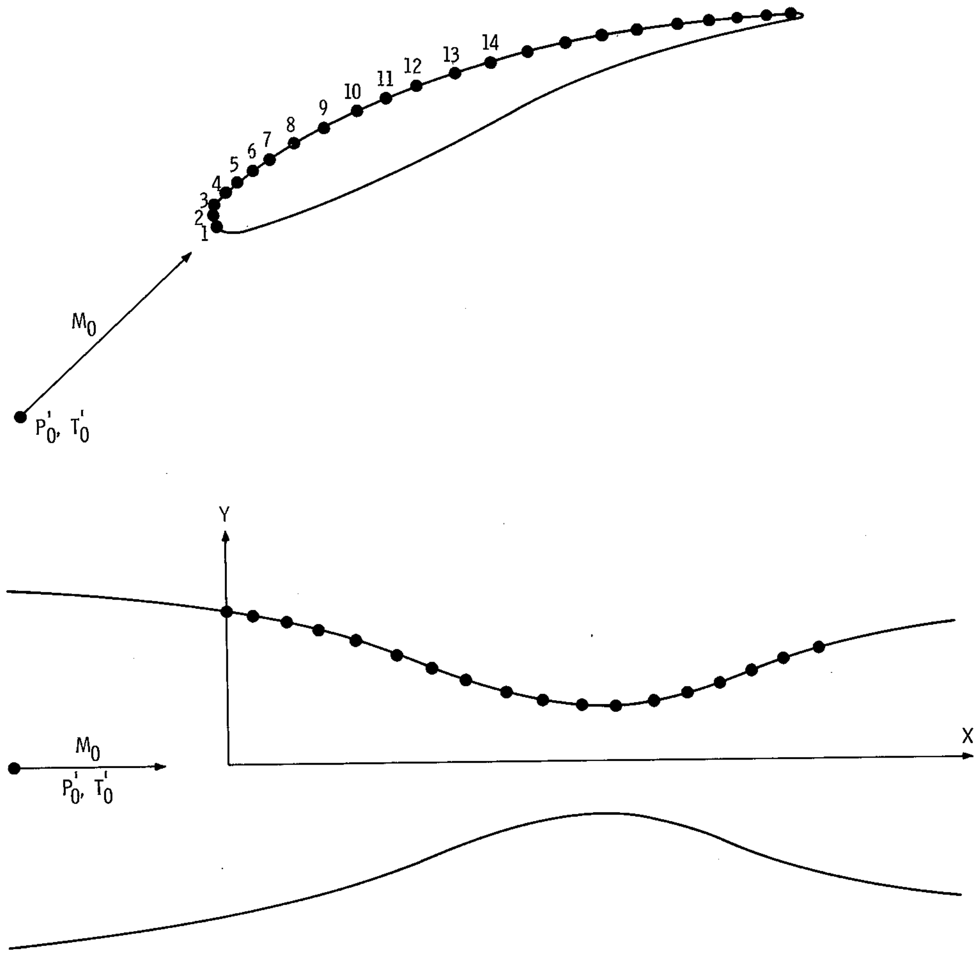

Figure 6. - Location of typical upstream input conditions.

Surface distributions of pressure, velocity, and Mach number. - Any one of five parameters may be used as input to describe flow along the boundary-layer surface: pressure, free-stream velocity, free-stream Mach number, ratio of pressure to total pressure, or ratio of free-stream velocity to free-stream critical velocity. Enough input points (<100) should be used (especially in areas where surface curvature is high or distribution of pressure changes rapidly) so that gradients of velocity will be computed accurately. Also, since answers are only printed at points where input was given, enough points should be used to satisfy the requirements for output.

Smoothing of surface distributions. - The velocity and Mach number gradients used in the solution of the boundary-layer equations are computed from the surface distributions of pressure (or velocity, etc.). These gradients are a function of the smoothness of the input distributions. Distributions of pressure calculated from experimental readings or read from a plot of pressure tend to be less smooth than those computed analytically. The user must therefore use his judgement and, by setting KSMTH = 1, 2, . . ., etc., smooth his input data whenever necessary. (An examination of the gradients computed by the program will indicate whether smoothing is required on a subsequent run.) Smoothing affects peaks and valleys most strongly, the effect diminishing as more points are used to describe the peak or valley.

Calculation of surface gradients. - Surface gradients of velocity and Mach number can be computed by two different methods in the program. The first (KSPLN = 0) is a difference method (see subroutine GRADNT) using weighted slopes between each point and adjacent points to compute slope at each reference point. The second method (KSPLN = 1) fits a spline curve to the data in order to compute slopes. The latter method gives more accurate results when the data are very smooth. Caution should be exercised, however, when using the spline since it forces the curve fit to pass exactly through each data point. The computed velocity and Mach number gradients should be checked in the program output when the spline fit is used. If these gradients are oscillating in magnitude or sign, the input data should be smoothed, or the weighted slope method (KSPLN = 0) should be used.

Arbitrary selection of transition point. - Ordinarily the point of transition is calculated by the Schlichting-Granville method. However, the user can control this point by means of the variable NTURB. If NTURB is given a number greater than zero, transition will be forced to occur at the station specified by that number, even if the Schlichting-Granville method would have predicted it sooner, or later, than that station.

If the user gives a value to NTURB, he may or may not choose to give values to DTURB and TTURB. If he does so, they are used at the station specified by NTURB. If values are not given to DTURB and TTURB, laminar values of displacement and momentum thickness at the station specified by NTURB are used as initial values. (When NTURB = 1, however, values must be given for DTURB and TTURB.)

Use of initial values of δ^* and θ. - The user may give initial values whenever he knows them. In the laminar case, only one value is required (either DLAM or TLAM). If both are given, TLAM is used. If initial turbulent values are given, both DTURB and TTURB must be given. Initial laminar values, if given, are used at station 1. Initial turbulent values are used at the station specified by NTURB, or (when NTURB = 0) at the station where transition is predicted.

Laminar separation and reattachment. - The user may simulate laminar separation and reattachment by use of the variables KATCH and CTHET. Ordinarily the program stops calculating when it locates either laminar or turbulent separation at any station. However, if laminar separation is located and KATCH = 1, the program multiplies the final laminar momentum thickness by CTHET and uses this as an initial value for starting the turbulent boundary layer at that point.

English or metric units. - Either English or metric units may be used for input and output. However, due to certain curve fits in the program, only one consistent set of each type of units may be used. If English units are used (KEM = 0), they must be the following:

Force, lbf
Length, ft
Time, sec
Mass, slug (or (lbf)(sec^2)/ft)
Temperature, $^{\text{o}}$R

If metric units are used (KEM = 1), they are as follows:

Force, N
Length, m
Time, sec
Mass, kg (or N-sec^2/m)
Temperature, K

With these consistent sets of units, density is mass per unit volume, energy is force times distance, and the gas constant $\mathscr{R}$ has units of energy per unit mass per degree. Since either of these two sets of units can be employed, the output is not labeled with any units.

Format for input data. - All the numbers (except CTHET) on the card beginning with NST and on the card beginning with KPRE are integers (no decimal point) in five-column fields (see fig. 3). They must all be right adjusted. The input variables on all other data cards are real numbers (punch decimal point) in 10-column fields.

OUTPUT

Sample output for the NACA 0012 airfoil example is given in table III. The entire output would be lengthy, so only a few lines from each section are included in the table. Most of the output is optional and is controlled by the input card which begins with KPRE. In most cases, output labels agree with internal variable names which are defined in the DICTIONARY OF VARIABLES IN MAIN SUBROUTINES.

Each section of output in table III has been numbered to correspond to the following descriptions:

(1) Output numbered 1 is a listing of the input data, which is always printed automatically. All items are labeled as on the input form (fig. 3). In this example, velocity (UE) has been used to describe flow along the surface.

(2) Output numbered 2 corresponds to KPRE. It includes the other four variables describing flow along the surface (PRES, ME, POPTZ, and VOVCR), geometrical variables (S, XOM, YOM, and SOL), local speed of sound (AE), static, recovery, and reference temperature distributions (TSE, TAWL, TAWT, and TBAR), local Reynolds number at the wall ($RW = u_e x/\nu_w$), and local density distributions (RHSW and RHSE). These are all variables which are used in the calculation of the boundary layer. If any smoothing is done, smoothed distributions of the five variables describing pressure and velocity (PRES, UE, ME, POPTZ, and VOVCR) are also included here.

(3) Output numbered 3 corresponds to KGRAD. It contains the three gradients of velocity and Mach number along the surface ($DUDS = du_e/dx$, $DMDS = dM_e/dx$, and $DMDL = dM_e/d(x/L)$) computed by either the weighted-slope method or the spline curve-fit method.

(4) Output numbered 4 corresponds to KSDE. It contains the numerical solutions of the laminar and turbulent differential boundary-layer equations. In the laminar case, the solution is correlation number (CORLN). In the turbulent case, the solution is incompressible form factor (FORMI), and a function (F) of the momentum thickness. These solutions are printed with respect to the surface length x.

(5) Output numbered 5 corresponds to KLAM. It contains the variables used in the laminar subroutine to check for position of instability and transition. The three variables, RTHI (increasing from station to station) and RCRIT and RTRAN (decreasing from station to station), are used in this analysis. When RTHI grows larger than RCRIT, instability has occurred. When RTHI bypasses RTRAN, transition is assumed to occur.

Instability and transition stations are located no matter how far RTHI has gone past RCRIT or RTRAN, respectively. Examination of the values of RCRIT and RTRAN sometimes indicates that instability or transition occurs closer to the previous station than the indicated station. Since final answers for turbulent thickness parameters are affected by the station at which transition occurs, the user may wish to rerun the program with additional points in this area so as to locate transition more exactly.

(6) Output numbered 6 corresponds to KMAIN. It indicates the regions of laminar and turbulent boundary layers, and the stations at which instability, transition, and separation occur. It contains all the principal boundary-layer output parameters: boundary-layer thickness (DELTA), displacement thickness (DELSR), momentum thickness (THET), form factors (FORM and FORMI), skin friction parameters (CF and TAUW), and momentum-thickness Reynolds number (RTH). It also contains the heat-transfer parameters (DTDY, NUSS, HTRAN, and CRN).

(7) Output numbered 7 corresponds to KPROF. It contains the velocity profiles at each of the stations along the surface. Profiles (U/UE) are given at equally spaced increments of distance (Y/DELTA) away from the surface.

TABLE III. - SAMPLE OUTPUT

1

```
BECKER  (WPI-682)  NACA 0012  C=5.0  CL=0.0

GAM        R          PTZ        TTZ        UPMACH
1.400      1716.48    2645.00    600.00     0.2840

NST        NVP        NTURB      KPVM       KEM        KSMTH      KSPLN      KLE        KATCH      CTHET
 33         20          0          2          0          0          0          0          0         0.

DLAM       TLAM       DTURB      TTURB
0.         0.         0.         0.

KPRF       KGRAD      KSDF       KLAM       KMAIN      KPROF
 1          1          1          1          1          1

   X          Y          UF          TWAL
0.         0.          0.          600.0000
0.02500    0.05765     270.64000   600.0000
0.06250    0.09470     339.99200   600.0000
0.12500    0.13075     376.86600   600.0000
0.25000    0.17775     397.16400   600.0000
0.37500    0.21000     400.54700   600.0000
0.50000    0.23415     401.90000   600.0000
0.75000    0.26725     401.90000   600.0000
1.00000    0.28685     400.20900   600.0000
1.25000    0.29705     397.16400   600.0000
1.30000    0.29810     396.38600   600.0000
1.35000    0.29890     395.57400   600.0000
1.40000    0.29960     394.79600   600.0000
1.45000    0.29995     393.95000   600.0000
1.50000    0.30010     393.10500   600.0000
1.55000    0.29995     392.22500   600.0000
1.60000    0.29960     391.31200   600.0000
1.65000    0.29900     390.39800   600.0000
1.70000    0.29825     389.48500   600.0000
1.75000    0.29735     388.53800   600.0000
2.00000    0.29015     383.97100   600.0000
2.25000    0.27905     379.43700   600.0000
2.50000    0.26470     374.83600   600.0000
2.75000    0.24760     370.10000   600.0000
3.00000    0.22815     365.36400   600.0000
3.25000    0.20685     360.89800   600.0000
3.50000    0.18320     356.23000   600.0000
3.75000    0.15805     351.15500   600.0000
4.00000    0.13115     345.74300   600.0000
4.25000    0.10275     338.63800   600.0000
4.50000    0.07240     330.85700   600.0000
4.75000    0.04035     322.06200   600.0000
5.00000    0.00630     311.91300   600.0000

ENGLISH UNITS USED FOR INPUT AND OUTPUT.
POUNDS FORCE, SLUGS, FEET, SECONDS, DEGREES RANKINE, AND FOOT-POUNDS.
```

TABLE III. - Continued. SAMPLE OUTPUT

2

```
PRELIMINARY CALCULATIONS


  PSZ     =     2500.92926
  TSZ     =     590.4749
  UZ      =     338.30036

  ASZ     =     1151.1985
  ATZ     =     1200.7677

  RHSZ    =     0.2467523E-02
  RHTZ    =     0.2568240E-02

  MUSZ    =     0.4107248E-06
  MUTZ    =     0.4158049E-06

  NUSZ    =     0.1664523E-03
  NUTZ    =     0.1619026E-03

  CP      =     6007.6799
  PR      =     0.69750
  TC      =     0.3584026E-02
  ARCL    =     5.0955


STATION       PRES            UE             ME          POPTZ         VOVCR
   1       2645.00000      0.             0.           1.000000      -0.
   2       2552.13171    270.64000      0.226543      0.964889      1.518034
   3       2499.51392    339.99200      0.285443      0.944996      1.618573
   4       2467.06619    376.86600      0.316992      0.932728      1.665801
   5       2447.92334    397.16400      0.334437      0.925491      1.690296
   6       2444.64600    400.54700      0.337351      0.924252      1.694283
   7       2443.32828    401.90000      0.338516      0.923754      1.695870
   8       2443.32828    401.90000      0.338516      0.923754      1.695870
   9       2444.97455    400.20900      0.337060      0.924376      1.693886
  10       2447.92334    397.16400      0.334437      0.925491      1.690296
  11       2448.67355    396.38600      0.333768      0.925774      1.689375
  12       2449.45514    395.57400      0.333069      0.926070      1.688413
```

2

```
STATION     X           Y           S           XCM         YOM         SOL
   1      0.          0.          0.          0.          0.          0.
   2      0.02500     0.05765     0.06284     0.00500     0.01153     0.01233
   3      0.06250     0.09470     0.11555     0.01250     0.01894     0.02268
   4      0.12500     0.13075     0.18770     0.02500     0.02615     0.03684
   5      0.25000     0.17775     0.32125     0.05000     0.03555     0.06305
   6      0.37500     0.21000     0.45034     0.07500     0.04200     0.08838
   7      0.50000     0.23415     0.57765     0.10000     0.04683     0.11337
   8      0.75000     0.26725     0.82984     0.15000     0.05345     0.16286
   9      1.00000     0.28685     1.08060     0.20000     0.05737     0.21207
  10      1.25000     0.29705     1.33081     0.25000     0.05941     0.26117
  11      1.30000     0.29810     1.38082     0.26000     0.05962     0.27099
  12      1.35000     0.29890     1.43083     0.27000     0.05978     0.28080
```

2

```
STATION      AF          TSF          TWAL         TAWL         TAWT         TBAR
   1      1200.768     600.000      600.000      600.000      600.000      600.000
   2      1194.652     593.904      600.000      598.995      599.310      598.141
   3      1191.102     590.379      600.000      598.414      598.911      597.067
   4      1188.881     588.179      600.000      598.052      598.662      596.396
   5      1187.559     586.872      600.000      597.836      598.514      595.997
   6      1187.331     586.647      600.000      597.799      598.489      595.929
   7      1187.240     586.557      600.000      597.784      598.479      595.901
   8      1187.240     586.557      600.000      597.784      598.479      595.901
   9      1187.354     586.670      600.000      597.803      598.492      595.936
  10      1187.559     586.872      600.000      597.836      598.514      595.997
  11      1187.611     586.923      600.000      597.844      598.520      596.013
  12      1187.665     586.977      600.000      597.853      598.526      596.029
```

2

```
STATION          RW       SW      SUTHL        RHSW             RHSE          HEADW      HEADE          NUW              MUBAR
   1               0.     0.      1.000     0.256824E-02     0.256824E-02    0.         0.        0.161903E-03     0.415805E-06
   2         101352.1     0.      1.000     0.247807E-02     0.250350E-02    90.754     91.686    0.167794E-03     0.414517E-06
   3         229311.6     0.      1.000     0.242693E-02     0.246653E-02   140.273    142.558    0.171326E-03     0.413772E-06
   4         407533.4     0.      1.000     0.239547E-02     0.244361E-02   170.112    173.531    0.173580E-03     0.413307E-06
   5         729339.0     0.      1.000     0.237688E-02     0.243005E-02   187.464    191.657    0.174937E-03     0.413031E-06
   6        1029751.0     0.      1.000     0.237370E-02     0.242773E-02   190.416    194.750    0.175172E-03     0.412984E-06
   7        1324608.7     0.      1.000     0.237242E-02     0.242679E-02   191.601    195.992    0.175266E-03     0.412964E-06
   8        1902883.0     0.      1.000     0.237242E-02     0.242679E-02   191.601    195.992    0.175266E-03     0.412964E-06
   9        2469150.1     0.      1.000     0.237402E-02     0.242796E-02   190.120    194.440    0.175148E-03     0.412988E-06
  10        3021572.4     0.      1.000     0.237688E-02     0.243005E-02   187.464    191.657    0.174937E-03     0.413031E-06
  11        3129731.7     0.      1.000     0.237761E-02     0.243059E-02   186.787    190.949    0.174883E-03     0.413042E-06
  12        3237464.3     0.      1.000     0.237837E-02     0.243114E-02   186.082    190.211    0.174828E-03     0.413053E-06
```

3

```
                SURFACE GRADIENTS

STATION         DUDS            DMDS            DMDL
   1         6688.219971      5.467127       27.857624
   2         1925.776199      1.743336        8.883132
   3          790.334671      0.776647        3.957390
   4          279.477798      0.324124        1.651568
   5           44.186589      0.075462        0.384517
   6           15.089687      0.015814        0.080580
   7            1.663434      0.006084        0.031001
   8           -0.866881     -0.002913       -0.014842
   9           -8.690249     -0.008147       -0.041513
  10          -14.868075     -0.012907       -0.065766
  11          -15.839995     -0.013684       -0.069727
  12          -15.830965     -0.013683       -0.069722
```

TABLE III. - Continued. SAMPLE OUTPUT

4

```
LAMINAR DIFFERENTIAL EQUATION - SOLUTION FOR CORLN


  S          CORLN          S          CORLN          S          CORLN          S          CORLN          S          CORLN

0.         -0.0818      0.01019   -0.0777      0.02038   -0.0733      0.03057   -0.0690      0.04076   -0.0650
0.05095    -0.0613      0.06115   -0.0581      0.07134   -0.0554      0.08153   -0.0533      0.09172   -0.0521
0.10191    -0.0519      0.11210   -0.0527      0.12229   -0.0507      0.13248   -0.0473      0.14267   -0.0445
0.15286    -0.0422      0.16306   -0.0408      0.17325   -0.0400      0.18344   -0.0401      0.19363   -0.0388
0.20382    -0.0360      0.21401   -0.0333      0.22420   -0.0308      0.23439   -0.0284      0.24458   -0.0263
0.25477    -0.0245      0.26496   -0.0229      0.27516   -0.0216      0.28535   -0.0206      0.29554   -0.0199
0.30573    -0.0195      0.31592   -0.0194      0.32611   -0.0186      0.33630   -0.0169      0.34649   -0.0153
0.35668    -0.0139      0.36687   -0.0126      0.37707   -0.0113      0.38726   -0.0103      0.39745   -0.0093
```

5

```
LAMINAR CALCULATION OF INSTABILITY AND TRANSITION LOCATIONS


STATION   CORLN      SHEAR      DTH       FORMTR     SHAPL        RTHI      SHAPK       RCRIT       KBAR          DIFF        RTRAN
   1     -0.0818    0.3522    8.9549    2.4890    7.0081        0.      0.07000     4015.7
   2     -0.0576    0.3150    8.5641    2.4620    3.9346      107.6     0.05682     3064.3
   3     -0.0522    0.3072    8.4903    2.4559    3.1931      188.7     0.04932     2366.4
   4     -0.0397    0.2993    8.3376    2.4420    1.9693      278.6     0.03648     1361.1
   5     -0.0191    0.2579    8.1424    2.4188    0.6089      417.9     0.01720      531.2
   6     -0.0064    0.2364    8.0597    2.4047    0.3255      532.2     0.00575      307.5
   7     -0.0034    0.2308    8.0438    2.4013    0.0493      626.7                            0.00362      1004.2      1536.4
   8      0.0026    0.2194    8.0176    2.3946   -0.0402      786.5                            0.00250       972.1      1504.3
   9      0.0101    0.2040    7.9538    2.3862   -0.5573      924.4                           -0.00111       874.9      1407.1
  10      0.0210    0.1793    7.9773    2.3740   -1.2456     1051.7                           -0.00468       788.6      1320.8
  11      0.0234    0.1734    7.9765    2.3713   -1.3995     1076.5                           -0.00553       769.6      1301.7
  12      0.0246    0.1704    7.9766    2.3700   -1.4698     1101.2                           -0.00634       751.7      1283.9
  13      0.0263    0.1661    7.9770    2.3681   -1.5714     1125.4                           -0.00707       736.1      1268.3
  14      0.0286    0.1599    7.9786    2.3654   -1.7175     1149.5                           -0.00788       719.4      1251.5
  15      0.0306    0.1548    7.9806    2.3633   -1.8327     1173.4                           -0.00875       701.7      1233.9
  16      0.0332    0.1476    7.9844    2.3604   -1.9929     1197.2                           -0.00957       685.4      1217.6
  17      0.0353    0.1416    7.9884    2.3580   -2.1235     1221.0                           -0.01043       669.0      1201.1
```

