A.R.A.P. WORKING PAPER NUMBER 76-2

GYC: A PROGRAM TO COMPUTE THE TURBULENT BOUNDARY LAYER ON A ROTATING CONE

Roger D. Sullivan

August 1976

Prepared under Contract No. NAS2-8014

by

Aeronautical Research Associates of Princeton, Inc.
50 Washington Road, Princeton, New Jersey 08540

for

AMES RESEARCH CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

TABLE OF CONTENTS

1.	INTRODUCTION	1
2.	ANALYSIS	3
3.	NUMERICAL METHOD	15
4.	AUXILIARY COMPUTATIONS	25
5.	CONTROL OF THE MESH	29
6.	PROGRAM STRUCTURE	32
7.	INPUT AND OUTPUT	37
8.	FINAL REMARKS	42
9.	REFERENCES	43
	APPENDIX A. THE EQUATIONS BEFORE MODELING	44
	APPENDIX B. THE MODELS	48
	APPENDIX C. THE EQUATIONS AFTER MODELING	52
	APPENDIX D. THE FINAL EQUATIONS	56
	APPENDIX E. TENSR NOTATION EXAMPLES	68
	APPENDIX F. GLOSSARY	70
	TABLE 1. INPUT CARDS	96
	TABLE 2. EDGE VALUES AND WALL VALUES	98
	TABLE 3. INITIAL CONDITIONS, FIRST SET	99
	TABLE 4. INITIAL CONDITIONS, SECOND SET	99
	TABLE 5. DEPENDENT VARIABLES	100

1. INTRODUCTION

The Ames Research Center of NASA is engaged in an effort to enhance its understanding of the behavior of turbulent boundary layers and to improve its ability to predict their effect on bodies in supersonic flow. As a part of this effort, it has entered into a contract with A.R.A.P. Under this contract, A.R.A.P. has delivered to ARC a computer program, GYC, which is capable of computing the properties of a compressible turbulent boundary layer on a rotating axisymmetric cone-cylinder body, according to the principles of invariant modeling. The program, which was delivered at the beginning of 1976, is to be extended during the current year to include the calculation of the turbulence scale by a differential equation.

In the meantime, GYC has been in operation on the ARC CDC-7600 computer and has undergone several corrections and improvements as a result of the experience gained with it there and at A.R.A.P. However, so far there has been lacking a comprehensive document giving the theoretical basis for the program and the method of implementation, as well as information on its operation. This paper is intended to fulfill this need, but in a tentative way. When an extended version of the program is delivered in a few months, it is hoped that a report describing it — an improved version of this — will be included. The anticipated improvements will be generated largely by the reactions to this paper.

Section 2 is a concise review of the means used to derive the equations. The models used are listed but not justified. Some background on the models can be found in Reference 1 for the models that apply to incompressible flows, and in Reference 2 for others.

In Section 3, the numerical method used to solve the equations is set forth. Some extra computations are described in Section 4, while the means for controlling the finite-difference mesh are reported in Section 5.

Sections 2 through 5 are essentially free of program considerations, which are covered in Sections 6 and 7. Section 6 describes GYC as a whole and then considers in more or less detail the main sections of the program. The part concerned with the numerical integration of the equations is given the most attention. External aspects of the program operation are described in Section 7. The paper is brought to a close with a few remarks in Section 8.

The full equations under consideration are set forth in appendices. An extensive glossary of Fortran names and Fortran-like names is included as Appendix F.

2. ANALYSIS

The basic equations used in this study are the following:

Continuity

$$\rho_t + \left(\rho u^{\ell}\right)_{,\ell} = 0$$

Momentum

$$\rho u_{i_t} + \rho u^{\ell} u_{i,\ell} = -p_{,i} + \tau^{\ell}_{i,\ell}$$

where

$$\tau^{j}_{i} = \mu\left(u_i^{,j} + u^{j}_{,i}\right) + \mu^{*}\delta^{j}_{i}u^{\ell}_{,\ell}$$

Enthalpy Version of Energy

$$\rho h_t + \rho u^{\ell} h_{,\ell} = p_t + u^{\ell} p_{,\ell} + \tau^{m}_{\ell} u^{\ell}_{,m} + \left(kT_{,\ell}\right)^{,\ell}$$

The thermodynamic relations of a "calorically perfect" gas are used

$$p = \rho RT$$

$$h = c_p T$$

where R and c_p are constants.

Writing the dependent variables in these equations as the sum of mean and fluctuating parts ($\rho = \bar{\rho} + \rho'$, for example), we can deduce equations for the mean quantities and for the second-order correlations of the fluctuations by protracted manipulations. Two assumptions are made immediately:

1) Fourth-order correlations are neglected.