4

```
TURBULENT DIFFERENTIAL EQUATIONS - SOLUTION FOR  F  AND  FORMI


   S          F        FORMI        S          F        FORMI        S          F        FORMI        S          F        FORMI

1.68084     8727.6    1.3607     1.69103     9163.3    1.3762     1.70122     9590.4    1.3891     1.71141    10010.5    1.3996
1.72160    10425.1    1.4080     1.73179    10835.6    1.4148     1.74198    11242.9    1.4202     1.75217    11647.8    1.4245
1.76236    12051.0    1.4278     1.77255    12453.0    1.4303     1.78274    12854.2    1.4323     1.79294    13255.0    1.4337
1.80313    13655.5    1.4347     1.81332    14056.1    1.4354     1.82351    14456.9    1.4359     1.83370    14858.0    1.4361
1.84389    15259.5    1.4362     1.85408    15661.5    1.4361     1.86427    16064.2    1.4359     1.87446    16467.4    1.4356
1.88465    16871.4    1.4352     1.89485    17276.0    1.4348     1.90504    17631.4    1.4343     1.91523    18087.5    1.4338
1.92542    18494.5    1.4332     1.93561    18902.1    1.4327     1.94580    19310.6    1.4321     1.95599    19719.9    1.4315
1.96618    20129.9    1.4309     1.97637    20540.8    1.4303     1.98656    20952.4    1.4297     1.99676    21364.8    1.4290
```

```
  PRINCIPAL BOUNDARY LAYER INFORMATION


          INSTABILITY OCCURS AT STATION     6

          TRANSITION OCCURS AT STATION    17

          SEPARATION DOES NOT OCCUR

          LAMINAR BOUNDARY LAYER - STATIONS   1  TO   16

          TURBULENT BOUNDARY LAYER - STATIONS   17  TO   33
6

  STATION        X           S          DELSR         THET         DELTA         FORM        FORMI
     1        0.          0.          0.000116      0.000046     0.000412      2.5466      2.4890
     2        0.025000    0.062837    0.000171      0.000068     0.000586      2.5190      2.4620
     3        0.062500    0.115553    0.000245      0.000097     0.000832      2.5129      2.4559
     4        0.125000    0.187705    0.000329      0.000132     0.001106      2.4987      2.4420
     5        0.250000    0.321249    0.000469      0.000189     0.001553      2.4751      2.4188
     6        0.375000    0.450342    0.000589      0.000239     0.001944      2.4608      2.4047
     7        0.500000    0.577653    0.000691      0.000281     0.002279      2.4573      2.4013
     8        0.750000    0.829835    0.000865      0.000353     0.002850      2.4505      2.3946
     9        1.000000    1.080602    0.001016      0.000416     0.003351      2.4419      2.3862
    10        1.250000    1.330810    0.001157      0.000476     0.003828      2.4295      2.3740
    11        1.300000    1.380821    0.001185      0.000488     0.003925      2.4268      2.3713
    12        1.350000    1.430828    0.001214      0.000500     0.004021      2.4255      2.3700


  STATION       CF          TAUW        RTH          DTDY          NUSS        HTRAN         CRN
     1       0.          0.           0.           0.           0.          0.           4.238
     2       0.00573     0.52040     109.5      -2494.59      156.00      -8.9407        3.344
     3       0.00313     0.43964     193.1      -2790.34      202.59      -9.9648        3.184
     4       0.00198     0.33646     286.0      -2631.15      253.47      -9.4301        2.854
     5       0.00117     0.21894     429.8      -2148.27      318.92      -7.6955        2.397
6    6       0.00084     0.15954     547.5      -1767.25      361.59      -6.3339        2.147
     7       0.00070     0.13336     644.9      -1521.22      396.56      -5.4521        2.087
     8       0.00053     0.10101     809.3      -1221.21      457.33      -4.3768        1.969
     9       0.00042     0.07935     951.0      -1034.98      508.99      -3.7094        1.817
    10       0.00032     0.06048    1081.6       -898.83      552.76      -3.2214        1.583
    11       0.00030     0.05696    1107.1       -874.99      560.52      -3.1360        1.528
    12       0.00029     0.05454    1132.4       -851.45      567.51      -3.0516        1.501
```

TABLE III. - Concluded. SAMPLE OUTPUT

```
  VELOCITY PROFILES


  STATION       2   PROFILE
    Y/DELTA           Y              Y/XMAX            U           U/UE
    0.            0              0                   0.          0.
    0.0500        0.292757E-04   0.585514E-05       34.61        0.1279
    0.1000        0.585514E-04   0.117103E-04       66.55        0.2459
    0.1500        0.878271E-04   0.175654E-04       95.85        0.3542
    0.2000        0.117103E-03   0.234205E-04      122.53        0.4528
    0.2500        0.146378E-03   0.292757E-04      146.64        0.5418
    0.3000        0.175654E-03   0.351308E-04      168.22        0.6216
7   0.3500        0.204930E-03   0.409860E-04      187.36        0.6923
    0.4000        0.234205E-03   0.468411E-04      204.13        0.7543
    0.4500        0.263481E-03   0.526962E-04      218.64        0.8079
    0.5000        0.292757E-03   0.585514E-04      230.99        0.8535
    0.5500        0.322033E-03   0.644065E-04      241.31        0.8916
    0.6000        0.351308E-03   0.702616E-04      249.74        0.9228
    0.6500        0.380584E-03   0.761168E-04      256.44        0.9475
    0.7000        0.409860E-03   0.819719E-04      261.57        0.9665
    0.7500        0.439135E-03   0.878271E-04      265.32        0.9803
    0.8000        0.468411E-03   0.936822E-04      267.88        0.9898
    0.8500        0.497687E-03   0.995373E-04      269.46        0.9956
    0.9000        0.526962E-03   0.105392E-03      270.29        0.9987
    0.9500        0.556238E-03   0.111248E-03      270.60        0.9998
    1.0000        0.585514E-03   0.117103E-03      270.64        1.0000


  STATION      17   PROFILE                                 N=    5.54
    Y/DELTA           Y              Y/XMAX            U           U/UE
    0.            0              0                   0.          0.
    0.0500        0.258083E-03   0.516165E-04      227.98        0.5826
    0.1000        0.516165E-03   0.103233E-03      258.33        0.6602
    0.1500        0.774248E-03   0.154850E-03      277.93        0.7103
    0.2000        0.103233E-02   0.206466E-03      292.73        0.7481
    0.2500        0.129041E-02   0.258083E-03      304.75        0.7788
    0.3000        0.154850E-02   0.309699E-03      314.94        0.8048
    0.3500        0.180658E-02   0.361316E-03      323.82        0.8275
    0.4000        0.206466E-02   0.412932E-03      331.71        0.8477
7   0.4500        0.232274E-02   0.464549E-03      338.83        0.8659
    0.5000        0.258083E-02   0.516165E-03      345.33        0.8825
    0.5500        0.283891E-02   0.567782E-03      351.32        0.8978
    0.6000        0.309699E-02   0.619398E-03      356.87        0.9120
    0.6500        0.335508E-02   0.671015E-03      362.06        0.9253
    0.7000        0.361316E-02   0.722632E-03      366.93        0.9377
    0.7500        0.387124E-02   0.774248E-03      371.53        0.9494
    0.8000        0.412932E-02   0.825865E-03      375.88        0.9606
    0.8500        0.438741E-02   0.877481E-03      380.01        0.9711
    0.9000        0.464549E-02   0.929098E-03      383.95        0.9812
    0.9500        0.490357E-02   0.980714E-03      387.71        0.9908
    1.0000        0.516165E-02   0.103233E-02      391.31        1.0000
```

ERROR CONDITIONS

This section lists the error messages given by the program and tells what to do when they are encountered:

(1) ERROR IN INPUT DATA. RECHECK INPUT INSTRUCTIONS.

This message is printed by subroutine INPUT if one of the following conditions is not met:

$$\begin{gathered} NST \leq 100 \\ NTURB \leq NST \\ KEM = 0 \text{ or } 1 \\ KSPLN = 0 \text{ or } 1 \\ KLE = 0 \text{ or } 1 \\ KATCH = 0 \text{ or } 1 \\ KPVM = 1,\ 2,\ 3,\ 4, \text{ or } 5 \end{gathered}$$

(2) THERE IS A STAGNATION POINT AT A STATION OTHER THAN STATION 1.
THIS IS NOT ALLOWED.

This message is printed by subroutine PRECAL if any of the following conditions is met at a station other than station 1:

$$\begin{gathered} P = P'_0 \\ u_e = 0. \\ M_e = 0. \\ P/P'_0 = 1. \\ u_e/u_{cr} = 0. \end{gathered}$$

If a stagnation point occurs in the input pressure or velocity distribution, it must occur at station 1.

(3) AN INPUT PRESSURE, VELOCITY, OR MACH NUMBER IS EITHER LESS
THAN ZERO OR GREATER THAN ITS MAXIMUM ALLOWABLE VALUE.

This message is printed in subroutine PRECAL if any of the following conditions is met at any station:

$$P < 0. \text{ or } P > P'_0$$
$$u_e < 0.$$
$$M_e < 0.$$
$$P/P'_0 < 0. \text{ or } P/P'_0 > 1.$$
$$u_e/u_{cr} < 0. \text{ or } u_e/u_{cr} > \sqrt{(\gamma + 1)/(\gamma - 1)}$$

(4) A NEGATIVE INITIAL VALUE HAS BEEN GIVEN. THIS IS NOT ALLOWED.

This message is printed by subroutine LAMNAR if one of the four initial values (DLAM, TLAM, DTURB, or TTURB) is less than zero. They can be positive or zero, but not negative.

(5) INITIAL VALUES WERE NOT GIVEN FOR THE TURBULENT BOUNDARY LAYER AT STATION 1.

This message is printed by subroutine LAMNAR if NTURB = 1 and values are not given for both DTURB and TTURB. If the turbulent boundary layer is to start at station 1, initial values must be given for both displacement thickness and momentum thickness.

(6) INITIAL VALUES WERE GIVEN FOR THE TURBULENT BOUNDARY LAYER AT A STAGNATION POINT.

This message is printed by subroutine LAMNAR if NTURB = 1 and DTURB and TTURB have been given values, but the pressure, velocity, or Mach number at station 1 indicates a stagnation condition at that point. The program will not allow a turbulent boundary layer to begin at a stagnation point. A pressure less than P'_0, or a velocity or Mach number greater than zero, should be given for this case at station 1.

(7) INITIAL VALUES OTHER THAN ZERO WERE GIVEN FOR THE LAMINAR BOUNDARY LAYER AT A STAGNATION POINT.

This message is printed by subroutine LAMNAR if initial values are given for DLAM and/or TLAM but the pressure, velocity, or Mach number at station 1 indicates a stagnation condition at that point. A pressure less than P'_0, or a velocity or Mach number greater than zero, should be given at station 1 if the laminar boundary layer has a thickness there.

(8) FOR THIS INPUT DATA STATION 1 IS ASSUMED TO BE A STAGNATION POINT, SINCE NO INITIAL THICKNESSES ARE GIVEN. IN THIS CASE PRESSURE SHOULD DECREASE INITIALLY. EITHER GIVE AN INITIAL VALUE FOR DISPLACEMENT OR MOMENTUM THICKNESS, OR BEGIN WITH A SHORT REGION OF FAVORABLE PRESSURE GRADIENT.

This message is printed by subroutine LAMNAR. If initial values are not given, and input does not indicate a sharp leading edge, the program assumes it has a stagntion point at station 1. The initial pressure ordinarily given in this case would be P_0' (or $u_e = 0$, or $M_e = 0$). However, the program will accept other values of pressure or velocity at station 1, as long as $dP/dx < 0$ initially.

(9) LAMINAR SOLUTION HAS PROCEEDED BEYOND THE RANGE WHERE IT IS VALID.

This message is printed by subroutine LAMNAR if the value of correlation number calculated in the solution of the laminar-boundary-layer equation is beyond the range $-0.32 \leq n \leq 0.16$. This can be caused by a strong adverse or favorable pressure gradient, generally occurring rapidly over one or two stations along the surface. Several input stations should be used to pass through any rapid change in pressure or velocity distribution.

(10) IF INITIAL TURBULENT VALUES ARE GIVEN, THEY BOTH MUST BE NONZERO.

This message is printed by subroutine LAMNAR if one of the following conditions exists:

DTURB > 0. and TTURB = 0.
DTURB = 0. and TTURB > 0.

If initial values are given for the turbulent boundary layer at any station, both displacement thickness and momentum thickness must be given.

(11) ERROR IN COMPUTING INTEGRAL FOR CORLN or ERROR IN COMPUTING INTEGRAL FOR KBAR.

One of these messages is printed by subroutine LAMNAR if an error occurs in a call on the SIMPS1 integration routine. This condition should generally not occur with the kinds of distributions being integrated in this program.

INFORMATION FOR PROGRAMMER

DESCRIPTION OF PROGRAM

The program BLAYER is segmented into five principal parts; the subroutines INPUT, PRECAL, LAMNAR, TURBLN, and PROFIL. These, in turn, call several other subroutines. All the subroutines and their relation are shown in figure 7.

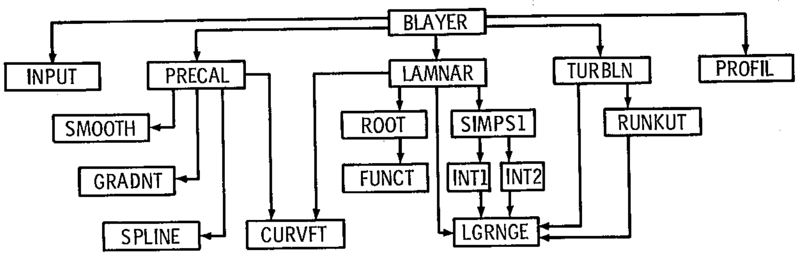

Figure 7. - Calling relation of program subroutines.

Most of the subroutines in BLAYER use the same set of variables. These are all in COMMON blocks which transmit information between routines. These variables are all defined in the DICTIONARY OF VARIABLES IN MAIN SUBROUTINES. All subroutines using these variables are described in the section MAIN SUBROUTINES. The remaining subroutines are described after the main dictionary in the section AUXILIARY SUBROUTINES. Their variables are described with each subroutine.

The program can handle as many as 100 points along the boundary-layer surface (NST) and as many points as the user desires normal to the surface in any of the velocity profiles (NVP). The program is run at Lewis on the IBM-7094/7044 direct-coupled system with a 32 768-word core ($77777_{(8)}$). The total program storage requirement is $50511_{(8)}$ of which $24627_{(8)}$ is used in storage of variables. The main subroutines require $42305_{(8)}$ words of storage of which $14336_{(8)}$ is used for COMMON blocks. The auxiliary subroutines use $6204_{(8)}$ words.

The first principal routine called by BLAYER is INPUT. This reads in and prints the input data and checks for errors. PRECAL then calculates all quantities which remain constant, and are subsequently used by the laminar- and turbulent-boundary-layer routines. The subroutine LAMNAR solves the laminar differential equation, computes the laminar-boundary-layer parameters, and checks for transition. After transition, it computes the initial values for the turbulent boundary layer. TURBLN solves the turbulent-boundary-layer equations by using RUNKUT, and computes the turbulent-boundary-layer parameters. Finally, PROFIL computes both laminar and turbulent velocity profiles.

Conventions Used in Program

For clarity, a number of conventions are used in naming variables in the program. In most cases, E at the end of a variable name refers to conditions external to the boundary layer, whereas W refers to conditions at the wall or boundary-layer surface; Z refers to station 0, where total conditions are given in the input; T means total conditions, and S static; P represents pressure, T temperature, V velocity, A speed of sound, RH density, MU dynamic viscosity, and NU kinematic viscosity. Thus, for example, POPTZ represents pressure over total pressure at station 0.

Labeled COMMON Blocks

Most variables which are used in more than one subroutine are placed in labeled COMMON blocks. The same variable names are used in the different subroutines for all the variables in COMMON blocks. The labeled COMMON blocks are briefly described by the following:

/C1/	contains all input variables
/C2/	contains single variables computed in PRECAL
/C3/	contains all arrays computed in PRECAL
/C4/	contains arrays of all principal boundary-layer parameters computed in LAMNAR and TURBLN
/C5/	contains variables transferred between LAMNAR and PROFIL, and between LAMNAR and INT1 and INT2
/C6/	contains initial turbulent values transferred between LAMNAR, TURBLN, and RUNKUT
/C7/	contains variables indicating position of instability, transition, and separation
/C8/	contains tables for solution of turbulent boundary-layer equations (These are transferred from RUNKUT to TURBLN.)
/C9/	contains variables indicating whether transition or separation has been encountered, or whether any errors have been found

MAIN SUBROUTINES

The following subroutines are particular to the BLAYER program, and all use the

COMMON blocks described in the previous section. More general subroutines (i. e., they are independent of BLAYER) are discussed in the section AUXILIARY SUBROUTINES.

Subroutine INPUT

Subroutine INPUT reads and prints all input data. It also checks for errors in the input.

Subroutine PRECAL

Subroutine PRECAL performs all the calculations required prior to the solution of the boundary-layer differential equations. All the variables it computes remain constant for the remainder of the program.

Reading of coefficients for curve fits. - PRECAL uses several curve fits, and the coefficients for these are stored into arrays at the beginning of the routine. Coefficients are given for the following functional relations (see appendix B):

$$\frac{\mu}{\mu_{sl}} = f\left(\frac{T}{T_{sl}}\right)$$

$$Pr = f\left(\frac{T}{T_{sl}}\right)$$

$$\frac{k}{k_{sl}} = f\left(\frac{T}{T_{sl}}\right)$$

These relations have been nondimensionalized so that they can be used with either English or metric units.

Initial calculations. - Using the total parameters and Mach number at station 0, other total and static parameters required by the program are computed. The curve fits are used in these calculations. Geometrical parameters are also obtained, and surface length computed from equation (11).

Calculation and smoothing of surface flow distributions. - One of five possible variables is used as input for flow past the boundary-layer surface. In this section of PRECAL, the remaining four variables are calculated from the variable given as input. If the user has requested smoothing, SMOOTH is then called for the velocity distribution UE. Each time smoothing is performed the other four surface distributions are recomputed from UE, and all five distributions are reprinted. After surface distributions have been finalized, other constant parameters depending on them are calculated.

Calculation of surface gradients. - Surface gradients of velocity and Mach number (du_e/dx, dM_e/dx, and $dM_e/d(x/L)$) are calculated by means of calls on either GRADNT or SPLINE, depending on which the user has requested.

Subroutine LAMNAR

Subroutine LAMNAR solves the laminar differential boundary-layer equation, computes laminar-boundary-layer parameters, checks for instability and transition to turbulent flow, and computes initial values for the turbulent boundary layer when transition occurs.

Reading of coefficients for curve fits. - Several curve fits are used in the LAMNAR routine, and the coefficients for these are stored into arrays at the beginning of the routine. Coefficients are given for the following functional relations:

$$\eta_{sp} = f(S_w)$$

$$\left(R_\theta\right)_{cr} = f(K)$$

$$\left(R_\theta\right)_{i,\Delta} = f(\overline{K})$$

$$l = f(n, S_w)$$

$$\left(\frac{C_f R_w}{Nu_x}\right)_{Pr=1} = f(n, S_w)$$

$$\left(\frac{\delta_{tr}}{\theta_{tr}}\right) = f(n, S_w)$$

These functional relations are given in appendix B.

Initial values. - A check is made for initial values given by the user. If initial values are given for the turbulent boundary layer at station 1, the program transfers to the end of LAMNAR, where transformed quantities are computed for use by the TURBLN routine. If an initial value is given for the laminar boundary layer, the corresponding correlation number n_1 is computed, since the laminar equation is solved in terms of n. If a laminar momentum thickness is given, the corresponding correlation number n is computed from the following relation obtained by rearranging equation (15), which is derived in appendix A:

$$\frac{n_1}{\left[\dfrac{dM_e}{d\left(\dfrac{x}{L}\right)}\right]_1} = \frac{-a_0' \theta_1^2}{\nu_0' \left(k_{su}\right)_1 L} \frac{1}{\left[1 + \dfrac{\gamma - 1}{2}\left(M_e^2\right)_1\right]^{(3-\gamma)/(2\gamma-2)}} \tag{12}$$

If an initial laminar displacement thickness is given, the corresponding correlation number is computed by calling the subroutine ROOT. Two calls are given in LAMNAR for ROOT, depending on whether the Mach number gradient $dM_e/d(x/L)$ is positive or negative. Since $\delta^* = \theta H$ and θ and H are both functions of n, δ^* can be expressed as a function of n. Subroutine ROOT solves the following equation for n_1, given δ_1^*:

$$\delta_1^* = \left\{ n_1 \frac{-\nu_0'\left(k_{su}\right)_1 L}{a_0' \left[\dfrac{dM_e}{d\left(\dfrac{x}{L}\right)}\right]_1} \left[1 + \frac{\gamma - 1}{2}\left(M_e^2\right)_1\right]^{(3-\gamma)/(2\gamma-2)} \right\}^{1/2}$$

$$\times (-1.1138\, n_1 + 2.38411)\left[1 + \left(2.79 - 1.78\, Pr^{1/2}\right)\left\{\left[1 + \left(S_w\right)_1\right]\left[1 + \frac{\gamma - 1}{2}\left(M_e^2\right)_1\right] - 1\right\}\right]$$

$$+ \left(4.65\, Pr^{1/3} - 3.65\, Pr^{1/2}\right)\left[Pr^{1/2} \frac{\gamma - 1}{2}\left(M_e^2\right)_1\right] \tag{13}$$

This relation is obtained from equations (15) and (16).

If no initial laminar values are given, LAMNAR obtains n_1 in one of two ways. For a sharp leading edge, $n_1 = 0$. For stagnation point flow, n_1 is obtained from the curve fit of n_{sp} against S_w.

Solution of the laminar differential equation. - Using n_1 as an initial value, the solution to the laminar differential equation can be obtained. This solution is expressed in equation (4). Numerically, the solution is obtained from point to point by the following formula (see appendix A):

$$(n)_{x_2/L} = \left[-AM_e^{-B} \frac{dM_e}{d\left(\frac{x}{L}\right)} \left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{(3\gamma-1)/(2\gamma-2)} \right]_{x_2/L}$$

$$\times \left[\int_{x_1/L}^{x_2/L} \frac{M_e^{B-1}}{\left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{(3\gamma-1)/(2\gamma-2)}} \, d\left(\frac{x}{L}\right) \right]$$

$$+ \frac{\left[M_e^{-B} \frac{dM_e}{d\left(\frac{x}{L}\right)} \left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{(3\gamma-1)/(2\gamma-2)} \right]_{x_2/L}}{\left[M_e^{-B} \frac{dM_e}{d\left(\frac{x}{L}\right)} \left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{(3\gamma-1)/(2\gamma-2)} \right]_{x_1/L}} \times (n)_{x_1/L} \qquad (14)$$

A spacing of 1/500th of surface length L is used to step from point to point. This spacing represents a compromise between use of machine time and excessive accuracy in the solution of the equation. The resulting surface lengths and correlation numbers are stored in tables.

Calculation of laminar-boundary-layer parameters. Through interpolation, correlation numbers are obtained at the input-output stations on the boundary-layer surface. Using these values of n, the boundary-layer parameters are calculated.