2) Third-order correlations involving μ , μ^{*} , or k are neglected.

The resulting equations for a steady mean flow are displayed in Appendix A.

Fluctuations in the transport parameters are handled by assuming

$$\mu' = \mu_T T' \qquad \mu'_{,i} = \mu_T T'_{,i}$$

$$\mu^{*\prime} = \mu^*_T T' \qquad \mu^{*\prime}_{,i} = \mu^*_T T'_{,i}$$

$$k' = k_T T' \qquad k'_{,i} = k_T T'_{,i}$$

where the subscript T denotes a derivative with respect to temperature evaluated at $\bar{T}$.

The models used to close the equations are given in Appendix B. As in previous A.R.A.P. models, these are invariant under coordinate and Galilean transformations and, of course, are dimensionally consistent. The goal of simplicity has been sacrificed to some extent in favor of generality. An obvious feature of these models is that many of them are expressed in terms of quantities that are themselves modeled. This results in some of the parameters (coefficients) appearing only in products with other parameters in the applications. (This is exemplified by the models for $\overline{p'h'_{,i}}$ and $\overline{p'\rho'_{,i}}$ where the substitution has already been carried out.) Thus some of the parameters are redundant, but they are left that way for the sake of flexibility in case it is found necessary to modify them in the future.

Three- or four-character names, suitable for direct use in Fortran, have been chosen for the modeling parameters except for a , b , and β , in the velocity-dissipation model, which are left in that form out of nostalgia.

It will be seen later that all of the terms containing models for correlations involving the divergence of the velocity

fluctuations, $u'^{\ell}_{,\ell}$, drop out. For this reason, the following parameters do not appear in the final equations: BDV, VBS, WMD, WWU2, WWD1, WWD2, WWH2, and WWR2. In addition, terms involving PMH, PMR, PTHM, and PTUM do not occur in the final set. Of the remaining 38 independent parameters, about a quarter are set to zero until better information on them is available, and the majority of the others are evaluated by analogy. Thus 0.1 has been found a good value for VUU in constant density tests; VRU, VUH, VRR, VRH, and VHH don't occur in constant-density modeling but, by analogy, 0.1 is used for all of them, until better information is available.

So far the equations have been written using mechanical units for heat quantities. In order to introduce additional flexibility, a conversion factor is introduced, allowing arbitrary units for heat. This factor is

$$\text{GMOMS} = 1/J$$

where J is the mechanical equivalent of heat. For example, if English units are used with heat measured in Btu, GMOMS = 0.001285 Btu/ft-lb (= 1/(778 ft-lb/Btu)). If mechanical units are used for heat, GMOMS = 1 . It is often desirable to make the equations nondimensional. If this is done using u_r and h_r as reference values for velocity and enthalpy per unit mass, the nondimensional version of GMOMS is

$$\text{GMOMS}_{ND} = \text{GMOMS}\,\frac{u_r^2}{h_r} = \frac{u_r^2}{Jh_r} = (\gamma - 1)M_r^2$$

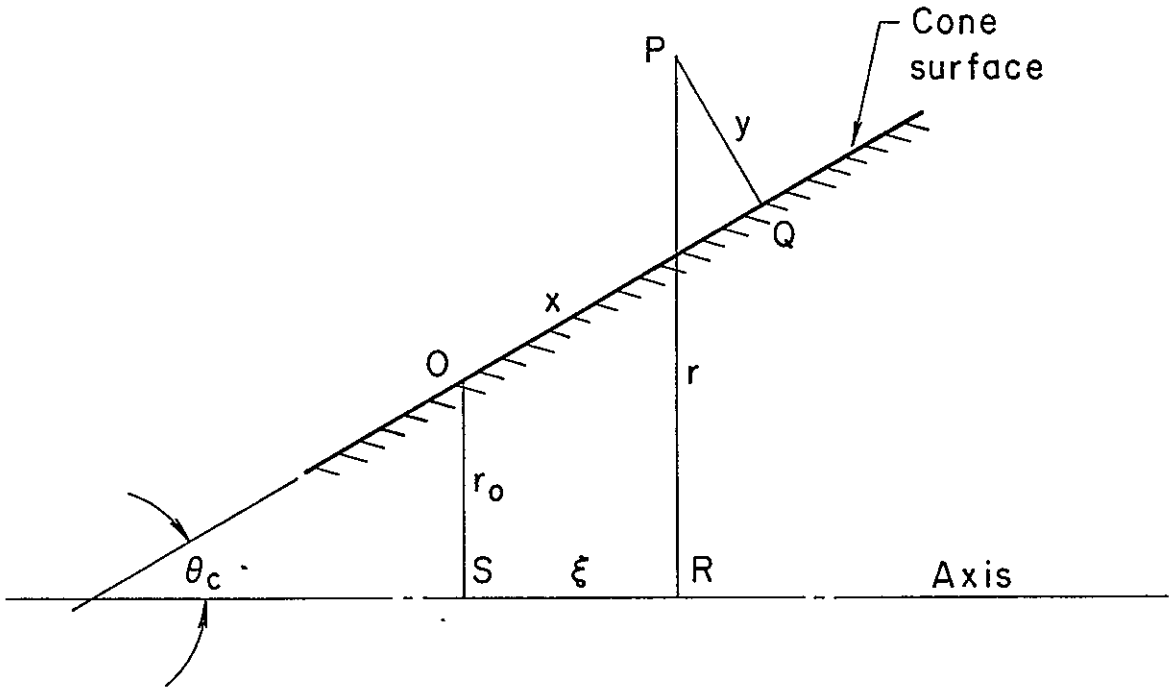
(Hence the name GMOMS - Gamma Minus One M_r Squared.)