Momentum thickness is obtained from the following equation (see appendix A):

$$\theta = \left\{ \frac{\nu_0' k_{su} L}{a_0'} \left[\frac{-n}{\dfrac{dM_e}{d\left(\dfrac{x}{L}\right)}} \right] \left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{(3-\gamma)/(2\gamma-2)} \right\}^{1/2} \tag{15}$$

Form factor is obtained from the following relation, obtained from reference 12:

$$H = (-1.1138\, n + 2.38411)\left\{1 + (2.79 - 1.78\, Pr^{1/2})\left[(1 + S_w)\left(1 + \frac{\gamma - 1}{2} M_e^2\right) - 1\right]\right\}$$

$$+ (4.65\, Pr^{1/3} - 3.65\, Pr^{1/2})\left(Pr^{1/2} \frac{\gamma - 1}{2} M_e^2\right) \tag{16}$$

Displacement thickness follows from

$$\delta^* = \theta H \tag{17}$$

Adiabatic form factor is then obtained from the following equation (see appendix A):

$$H_i = \frac{H - Pr^{1/2} \dfrac{\gamma - 1}{2} M_e^2}{(1 + S_w)\left(1 + \dfrac{\gamma - 1}{2} M_e^2\right)} \tag{18}$$

Boundary-layer thickness δ is obtained from equation (41) of reference 4:

$$\delta = \theta\left\{\left(\frac{\delta_{tr}}{\theta_{tr}}\right) + \frac{\gamma - 1}{2} M_e^2\left[(1 + S_w)H_i + 1\right]\right\} \tag{19}$$

The shear-stress parameter l is obtained from the curve fit of l as a function of n and S_w (see appendix B). The skin friction coefficient is then calculated from the following equation (see appendix A):

$$C_f = \frac{2l\sqrt{-\frac{1}{n}\frac{1}{M_e}\frac{dM_e}{d\left(\frac{x}{L}\right)}\frac{x}{L}}}{\sqrt{R_w}} \tag{20}$$

Shear stress at the wall follows from

$$\tau_w = C_f \frac{1}{2} \rho_w u_e^2 \tag{21}$$

Laminar separation occurs when C_f or τ_w attains a negative value.

The Reynolds analogy heat-transfer parameter $C_f R_w/Nu_x$ for $Pr = 1$, is obtained from the curve fit of $C_f R_w/Nu_x$ as a function of n and S_w. The local Nusselt number is obtained from

$$Nu_x = \frac{C_f R_w Pr^{0.3}}{\left(\frac{C_f R_w}{Nu_x}\right)_{Pr=1}} \tag{22}$$

which is a form of equation (38) of reference 4. The slope of the wall temperature profile is then calculated by using the definition of local Nusselt number

$$Nu_x = \frac{x\left(\frac{\partial T}{\partial y}\right)_w}{T_{aw} - T_w} \tag{23}$$

and heat transfer per unit area is obtained from

$$q = k\left(\frac{\partial T}{\partial y}\right)_w \tag{24}$$

Check for instability and transition. - Position of instability is checked using Schlichting and Ulrich's curve of critical momentum-thickness Reynolds number $\left(R_\theta\right)_{cr}$ as a function of momentum-thickness shape factor K (refs. 7 and 8). At each point in the laminar boundary layer, $\left(R_\theta\right)_{cr}$ is obtained and compared with local incompressible momentum-thickness Reynolds number $\left(R_\theta\right)_i$ (see fig. 8). When $\left(R_\theta\right)_i$ surpasses $\left(R_\theta\right)_{cr}$, the boundary layer is assumed to be unstable. At each station after the point of instability, the mean pressure gradient parameter $\overline{K}$ is computed from the following equation:

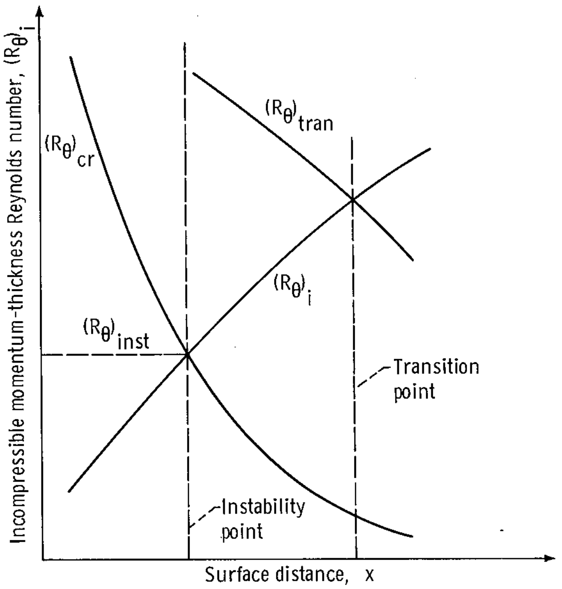

Figure 8. - Variation of variables used in prediction of transition.

$$\overline{K} = \frac{1}{\frac{x}{L} - \left(\frac{x}{L}\right)_{inst}} \int_{(x/L)_{inst}}^{x/L} K_i d\left(\frac{x}{L}\right) \tag{25}$$

Using Granville's results (ref. 9), the difference $\left(R_\theta\right)_{i,\Delta}$ between instability and transition momentum-thickness Reynolds numbers is obtained from a curve as a function of $\overline{K}$. This difference $\left(R_\theta\right)_{i,\Delta}$ is added to Reynolds number at instability $\left(R_\theta\right)_{inst}$ to obtain a local transition Reynolds number $\left(R_\theta\right)_{tran}$. Transition is assumed to occur when local incompressible Reynolds number $\left(R_\theta\right)_i$ attains a value larger than $\left(R_\theta\right)_{tran}$ (see fig. 8).

Initial values for turbulent boundary layer. - After transition is located, initial values of f and H_i are calculated for the solution of the turbulent-boundary-layer equations.

Subroutine TURBLN

Subroutine TURBLN solves the turbulent-boundary-layer equations, and computes the turbulent-boundary-layer parameters.

Solution of turbulent differential equations. - The initial values of f and H_i are transferred to TURBLN from LAMNAR. TURBLN calls RUNKUT, which uses these initial values for the solution of the coupled differential equations (eqs. (8) and (9)) of the turbulent boundary layer. Once again a step size of 1/500th of the arc length L is used in the solution. The resulting distributions of f and H_i are returned to TURBLN in tables.

Calculation of turbulent-boundary-layer parameters. - By means of interpolation, values of f and H_i are obtained at the input and output stations. By using these values of f and H_i, the boundary-layer parameters are calculated.

Transformed momentum thickness θ_{tr} is obtained by using the definition of the function f:

$$f = \left(\frac{M_e a'_0 \theta_{tr}}{\nu'_0}\right)^{1.268} \tag{26}$$

From transformed momentum thickness, actual momentum thickness is then calculated as follows:

$$\theta = \left(\frac{T'_0}{T_e}\right)^{(\gamma+1)/(2\gamma-2)} \theta_{tr} \tag{27}$$

The form factor is obtained from the incompressible form factor by the equation

$$H = H_i(1 + S_w)\left(1 + \frac{\gamma - 1}{2} M_e^2\right) + Pr^{1/3} \frac{\gamma - 1}{2} M_e^2 \tag{28}$$

Displacement thickness follows from

$$\delta^* = \theta H \tag{29}$$

The power law is used for velocity profiles in the turbulent boundary layer (see p. 47). When the power law is used, the boundary-layer thickness δ can be obtained from δ^* through the relation

$$\delta = (1 + n_p)\delta^* \tag{30}$$

The Ludwieg-Tillman skin friction relation, transformed for compressible flow, is used to obtain C_f. The following relation is obtained from equation (10) of reference 5:

$$C_f = 0.246 \, e^{-1.561 \, H_i} \left[\frac{u_e \theta}{\nu'_0 \left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{1/(\gamma - 1)}} \right]^{-0.268} \left(\frac{T_e}{\bar{T}}\right)\left(\frac{\bar{\mu}}{\mu'_0}\right)^{0.268} \tag{31}$$

Shear stress at the wall follows from

$$\tau_w = C_f \frac{1}{2} \rho_e u_e^2 \tag{32}$$

Heat transfer is obtained from C_f, as in reference 6,

$$q = \frac{C_f}{2} \frac{\rho_e u_e c_p}{Pr^{2/3}} (T_{aw} - T_w) \tag{33}$$

The slope of the temperature profile at the wall $(\partial T/\partial y)_w$ is obtained from q/k, and the local Nusselt number is computed as defined:

$$Nu_x = \frac{x\left(\frac{\partial T}{\partial y}\right)_w}{T_{aw} - T_w} \tag{34}$$

Subroutine PROFIL

Subroutine PROFIL prints all the principal boundary-layer parameters computed by LAMNAR and TURBLN, and calculates and prints the laminar and turbulent velocity profiles.

Laminar profiles. - Laminar profiles are computed using Pohlhausen's fourth-degree equation

$$\frac{u}{u_e} = a\eta + b\eta^2 + c\eta^3 + d\eta^4 \tag{35}$$

where

$$\eta = \frac{y}{\delta}$$

$$a = 2 + \frac{\lambda}{6}$$

$$b = -\frac{\lambda}{2}$$

$$c = -2 + \frac{\lambda}{2}$$

$$d = 1 - \frac{\lambda}{6}$$

and the shape factor

$$\lambda = \frac{\delta^2}{\nu_w} \frac{du_e}{dx}$$

Turbulent profiles. - Turbulent profiles are computed using the power law

$$\frac{u}{u_e} = \left(\frac{y}{\delta}\right)^{1/n_p} \tag{36}$$

The power factor n_p is obtained in TURBLN from the relation

$$H_i = \frac{2 + n_p}{n_p} \tag{37}$$

which follows from the definition of H_i and equation (36).

Subroutine RUNKUT

Subroutine RUNKUT solves the coupled ordinary differential equations of the turbulent boundary layer using a fourth-order Runge-Kutta method. Initial values of f and H_i are transferred from TURBLN. A step size Δx of 1/500th of L is used. The two differential equations (8) and (9), are expressed as functions of x, f, H_i, and some other known quantities as follows:

$$\frac{df}{dx} = F_1(x,\ f,\ H_i) \tag{38}$$

$$\frac{dH_i}{dx} = F_2(x,\ f,\ H_i)$$

The following Runge-Kutta equations are then used from step to step

$$f_{N+1} = f_N + \frac{1}{6}(c_1 + 2c_2 + 2c_3 + c_4) \tag{39}$$

$$\left(H_i\right)_{N+1} = \left(H_i\right)_N + \frac{1}{6}(d_1 + 2d_2 + 2d_3 + d_4) \tag{40}$$

where

$$\left.\begin{aligned} c_1 &= \Delta x\ F_1(x_N,\ f_N,\ H_{i,N}) \\ c_2 &= \Delta x\ F_1\left(x_N + \frac{\Delta x}{2},\ f_N + \frac{c_1}{2},\ H_{i,N} + \frac{d_1}{2}\right) \\ c_3 &= \Delta x\ F_1\left(x_N + \frac{\Delta x}{2},\ f_N + \frac{c_2}{2},\ H_{i,N} + \frac{d_2}{2}\right) \\ c_4 &= \Delta x\ F_1(x_N + \Delta x,\ f_N + c_3,\ H_{i,N} + d_3) \end{aligned}\right\} \tag{41}$$

and

$$\left.\begin{aligned}
d_1 &= \Delta x\, F_2(x_N,\ f_N,\ H_{i,N}) \\
d_2 &= \Delta x\, F_2\left(x_N + \frac{\Delta x}{2},\ f_N + \frac{c_1}{2},\ H_{i,N} + \frac{d_1}{2}\right) \\
d_3 &= \Delta x\, F_2\left(x_N + \frac{\Delta x}{2},\ f_N + \frac{c_2}{2},\ H_{i,N} + \frac{d_2}{2}\right) \\
d_4 &= \Delta x\, F_2(x_N + \Delta x,\ f_N + c_3,\ H_{i,N} + d_3)
\end{aligned}\right\} \tag{42}$$

Final tabulated values of f and H_i are returned to TURBLN.

Subroutine FUNCT

Subroutine FUNCT is called by the ROOT subroutine in LAMNAR when displacement thickness is given as an initial value for the laminar boundary layer. The ROOT subroutine is called in order to obtain an initial correlation number corresponding to the displacement thickness given. Since δ^* is a function of θ and H, and θ and H are both functions of correlation number n (see eqs. (15) and (16)), then δ^* can be expressed as a function of n. FUNCT expresses this functional relation of δ^* and n for use by the ROOT routine.

Function INT1

Function INT1 computes the integrand used in the first call on the SIMS1 integration routine. The integral is the following:

$$\int_{x_1/L}^{x_2/L} \frac{M_e^{B-1}}{\left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{(3\gamma-1)/(2\gamma-2)}}\, d\left(\frac{x}{L}\right)$$

which is part of equation (14).

Function INT2

Function INT2 computes the integrand used in the second call on the SIMPS1 integration routine. The integral is the following:

$$\int_{(x/L)_{inst}}^{x/L} K_i \, d\left(\frac{x}{L}\right)$$

which appears in equation (25).

DICTIONARY OF VARIABLES IN MAIN SUBROUTINES

A,A1,A2,A3,A4	temporary variables in LAMNAR
AA	combination of several terms in turbulent-boundary-layer equations
AAA	coefficient in expression for Pohlhausen laminar velocity profiles
AE	speed of sound external to boundary layer a_e based on T_e, ft/sec; m/sec
ANS	value of Mach number M_e interpolated by LGRNGE in INT1 routine
ANS1,ANS2, ANS3,ANS4, ANS5,ANS6	values of SW, ME, DMDL, DMDS, AA, BB, and TBAR interpolated by LGRNGE calls in LAMNAR and RUNKUT routines
ARCL	total distance along surface in x-direction, L, ft; m
ASZ(ATZ)	speed of sound $a_0(a_0')$ based on static (total) temperature at station 0, ft/sec; m/sec
B	temporary variable in LAMNAR and INT1
B1,B2,B3,B4,B5	temporary variables in FUNCT
BB	combination of several terms in turbulent-boundary-layer equations
BBB	coefficient in expression for Pohlhausen laminar velocity profiles
CCC	coefficient in expression for Pohlhausen laminar velocity profiles
CCN	array of coefficients for curve fit of n_{sp} against S_w in LAMNAR
CCRN	array of coefficients for curve fit of $C_f R_w / Nu_x$ against n and S_w in LAMNAR

CDIF	array of coefficients for curve fit of $\left(R_{\theta}\right)_{i,\Delta}$ against $\bar{K}$ in LAMNAR
CDTH	array of coefficients for curve fit of δ_{tr}/θ_{tr} against n and S_w in LAMNAR
CF	skin friction coefficient at wall, C_f
CFRW	$C_f \sqrt{R_w}$
CMU	array of coefficients for curve fit of μ/μ_{sl} against T/T_{sl} in PRECAL
CORLN	correlation number, or pressure gradient parameter, n
CORML	CORLN/DMDL or $n/\left[dM_e/d(x/L)\right]$
CP	specific heat at constant pressure, c_p, (ft)(lbf)/(slug)(^{o}R); J/(kg)(K)
CPR	array of coefficients for curve fit of Pr against T/T_{sl} in PRECAL
CRCR	array of coefficients for curve fit of $\left(R_{\theta}\right)_{cr}$ against K in LAMNAR
CRN	Reynolds analogy parameter, $C_f R_w/Nu_x$
CSHR	array of coefficients for curve fit of l against n and S_w in LAMNAR
CTAB1	table of correlation numbers (CORLN) obtained from CORML while solving laminar differential equation
CTAB2	table of values of variable CORML computed by solving laminar differential equation
CTC	array of coefficients for curve fit of k/k_{sl} against T/T_{sl} in PRECAL
CTHET	see INPUT
DDD	coefficient in expression for Pohlhausen laminar velocity profiles
DEL	small increment along x- or y-axis
DELSR	displacement thickness, δ^*, ft; m
DELTA	full boundary-layer thickness, δ, ft; m
DFX	derivative of FX with respect to XX in FUNCT
DIFF	difference between transition and instability momentum-thickness Reynolds numbers $\left(R_{\theta}\right)_{i,\Delta}$ in LAMNAR
DLAM	see INPUT
DMDL	$dM_e/[d(x/L)]$

DMDS	dM_e/dx, ft^{-1}; m^{-1}
DOT	two-element array containing expressions for coupled ordinary differential equations for f and H_i in RUNKUT
DTDY	slope of temperature profile at wall, $(\partial T/\partial y)_w$, $^{o}R/ft$; K/m
DTH	δ_{tr}/θ_{tr} in LAMNAR
DTURB	see INPUT
DUDS	du_e/dx, sec^{-1}
ERROR	logical variable indicating error in program due to improper input
ETA	y/δ
F	$f = \left[(U_e \theta_{tr})/(\nu_0')\right]^{1.268}$
FF	polynomial function of M_e and S_w
FORM	form factor, H
FORMI	adiabatic form factor, H_i
FORMS	initial value of H_i for turbulent boundary-layer equations
FORMTR	transformed form factor, H_{tr}
FTRAN	initial value of f for turbulent boundary-layer equations
FX	value of function calculated in FUNCT
GAM	ratio of specific heats, γ
HEADE	velocity head based on density external to boundary layer, $1/2\left(\rho_e u_e^2\right)$, lbf/ft^2; N/m^2
HEADW	velocity head based on density at wall, $1/2\left(\rho_w u_e^2\right)$, lbf/ft^2; N/m^2
HTRAN	heat transfer per unit area, $(ft)(lbf)/(sec)(ft^2)$; $J/(sec)(m^2)$
I	integer iteration counter
IEND	integer location of station which ends boundary layer
IL1(IL2)	integer location of station at beginning (end) of laminar portion of boundary layer
INF	indicator of an infinite derivative (DFX) in FUNCT
INST	integer location of station where laminar instability occurs
ISEP	integer location of station where laminar or turbulent separation occurs

IT1(IT2)	integer location of station at beginning (end) of turbulent portion of boundary layer
ITRAN	integer location of station where transition occurs from laminar to turbulent boundary layer
K1	error indicator in call on SIMPS1
KATCH	see INPUT
KBAR	mean shape factor based on momentum thickness, $\overline{K}$
KDONE	integer counter indicating how many times input data has been smoothed by calls on SMOOTH in PRECAL
KEM	see INPUT
KGRAD	see INPUT
KLAM	see INPUT
KLE	see INPUT
KMAIN	see INPUT
KPRE	see INPUT
KPROF	see INPUT
KPVM	see INPUT
KSDE	see INPUT
KSMTH	see INPUT
KSPLN	see INPUT
L	integer iteration counter
ME	Mach number external to boundary layer, M_e
MUBAR	dynamic viscosity based on reference temperature $\overline{T}$, $\overline{\mu}$, (lbf)(sec)/ft^2; (N)(sec)/m^2
MUSLE(MUSLM)	dynamic viscosity in English (metric) units at sea-level conditions, μ_{sl}, (lbf)(sec)/ft^2; (N)(sec)/m^2
MUSZ(MUTZ)	dynamic viscosity based on static (total) temperature at station 0, $\mu_0(\mu_0')$, (lbf)(sec)/ft^2; (N)(sec)/m^2
NS	integer station number used in call on SIMPS1 in LAMNAR
NST	see INPUT
NTAB	counter on STAB, CTAB, XTAB, and YTAB tables

NTURB	see INPUT
NURW	$\mathrm{Nu}_x / \sqrt{R_w}$
NUSS	local Nusselt number, Nu_x
NUSZ(NUTZ)	kinematic viscosity based on static (total) temperature at station 0, $\nu_0(\nu_0')$, $\mathrm{ft}^2/\mathrm{sec}$; $\mathrm{m}^2/\mathrm{sec}$
NUW	kinematic viscosity at wall, ν_w, $\mathrm{ft}^2/\mathrm{sec}$; $\mathrm{m}^2/\mathrm{sec}$
NV	number of coupled differential equations being solved by RUNKUT routine - two in this program
NVP	see INPUT
NVP1	NVP + 1
POPTZ	see INPUT
POWER	reciprocal of power on power-law turbulent velocity profiles, n_p
PR	Prandtl number, Pr
PRES	see INPUT
PSZ	static pressure at station 0, P_0, $\mathrm{lbf/ft}^2$; $\mathrm{N/m}^2$
PTZ	see INPUT
R	see INPUT
RCRIT	critical incompressible momentum-thickness Reynolds number, $\left(R_\theta\right)_{i,cr}$
RHSE	static density based on temperature external to boundary layer, ρ_e, $\mathrm{slug/ft}^3$; $\mathrm{kg/m}^3$
RHSW	static density based on wall temperature, ρ_w, $\mathrm{slug/ft}^3$; $\mathrm{kg/m}^3$
RHSZ(RHTZ)	static (total) density based on static (total) temperature at station 0, $\rho_0(\rho_0')$, $\mathrm{slug/ft}^3$; $\mathrm{kg/m}^3$
RINS	incompressible momentum-thickness Reynolds number at instability point, $\left(R_\theta\right)_{i,inst}$
RTH	momentum-thickness Reynolds number, R_θ
RTHI	incompressible momentum-thickness Reynolds number, $\left(R_\theta\right)_i$
RTRAN	incompressible momentum-thickness Reynolds number used in checking for transition point, $\left(R_\theta\right)_{i,tran}$

RUK	two-dimensional array of parameters used in Runge-Kutta scheme in RUNKUT
RW	Reynolds number at wall, $R_w = u_e x/\nu_w$
RX, RY	temporary values of independent and dependent variables in RUNKUT routine
S	distance along surface from station 1, x, ft; m
SDER	second derivatives generated in call on SPLINE
SEPRN	logical variable indicating whether separation has occurred
SHAPK	dimensionless shape factor based on momentum thickness, K
SHAPL	Pohlhausen shape factor based on boundary-layer thickness, λ
SHEAR	shear parameter, l, in LAMNAR
SL	derivative of FUNCT in call on ROOT in LAMNAR
SOL	ratio of surface distance to total arc length, x/L
SOL1,SOL2	limits of integration for calls on SIMPS1 in LAMNAR
SS	distance along surface from station 1, x
SSDEL	SS + DEL
STAB	table of surface distances x obtained in solving laminar differential equation
SUTHL	value of coefficient k_{su} in Sutherland's viscosity temperature formula
SW	temperature function at wall, $(T_w/T'_0) - 1$
TAUW	shear stress at wall, τ_w, lbf/ft^2; N/m^2
TAWL	adiabatic wall temperature T_{aw} based on laminar recovery factor $Pr^{1/2}$, ^oR; K
TAWT	adiabatic wall temperature T_{aw} based on turbulent recovery factor $Pr^{1/3}$, ^oR; K
TBAR	reference temperature, $\overline{T}$, ^oR; K
TC	thermal conductivity, k, (ft)(lbf)/(ft)(sec)(^oR); J/(m)(sec)(K)
TCON	temperature constant in Sutherland's temperature-viscosity formula
TCSLE(TCSLM)	thermal conductivity in English (metric) units at sea-level conditions, k_{sl}, (ft)(lbf)/(ft)(sec)(^oR); J/(m)(sec)(K)

TEM, TEM1, TEM2, TEM3, TEM4, TEM5, TEM6, TEM7,	temporary variables
THET	momentum thickness, θ, ft; m
THETTR	transformed momentum thickness, θ_{tr}
TLAM	see INPUT
TR1, TR2	temperature ratios in PRECAL
TRANS	logical variable indicating whether transition has occurred
TSE	static temperature external to boundary layer, T_e, ^oR; K
TSLE(TSLM)	total temperature in English (metric) units at sea-level conditions, $T_{s l}$, ^oR; K
TSZ	static temperature at station 0, T_0, ^oR; K
TTURB	see INPUT
TTZ	see INPUT
TWAL	see INPUT
U	velocity within boundary layer, ft/sec; m/sec
UE	see INPUT
UPMACH	relative Mach number at station 0, M_0
UUE	u/u_e
UZ	relative velocity at station 0, u_0, ft/sec; m/sec
VOVCR	see INPUT
X	see INPUT
XOM	ratio of X to maximum X-coordinate
XTAB	table of surface distances x obtained in solving turbulent differential equations
XX	distance along surface from station 1, x
Y	see INPUT
YINC	dependent variable increments in Runge-Kutta scheme in RUNKUT
YOM	ratio of Y to maximum X-coordinate

YP	distance normal to surface in y-direction in boundary-layer profile, ft; m
YTAB1	table of values of f computed by solving turbulent differential equations
YTAB2	table of values of H_i computed by solving turbulent differential equations
YXMAX	ratio of YP to maximum X-coordinate
YY	dependent variables in solution of turbulent differential equations in RUNKUT

AUXILIARY SUBROUTINES

The following general subroutines are called by the main subroutines, but do not use the COMMON variables. Each has its own variables, which are defined below instead of in the main dictionary.

Subroutine SMOOTH

Subroutine SMOOTH is a simple data-smoothing routine. It smoothes an array y of length n by the following formula

$$y_i = \frac{y_{i-1} + 2y_i + y_{i+1}}{4} \qquad i = 2, n - 1 \tag{43}$$

If $K = 1$, the end points are also smoothed as follows:

$$y_1 = \frac{y_1 + 2y_2 - y_3}{2} \tag{44}$$

$$y_n = \frac{y_n + 2y_{n-1} - y_{n-2}}{2} \tag{45}$$

but this is not done in the BLAYER program.

The input variables for SMOOTH are as follows:

Y array y to be smoothed

N number of elements in array Y

K counter indicating whether end points are to be smoothed

The output variable of SMOOTH is

Y array y after smoothing

Subroutine GRADNT

Subroutine GRADNT computes the gradient of f(x) with respect to x using difference techniques. Referring to figure 9, the slope of the function at any point $f_i'(x)$ is computed by weighting slopes between adjacent points

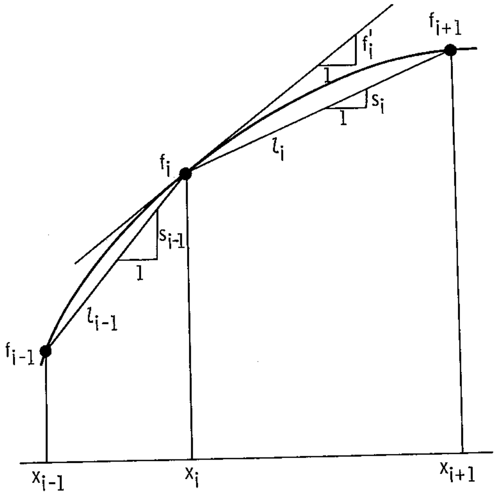

Figure 9. - Parameters used in calculation of gradients at input points.

$$f_i'(x) = \frac{s_i l_{i-1} + s_{i-1} l_i}{l_{i-1} + l_i} \tag{46}$$

where

$$s_i = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} \tag{47}$$

and

$$l_i = \sqrt{\left(f_{i+1} - f_i\right)^2 + \left(x_{i+1} - x_i\right)^2} \qquad (48)$$

Slopes at the end points are given by

$$f_1'(x) = s_1 + \frac{l_1}{l_1 + l_2}\left(s_1 - s_2\right) \qquad (49)$$

$$f_n'(x) = s_{n-1} + \frac{l_{n-1}}{l_{n-1} + l_{n-2}}\left(s_{n-1} - s_{n-2}\right) \qquad (50)$$

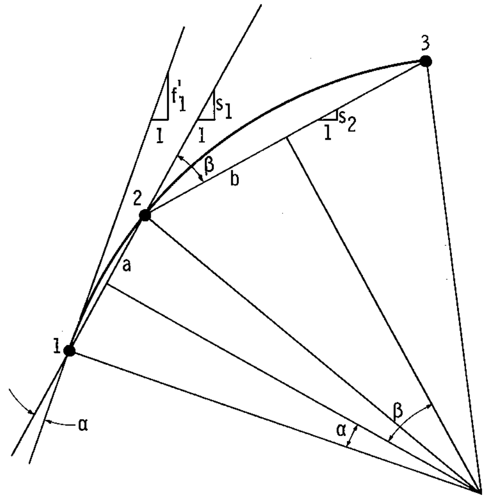

Figure 10. - Parameters used in calculation of gradients at end points.

Equations (49) and (50) can be derived from figure 10 where

$$f_1' = s_1 + \text{Change in slope due to angle } \alpha$$

Assuming a circle

$$\frac{\alpha}{\beta} = \frac{\text{arc } \alpha}{\text{arc } \beta} \approx \frac{a}{a + b} \qquad (51)$$

then,

$$\text{Change in slope in } \alpha = \frac{a}{a+b} (\text{Change in slope in } \beta) = \frac{a}{a+b} (s_1 - s_2) \qquad (52)$$

The input variables for GRADNT are as follows:

X	independent variable, x
FX	f(x)
N	number of elements in array FX

The output variable of GRADNT is

DFDX	gradient of f(x) with respect to x, f'(x)

Internal variables for GRADNT are as follows:

SL	slopes between adjacent points, s
DIST	distances between adjacent points, l

Subroutine SPLINE

Subroutine SPLINE is based on the cubic spline curve. It solves the tridiagonal matrix equation given in reference 17 to obtain coefficients for the piecewise cubic polynomial function giving the spline fit curve. SPLINE uses the end condition that the second derivative at either end point is equal to that at the adjacent spline point.

The input variables for SPLINE are as follows:

X	array of x-coordinates
Y	array of function values corresponding to X
N	number of elements in X or Y array

The output variables of SPLINE are as follows:

DYDX	array of first derivatives of Y with respect to X at spline points
D2YDX2	array of second derivatives of Y with respect to X at spline points

Subroutine ROOT

Subroutine ROOT locates a root for f(x) = y in the interval a, b (see fig. 11). The values of f(x) are calculated by another subroutine FUNCT. ROOT divides the interval

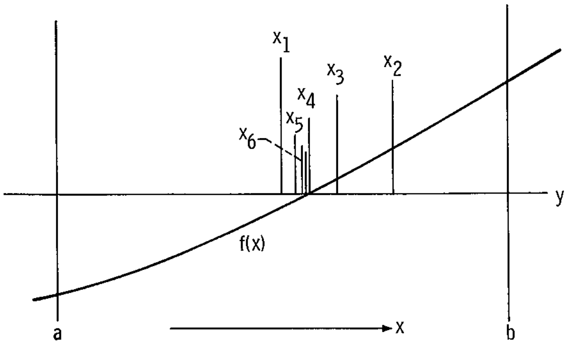

Figure 11. - Division of interval for location of a root.

a, b in half and checks which half contains the root. It then divides that half and checks again. By repeating this procedure 20 times the root is located to within $1/(2^{20})$ of the original interval.

ROOT contains an error message - ROOT HAS FAILED TO CONVERGE IN THE GIVEN INTERVAL - which is given if a root cannot be located within the tolerance (TOLERY) in 20 iterations. This should never occur in the BLAYER program.

The input variables for ROOT are as follows:

A	a (see fig. 11)
B	b (see fig. 11)
Y	y (see fig. 11)
FUNCT	external subroutine to calculate f(x)
TOLERY	tolerance on solution (x is accepted if $\lvert f(x) - y \rvert < \text{TOLERY}$)

The output arguments for ROOT are as follows:

X	value of x such that f(x) = y
DFX	derivative of f(x) with respect to x at the point f(x) = y

Internal variables for ROOT are as follows:

X1,X2	left and right boundaries of interval in which root is located
FX,FX1	f(x)
INF	used to indicate infinite derivative, DFX 0 if finite 1 if infinite

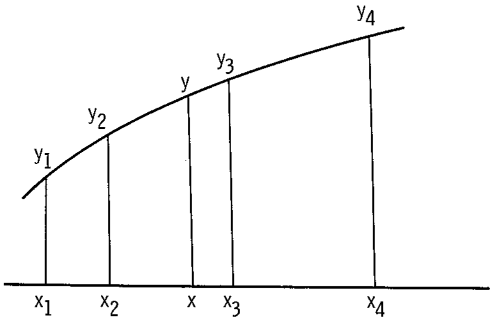

Figure 12. - Parameters used in Lagrangian interpolation.

Subroutine LGRNGE

Subroutine LGRNGE performs four-point interpolation using Lagrange's formula. Referring to figure 12, the formula is

$$y = f(x) = \frac{(x - x_2)(x - x_3)(x - x_4)}{(x_1 - x_2)(x_1 - x_3)(x_1 - x_4)} y_1 + \frac{(x - x_1)(x - x_3)(x - x_4)}{(x_2 - x_1)(x_2 - x_3)(x_2 - x_4)} y_2$$

$$+ \frac{(x - x_1)(x - x_2)(x - x_4)}{(x_3 - x_1)(x_3 - x_2)(x_3 - x_4)} y_3 + \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_4 - x_1)(x_4 - x_2)(x_4 - x_3)} y_4 \qquad (53$$

The input variables for LGRNGE are as follows:

X	array of values of independent variable x
Y	array of values of dependent variable y
N	number of elements in X or Y array
ARG	value of independent variable at which value of dependent variable is desired

The output variable for LGRNGE is

ANS	value of dependent variable corresponding to ARG

Subroutine CURVFT

Subroutine CURVFT evaluates the polynomial f(x,y) using the coefficients C. The general formula used is the following:

$$
\begin{array}{lllll}
f(x,y) = C_1 & + C_2 x & + C_3 x^2 & \ldots & + C_n x^{n-1} \\
+ C_{n+1} y & + C_{n+2} xy & + C_{n+3} x^2 y & \ldots & + C_{2n} x^{n-1} y \\
+ C_{2n+1} y^2 & + C_{2n+2} xy^2 & + C_{2n+3} x^2 y^2 & \ldots & + C_{3n} x^{n-1} y^2 \\
\vdots & \vdots & \vdots & & \vdots \\
+ C_{(m-1)n+1} y^{m-1} & + C_{(m-1)n+2} xy^{m-1} & + C_{(m-1)n+3} x^2 y^{m-1} & \ldots & + C_{mn} x^{n-1} y^{m-1}
\end{array}
\tag{54}
$$

This subroutine is also used for evaluating functions of one independent variable.

The input variables for CURVFT are as follows:

COEF	array of polynomial coefficients, C_1 to C_{mn}
X	independent variable x
Y	independent variable y
NX	n - 1
NY	m - 1

The output variable for CURVFT is

ANS	dependent variable, f(x,y)

Function SIMPS1

Function SIMPS1 performs numerical integration of a function of one variable between specified limits. A modification of Simpson's rule is used in which more intervals of smaller widths are placed in regions requiring greater accuracy. The interval specified by the limits of integration may be subdivided into as many as 200 subintervals in order to meet the criterion for accuracy built into the routine. The local values of the integrand are calculated by an external subroutine FUNC. In BLAYER, SIMP1 is used twice, with functions INT1 and INT2 used for FUNC.

The input variables for SIMPS1 are as follows:

X1,X2	limits of integration
FUNC	integrand, an external function

The output variables for SIMPS1 are as follows:

SIMPS1	value of the integral
KSIG	error parameter set to 1 if SIMPS1 is unable to achieve a value of the integral which will pass its tests for accuracy