The equations with the models substituted are exhibited in Appendix C. Here the notation is that devised for input to TENSR, A.R.A.P.'s software for the expansion of tensor equations. This notation is like Fortran in several respects. Thus,

variable names may consist of more than one character; multiplication is always made explicit with a * ; multiplication and division proceed from left to right so that A/B/C*D is to be evaluated as ((A/B)/C)*D = (A*D)/(B*C) . On the other hand, superscripts and subscripts are represented in a manner completely different from the Fortran representation of subscripts. Superscripts are preceded by " ; subscripts by ' . Superscripts and subscripts consist of a single letter or digit only; two or more together represent separate indices. The metric tensor is denoted by @ ; the Kronecker delta by & . Covariant differentiation is denoted by a , preceding an index; and "contravariant differentiation" (that is, covariant differentiation with the index raised) by ! preceding an index. All these forms are illustrated in Appendix E.

The Glossary (Appendix F) gives the meaning of the variable names used. Thus, for RS, US, and RU in the first equation of Appendix C, we find $\bar{\rho}$, $\bar{u}$, and $\overline{\rho' u'}$ respectively. With the help of Appendix E, we see that the first equation of Appendix C is a transcription of the first equation in Appendix A, as it should be since there are no terms needing modeling. Proceeding thus step by step, the formidable expressions of Appendix C can gradually be seen to make sense.

A cross-section of the axisymmetric coordinate system is shown in the sketch on the following page.



The coordinates of the point P are $x^1 = x = OQ$, $x^2 = y = PQ$, $x^3 = \phi$ = the azimuthal angle. The half angle of the cone, θ_c, and the "origin radius," r_o, are parameters of the system. The coordinates x and y are related to the cylindrical coordinates $\xi = SR$ and $r = PR$ by

$$\xi = \cos \theta_c x - \sin \theta_c y$$

$$r = r_o + \sin \theta_c x + \cos \theta_c y$$

It is understood that $y = 0$ always represents the cone surface. The location of the origin in the x direction is quite arbitrary and hence, in general, the value of r_o is arbitrary. The exception is the limiting case of a cylindrical surface ($\theta_c = 0$ or $\theta_c = 180°$) for which r_o represents the radius of the cylinder.

Note that for θ_c in the second and third quadrants, the flowfield (for which $y > 0$) is on the inside of a conical surface.

It is convenient to have the equations apply also for a Cartesian coordinate system (x,y,z). To accomplish this, another parameter is defined by

$$\text{DIMFL} = \begin{cases} 1. & \text{axisymmetric} \\ 0. & \text{Cartesian} \end{cases}$$

Now let

$$S = \text{DIMFL}* \sin \theta_c$$

$$C = \text{DIMFL}* \cos \theta_c$$

$$R = \text{DIMFL}*r_o + Sx + Cy + 1. - \text{DIMFL}$$

Then the metric tensor can be written

$$g_{ij} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & R^2 \end{pmatrix}$$

The nonvanishing Christoffel symbols of the second kind are

$$\$^1_{33} = -SR$$

$$\$^2_{33} = -CR$$

$$\$^3_{31} = \$^3_{13} = S/R$$

$$\$^3_{32} = \$^3_{23} = C/R$$

These apply for both Cartesian coordinates and the axisymmetric coordinates of the sketch.

Note that

$$R = \begin{cases} r & \text{axisymmetric} \\ 1. & \text{Cartesian} \end{cases}$$

so that it not only has different meaning but even different dimensions in the two cases.

The next stage of the analysis was carried out by TENSR, A.R.A.P.'s software for the expansion of tensor expressions. The following information was supplied to TENSR:

1) The values of the metric tensor and the Christoffel symbols of the second kind in terms of R , S , and C , as given above.