COMPLETE PROGRAM LISTING

```
$IBJOB
$IBFTC BLAYER

      COMMON/C9/ERROR,TRANS,SEPRN
      LOGICAL ERROR,TRANS,SEPRN
   10 CALL INPUT
      IF (ERROR) GO TO 10
      CALL PRECAL
      IF(ERROR) GO TO 10
      CALL LAMNAR
      IF (ERROR) GO TO 10
      IF (SEPRN) GO TO 20
      IF (.NOT.TRANS) GO TO 20
      CALL TURBLN
      IF (ERROR) GO TO 10
   20 CALL PROFIL
      GO TO 10
      END
```

```
$IBFTC INPU

      SUBROUTINE INPUT
      COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTURB,KPVM,KEM,KSMTH,
     1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTURB,TTURB,KPRE,KGRAD,KSDE,KLAM,
     2KMAIN,KPROF,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
     3VOVCR(100),TWAL(100)
      COMMON/C9/ERROR,TRANS,SEPRN
      LOGICAL ERROR,TRANS,SEPRN
      REAL ME
      ERROR= .FALSE.
      TRANS= .FALSE.
```

```
      SEPRN= .FALSE.
      WRITE(6,1000)
      READ (5,1050)
      WRITE(6,1050)
      READ (5,1020) GAM,R,PTZ,TTZ,UPMACH
      WRITE(6,1060) GAM,R,PTZ,TTZ,UPMACH
      READ (5,1040) NST,NVP,NTURB,KPVM,KEM,KSMTH,KSPLN,KLE,KATCH,CTHET
      WRITE(6,1070) NST,NVP,NTURB,KPVM,KEM,KSMTH,KSPLN,KLE,KATCH,CTHET
      READ (5,1020) DLAM,TLAM,DTURB,TTURB
      WRITE(6,1080) DLAM,TLAM,DTURB,TTURB
      READ (5,1010) KPRE,KGRAD,KSDE,KLAM,KMAIN,KPROF
      WRITE(6,1090) KPRE,KGRAD,KSDE,KLAM,KMAIN,KPROF
      IF(NST.GT.100.OR.NTURB.GT.NST.OR.KEM.LT.0.OR.KEM.GT.1.OR.KSPLN.LT.
     10.OR.KSPLN.GT.1.OR.KLE.LT.0.OR.KLE.GT.1.OR.KATCH.LT.0.OR.KATCH.GT.
     21) GO TO 70
      IF(KPVM.EQ.1) GO TO 10
      IF(KPVM.EQ.2) GO TO 20
      IF(KPVM.EQ.3) GO TO 30
      IF(KPVM.EQ.4) GO TO 40
      IF(KPVM.EQ.5) GO TO 50
      GO TO 70
   10 READ (5,1030) (X(I),Y(I),PRES(I),TWAL(I),I=1,NST)
      WRITE(6,1100) (X(I),Y(I),PRES(I),TWAL(I),I=1,NST)
      GO TO 60
   20 READ (5,1030) (X(I),Y(I),UE(I),TWAL(I),I=1,NST)
      WRITE(6,1110) (X(I),Y(I),UE(I),TWAL(I),I=1,NST)
      GO TO 60
   30 READ (5,1030) (X(I),Y(I),ME(I),TWAL(I),I=1,NST)
      WRITE(6,1120) (X(I),Y(I),ME(I),TWAL(I),I=1,NST)
      GO TO 60
   40 READ (5,1030) (X(I),Y(I),POPTZ(I),TWAL(I),I=1,NST)
      WRITE(6,1130) (X(I),Y(I),POPTZ(I),TWAL(I),I=1,NST)
      GO TO 60
   50 READ (5,1030) (X(I),Y(I),VOVCR(I),TWAL(I),I=1,NST)
      WRITE(6,1140) (X(I),Y(I),VOVCR(I),TWAL(I),I=1,NST)
   60 IF (KEM.EQ.0) WRITE(6,1150)
      IF (KEM.EQ.1) WRITE(6,1160)
      RETURN
   70 ERROR = .TRUE.
      WRITE(6,1170)
      RETURN
 1000 FORMAT(1H1////)
 1010 FORMAT(16I5)
 1020 FORMAT(8F10.5)
 1030 FORMAT(4F10.5)
 1040 FORMAT(9I5,F10.5)
 1050 FORMAT(80H
     1                              )
 1060 FORMAT(//6X,3HGAM,9X,1HR,11X,3HPTZ,8X,3HTTZ,8X,6HUPMACH/3X,F7.3,3X
     1,F9.2,3X,F10.2,2X,F9.2,4X,F8.4)
 1070 FORMAT(/6X,3HNST,8X,3HNVP,9X,5HNTURB,7X,4HKPVM,8X,3HKEM,3X,5HKSMTH
     1,7X,5HKSPLN,8X,3HKLE,8X,5HKATCH,6X,5HCTHET/5X,I3,8X,I3,1X,I3,9X,I
     22,9X,I2,10X,I2,10X,I2,10X,I2,10X,I2,8X,F7.4)
 1080 FORMAT(/6X,4HDLAM,7X,4HTLAM,8X,5HDTURB,7X,5HTTURB/4X,F10.6,1X,F10.
     16,2X,F10.6,2X,F10.6)
 1090 FORMAT(/6X,4HKPRE,7X,5HKGRAD,7X,4HKSDE,8X,4HKLAM,7X,5HKMAIN,7X,5HK
     1PROF/7X,I2,9X,I2,10X,I2,10X,I2,9X,I2,10X,I2)
 1100 FORMAT(/9X,1HX,11X,1HY,11X,4HPRES,10X,4HTWAL/(3X,F10.5,2X,F10.5,3X
     1,F12.5,3X,F10.4))
```

```
1110 FORMAT(/9X,1HX,11X,1HY,12X,2HUE,11X,4HTWAL/(3X,F10.5,2X,F10.5,3X,F
    112.5,3X,F10.4))
1120 FORMAT(/9X,1HX,11X,1HY,12X,2HME,11X,4HTWAL/(3X,F10.5,2X,F10.5,4X,F
    110.6,4X,F10.4))
1130 FORMAT(/9X,1HX,11X,1HY,10X,5HPOPTZ,10X,4HTWAL/(3X,F10.5,2X,F10.5,4
    1X,F10.6,4X,F10.4))
1140 FORMAT(/9X,1HX,11X,1HY,10X,5HVOVCR,10X,4HTWAL/(3X,F10.5,2X,F10.5,4
    1X,F10.6,4X,F10.4))
1150 FORMAT(//6X,40HENGLISH UNITS USED FOR INPUT AND OUTPUT./6X,69HPOUN
    1DS FORCE, SLUGS, FEET, SECONDS, DEGREES RANKINE, AND FOOT-POUNDS.)
1160 FORMAT(//6X,39HMETRIC UNITS USED FOR INPUT AND OUTPUT./6X,64HNEWTO
    1NS, KILOGRAMS, METERS, SECONDS, DEGREES KELVIN, AND JOULES.)
1170 FORMAT(////10X,48HERROR IN INPUT DATA.  RECHECK INPUT INSTRUCTIONS
    1)
     END
```

```
$IBFTC PRECA

      SUBROUTINE PRECAL
      COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTURB,KPVM,KEM,KSMTH,
     1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTURB,TTURB,KPRE,KGRAD,KSDE,KLAM,
     2KMAIN,KPROF,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
     3VOVCR(100),TWAL(100)
      COMMON/C2/PSZ,TSZ,UZ,ASZ,ATZ,RHSZ,RHTZ,MUSZ,MUTZ,NUSZ,NUTZ,CP,
     1PR,TC,ARCL
      COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
     1TAWL(100),TAWT(100),TBAR(100),RW(100),SW(100),SUTHL(100),
     2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
     3AA(100),3B(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
      COMMON/C9/ERROR,TRANS,SEPRN
      DIMENSION SDER(100),CMU(20),CPR(20),CTC(20)
      REAL MUSZ,MUTZ,NUSZ,NUTZ,MUSLE,MUSLM,ME,NUW,MUBAR
      LOGICAL ERROR,TRANS,SEPRN
C
C  READ DATA FOR MU, PR, AND TC CURVE FITS
C
      DATA(CMU(I),I=1,5)/-.01945170,1.3019531,-.34511323,
     1.068277826,-.00566593/
      DATA(CPR(I),I=1,5)/.8557,-.234136,.1078624,
     1-.0236214,.00202863/
      DATA(CTC(I),I=1,5)/-.03839323,1.2697427,-.30911252,
     1.08743781,-.009674725/
C
C  INITIALIZE STATIC AND TOTAL PARAMETERS
C
      TSLE= 518.688
      TSLM= 288.160
      MUSLE= 3.711402E-7
      MUSLM= 1.777029E-5
      TCSLE= 3.202206E-3
      TCSLM= 2.561796E-2
      TSZ= TTZ/(1.+(GAM-1.)/2.*UPMACH**2)
      PSZ= PTZ*(TSZ/TTZ)**(GAM/(GAM-1.))
      RHSZ= PSZ/R/TSZ
      RHTZ= PTZ/R/TTZ
```

```
      ASZ= SQRT(GAM*R*TSZ)
      ATZ= SQRT(GAM*R*TTZ)
      UZ= UPMACH*ASZ
      CP= R*GAM/(GAM-1.)
      IF (KEM.EQ.1) GO TO 10
      TCON= 198.60
      TR1= TSZ/TSLE
      TR2= TTZ/TSLE
      GO TO 20
   10 TCON= 110.33
      TR1= TSZ/TSLM
      TR2= TTZ/TSLM
   20 CALL CURVFT(CPR,PR,TR1,0,4,0)
      CALL CURVFT(CTC,TC,TR1,0,4,0)
      CALL CURVFT(CMU,MUSZ,TR1,0,4,0)
      CALL CURVFT(CMU,MUTZ,TR2,0,4,0)
      IF (KEM.EQ.1) GO TO 30
      TC= TC*TCSLE
      MUSZ= MUSZ*MUSLE
      MUTZ= MUTZ*MUSLE
      GO TO 40
   30 TC= TC*TCSLM
      MUSZ= MUSZ*MUSLM
      MUTZ= MUTZ*MUSLM
   40 NUSZ= MUSZ/RHSZ
      NUTZ= MUTZ/RHTZ
C
C  CALCULATE GEOMETRY RATIOS AND ARC LENGTHS
C
      XOM(1)= X(1)/X(NST)
      YOM(1)= Y(1)/X(NST)
      S(1)= 0.
      DO 50 I=2,NST
      XOM(I)= X(I)/X(NST)
      YOM(I)= Y(I)/X(NST)
   50 S(I)= S(I-1)+SQRT((X(I)-X(I-1))**2+(Y(I)-Y(I-1))**2)
      ARCL= S(NST)
      DO 60 I=1,NST
   60 SOL(I)= S(I)/ARCL
C
C  CALCULATE PRES,UE,ME,POPTZ,AND VOVCR AT EACH STATION
C
      KDONE=0
      IF(KPVM.EQ.1) GO TO 70
      IF(KPVM.EQ.2) GO TO 90
      IF(KPVM.EQ.3) GO TO 110
      IF(KPVM.EQ.4) GO TO 130
      IF(KPVM.EQ.5) GO TO 150
C  PRESSURE GIVEN AS INPUT
   70 DO 80 I=1,NST
      IF(PRES(I).LT.0..OR.PRES(I).GT.PTZ) GO TO 290
      UE(I)= SQRT(2.*GAM/(GAM-1.)*PTZ/RHTZ*(1.-(PRES(I)/PTZ)**((GAM-1.)/
     1GAM)))
      TSE(I)= TTZ-UE(I)**2/(2.*CP)
      AE(I)= SQRT(GAM*R*TSE(I))
      ME(I)= UE(I)/AE(I)
      POPTZ(I)= PRES(I)/PTZ
   80 VOVCR(I)= SQRT((GAM+1.)/(GAM-1.)*(1.-PRES(I)/PTZ)**((GAM-1.)/GAM))
      GO TO 170
```

```
C  VELOCITY GIVEN AS INPUT
  90 DO 100 I=1,NST
      IF(UE(I).LT.0.) GO TO 290
      PRES(I)= PTZ*(1.-((GAM-1.)*RHTZ*UE(I)**2)/(2.*GAM*PTZ))**(GAM/(GAM
     1-1.))
      TSE(I)= TTZ-UE(I)**2/(2.*CP)
      AE(I)= SQRT(GAM*R*TSE(I))
      ME(I)= UE(I)/AE(I)
      POPTZ(I)= PRES(I)/PTZ
 100 VOVCR(I)= SQRT((GAM+1.)/(GAM-1.)*(1.-PRES(I)/PTZ)**((GAM-1.)/GAM))
      IF(KDONE.GT.0) GO TO 190
      GO TO 170
C  MACH NUMBER GIVEN AS INPUT
 110 DO 120 I=1,NST
      IF(ME(I).LT.0.) GO TO 290
      TSE(I)= TTZ/(1.+(GAM-1.)/2.*ME(I)**2)
      AE(I)= SQRT(GAM*R*TSE(I))
      UE(I)= ME(I)*AE(I)
      PRES(I)= PTZ*(1.-((GAM-1.)*RHTZ*UE(I)**2)/(2.*GAM*PTZ))**(GAM/(GAM
     1-1.))
      POPTZ(I)= PRES(I)/PTZ
 120 VOVCR(I)= SQRT((GAM+1.)/(GAM-1.)*(1.-PRES(I)/PTZ)**((GAM-1.)/GAM))
      GO TO 170
C  PRESSURE OVER TOTAL PRESSURE GIVEN AS INPUT
 130 DO 140 I=1,NST
      IF(POPTZ(I).LT.0..OR.POPTZ(I).GT.1.)GO TO 290
      PRES(I)= POPTZ(I)*PTZ
      UE(I)= SQRT(2.*GAM/(GAM-1.)*PTZ/RHTZ*(1.-(PRES(I)/PTZ)**((GAM-1.)/
     1GAM)))
      TSE(I)= TTZ-UE(I)**2/(2.*CP)
      AE(I)= SQRT(GAM*R*TSE(I))
      ME(I)= UE(I)/AE(I)
 140 VOVCR(I)= SQRT((GAM+1.)/(GAM-1.)*(1.-PRES(I)/PTZ)**((GAM-1.)/GAM))
      GO TO 170
C  VELOCITY OVER CRITICAL VELOCITY GIVEN AS INPUT
 150 DO 160 I=1,NST
      IF(VOVCR(I).LT.0..OR.VOVCR(I).GT.SQRT((GAM+1.)/(GAM-1.)))GO TO 290
      PRES(I)= PTZ*(1.-((GAM-1.)/(GAM+1.)*VOVCR(I)**2)**(GAM/(GAM-1.)))
      UE(I)= SQRT(2.*GAM/(GAM-1.)*PTZ/RHTZ*(1.-(PRES(I)/PTZ)**((GAM-1.)/
     1GAM)))
      TSE(I)= TTZ-UE(I)**2/(2.*CP)
      AE(I)= SQRT(GAM*R*TSE(I))
      ME(I)= UE(I)/AE(I)
 160 POPTZ(I)= PRES(I)/PTZ
C
C  PRINT INITIAL CALCULATED PARAMETERS
C
 170 WRITE(6,1000)
      WRITE(6,1010) PSZ,TSZ,UZ,ASZ,ATZ,RHSZ,RHTZ,MUSZ,MUTZ,NUSZ,NUTZ,CP,
     1PR,TC,ARCL
      WRITE(6,1020) (I,PRES(I),UE(I),ME(I),POPTZ(I),VOVCR(I),I=1,NST)
C
C  SMOOTH INPUT DATA IF NECESSARY
C
      IF (KSMTH.LT.1) GO TO 200
 180 CALL SMOOTH(UE,NST,0)
      KDONE=KDONE+1
C     RECOMPUTE PRES,ME,POPTZ, AND VOVCR AT EACH STATION
      GO TO 90
```

```
  190 WRITE(6,1040)
      WRITE(6,1020) (I,PRES(I),UE(I),ME(I),POPTZ(I),VOVCR(I),I=1,NST)
      IF(KDONE.LT.KSMTH) GO TO 180
C
C  PRINT GEOMETRY PARAMETERS
C
  200 IF (KPRE.NE.1) GO TO 210
      WRITE(6,1030) (I,X(I),Y(I),S(I),XOM(I),YOM(I),SOL(I),I=1,NST)
C
C  CALCULATE OTHER NECESSARY PARAMETERS AT EACH STATION
C
  210 DO 220 I=1,NST
      TEM1= 1.+.5*(GAM-1.)*ME(I)**2
      RHSW(I)= PRES(I)/R/TWAL(I)
      RHSE(I)= PRES(I)/R/TSE(I)
      HEADW(I)= .5*RHSW(I)*UE(I)**2
      HEADE(I)= .5*RHSE(I)*UE(I)**2
      SW(I)= TWAL(I)/TTZ-1.
      SUTHL(I)= SQRT(TWAL(I)/TTZ)*(TTZ+TCON)/(TWAL(I)+TCON)
      NUW(I)= SUTHL(I)*NUTZ*(1.+SW(I))**2*TEM1**(GAM/(GAM-1.))
      RW(I)= UE(I)*S(I)/NUW(I)
      TAWL(I)= TSE(I)*(1.+PR**(1./2.)*(TEM1-1.))
      TAWT(I)= TSE(I)*(1.+PR**(1./3.)*(TEM1-1.))
      TBAR(I)= .5*(TWAL(I)+TSE(I))+.22*PR**(1./3.)*(TTZ-TSE(I))
      MUBAR(I)= MUTZ*SUTHL(I)*TBAR(I)/TTZ
      BB(I)= ME(I)*ATZ/NUTZ*(TSE(I)/TTZ)**((GAM+1.)/(2.*GAM-2.))
      AA(I)= BB(I)*TSE(I)/TBAR(I)*(MUBAR(I)/MUTZ)**.268
      FF(I)=1.+.1750*ME(I)**2+.60*SW(I)+.4217*SW(I)*ME(I)**2+.0088*ME(I)
     1**4+.0603*SW(I)*ME(I)**4+.1825*SW(I)**2+.0735*SW(I)**2*ME(I)**2
     2+.0073*SW(I)**2*ME(I)**4
  220 CONTINUE
C
C  COMPUTE VELOCITY AND MACH NUMBER GRADIENTS ALONG THE SURFACE
C
C  FINITE DIFFERENCE TECHNIQUE
      IF(KSPLN.EQ.1) GO TO 230
      CALL GRADNT(S,UE,NST,DUDS)
      CALL GRADNT(S,ME,NST,DMDS)
      GO TO 240
C  SPLINE CURVE TECHNIQUE
  230 CALL SPLINE(S,UE,NST,DUDS,SDER)
      CALL SPLINE(S,ME,NST,DMDS,SDER)
  240 DO 250 I=1,NST
  250 DMDL(I)= ARCL*DMDS(I)
C
C  PRINT OTHER CALCULATED PARAMETERS
C
      IF(KPRE.NE.1) GO TO 260
      WRITE(6,1050) (I,AE(I),TSE(I),TWAL(I),TAWL(I),TAWT(I),TBAR(I),
     1I=1,NST)
      WRITE(6,1060) (I,RW(I),SW(I),SUTHL(I),RHSW(I),RHSE(I),HEADW(I),
     1HEADE(I),NUW(I),MUBAR(I),I=1,NST)
  260 IF(KGRAD.NE.1) GO TO 270
      WRITE(6,1070)
      WRITE(6,1080) (I,DUDS(I),DMDS(I),DMDL(I),I=1,NST)
C
C  CHECK FOR IMPROPER INPUT
C
```

```
  270 DO 280 I=2,NST
      IF (UE(I).NE.0.) GO TO 280
      ERROR= .TRUE.
      WRITE(6,1090)
      RETURN
  280 CONTINUE
      RETURN
  290 ERROR= .TRUE.
      WRITE(6,1100)
      RETURN
C
C  FORMAT STATEMENTS
C
 1000 FORMAT(1H1///4X,24HPRELIMINARY CALCULATIONS///)
 1010 FORMAT(5X,10HPSZ       =  F12.5/5X,10HTSZ       =  F10.4/5X,10HUZ        =
     1 F11.5//5X,10HASZ       =  F11.4/5X,10HATZ       =  F11.4//5X,10HRHSZ
     2=  G15.7/5X,10HRHTZ      =  G15.7//5X,10HMUSZ      =  G15.7/5X,10HMUTZ
     3 =  G15.7//5X,10HNUSZ      =  G15.7/5X,10HNUTZ      =  G15.7//5X,10HCP
     4    =  F11.4/5X,10HPR        =  F9.5/5X,10HTC        =  G15.7/5X,10HARCL
     5 =  F8.4///)
 1020 FORMAT(/1X,7HSTATION,7X,4HPRES,13X,2HUE,12X,2HME,11X,5HPOPTZ,9X,5H
     1VOVCR/(2X,I3,5X,F12.5,3X,F12.5,4X,F10.6,4X,F10.6,4X,F10.5))
 1030 FORMAT(///1X,7HSTATION,7X,1HX,12X,1HY,12X,1HS,12X,3HXOM,9X,3HYOM,
     19X,3HSOL/(2X,I3,3X,F12.5,1X,F12.5,1X,F12.5,4X,F9.5,3X,F9.5,3X,F9.5
     2))
 1040 FORMAT(///1X,64HSMOOTHED SURFACE DISTRIBUTIONS OF PRES, UE, ME, PO
     1PTZ, AND VOVCR)
 1050 FORMAT(///1X,7HSTATION,5X,2HAE,10X,3HTSE,9X,4HTWAL,8X,4HTAWL,8X,4H
     1TAWT,8X,4HTBAR/(2X,I3,4X,F9.3,5(4X,F8.3)))
 1060 FORMAT(///1X,7HSTATION,11X,2HRW,6X,2HSW,4X,5HSUTHL,7X,4HRHSW,12X,4
     1HRHSE,8X,5HHEADW,4X,5HHEADE,9X,3HNUW,12X,5HMUBAR/(2X,I3,3X,F16.1,2
     2X,F4.1,1X,F7.3,2X,G14.6,2X,G14.6,1X,F8.3,1X,F8.3,2X,G14.5,2X,G14.6
     3))
 1070 FORMAT(1H1///21X,17HSURFACE GRADIENTS///)
 1080 FORMAT(1X,7HSTATION,13X,4HDUDS,15X,4HDMDS,15X,4HDMDL/(2X,I3,4X,F13
     1.6,1X,F18.6,1X,F18.6))
 1090 FORMAT(/////10X,83HTHERE IS A STAGNATION POINT AT A STATION OTHER
     1THAN STATION 1.  THIS IS NOT ALLOWED)
 1100 FORMAT(/////10X,111HAN INPUT PRESSURE,VELOCITY, OR MACH NUMBER IS
     1EITHER LESS THAN ZERO OR GREATER THAN ITS MAXIMUM ALLOWABLE VALUE)
      END
```

```
$IBFTC LAMNA

      SUBROUTINE LAMNAR
      COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTURB,KPVM,KEM,KSMTH,
     1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTURB,TTURB,KPRE,KGRAD,KSDE,KLAM,
     2KMAIN,KPROF,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
     3VOVCR(100),TWAL(100)
      COMMON/C2/PSZ,TSZ,UZ,ASZ,ATZ,RHSZ,RHTZ,MUSZ,MUTZ,NUSZ,NUTZ,CP,
     1PR,TC,ARCL
```

```
      COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
     1TAWL(100),TAWT(100),TBAR(100),RW(100),SW(100),SUTHL(100),
     2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
     3AA(100),BB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
      COMMON/C4/THET(100),DELSR(100),DELTA(100),FORM(100),
     1FORMI(100),FORMTR(100),RTH(100),RTHI(100),CF(100),
     1TAUW(100),NUSS(100),DTDY(100),HTRAN(100),CRN(100)
      COMMON/C5/SHAPL(100),SHAPK(100),B,NS
      COMMON/C6/FTRAN,FORMS
      COMMON/C7/INST,ITRAN,ISEP
      COMMON/C9/ERROR,TRANS,SEPRN
      DIMENSION CORLN(100),CORML(100),SHEAR(100),DTH(100)
      DIMENSION CCN(20),CRCR(20),CDIF(20),CSHR(20),CCRN(20),CDTH(20)
      DIMENSION STAB(505),CTAB1(505),CTAB2(505)
      REAL MUSZ,NUSZ,MUTZ,NUTZ,ME,NUW,MUBAR,NUSS,NURW,KBAR,INT1,INT2
      LOGICAL ERROR,TRANS,SEPRN
      EXTERNAL FUNCT,INT1,INT2
C
C  READ DATA FOR CORLN(1), RCRIT, DIFF, SHEAR, CRN, AND DTH CURVE FITS
C
      DATA(CCN(I),I=1,6)/-.08178,.06670,-.03143,
     1.00873,.01657,-.01052/
      DATA(CRCR(I),I=1,6)/5.47073,43.6053,227.198,
     1-2067.04,-27172.7,13691.2/
      DATA(CDIF(I),I=1,6)/903.785,26365.0,3.85695E+5,
     11.11044E+6,-4.53853E+7,-7.70276E+7/
      DATA(CSHR(I),I=1,16)/.224488,-1.91539,-9.894,-68.13488,
     1-.001512,-1.4768,-10.52925,-152.2781,-.002406,-.015629,
     1-1.45743,-126.23395,.000752,.005385,.917838,-39.40644/
      DATA(CCRN(I),I=1,16)/2.02056,-19.7211,-24.0495,-1400.002,
     1-.050979,-10.88012,62.4419,-5081.76,-.014343,2.279845,
     1129.7008,-6257.848,.0270567,-1.677051,57.4397,-2552.256/
      DATA(CDTH(I),I=1,16)/8.02829,-4.30978,88.8244,36.4336,
     12.71101,-7.42259,242.293,-16.293, -.16394,-7.61942,285.9795,
     164.11186,-.16758,-3.70289,130.8107,111.3276/
C
C  INITIALIZE PARAMETERS
C
      INST = 0
      ITRAN = 0
      ISEP = 0
      CF(1)= 0.
      TAUW(1)= 0.
      NUSS(1)= 0.
      DTDY(1)= 0.
      HTRAN(1)= 0.
      CRN(1)= 0.
      RTRAN= 0.
C
C  CHECK CONSISTENCY OF INITIAL VALUES
C
      IF (DLAM.GE.0..AND.TLAM.GE.0..AND.DTURB.GE.0..AND.TTURB.GE.0.)
     1GO TO 10
      ERROR = .TRUE.
      WRITE(6,1000)
      RETURN
   10 IF (NTURB.NE.1) GO TO 30
      ITRAN = 1
      IF (DTURB.GT.0..AND.TTURB.GT.0.) GO TO 20
```

```
      ERROR = .TRUE.
      WRITE(6,1010)
      RETURN
   20 IF (UE(1).GT.0.) GO TO 240
      ERROR = .TRUE.
      WRITE(6,1020)
      RETURN
C
C  BEGIN CALCULATION IN LAMINAR REGION - CHECK FOR INITIAL VALUES
C  CALCULATE INITIAL CORRELATION NUMBER
C
   30 IF (DLAM.EQ.0..AND.TLAM.EQ.0.) GO TO 70
      IF (UE(1).GT.0.) GO TO 40
      ERROR = .TRUE.
      WRITE(6,1030)
      RETURN
   40 IF (TLAM.EQ.0.) GO TO 50
C  INITIAL MOMENTUM THICKNESS WAS GIVEN
      TEM1= 1.+.5*(GAM-1.)*ME(1)**2
      CORML(1)= -ATZ*TLAM**2/NUTZ/SUTHL(1)/ARCL/TEM1**((3.-GAM)/
     1(2.*GAM-2.))
      CORLN(1) = CORML(1)*DMDL(1)
      GO TO 90
C  INITIAL DISPLACEMENT THICKNESS WAS GIVEN
   50 IF (ABS(DMDL(1)).GE..0001) GO TO 60
      CORLN(1)= 0.
      TEM1= 1.+.5*(GAM-1.)*ME(1)**2
      FORM(1)= 2.38411*(1.+(2.79-1.78*PR**.5)*((1.+SW(1))*TEM1-1.))+(4.5
     15*PR**(1./3.)-3.65*PR*.5)*PR**.5*(TEM1-1.)
      THET(1)= DLAM/FORM(1)
      CORML(1)= -ATZ*THET(1)**2/NUTZ/SUTHL(1)/ARCL/TEM1**((3.-GAM)/(2.*G
     1AM-2.))
      GO TO 90
   60 IF (DMDL(1).GT.0.) CALL ROOT(-1.,0.,DLAM,FUNCT,.5E-5,CORLN(1),SL)
      IF (DMDL(1).LT.0.) CALL ROOT( 0.,.2,DLAM,FUNCT,.5E-5,CORLN(1),SL)
      CORML(1) = CORLN(1)/DMDL(1)
      GO TO 90
C
C  NO INITIAL LAMINAR VALUES GIVEN
C  CALCULATE INITIAL CORRELATION NUMBER
C
C  SHARP LEADING EDGE
   70 IF(KLE.NE.1.AND.ABS(DMDL(1)).GE..0001) GO TO 80
      CORLN(1)= 0.
      CORML(1)= 0.
      GO TO 90
C  STAGNATION POINT
   80 CALL CURVFT(CCN,CORLN(1),SW(1),0,5,0)
      CORML(1)= CORLN(1)/DMDL(1)
      IF (CORML(1).LT.0.) GO TO 90
      ERROR= .TRUE.
      WRITE(6,1040)
      RETURN
C
C  SOLVE LAMINAR DIFFERENTIAL EQUATION
C  CALCULATE CORRELATION NUMBERS ALONG THE SURFACE
C
   90 TEM1= 1.+.5*(GAM-1.)*ME(1)**2
      TEM2= (3.*GAM-1.)/(2.*GAM-2.)
      DEL= 0.002*ARCL
```

```
      SS= -DEL
      NTAB=1
      CTAB1(1)= CORLN(1)
      CTAB2(1)= CORML(1)
      STAB(1)= 0.
  100 SS= SS+DEL
      SSDEL = SS+DEL
      CALL LGRNGE(S,SW,NST,SS,ANS1)
      CALL LGRNGE(S,ME,NST,SS,ANS2)
      CALL LGRNGE(S,ME,NST,SSDEL,ANS3)
      CALL LGRNGE(S,DMDL,NST,SSDEL,ANS4)
      A1= 0.43631-0.00367*ANS1+0.00481*ANS1**2+0.00651*ANS1**3
      A2= 5.43220+2.25400*ANS1-0.06672*ANS1**2-0.20637*ANS1**3
      A3= 4.51903-10.49775*ANS1-12.71732*ANS1**2-2.95270*ANS1**3
      A4= 19.01831+62.76597*ANS1+115.00986*ANS1**2+62.53113*ANS1**3
      A= A1-A3*CTAB1(NTAB)**2-2.*A4*CTAB1(NTAB)**3
      B= A2+2.*A3*CTAB1(NTAB)+3.*A4*CTAB1(NTAB)**2
      K1= 0
      SOL1 = SS/ARCL
      SOL2 = SSDEL/ARCL
      TEM3 = SIMPS1(SOL1,SOL2,INT1,K1)
      IF (TEM3.EQ.0..OR.K1.EQ.0) GO TO 110
      ERROR= .TRUE.
      WRITE(6,1050)
      RETURN
  110 IF (NTAB.GT.1) TEM4= ANS2**(-B)*TEM1**TEM2
      TEM1= 1.+.5*(GAM-1.)*ANS3**2
      TEM5= ANS3**(-B)*TEM1**TEM2
      TEM6= -A*TEM5*TEM3
      IF (NTAB.EQ.1) TEM7=0.
      IF (NTAB.GT.1) TEM7= TEM5/TEM4*CTAB2(NTAB)
      NTAB= NTAB+1
      CTAB2(NTAB)= TEM6+TEM7
      CTAB1(NTAB)= CTAB2(NTAB)*ANS4
      STAB(NTAB)= SSDEL
      IF (CTAB1(NTAB).LT.-.32.OR.CTAB1(NTAB).GT..16) GO TO 120
      IF (SS.LT.ARCL) GO TO 100
  120 IF (KSDE.NE.1) GO TO 130
      WRITE(6,1060)
      WRITE(6,1070) (STAB(I),CTAB1(I),I=1,NTAB)
C
C  CALCULATE LAMINAR BOUNDARY LAYER PARAMETERS AT EACH STATION
C
  130 IF (KLAM.NE.1) GO TO 140
      WRITE(6,1080)
  140 I= 0
  150 I= I+1
      IF (I.EQ.NTURB) ITRAN=-1
      IF (S(I).LE.STAB(NTAB)) GO TO 160
      ERROR= .TRUE.
      WRITE(6,1090)
      RETURN
  160 CALL LGRNGE(STAB,CTAB1,NTAB,S(I),CORLN(I))
      CALL LGRNGE(STAB,CTAB2,NTAB,S(I),CORML(I))
C  OBTAIN SHEAR, CRN, AND DTH FROM CURVE FITS VS CORLN AND SW
      CALL CURVFT(CSHR,SHEAR(I),CORLN(I),SW(I),3,3)
      CALL CURVFT(CCRN,CRN(I),CORLN(I),SW(I),3,3)
      CALL CURVFT(CDTH,DTH(I),CORLN(I),SW(I),3,3)
C  CALCULATE OTHER LAMINAR BOUNDARY LAYER PARAMETERS
```

```
      THET(I)= SQRT(-CORML(I)*NUTZ*SUTHL(I)*ARCL/ATZ*TEM1**((3.-GAM)/
     1(2.*GAM-2.)))
      FORM(I)= (-1.1138*CORLN(I)+2.38411)*(1.+(2.79-1.78*PR**.5)*((1.+
     1SW(I))*TEM1-1.))+(4.65*PR**(1./3.)-3.65*PR**.5)*PR**.5*(TEM1-1.)
      DELSR(I)= THET(I)*FORM(I)
      RTH(I)= UE(I)*THET(I)/NUW(I)
      FORMI(I)= (FORM(I)-SQRT(PR)*(TEM1-1.))/((1.+SW(I))*TEM1)
      FORMTR(I)= FORMI(I)*(1.+SW(I))
      DELTA(I)= THET(I)*(DTH(I)+(TEM1-1.)*(FORMTR(I)+1.))
      SHAPL(I)= DELTA(I)**2/NUW(I)*DUDS(I)
      IF (I.EQ.1) GO TO 180
      CFRW= 2.*SHEAR(I)*SQRT(-SOL(I)/ME(I)/CORML(I))
      CF(I)= CFRW/SQRT(RW(I))
      TAUW(I)= CF(I)*HEADW(I)
      NURW= CFRW*PR**.3/CRN(I)
      NUSS(I)= NURW*SQRT(RW(I))
      DTDY(I)= NUSS(I)*(TAWL(I)-TWAL(I))/S(I)
      HTRAN(I)= TC*DTDY(I)
      IF (TAUW(I).GT.0.) GO TO 180
      IF (KATCH.NE.0) GO TO 170
      ISEP= I
      SEPRN= .TRUE.
      RETURN
  170 ITRAN= -2
      GO TO 270
  180 IF (I.EQ.1.AND.UE(1).EQ.0.) GO TO 190
      SHAPK(I)= NUTZ*RTH(I)**2*SUTHL(I)**2*(1.+SW(I))**4/ATZ/ME(I)**2/
     1FF(I)/ARCL*DMDL(I)*TEM1**(1./(GAM-1.))
      GO TO 200
  190 SHAPK(I)= 0.07
  200 RTHI(I)= RTH(I)*SUTHL(I)*(1.+SW(I))**2/FF(I)/SQRT(TEM1)
C
C  CALCULATE RCRIT TO CHECK FOR INSTABILITY AND TRANSITION
C
      CALL CURVFT(CRCR,RCRIT,SHAPK(I),0,5,0)
      RCRIT= EXP(RCRIT)
      IF(INST.NE.0) GO TO 210
C
C  CHECK FOR INSTABILITY
C
      IF(RTHI(I).LT.RCRIT) GO TO 270
      RINS= RTHI(I)
      INST= I
      GO TO 270
C
C  CHECK FOR TRANSITION
C
  210 K1= 0
      NS= I
      TEM= SIMPS1(SOL(INST),SOL(I),INT2,K1)
      IF (TEM.EQ.0..OR.K1.EQ.0) GO TO 220
      ERROR= .TRUE.
      WRITE(6,1100)
      RETURN
  220 KBAR= TEM/(SOL(I)-SOL(INST))
      CALL CURVFT(CDIF,DIFF,KBAR,0,5,0)
      RTRAN= RINS+DIFF
      IF(RTHI(I).LT.RTRAN) GO TO 270
      IF (I.LT.NTURB) GO TO 270
      ITRAN= -1
```

```
      GO TO 270
  230 ITRAN= I
C
C  COMPUTE INITIAL VALUES FOR TURBULENT SOLUTION
C
  240 TRANS= .TRUE.
      IF (DTURB.EQ.0..AND.TTURB.EQ.0.) GO TO 260
      IF (DTURB.GT.0..AND.TTURB.GT.0.) GO TO 250
      ERROR = .TRUE.
      WRITE(6,1110)
      RETURN
  250 THET(ITRAN)= TTURB
      FORM(ITRAN)= DTURB/TTURB
      TEM1= 1.+.5*(GAM-1.)*ME(ITRAN)**2
      FORMI(ITRAN)= (FORM(ITRAN)-PR**(1./3.)*(TEM1-1.))/((1.+SW(ITRAN))
     1*TEM1)
  260 IF (CTHET.GT.0..AND.DTURB.EQ.0..AND.TTURB.EQ.0.) THET(ITRAN)=
     1CTHET*THET(ITRAN)
      THETTR= THET(ITRAN)*(TSE(ITRAN)/TTZ)**((GAM+1.)/(2.*GAM-2.))
      FTRAN= (ME(ITRAN)*ATZ*THETTR/NUTZ)**1.268
      IF (RTRAN.LE.0.) RTRAN=1000.
      FORMS= FORMI(ITRAN)-0.59389-0.06591*ALOG(RTRAN)+0.001272*(ALOG(RTR
     1AN))**2
      IF (DTURB.GT.0..AND.TTURB.GT.0.) FORMS=FORMI(ITRAN)
      RETURN
C
C  PRINT OUTPUT
C
  270 IF (KLAM.NE.1) GO TO 280
      IF(INST.EQ.0 .OR. INST.EQ.I) WRITE(6,1120) I,CORLN(I),SHEAR(I),
     1DTH(I),FORMTR(I),SHAPL(I),RTHI(I),SHAPK(I),RCRIT
      IF(INST.NE.0 .AND. INST.NE.I) WRITE(6,1130) I,CORLN(I),SHEAR(I),
     1DTH(I),FORMTR(I),SHAPL(I),RTHI(I),KBAR,DIFF,RTRAN
      IF (ITRAN.EQ.-2) WRITE(6,1140)
  280 IF(ITRAN.EQ.-1.OR.ITRAN.EQ.-2) GO TO 230
      IF (I.EQ.NST) RETURN
      GO TO 150
C
C  FORMAT STATEMENTS
C
 1000 FORMAT(/////,10X,60HA NEGATIVE INITIAL VALUE HAS BEEN GIVEN. THIS
     1IS NOT ALLOWED)
 1010 FORMAT(/////,10X,75HINITIAL VALUES WERE NOT GIVEN FOR THE TURBULEN
     1T BOUNDARY LAYER AT STATION 1)
 1020 FORMAT(/////,10X,80HINITIAL VALUES WERE GIVEN FOR THE TURBULENT BO
     1UNDARY LAYER AT A STAGNATION POINT)
 1030 FORMAT(/////,10X,94HINITIAL VALUES OTHER THAN ZERO WERE GIVEN FOR
     1THE LAMINAR BOUNDARY LAYER AT A STAGNATION POINT)
 1040 FORMAT(/////,10X,106HFOR THIS INPUT DATA STATION 1 IS ASSUMED TO B
     1E A STAGNATION POINT, SINCE NO INITIAL THICKNESSES ARE GIVEN./
     210X,118HIN THIS CASE PRESSURE SHOULD DECREASE INITIALLY.  EITHER G
     3IVE AN INITIAL VALUE FOR DISPLACEMENT OR MOMENTUM THICKNESS,/
     410X,60HOR BEGIN WITH A SHORT REGION OF FAVORABLE PRESSURE GRADIENT
     5.)
 1050 FORMAT(/////,10X,37HERROR IN COMPUTING INTEGRAL FOR CORLN)
 1060 FORMAT(1H1///7X,50HLAMINAR DIFFERENTIAL EQUATION - SOLUTION FOR CO
     1RLN///5(24H          S          CORLN     )//)
 1070 FORMAT((5(F12.5,2X,F7.4,3X)))
 1080 FORMAT(1H1///1X,59HLAMINAR CALCULATION OF INSTABILITY AND TRANSITI
     1ON LOCATIONS///1X,7HSTATION,2X,5HCORLN,5X,5HSHEAR,5X,3HDTH,6X,6HFO
```

```
    2RMTR,4X,5HSHAPL,9X,4HRTHI,6X,5HSHAPK,9X,5HRCRIT,6X,4HKBAR,10X,4HDI
    3FF,9X,5HRTRAN)
1090 FORMAT(/////,10X,65HLAMINAR SOLUTION HAS PROCEEDED BEYOND THE RANG
    1E WHERE IT IS VALID)
1100 FORMAT(/////,10X,36HERROR IN COMPUTING INTEGRAL FOR KBAR)
1110 FORMAT(/////,10X,64HIF INITIAL TURBULENT VALUES ARE GIVEN, THEY BO
    1TH MUST BE NONZERO)
1120 FORMAT(I4,1X,5F10.4,1X,F12.1,1X,F10.5,1X,F12.1)
1130 FORMAT(I4,1X,5F10.4,1X,F12.1,24X,F12.5,1X,F12.1,1X,F12.1)
1140 FORMAT(/////,10X,85HLAMINAR SEPARATION HAS OCCURRED. ASSUMED TO BE
    1 TRANSITION TO TURBULENT BOUNDARY LAYER)
      END




$IBFTC TURBL

      SUBROUTINE TURBLN
      COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTURB,KPVM,KEM,KSMTH,
     1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTURB,TTURB,KPRE,KGRAD,KSDE,KLAM,
     2KMAIN,KPROF,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
     3VOVCR(100),TWAL(100)
      COMMON/C2/PSZ,TSZ,UZ,ASZ,ATZ,RHSZ,RHTZ,MUSZ,MUTZ,NUSZ,NUTZ,CP,
     1PR,TC,ARCL
      COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
     1TAWL(100),TAWT(100),TBAR(100),RW(100),SW(100),SUTHL(100),
     2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
     3AA(100),BB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
      COMMON/C4/THET(100),DELSR(100),DELTA(100),FORM(100),
     1FORMI(100),FORMTR(100),RTH(100),RTHI(100),CF(100),
     1TAUW(100),NUSS(100),DTDY(100),HTRAN(100),CRN(100)
      COMMON/C6/FTRAN,FORMS
      COMMON/C7/INST,ITRAN,ISEP
      COMMON/C8/XTAB(505),YTAB1(505),YTAB2(505),NTAB
      COMMON/C9/ERROR,TRANS,SEPRN
      REAL MUSZ,NUSZ,MUTZ,NUTZ,ME,NUW,MUBAR,NUSS
      LOGICAL ERROR,TRANS,SEPRN
C
C  SOLVE TURBULENT BOUNDARY LAYER DIFFERENTIAL EQUATIONS
C  USING RUNGA-KUTTA
      CALL RUNKUT
      IF (KSDE.NE.1) GO TO 10
      WRITE(6,1000)
      WRITE(6,1010) (XTAB(I),YTAB1(I),YTAB2(I),I=1,NTAB)
C
C  CALCULATE TURBULENT BOUNDARY LAYER PARAMETERS AT EACH STATION
C
   10 DO 30 I=ITRAN,NST
      IF (S(I).LE.XTAB(NTAB)) GO TO 20
      ISEP = I-1
      SEPRN= .TRUE.
      RETURN
   20 TEM1 = 1.+.5*(GAM-1.)*ME(I)**2
      CALL LGRNGE(XTAB,YTAB1,NTAB,S(I),F)
      THETTR= NUTZ*F**.7886/ME(I)/ATZ
      THET(I)= THETTR*(TTZ/TSE(I))**((GAM+1.)/(2.*GAM-2.))
      RTH(I)= UE(I)*THET(I)/NUW(I)
```

```
      CALL LGRNGE(XTAB,YTAB2,NTAB,S(I),FORMI(I))
      FORMTR(I)= FORMI(I)*(1.+SW(I))
      FORM(I)= FORMTR(I)*TEM1+PR**(1./3.)*(TEM1-1.)
      DELSR(I)= THET(I)*FORM(I)
      POWER= 2.0/(FORMI(I)-1.0)
      IF (FORMI(I).LT.1.02) POWER=100.
      DELTA(I)= (1.+POWER)*DELSR(I)
      CF(I)= 0.246*EXP(-1.561*FORMI(I))*(UE(I)*THET(I)/NUTZ/(TEM1**(1./(
     1GAM-1.))))**(-.268)*TSE(I)/TBAR(I)*(MUBAR(I)/MUTZ)**(.263)
      TAUW(I)= CF(I)*HEADE(I)
      IF (I.EQ.1) GO TO 30
      HTRAN(I)= CF(I)/2./PR**(2./3.)*RHSE(I)*UE(I)*CP*(TAWT(I)-TWAL(I))
      DTDY(I)= HTRAN(I)/TC
      NUSS(I)= S(I)*DTDY(I)/(TAWT(I)-TWAL(I))
      CRN(I)= CF(I)*RW(I)/NUSS(I)
   30 CONTINUE
      RETURN
 1000 FORMAT(1H1///5X,62HTURBULENT DIFFERENTIAL EQUATIONS - SOLUTION FOR
     1 F  AND  FORMI///4(31H       S           F         FORMI   )//)
 1010 FORMAT((4(F10.5,2X,F8.1,2X,F7.4,2X)))
      END
```

```
$IBFTC PROFI

       SUBROUTINE PROFIL
       COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTURB,KPVM,KEM,KSMTH,
      1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTURB,TTURB,KPRE,KGRAD,KSDE,KLAM,
      2KMAIN,KPROF,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
      3VOVCR(100),TWAL(100)
       COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
      1TAWL(100),TAWT(100),TBAR(100),RW(100),SW(100),SUTHL(100),
      2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
      3AA(100),BB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
       COMMON/C4/THET(100),DELSR(100),DELTA(100),FORM(100),
      1FORMI(100),FORMTR(100),RTH(100),RTHI(100),CF(100),
      1TAUW(100),NUSS(100),DTDY(100),HTRAN(100),CRN(100)
       COMMON/C5/SHAPL(100),SHAPK(100),B,NS
       COMMON/C7/INST,ITRAN,ISEP
       REAL ME,NUSS
C
C  PRINT LOCATIONS OF INSTABILITY, TRANSITION, AND SEPARATION
C
       IF (KMAIN.NE.1) GO TO 60
       WRITE(6,1000)
       IF(INST.EQ.0) GO TO 10
       WRITE(6,1010) INST
       GO TO 20
   10 WRITE(6,1020)
   20 IF (ITRAN.LE.1) GO TO 30
       WRITE(6,1030) ITRAN
       GO TO 40
   30 WRITE(6,1040)
   40 IF(ISEP.EQ.0) GO TO 50
       WRITE(6,1050) ISEP
```

```
      GO TO 60
   50 WRITE(6,1060)
C
C  PRINT LOCATIONS OF LAMINAR AND TURBULENT BOUNDARY LAYERS
C
   60 IEND = ITRAN-1
      IF (IEND.EQ.-1.OR.IEND.EQ.0) IEND=ISEP
      IF (IEND.EQ.0) IEND=NST
      IF (KMAIN.NE.1) GO TO 70
      IF (ITRAN.EQ.1) WRITE(6,1070)
      IF (ITRAN.NE.1) WRITE(6,1080) IEND
      IF (ITRAN.EQ.0) WRITE(6,1090)
      IF (ITRAN.EQ.1) WRITE(6,1100) ITRAN,IEND
   70 IF (ITRAN.LE.1) GO TO 80
      IEND = ISEP
      IF(IEND.EQ.0) IEND=NST
      IF (KMAIN.NE.1) GO TO 90
      WRITE(6,1100) ITRAN,IEND
C
C  PRINT CALCULATED BOUNDARY LAYER PARAMETERS
C
   80 IF (KMAIN.NE.1) GO TO 90
      WRITE(6,1110)
      WRITE(6,1120) (I,X(I),S(I),DELSR(I),THET(I),DELTA(I),FORM(I),
     1FORMI(I), I=1,IEND)
      WRITE(6,1130)
      WRITE(6,1140) (I,CF(I),TAUW(I),RTH(I),DTDY(I),NUSS(I),HTRAN(I),
     1CRN(I), I=1,IEND)
C
C  COMPUTE BOUNDS ON VELOCITY PROFILES
C
   90 IF (KPROF.NE.1) RETURN
      WRITE(6,1150)
      IF(ITRAN.NE.0) GO TO 100
      IL1= 2
      IL2= IEND
      IT1= 0
      IT2= 0
      GO TO 110
  100 IL1= 2
      IL2= ITRAN-1
      IT1= ITRAN
      IT2= IEND
      IF (IT1.EQ.1) IT1=2
C
C  CALCULATE AND PRINT LAMINAR BOUNDARY LAYER VELOCITY PROFILES
C
  110 NVP1= NVP+1
      IF (IL2.LT.IL1) GO TO 140
      DO 130 I=IL1,IL2
      WRITE(6,1160) I
      AAA= 2.+SHAPL(I)/6.
      BBB= -.5*SHAPL(I)
      CCC= -2.+.5*SHAPL(I)
      DDD= 1.-SHAPL(I)/6.
      DEL= DELTA(I)/FLOAT(NVP)
      YP= -DEL
      DO 120 J=1,NVP1
      YP= YP+DEL
```

```
      ETA= YP/DELTA(I)
      YXMAX= YP/X(NST)
      UUE= AAA*ETA+BBB*ETA**2+CCC*ETA**3+DDD*ETA**4
      U= UUE*UE(I)
  120 WRITE(6,1180) ETA,YP,YXMAX,U,UUE
  130 CONTINUE
C
C  CALCULATE AND PRINT TURBULENT BOUNDARY LAYER VELOCITY PROFILES
C
  140 IF(IT1.EQ.0) RETURN
      DO 160 I=IT1,IT2
      POWER= DELTA(I)/DELSR(I)-1.
      WRITE(6,1170) I,POWER
      DEL= DELTA(I)/FLOAT(NVP)
      YP= -DEL
      DO 150 J=1,NVP1
      YP= YP+DEL
      ETA= YP/DELTA(I)
      YXMAX= YP/X(NST)
      UUE= ETA**(1./POWER)
      U= UUE*UE(I)
  150 WRITE(6,1180) ETA,YP,YXMAX,U,UUE
  160 CONTINUE
      RETURN
C
C  FORMAT STATEMENTS
C
 1000 FORMAT(1H1///1X,36HPRINCIPAL BOUNDARY LAYER INFORMATION///)
 1010 FORMAT (/10X,31HINSTABILITY OCCURS AT STATION  ,I3)
 1020 FORMAT (/10X,26HINSTABILITY DOES NOT OCCUR)
 1030 FORMAT (/10X,30HTRANSITION OCCURS AT STATION  ,I3)
 1040 FORMAT (/10X,25HTRANSITION DOES NOT OCCUR)
 1050 FORMAT (/10X,30HSEPARATION OCCURS AT STATION  ,I3)
 1060 FORMAT (/10X,25HSEPARATION DOES NOT OCCUR)
 1070 FORMAT (/10X,37HLAMINAR BOUNDARY LAYER DOES NOT OCCUR)
 1080 FORMAT (/10X,42HLAMINAR BOUNDARY LAYER - STATIONS  1  TO  ,I3)
 1090 FORMAT (/10X,39HTURBULENT BOUNDARY LAYER DOES NOT OCCUR///)
 1100 FORMAT (/10X,35HTURBULENT BOUNDARY LAYER - STATIONS,2X,
     1I3,6H  TO  ,I3///)
 1110 FORMAT(/1X,7HSTATION,8X,1HX,12X,1HS,12X,5HDELSR,10X,4HTHET,11X,
     15HDELTA,11X,4HFORM,10X,5HFORMI)
 1120 FORMAT(2X,I3,3X,2F13.6,F14.6,1X,F14.6,1X,F14.5,1X,2F14.4)
 1130 FORMAT(///1X,7HSTATION,6X,2HC=,13X,4HTAUW,11X,3HRTH,14X,4HDTDY,
     113X,4HNUSS,10X,5HHTRAN,12X,3HCRN)
 1140 FORMAT(I5,F14.5,2X,F14.5,1X,F12.1,5X,F14.2,2X,F14.2,1X,
     1F14.4,2X,F13.3)
 1150 FORMAT(1H1///1X,17HVELOCITY PROFILES///)
 1160 FORMAT (/1X,7HSTATION,1X,I5,2X,7HPROFILE/3X,7HY/DELTA,9X,
     11HY,12X,6HY/XMAX,10X,1HU,12X,4HU/UE)
 1170 FORMAT (/1X,7HSTATION,1X,I5,2X,7HPROFILE,28X,2HN=,1X,F6.2/3X,7HY/D
     1ELTA,9X,1HY,12X,6HY/XMAX,10X,1HU,12X,4HU/UE)
 1180 FORMAT(1X,F8.4,2X,2G15.6,2X,F9.2,6X,F8.4)
      END
```

```
$IBFTC RUNKU

      SUBROUTINE RUNKUT
C
C  RUNKUT SOLVES SIMULTANEOUS FIRST ORDER INITIAL VALUE
C  ORDINARY DIFFERENTIAL EQUATIONS
C
      COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTURB,KPVM,KEM,KSMTH,
     1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTURB,TTURB,KPRE,KGRAD,KSDE,KLAM,
     2KMAIN,KPROF,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
     3VOVCR(100),TWAL(100)
      COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
     1TAWL(100),TAWT(100),TBAR(100),RW(100),SW(100),SUTHL(100),
     2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
     3AA(100),BB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
      COMMON/C6/FTRAN,FORMS
      COMMON/C7/INST,ITRAN,ISEP
      COMMON/C8/XTAB(505),YTAB1(505),YTAB2(505),NTAB
      DIMENSION YY(2),RY(2),YINC(2),DOT(2),RUK(2,4)
      DOUBLE PRECISION XX,RX,YY,RY,RUK,DEL,DOT,
     1TEM1,TEM2,TEM3,TEM4,TEM5,TEM6
      REAL ME,NUW,MUBAR
C
C  SET DEL SPACING AND STORE INITIAL VALUES
C
      DEL= 0.002*S(NST)
   10 YY(1)=FTRAN
      YY(2)= FORMS
      XX= S(ITRAN)
      NV=2
      NTAB = 1
      YTAB1(1)= YY(1)
      YTAB2(1)= YY(2)
      XTAB(1)= XX
C
C  SOLVE FOR YY(1) AND YY(2) AT NEXT XX INCREMENT
C
C  SAVE PREVIOUS YY(1) AND YY(2)
   20 DO 30 J=1,NV
   30 RY(J)= YY(J)
      RX= XX
C
C  CALCULATE NEW YY(1) AND YY(2)
C
      DO 90 L=1,4
C  PUT DIFFERENTIAL EQUATIONS IN THE FORM OF
C  FIRST DERIVATIVE = REMAINDER OF EQUATION
      CALL LGRNGE(S,ME,NST,XX,ANS1)
      CALL LGRNGE(S,SW,NST,XX,ANS2)
      CALL LGRNGE(S,AA,NST,XX,ANS3)
      CALL LGRNGE(S,BB,NST,XX,ANS4)
      CALL LGRNGE(S,DMDS,NST,XX,ANS5)
      CALL LGRNGE(S,TBAR,NST,XX,ANS6)
      TEM1= 1.+(1.+ANS2)*YY(2)
      TEM2= .123*EXP(-1.561*YY(2))*ANS3
      DOT(1)= 1.268*(-YY(1)/ANS1*ANS5*TEM1+TEM2)
      TEM3= YY(2)*(YY(2)+1.)**2*(YY(2)-1.)
      TEM4= 1.+ANS2*(YY(2)*YY(2)+4.*YY(2)-1.)/((YY(2)+1.)*(YY(2)+3.))
      TEM5= (YY(2)*YY(2)-1.)*YY(2)/YY(1)*(.123*EXP(-1.561*YY(2))*ANS3)
```

```
      TEM6= (YY(2)*YY(2)-1.)/YY(1)**(.7886)*(.011*(YY(2)+1.)*(YY(2)-1.)
     1**2/YY(2)**2*TTZ/ANS6)*ANS4
      DOT(2)= -ANS5*.5/ANS1*TEM3*TEM4+TEM5-TEM6
C   APPLY THE RUNGA-KUTTA SCHEME
      DO 40 J=1,NV
   40 RUK(J,L) = DEL*DOT(J)
      GO TO (50,50,70,90), L
   50 DO 60 J=1,NV
   60 YY(J)= RY(J)+RUK(J,L)/2.
      XX= RX+DEL/2.
      GO TO 90
   70 DO 80 J=1,NV
   80 YY(J)= RY(J)+RUK(J,L)
      XX= RX+DEL
   90 CONTINUE
C   INCREMENT THE DEPENDENT VARIABLES TO OBTAIN NEW YY(1) AND YY(2)
      DO 100 J=1,NV
      YINC(J) = (RUK(J,1)+2.*RUK(J,2)+2.*RUK(J,3)+RUK(J,4))/6.
  100 YY(J)= RY(J)+YINC(J)
      IF (YY(2).GT.2.8) RETURN
C
C   STORE NEW COMPUTED VALUES IN A TABLE
C
      NTAB = NTAB+1
      YTAB1(NTAB)= YY(1)
      YTAB2(NTAB)= YY(2)
      XTAB(NTAB)= XX
      IF (XX.LT.S(NST)) GO TO 20
      RETURN
      END
```

```
$IBFTC FUNC

      SUBROUTINE FUNCT(XX,FX,DFX,INF)
      COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTURB,KPVM,KEM,KSMTH,
     1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTURB,TTURB,KPRE,KGRAD,KSDE,KLAM,
     2KMAIN,KPROF,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
     3VOVCR(100),TWAL(100)
      COMMON/C2/PSZ,TSZ,UZ,ASZ,ATZ,RHSZ,RHTZ,MUSZ,MUTZ,NUSZ,NUTZ,CP,
     1PR,TC,ARCL
      COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
     1TAWL(100),TAWT(100),TBAR(100),RW(100),SW(100),SUTHL(100),
     2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
     3AA(100),BB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
      REAL MUSZ,NUSZ,MUTZ,NUTZ,ME,NUW,MUBAR
      INF = 0
      B1= 1.+.5*(GAM-1.)*ME(1)**2
      B2= 1.+(2.79-1.78*PR**.5)*((1.+SW(1))*B1-1.)
      B3= -NUTZ*SUTHL(1)*ARCL/ATZ/DMDL(1)*B1**((3.-GAM)/(2.*GAM-2.))
      B4= -1.1138*B2
      B5= 2.38411*B2+(4.65*PR**(1./3.)-3.65*PR**.5)*PR**.5*(B1-1.)
      FX= (B3*XX)**.5*(B4*XX+B5)
      IF (XX.EQ.0.) GO TO 10
      DFX= .5*(B3*XX)**(-.5)*B3*(B4*XX+B5)+B4*(B3*XX)**.5
```

```
      RETURN
   10 INF = 1
      DFX = 1.E10
      RETURN
      END
```

```
$IBFTC INTG1

      REAL FUNCTION INT1(XX)
      COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTURB,KPVM,KEM,KSMTH,
     1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTURB,TTURB,KPRE,KGRAD,KSDE,KLAM,
     2KMAIN,KPROF,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
     3VOVCR(100),TWAL(100)
      COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
     1TAWL(100),TAWT(100),TBAR(100),RW(100),SW(100),SUTHL(100),
     2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
     3AA(100),BB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
      COMMON/C5/SHAPL(100),SHAPK(100),B,NS
      REAL ME,NUW,MUBAR,INT1
      CALL LGRNGE(SOL,ME,NST,XX,ANS)
      INT1= ANS**(B-1.)/((1.+.5*(GAM-1.)*ANS**2)**
     1((3.*GAM-1.)/(2.*GAM-2.)))
      RETURN
      END
```

```
$IBFTC INTG2

      REAL FUNCTION INT2(XX)
      COMMON/C1/GAM,R,PTZ,TTZ,UPMACH,NST,NVP,NTURB,KPVM,KEM,KSMTH,
     1KSPLN,KLE,KATCH,CTHET,DLAM,TLAM,DTURB,TTURB,KPRE,KGRAD,KSDE,KLAM,
     2KMAIN,KPROF,X(100),Y(100),PRES(100),UE(100),ME(100),POPTZ(100),
     3VOVCR(100),TWAL(100)
      COMMON/C3/XOM(100),YOM(100),S(100),SOL(100),AE(100),TSE(100),
     1TAWL(100),TAWT(100),TBAR(100),RW(100),SW(100),SUTHL(100),
     2RHSW(100),RHSE(100),HEADW(100),HEADE(100),NUW(100),MUBAR(100),
     3AA(100),BB(100),FF(100),DUDS(100),DMDS(100),DMDL(100)
      COMMON/C5/SHAPL(100),SHAPK(100),B,NS
      REAL ME,NUW,MUBAR,INT2
      IF (NS.LT.4) GO TO 10
      CALL LGRNGE(SOL,SHAPK,NS,XX,INT2)
      RETURN
   10 DO 20 J=2,NS
      IF (SOL(J).LT.XX) GO TO 20
      INT2= SHAPK(J-1)+(SHAPK(J)-SHAPK(J-1))*(XX-SOL(J-1))/(SOL(J)-SOL(J
     1-1))
      RETURN
   20 CONTINUE
      RETURN
      END
```

```
$IBFTC SMOTH

      SUBROUTINE SMOOTH(Y,N,K)
C
C  SMOOTH IS A SIMPLE DATA SMOOTHING ROUTINE.
C  IF K=1, THE END POINTS ARE ALSO SMOOTHED.
C
      DIMENSION Y(100),Z(100)
      N1=N-1
      DO 10 I=2,N1
   10 Z(I)= (Y(I-1)+2.*Y(I)+Y(I+1))/4.
      Z(1)= Y(1)
      Z(N)= Y(N)
      IF(K.NE.1) GO TO 20
      Z(1)=(Y(1)+2.*Y(2)-Y(3))/2.
      Z(N)=(Y(N)+2.*Y(N1)-Y(N-2))/2.
   20 DO 30 I=1,N
   30 Y(I)= Z(I)
      RETURN
      END
```

```
$IBFTC GRADN

      SUBROUTINE GRADNT(X,FX,N,DFDX)
C
C  GRADNT CALCULATES THE GRADIENT OF FX WITH RESPECT TO X
C  USING FINITE DIFFERENCE TECHNIQUES
C
      DIMENSION X(N),FX(N),DFDX(N)
      DIMENSION SL(100),DIST(100)
      N1= N-1
      DO 10 I=1,N1
      SL(I)= (FX(I+1)-FX(I))/(X(I+1)-X(I))
   10 DIST(I)= SQRT((FX(I+1)-FX(I))**2+(X(I+1)-X(I))**2)
      DO 20 I=2,N1
   20 DFDX(I)=(SL(I)*DIST(I-1)+SL(I-1)*DIST(I))/(DIST(I-1)+DIST(I))
      DFDX(1)= SL(1)+(SL(1)-SL(2))*DIST(1)/(DIST(1)+DIST(2))
      DFDX(N)= SL(N1)+(SL(N1)-SL(N-2))*DIST(N1)/(DIST(N1)+DIST(N-2))
      RETURN
      END
```

```
$IBFTC SPLIN

      SUBROUTINE SPLINE(X,Y,N,DYDX,D2YDX2)
C
C  SPLINE FITS A SPLINE CURVE TO  X  AND  Y
C  AND CALCULATES FIRST AND SECOND DERIVATIVES AT THE SPLINE POINTS
C  END POINT SECOND DERIVATIVES EQUAL THOSE AT ADJACENT POINTS
C
```

```
      DIMENSION X(N),Y(N),DYDX(N),D2YDX2(N)
      DIMENSION G(100),H(100)
      G(1)= -1.
      H(1)= 0.
      N1= N-1
      IF (N1.LT.2) GO TO 20
      DO 10 I=2,N1
      A= (X(I)-X(I-1))/6.
      B= (X(I+1)-X(I))/6.
      C= 2.*(A+B)-A*G(I-1)
      D= (Y(I+1)-Y(I))/(X(I+1)-X(I))-(Y(I)-Y(I-1))/(X(I)-X(I-1))
      G(I)= B/C
   10 H(I)= (D-A*H(I-1))/C
   20 D2YDX2(N)= H(N1)/(1.+G(N1))
      DO 30 I=2,N
      K= N+1-I
   30 D2YDX2(K)= H(K)-G(K)*D2YDX2(K+1)
      DYDX(1)= (X(1)-X(2))/6.*(2.*D2YDX2(1)+D2YDX2(2))+(Y(2)-Y(1))/(X(2)
     1-X(1))
      DO 40 I=2,N
   40 DYDX(I)= (X(I)-X(I-1))/6.*(2.*D2YDX2(I)+D2YDX2(I-1))+(Y(I)-Y(I-1))
     1/(X(I)-X(I-1))
      RETURN
      END




$IBFTC ROO

      SUBROUTINE ROOT(A,B,Y,FUNCT,TOLERY,X,DFX)
C
C  ROOT FINDS A ROOT FOR (FUNCT-Y) IN THE INTERVAL (A,B)
C
      X1= A
      X2= B
      CALL FUNCT(X1,FX1,DFX,INF)
   10 DO 30 I=1,20
      X= (X1+X2)/2.
      CALL FUNCT(X,FX,DFX,INF)
      IF ((FX1-Y)*(FX-Y).GT.0.) GO TO 20
      X2= X
      GO TO 30
   20 X1= X
      FX1= FX
   30 CONTINUE
      IF (ABS(Y-FX).LT.TOLERY) RETURN
      WRITE(6,1000) A,B,Y
      STOP
 1000 FORMAT(////4X,49HROOT HAS FAILED TO CONVERGE IN THE GIVEN INTERVAL
     1/4X,3HA =,G14.6,10X,3HB =,G14.6,10X,3HY =,G14.6)
      END
```

```
$IBFTC LGRNG
      SUBROUTINE LGRNGE(X,Y,N,ARG,ANS)
C
C  LGRNGE PERFORMS 4 POINT LAGRANGIAN INTERPOLATION
C
      DIMENSION X(N),Y(N),XX(4),YY(4)
      IF (ARG-X(2)) 10,10,20
   10 MM = 1
      GO TO 70
   20 IF (ARG-X(N-1)) 40,40,30
   30 MM = N-3
      GO TO 70
   40 N1 = N-1
      DO 60 I=2,N1
      IF (ARG-X(I)) 50,50,60
   50 MM = I-2
      GO TO 70
   60 CONTINUE
   70 DO 80 I=1,4
      MMM = MM+I-1
      XX(I) = X(MMM)
   80 YY(I) = Y(MMM)
      C1 = ((ARG-XX(2))*(ARG-XX(3))*(ARG-XX(4)))/
     1(XX(1)-XX(2))/(XX(1)-XX(3))/(XX(1)-XX(4))
      C2 = ((ARG-XX(1))*(ARG-XX(3))*(ARG-XX(4)))/
     1(XX(2)-XX(1))/(XX(2)-XX(3))/(XX(2)-XX(4))
      C3 = ((ARG-XX(1))*(ARG-XX(2))*(ARG-XX(4)))/
     1(XX(3)-XX(1))/(XX(3)-XX(2))/(XX(3)-XX(4))
      C4 = ((ARG-XX(1))*(ARG-XX(2))*(ARG-XX(3)))/
     1(XX(4)-XX(1))/(XX(4)-XX(2))/(XX(4)-XX(3))
      ANS = C1*YY(1)+C2*YY(2)+C3*YY(3)+C4*YY(4)
      RETURN
      END
```

```
$IBFTC CURVF
      SUBROUTINE CURVFT(COEF,ANS,X,Y,NX,NY)
C
C  EVALUATE THE POLYNOMIAL FUNCTION, ANS=F(X,Y), USING COEFFICIENTS, COE
C
      DIMENSION COEF(20)
      NX1 = NX+1
      NY1 = NY+1
      ANS = COEF(1)
      IF (X.EQ..0.AND.Y.EQ..0) RETURN
      IF (Y.EQ..0) GO TO 10
      IF (X.EQ..0) GO TO 30
      GO TO 50
   10 DO 20 I=2,NX1
   20 ANS = ANS+COEF(I)*X**(I-1)
      RETURN
   30 DO 40 I=2,NY1
      K = (I-1)*NX1+1
   40 ANS = ANS+COEF(K)*Y**(I-1)
```

```
      RETURN
   50 ANS = .0
      DO 60 I=1,NY1
      DO 60 J=1,NX1
      K = (I-1)*NX1+J
   60 ANS = ANS+COEF(K)*Y**(I-1)*X**(J-1)
      RETURN
      END
```

```
$IBFTC SIMP

      FUNCTION SIMPS1(X1,X2,FUNC,KSIG)
      DIMENSION V(200),H(200),A(200),B(200),C(200),P(200),E(200)
      LOGICAL SPILL
      DOUBLE PRECISION ANS,Q
      DATA TWO,THREE,FOUR,THIRTY/2.0,3.0,4.0,30.0/
      DATA T,NMAX,NSIG/3.0E-5,200,1/
C  INITIALIZE FIRST ELEMENTS OF ARRAYS.
      V=X1
      H=(X2-V)/TWO
      A=FUNC(V)
      B=FUNC(V+H)
      C=FUNC(X2)
      P=H*(A+FOUR*B+C)
      E=P
      ANS=P
      N=1
      FRAC=T
      SPILL=.FALSE.
   10 TEST=ABS(FRAC*ANS)
      K=N
      DO 30 I=1,K
C  TEST MAGNITUDE OF 4TH ORDER ERROR IN THIS INTERVAL.
      IF (ABS(E(I)).LE.TEST) GO TO 30
      IF (N.LT.NMAX) GO TO 20
C  GO TO FINISH IF STORAGE IS FILLED UP.
      SPILL=.TRUE.
      KSIG=KSIG+NSIG
      GO TO 40
C  SUBDIVIDE INTERVAL AGAIN TO REDUCE 4TH ORDER ERROR.
   20 N=N+1
      V(N)=V(I)+H(I)
      H(N)=H(I)/TWO
      A(N)=B(I)
      B(N)=FUNC(V(N)+H(N))
      C(N)=C(I)
      P(N)=H(N)*(A(N)+FOUR*B(N)+C(N))
      H(I)=H(N)
      B(I)=FUNC(V(I)+H(I))
      C(I)=A(N)
      Q=P(I)
      P(I)=H(I)*(A(I)+FOUR*B(I)+C(I))
      Q=P(I)+P(N)-Q
      ANS=ANS+Q
```

```
      E(I)=Q
      E(N)=Q
   30 CONTINUE
C  TEST ALL INTERVALS AGAIN IF ANY WERE SUBDIVIDED THE LAST TIME.
      IF (N.GT.K) GO TO 10
   40 Q=0.0
      DO 50 I=1,N
   50 Q=Q+E(I)
C  TIGHTEN ERROR LIMIT IF TOTAL ACCUMULATED ERROR TOO LARGE.
      IF (ABS(Q/T).LE.ABS(ANS).OR.SPILL) GO TO 60
      FRAC=FRAC/TWO
      GO TO 10
C  FINISH CALCULATION.
   60 SIMPS1=(ANS+Q/THIRTY)/THREE
      RETURN
      END
```

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, January 8, 1970,
126-15.

APPENDIX A

DERIVATION OF EQUATIONS

DERIVATION OF EQUATION (14)

Equation (3) (which is eq. (32) of ref. 4) is the solution of equation (1), the laminar differential equation for correlation number n. When equation (3) is transformed to physical quantities, equation (33) of reference 4 results

$$\frac{n}{-\frac{L}{u_e}\frac{du_e}{dx}\frac{T_0'}{T_e}} = A\left(\frac{T_e}{T_0'}\right)^{\frac{1-3\gamma}{2\gamma-2}} M_e^{1-B}\int_0^{x/L}\left(\frac{T_e}{T_0'}\right)^{\frac{3\gamma-1}{2\gamma-2}} M_e^{B-1}\, d\left(\frac{x}{L}\right) \qquad \text{(A1)}$$

Rearranging gives

$$n = -AM_e^{-B}\frac{T_0'}{T_e}\frac{M_e}{u_e}\frac{du_e}{d\left(\frac{x}{L}\right)}\left(\frac{T_0'}{T_e}\right)^{\frac{3\gamma-1}{2\gamma-2}}\int_0^{x/L}\frac{M_e^{B-1}}{\left(\frac{T_0'}{T_e}\right)^{\frac{3\gamma-1}{2\gamma-2}}}\, d\left(\frac{x}{L}\right) \qquad \text{(A2)}$$

By using the isentropic equations and the compressible Bernoulli equation, it is possible to obtain the relation

$$\frac{dM_e}{dx} = \frac{1}{a_e}\frac{T_0'}{T_e}\frac{du_e}{dx} \qquad \text{(A3)}$$

When this substitution is made, equation (A2) becomes

$$n = -AM_e^{-B} \frac{dM_e}{d\left(\frac{x}{L}\right)} \left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{\frac{3\gamma-1}{2\gamma-2}} \int_0^{x/L} \frac{M_e^{B-1}}{\left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{\frac{3\gamma-1}{2\gamma-2}}} d\left(\frac{x}{L}\right) \qquad \text{(A4)}$$

This agrees with equation (4).

In the program n is obtained at small incremental steps along the boundary-layer surface. Applying equation (A4) at any two adjacent steps yields

$$(n)_{x_1/L} = \left[-AM_e^{-B} \frac{dM_e}{d\left(\frac{x}{L}\right)} \left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{\frac{3\gamma-1}{2\gamma-2}}\right]_{x_1/L} \int_0^{x_1/L} \frac{M_e^{B-1}}{\left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{\frac{3\gamma-1}{2\gamma-2}}} d\left(\frac{x}{L}\right) \qquad \text{(A5)}$$

$$(n)_{x_2/L} = \left[-AM_e^{-B} \frac{dM_e}{d\left(\frac{x}{L}\right)} \left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{\frac{3\gamma-1}{2\gamma-2}}\right]_{x_2/L} \int_0^{x_2/L} \frac{M_e^{B-1}}{\left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{\frac{3\gamma-1}{2\gamma-2}}} d\left(\frac{x}{L}\right) \qquad \text{(A6)}$$

But equation (A6) can be rewritten

$$(n)_{x_2/L} = \left[-AM_e^{-B}\frac{dM_e}{d\left(\frac{x}{L}\right)}\left(1+\frac{\gamma-1}{2}M_e^2\right)^{\frac{3\gamma-1}{2\gamma-2}}\right]_{x_2/L}\int_0^{x_1/L}\frac{M_e^{B-1}}{\left(1+\frac{\gamma-1}{2}M_e^2\right)^{\frac{3\gamma-1}{2\gamma-2}}}\,d\left(\frac{x}{L}\right)$$

$$+\left[-AM_e^{-B}\frac{dM_e}{d\left(\frac{x}{L}\right)}\left(1+\frac{\gamma-1}{2}M_e^2\right)^{\frac{3\gamma-1}{2\gamma-2}}\right]_{x_2/L}\int_{x_1/L}^{x_2/L}\frac{M_e^{B-1}}{\left(1+\frac{\gamma-1}{2}M_e^2\right)^{\frac{3\gamma-1}{2\gamma-2}}}\,d\left(\frac{x}{L}\right) \tag{A7}$$

Then substituting for the first integral from equation (A5) gives

$$(n)_{x_2/L} = \frac{\left[M_e^{-B}\frac{dM_e}{d\left(\frac{x}{L}\right)}\left(1+\frac{\gamma-1}{2}M_e^2\right)^{\frac{3\gamma-1}{2\gamma-2}}\right]_{x_2/L}}{\left[M_e^{-B}\frac{dM_e}{d\left(\frac{x}{L}\right)}\left(1+\frac{\gamma-1}{2}M_e^2\right)^{\frac{3\gamma-1}{2\gamma-2}}\right]_{x_1/L}}(n)_{x_1/L}$$

$$-\left[AM_e^{-B}\frac{dM_e}{d\left(\frac{x}{L}\right)}\left(1+\frac{\gamma-1}{2}M_e^2\right)^{\frac{3\gamma-1}{2\gamma-2}}\right]_{x_2/L}\int_{x_1/L}^{x_2/L}\frac{M_e^{B-1}}{\left(1+\frac{\gamma-1}{2}M_e^2\right)^{\frac{3\gamma-1}{2\gamma-2}}}\,d\left(\frac{x}{L}\right) \tag{14}$$

DERIVATION OF EQUATION (15)

Momentum thickness is calculated by equation (39) of reference 4:

$$\frac{\theta}{L}\sqrt{R_w} = \frac{T_e}{T_w}\sqrt{\frac{n\dfrac{x}{L}}{-\dfrac{L}{u_e}\,\dfrac{du_e}{dx}\,\dfrac{T_0'}{T_e}}} \tag{A8}$$

Rearranging gives

$$\theta = \left(-\frac{L^2}{R_w}\,\frac{x}{L}\,\frac{u_e}{L}\,\frac{T_e^2}{T_w^2}\,\frac{n}{\dfrac{T_0'}{T_e}\,\dfrac{du_e}{dx}}\right)^{1/2} = \left[-L\nu_w\,\frac{T_e^2}{T_w^2}\,\frac{n}{\dfrac{T_0'}{T_e}\,\dfrac{du_e}{d\left(\dfrac{x}{L}\right)}}\right]^{1/2}$$

Using equation (A3), this becomes

$$\theta = \left\{\frac{L\nu_w}{a_e}\,\frac{T_e^2}{T_w^2}\left[\frac{-n}{\dfrac{dM_e}{d\left(\dfrac{x}{L}\right)}}\right]\right\}^{1/2} \tag{A9}$$

From the viscosity law of the form

$$\frac{\mu_w}{\mu_0'} = k_{su}\,\frac{T_w}{T_0'} \tag{A10}$$

it is easy to obtain the following relation:

$$\nu_w = k_{su}\nu_0' \frac{T_w}{T_0'} \frac{\rho_0'}{\rho_w} \tag{A11}$$

Substituting equation (A11) into equation (A9) yields

$$\theta = \left\{ \frac{Lk_{su}\nu_0'}{a_e} \frac{\rho_0'}{\rho_w} \frac{T_w}{T_0'} \frac{T_e^2}{T_w^2} \left[\frac{-n}{\frac{dM_e}{d\left(\frac{x}{L}\right)}} \right] \right\}^{1/2} = \left\{ \frac{Lk_{su}\nu_0'}{a_0'} \left(\frac{T_0'}{T_e} \right)^{1/2} \frac{P_0'T_w}{P_wT_0'} \frac{T_w}{T_0'} \frac{T_e^2}{T_w^2} \left[\frac{-n}{\frac{dM_e}{d\left(\frac{x}{L}\right)}} \right] \right\}^{1/2}$$

Using $P_w = P_e$ and the isentropic relations results in

$$\theta = \left\{ \frac{Lk_{su}\nu_0'}{a_e'} \left(\frac{T_0'}{T_e} \right)^{\frac{3-\gamma}{2\gamma-2}} \left[\frac{-n}{\frac{dM_e}{d\left(\frac{x}{L}\right)}} \right] \right\}^{1/2} \tag{A12}$$

$$\theta = \left\{ \frac{\nu_0' k_{su} L}{a_0'} \left[\frac{-n}{\frac{dM_e}{d\left(\frac{x}{L}\right)}} \right] \left(1 + \frac{\gamma - 1}{2} M_e^2 \right)^{\frac{3-\gamma}{2\gamma-2}} \right\}^{1/2} \tag{15}$$

DERIVATION OF EQUATION (20)

Skin friction coefficient is calculated by equation (35) of reference 4:

$$C_f\sqrt{R_w} = 2l\sqrt{\frac{\frac{x}{L}\left(-\frac{L}{u_e}\frac{du_e}{dx}\right)\frac{T_0}{T_e}}{n}} \tag{A13}$$

Rearranging and using equation (A3), this becomes

$$C_f\sqrt{R_w} = 2l\left[-\frac{1}{n}\frac{1}{u_e}\frac{T_0}{T_e}\frac{T_e a_e}{T_0}\frac{dM_e}{d\left(\frac{x}{L}\right)}\frac{x}{L}\right]^{1/2}$$

This equation reduces to

$$C_f = \frac{2l\left[-\frac{1}{n}\frac{1}{M_e}\frac{dM_e}{d\left(\frac{x}{L}\right)}\frac{x}{L}\right]^{1/2}}{\sqrt{R_w}} \tag{20}$$

DERIVATION OF EQUATIONS (18) AND (28)

In reference 18 the following expression is derived, relating form factor H to equivalent incompressible form factor H_i for flows over insulated surfaces with $Pr = 1$:

$$H = H_i + \frac{\gamma - 1}{2} M_e^2(H_i + 1) \tag{A14}$$

For noninsulated surfaces, equation (A14) becomes

$$H = H_{tr} + \frac{\gamma - 1}{2} M_e^2(H_{tr} + 1) \tag{A15}$$

This is equation (40) of reference 4. Using the Crocco relation for a flat plate

$$\frac{h' - h_w}{h_0' - h_w} = \frac{u}{u_e} = \frac{U}{U_e} \tag{A16}$$

it can be shown that

$$\int_0^{\delta_{tr}} \left(\frac{h'}{h'_0} - 1\right) dY_{tr} = \int_0^{\delta_{tr}} S \, dY_{tr} = S_w \theta_{tr} H_i \tag{A17}$$

Then the relation between the adiabatic and nonadiabatic transformed form factors

$$H_{tr} = \frac{\delta^*_{tr}}{\theta_{tr}} = \frac{\int_0^{\delta_{tr}} \left(1 - \frac{U}{U_e} + S\right) dY_{tr}}{\theta_{tr}} = H_i + \frac{\int_0^{\delta_{tr}} S \, dY_{tr}}{\theta_{tr}} \tag{A18}$$

is used to obtain the relation

$$H_{tr} = H_i + \frac{S_w \theta_{tr} H_i}{\theta_{tr}} = (1 + S_w) H_i \tag{A19}$$

Substituting equation (A19) into equation (A15) gives

$$H = (1 + S_w)\left(1 + \frac{\gamma - 1}{2} M_e^2\right) H_i + \frac{\gamma - 1}{2} M_e^2 \tag{A20}$$

Solving for H_i results in

$$H_i = \frac{H - \frac{\gamma - 1}{2} M_e^2}{(1 + S_w)\left(1 + \frac{\gamma - 1}{2} M_e^2\right)} \tag{A21}$$

For nonunity Prandtl number and laminar flow, equation (A21) becomes

$$H_i = \frac{H - Pr^{1/2} \frac{\gamma - 1}{2} M_e^2}{(1 + S_w)\left(1 + \frac{\gamma - 1}{2} M_e^2\right)} \tag{18}$$

For nonunity Prandtl number and turbulent flow,

$$H_i = \frac{H - Pr^{1/3} \frac{\gamma - 1}{2} M_e^2}{(1 + S_w)\left(1 + \frac{\gamma - 1}{2} M_e^2\right)} \tag{28}$$

APPENDIX B

CURVE FITS USED IN PROGRAM

Polynomial curve fits are used in the PRECAL and LAMNAR routines. Coefficients for the curve fits are read in at the beginning of each of these routines, and values are calculated by means of calls on the CURVFT routine.

In PRECAL the following curve fits are used:

(1) Nondimensional dynamic viscosity of air as a function of nondimensional static temperature, $\mu/\mu_{sl} = f(T/T_{sl})$:

$$\frac{\mu}{\mu_{sl}} = -0.0194517 + 1.3019531\left(\frac{T}{T_{sl}}\right) - 0.34511323\left(\frac{T}{T_{sl}}\right)^2 + 0.068277826\left(\frac{T}{T_{sl}}\right)^3 - 0.00566593\left(\frac{T}{T_{sl}}\right)^4 \tag{B1}$$

CMU is used for the storage of these coefficients. The following values are used for the sea-level reference conditions T_{sl} and μ_{sl}:

$$T_{sl} = 518.688^{\circ}\ \mathrm{R} = 288.160\ \mathrm{K}$$

$$\mu_{sl} = 0.3711402\times10^{-6}\ \frac{(\mathrm{lbf})(\mathrm{sec})}{\mathrm{ft}^2} = 0.1777029\times10^{-4}\ \frac{(\mathrm{N})(\mathrm{sec})}{\mathrm{m}^2}$$

(2) Prandtl number of air as a function of nondimensional static temperature, $Pr = f(T/T_{sl})$:

$$Pr = 0.85570 - 0.234136\left(\frac{T}{T_{sl}}\right) + 0.1078624\left(\frac{T}{T_{sl}}\right)^2 - 0.0236214\left(\frac{T}{T_{sl}}\right)^3 + 0.00202863\left(\frac{T}{T_{sl}}\right)^4 \tag{B2}$$

CPR is used for the storage of these coefficients.

(3) Nondimensional thermal conductivity of air as a function of nondimensional static temperature, $(k/k_{sl}) = f(T/T_{sl})$:

$$\frac{k}{k_{sl}} = -0.03839323 + 1.2697427\left(\frac{T}{T_{sl}}\right) - 0.30911252\left(\frac{T}{T_{sl}}\right)^2 + 1.08743781\left(\frac{T}{T_{sl}}\right)^3 - 0.009674725\left(\frac{T}{T_{sl}}\right)^4 \tag{B3}$$

CTC is used for the storage of these coefficients. The following values are used for k_{sl}:

$$k_{sl} = 0.3202206\times10^{-2}\ \frac{(\mathrm{ft})(\mathrm{lbf})}{(\mathrm{ft})(\mathrm{sec})(^{\circ}\mathrm{R})} = 2.561796\times10^{-2}\ \frac{\mathrm{J}}{(\mathrm{m})(\mathrm{sec})(\mathrm{K})}$$

In LAMNAR the following curve fits are used:

(1) Stagnation point correlation number against wall temperature function, $n_{sp} = f(S_w)$:

$$n_{sp} = -0.08178 + 0.06670\ S_w - 0.03143\ S_w^2 + 0.00873\ S_w^3 + 0.01657\ S_w^4 - 0.01052\ S_w^5 \tag{B4}$$

CCN is used for the storage of these coefficients. This is an expression for the "two-dimensional body" curve in figure 6 of reference 4.

(2) Critical momentum-thickness Reynolds number against shape factor based on momentum thickness, $\left(R_\theta\right)_{cr} = f(K)$:

$$\left(R_\theta\right)_{cr} = \exp(5.47073 + 43.6053\ K + 227.198\ K^2 - 2067.04\ K^3 - 27\,172.7\ K^4 + 13\,691.2\ K^5) \tag{B5}$$

CRCR is used for the storage of these coefficients. This curve is plotted as figure 3 of reference 9.

(3) Difference between instability and transition momentum-thickness Reynolds numbers against mean shape factor based on momentum thickness, $\left(R_\theta\right)_{i,\Delta} = f\left(\overline{K}\right)$:

$$\left(R_\theta\right)_{i,\Delta} = 903.785 + 26\,365.0\,\overline{K} + 385\,695.0\,\overline{K}^2 + 1\,110\,440.0\,\overline{K}^3 - 45\,385\,300.0\,\overline{K}^4 - 77\,027\,600.0\,\overline{K}^5 \tag{B6}$$

CDIF is used for the storage of these coefficients. This curve is plotted as figure 4 of reference 9.

(4) Shear parameter against correlation number and wall temperature function, $l = f(n, S_w)$:

$$\begin{aligned} l = {} & 0.224488 - 1.91539\,n - 9.89400\,n^2 - 68.13488\,n^3 - 0.001512\,S_w \\ & - 1.47680\,S_w n - 10.52925\,S_w n^2 - 152.2781\,S_w n^3 - 0.002406\,S_w^2 \\ & - 0.015629\,S_w^2 n - 1.45743\,S_w^2 n^2 - 126.23395\,S_w^2 n^3 + 0.000752\,S_w^3 \\ & + 0.005385\,S_w^3 n + 0.917838\,S_w^3 n^2 - 39.40644\,S_w^3 n^3 \end{aligned} \tag{B7}$$

CSHR is used for the storage of these coefficients. This is a representation of the curves of figure 2 in reference 4. The double-valued portions of the curves for $S_w = -0.8$ and -1.0, as well as the reversal on the $S_w = 1.0$ curve, are not reflected in equation (B7).

(5) Reynolds analogy parameter against correlation number and wall temperature function, $C_f R_w / Nu_x = f(n, S_w)$:

$$\begin{aligned} \frac{C_f R_w}{Nu_x} = {} & 2.02056 - 19.7211\,n - 24.0495\,n^2 - 1400.002\,n^3 - 0.050979\,S_w \\ & - 10.88012\,S_w n + 62.4419\,S_w n^2 - 5081.76\,S_w n^3 - 0.014343\,S_w^2 \\ & + 2.279845\,S_w^2 n + 129.7008\,S_w^2 n^2 - 6257.848\,S_w^2 n^3 + 0.0270567\,S_w^3 \\ & - 1.677051\,S_w^3 n + 57.4397\,S_w^3 n^2 - 2552.266\,S_w^3 n^3 \end{aligned} \tag{B8}$$

CCRN is used for the storage of these coefficients. This is a representation of the curves in figure 3 of reference 4. The double-valued regions at the ends of the $S_w = 1.0$, -0.8, and -1.0 curves are not reflected in equation (B8).

(6) Thickness ratio parameter against correlation number and wall temperature function, $\delta_{tr}/\theta_{tr} = f(n, S_w)$:

$$\frac{\delta_{tr}}{\theta_{tr}} = 8.02829 - 4.30978\,n + 88.8244\,n^2 + 36.4336\,n^3 + 2.71101\,S_w$$

$$- 7.42259\,S_w n + 242.293\,S_w n^2 - 16.293\,S_w n^3 - 0.16394\,S_w^2$$

$$- 7.61942\,S_w^2 n + 286.9795\,S_w^2 n^2 + 64.11186\,S_w^2 n^3 - 0.16758\,S_w^3$$

$$- 3.70289\,S_w^3 n + 130.8107\,S_w^3 n^2 + 111.3276\,S_w^3 n^3 \qquad \text{(B9)}$$

CDTH is used for the storage of these coefficients. This is a representation of the curves in figure 8 of reference 4. The small double-valued portions of the curves on the figure are not reflected in equation (B9).

(7) In addition to these curve fits, LAMNAR uses another curve fit indirectly - momentum parameter as a function of correlation number and wall temperature function, $N = f(n, S_w)$:

$$N = 0.04631 + 5.43220\,n + 4.51903\,n^2 + 19.01831\,n^3 - 0.00367\,S_w + 2.25400\,S_w n$$

$$- 10.49775\,S_w n^2 + 62.76597\,S_w n^3 + 0.00481\,S_w^2 - 0.06672\,S_w^2\,n$$

$$- 12.71732\,S_w^2 n^2 + 115.00986\,S_w^2 n^3 + 0.00651\,S_w^3 - 0.20637\,S_w^3\,n$$

$$- 2.95270\,S_w^3 n^2 + 62.53113\,S_w^3 n^3 \qquad \text{(B10)}$$

This is a representation of figure 4 of reference 4. The momentum parameter (see eq. (1)) of this report) for a given S_w is assumed to be represented by the following linear function:

$$N = A + Bn \qquad \text{(B11)}$$

The coefficients A and B are calculated near statement 100 in LAMNAR. In these calculations, the terms in equation (B10) are used.

APPENDIX C

CHANGES TO PROGRAM FOR A GAS OTHER THAN AIR

BLAYER is easily altered so that it applies for gases other than air. The necessary changes are all made in the PRECAL subroutine. These changes are the following:

(1) At the beginning of PRECAL, new curve-fit coefficients must be read in by DATA statements for the CMU, CPR, and CTC arrays. These are the coefficients of the following curve fits (see appendix B):

$$\frac{\mu}{\mu_{sl}} = f\left(\frac{T}{T_{sl}}\right)$$

$$Pr = f\left(\frac{T}{T_{sl}}\right)$$

$$\frac{k}{k_{sl}} = f\left(\frac{T}{T_{sl}}\right)$$

If the number of coefficients changes from five in any case, this must be reflected both in the DATA statements and later in PRECAL in the calls on CURVFT where CMU, CPR, and CTC are used.

(2) The sea-level reference values of temperature (TSLE, TSLM), viscosity (MUSLE, MUSLM), and thermal conductivity (TCSLE, TCSLM) must be changed. These values are stored at the beginning of PRECAL after the DATA statements for CMU, CPR, and CTC.

(3) Near statement 10 in PRECAL, the two statements storing values into TCON should be removed. The computation of TR1 using TSLM would then become statement 10.

(4) Shortly after statement 210 in PRECAL, the computation of SUTHL(I) should be changed. A temperature-viscosity law of the following form is used in the program.

$$\frac{\mu}{\mu_0} = k_{su} \frac{T}{T_0} \tag{C1}$$

with the constant k_{su} having the famous Sutherland value for air

$$k_{su} = \left(\frac{T}{T_0}\right)^{1/2} \frac{T_0 + \mathcal{S}}{T + \mathcal{S}} \tag{C2}$$

where $\mathcal{S}$ is Sutherland's constant (TCON). SUTHL(I) is the k_{su} of formulas (C1) and (C2). A temperature-viscosity law agreeing with formula (C1) must be used in the program, but the computation of k_{su} would have to change for a gas other than air.

APPENDIX D

SYMBOLS

A, B	constants in eqs. (2) to (4) and (14)
a	speed of sound, ft/sec; m/sec
C_f	local skin friction coefficient
c_p	specific heat at constant pressure, (ft)(lbf)/(slug)($^\circ$R); J/(kg)(K)
f	function of transformed momentum thickness, eq. (7)
H	form factor, δ^*/θ
H_i	transformed form factor for adiabatic flow, also called incompressible form factor, $\int_0^{\delta_{tr}} \left[1 - \left(\frac{U}{U_e}\right)\right] dY_{tr} \Big/ \theta_{tr}$
H_{tr}	transformed form factor for nonadiabatic flow, $\delta^*_{tr}/\theta_{tr}$
h	enthalpy, (ft)(lbf)/slug; J/kg
K	shape factor based on momentum thickness, $(\theta^2/\nu_w)\,(du_e/dx)$
$\overline{K}$	mean shape factor based on momentum thickness, eq. (25)
k	coefficient of thermal conductivity, (ft)(lbf)/(ft)(sec)($^\circ$R); J/(m)(sec)(K)
k_{su}	constant in Sutherland's viscosity-temperature relation, eqs. (C1) and (C2)
L	total distance along boundary-layer surface, ft; m
l	shear-stress parameter, eqs. (20) and (B7)
M	Mach number
N	momentum parameter, eqs. (1) and (2)
Nu_x	local Nusselt number, $x\left(\frac{\partial T}{\partial y}\right)_w \Big/ (T_{aw} - T_w)$
n	correlation number (pressure gradient parameter), eqs. (4) and (14)
n_p	reciprocal of power in power law, eq. (36)
P	pressure, lbf/ft^2; N/m^2

Pr Prandtl number, $\mu c_p/k$

q heat transfer per unit area, (ft)(lbf)/(ft^2)(sec); J/(m^2)(sec)

R_w Reynolds number at the wall based on surface length, $u_e x/\nu_w$

R_θ Reynolds number based on momentum thickness, $u_e \theta/\nu_w$

$\mathscr{R}$ universal gas constant, (ft)(lbf)/(slug)($^\circ$R); J/(kg)(K)

S enthalpy function, $(h'/h'_0) - 1$

T temperature, $^\circ$R; K

T_{aw} adiabatic or recovery wall temperature, $^\circ$R; K

U transformed longitudinal velocity, $(a'_0/a_e)u$, ft/sec; m/sec

u longitudinal velocity parallel to boundary-layer surface in x-direction (see fig. 4), ft/sec; m/sec

X coordinate for input of surface geometry (see figs. 4 and 5), ft; m

X_{tr} transformed x-coordinate along body surface, ft; m:

$$X_{tr} = \int_0^x k_{su} \frac{P_e}{P'_0}\left(\frac{T_e}{T'_0}\right)^{1/2} dx \qquad \text{in laminar flow}$$

$$X_{tr} = \int_0^x \frac{T'_0}{\overline{T}}\left(\frac{T_e}{T'_0}\right)^{\frac{\gamma+1}{2\gamma-2}} dx \qquad \text{in turbulent flow}$$

x coordinate parallel to body surface in streamwise direction (see fig. 4), ft; m

Y coordinate for input of surface geometry (see figs. 4 and 5), ft; m

Y_{tr} transformed Y-coordinate normal to body surface, ft; m:

$$Y_{tr} = \left(\frac{T_e}{T'_0}\right)^{1/2} \int_0^y \frac{\rho}{\rho'_0}\, dy \qquad \text{for both laminar and turbulent flow}$$

y coordinate normal to body surface (see fig. 4), ft; m

γ	ratio of specific heats
δ	boundary-layer thickness, ft; m
δ_{tr}	transformed boundary-layer thickness, ft; m
δ^*	displacement thickness, $\int_0^{\delta} \left(1 - \frac{\rho u}{\rho_e u_e}\right) dy$, ft; m
δ^*_{tr}	transformed displacement thickness, $\int_0^{\delta_{tr}} \left(1 - \frac{U}{U_e} + S\right) dY_{tr}$, ft; m
η	normalized distance from wall, y/δ
θ	momentum thickness, $\int_0^{\delta} \frac{\rho u}{\rho_e u_e} \left(1 - \frac{u}{u_e}\right) dy$, ft; m
θ_{tr}	transformed momentum thickness, $\int_0^{\delta_{tr}} \frac{U}{U_e} \left(1 - \frac{U}{U_e}\right) dY_{tr}$, ft; m
λ	Pohlhausen shape factor based on boundary-layer thickness, $(\delta^2/\nu_w)(du_e/dx)$
μ	dynamic viscosity, (lbf)(sec)/ft^2; (N)(sec)/m^2
ν	kinematic viscosity, μ/ρ, ft^2/sec; m^2/sec
ρ	density, slug/ft^3; kg/m^3
τ	shear stress, lbf/ft^2; N/m^2

Subscripts:

cr	critical
e	value external to boundary layer
i	incompressible quantity
inst	instability
sl	sea-level value
sp	stagnation point value
tr	transformed quantity
tran	transition

w	wall or surface value
x	local value based on x
Δ	increment or difference in some quantity
0	free-stream value; condition external to boundary layer, usually near $x = 0$ (see fig. 6)
1, 2, 3, . . .	at station 1, 2, 3, . . .

Superscripts:

'	total or stagnation condition, or quantity based on total or stagnation condition
—	evaluated at reference conditions (see ref. 19)

REFERENCES

1. Katsanis, Theodore; and McNally, William D.: Revised FORTRAN Program for Calculating Velocities and Streamlines on a Blade-to-Blade Stream Surface of a Turbomachine. NASA TM X-1764, 1969.
2. Katsanis, Theodore; and McNally, William D.: FORTRAN Program for Calculating Velocities and Streamlines on a Blade-to-Blade Stream Surface of a Tandem Blade Turbomachine. NASA TN D-5044, 1969.
3. Katsanis, Theodore; and McNally, William D.: FORTRAN Program for Calculating Velocities in a Magnified Region on a Blade-to-Blade Stream Surface of a Turbomachine. NASA TN D-5091, 1969.
4. Cohen, Clarence B.; and Reshotko, Eli: The Compressible Laminar Boundary Layer with Heat Transfer and Arbitrary Pressure Gradient. NACA TR 1294, 1956.
5. Sasman, Philip K.; and Cresci, Robert J.: Compressible Turbulent Boundary Layer with Pressure Gradient and Heat Transfer. AIAA J., vol. 4, no. 1, Jan. 1966, pp. 19-25.
6. Reshotko, Eli; and Tucker, Maurice: Approximate Calculation of the Compressible Turbulent Boundary Layer with Heat Transfer and Arbitrary Pressure Gradient. NACA TN 4154, 1957.
7. Schlichting, Hermann (J. Kestin, trans.): Boundary Layer Theory. Sixth ed., McGraw-Hill Book Co., Inc., 1968.
8. Schlichting, H.; and Ulrich, A.: Zur Berechnung des Umschlages Laminar-Turbulent. Jahrbuch d. dt. Luftfahrtforschung, no. 1, 1942, pp. 8-35.
9. Granville, Paul S.: The Calculation of the Viscous Drag of Bodies of Revolution. Rep. 849, David Taylor Model Basin, July 1953.
10. Stewartson, K.: Correlated Incompressible and Compressible Boundary Layers. Proc. Roy. Soc. (London), Ser. A, vol. 200, no. 1060, Dec. 22, 1949, pp. 84-100.
11. Thwaites, B.: Approximate Calculation of the Laminar Boundary Layer. Aeron. Quart., vol. 1, Nov. 1949, pp. 245-280.
12. Bennett, J. A.; and Goradia, S. H.: Methods for Analysis of Two-Dimensional Airfoils with Subsonic and Transonic Applications. Rep. ER-8591, Lockheed-Georgia Co. (AROD-6074-1, DDC No. AD-637603), July 21, 1966.
13. Mager, Artur: Transformation of the Compressible Turbulent Boundary Layer. J. Aeron. Sci., vol. 25, no. 5, May 1958, pp. 305-311.

14. Tetervin, Neal; and Lin, Chig Chiao: A General Integral Form of the Boundary-Layer Equation for Incompressible Flow with an Application to the Calculation of the Separation Point of Turbulent Boundary Layers. NACA TR 1046, 1951.

15. Ludwieg, Hubert; and Tillmann, W.: Investigations of the Wall-Shearing Stress in Turbulent Boundary Layers. NACA TM 1285, 1950.

16. Becker, John V.: Boundary-Layer Transition on the N.A.C.A. 0012 and 23012 Airfoils in the 8-Foot High-Speed Wind Tunnel. NACA WR-L-682, 1940.

17. Walsh, J. L.; Ahlberg, J. H.; and Nilson, E. N.: Best Approximation Properties of the Spline Fit. J. Math. Mech., vol. 11, no. 2, 1962, pp. 225-234.

18. Rott, Nicholas; and Crabtree, L. F.: Simplified Laminar Boundary-Layer Calculations for Bodies of Revolution and for Yawed Wings. J. Aeron. Sci., vol. 19, no. 8, Aug. 1952, pp. 553-565.

19. Eckert, E. R. G.: Engineering Relations for Friction and Heat Transfer to Surfaces in High Velocity Flow. J. Aeron. Sci., vol. 22, no. 8, Aug. 1955, pp. 585-587.

NASA-Langley, 1970 —— 12 E-5256

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D. C. 20546
OFFICIAL BUSINESS

FIRST CLASS MAIL

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS A
SPACE ADMINISTRATION

POSTMASTER: If Undeliverable (Section 15
Postal Manual) Do Not Retu

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

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

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

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

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

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

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

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

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

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

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546