2) The information that terms involving derivatives with respect to x^3 (ϕ) are zero, since the flow is assumed to be axisymmetric.

3) The substitutions required so the output involving the mean velocity could be expressed in terms of the physical components, $\bar{u}$, $\bar{v}$, and $\bar{w}$ (or US , VS , and WS) , instead of the covariant and contravariant components, and similarly for correlations involving u' .

4) The equations of Appendix C.

The output for the whole set of 18 equations consisted of 9843 terms. This number is somewhat deceptive, since TENSR is naive algebraically and does not combine like terms.

These equations were submitted to more A.R.A.P. software known as TATTR. The first stage of TATTR does combine terms and reduced the 18 equations to a total of only (!) 4348 terms.

The main function of TATTR is to drop terms according to order-of-magnitude assumptions. Its operation in this case can be described as follows.

The equations were interpreted to be written in nondimensional form. The (nondimensional) boundary-layer thickness was assumed to have a small value, δ . The mean quantities and the radius, R , were assumed to be of order one except for the normal component of the velocity, $\bar{v}$ (VS) , which was taken to be of order δ . Derivatives with respect to x and y were assumed to be of order one and δ^{-1} , respectively. The transport coefficients μ , μ^* , and k were assumed to be of order δ^2 . All of the above corresponds to standard boundary layer assumptions. In addition, correlations involving a velocity fluctuation were taken to be of order $\delta^{1/2}$ and those involving ρ' or h' of order $\delta^{1/3}$ for each such appearance. (For example, $\overline{u'u'}$, $\overline{u'v'}$, $\overline{\rho'w'}$, and $\overline{h'h'}$ (UU , UV , RW , and HH) were assigned orders δ , δ , $\delta^{5/6}$, and $\delta^{2/3}$, respectively.) The scale length Λ was assumed to be of order $\delta^{4/3}$. Finally, GMOMS , which, as has been seen, is $(\gamma - 1) M_r^2$ when the equations are interpreted nondimensionally, was taken to be of order $\delta^{-1/3}$. All quantities not mentioned, including the modeling parameters, were assumed to be of order one. These ratios were arrived at after some trial and error, using Cartesian coordinates. They were chosen so as to insure the retention of terms known to be important, while allowing the elimination of terms known to be unimportant.

TATTR computed the exponent, T , representing the order of magnitude δ^T , for each term of each equation, given the information in the paragraph above concerning the factors that can appear. For example, the continuity equation was analyzed as follows (where :1 and :2 stand for partial differentiation with respect to x^1 and to x^2).

	T
(RS*US):1	0
+ (RS*VS):2	0
+ S*RS/R*US	0
+ C*RS/R*VS	1
+ (RU):1	5/6
+ (RV):2	-1/6
+ S/R*RU	5/6
+ C/R*RV	5/6
= 0	

The final stage of TATTR dropped terms for which the value of T is greater than a maximum appropriate for the particular equation. In the case of the continuity equation, the cutoff value is zero, so the final form is

$$(RS*US):1 + (RS*VS):2 + S*RS/R*US + (RV):2 = 0$$

or

$$(\bar{\rho}\bar{u})_x + (\bar{\rho}\bar{v})_y + \frac{S}{R}\bar{\rho}\bar{u} + (\overline{\rho' v'})_y = 0$$

The whole set of equations was reduced from 4348 terms to 529 terms by the final stage of TATTR. The second component (normal to the surface) of the momentum equation reduced to

	T
-C*RS/R*WS*WS	0
+ (RS*VV):2	0
= -(PS):2	-1

The values of T indicate that, according to TATTR, the remaining terms are seriously out of balance. But it is well known in boundary-layer theory, though not to TATTR, that the derivative of pressure in the normal direction is not large like other normal derivatives; (PS):2 is really of order one $\left(\delta^0\right)$, not δ^{-1}. Therefore, the terms containing (PS):2 in

the other equations were dropped manually if the T value for that term was within one of the cutoff value for the equation. In this way, 63 terms were dropped leaving 466 in the 18 equations. Two of the final equations have been presented above. The others, which have been slightly edited manually, are presented in Appendix D. (The reason for the organization of the equations into "passes" is given in the next section.)

As has been indicated, the criteria used to drop terms were arrived at pragmatically. This can be defended on the basis that (1) the process is more systematic than that heretofore used for modeled equations for compressible turbulent flow and (2) the results are not sensitive to the details of the assumptions used — e.g., it makes little difference whether ρ' is assumed to be of order $\delta^{1/3}$ or order $\delta^{1/2}$. Nevertheless, it would be valuable to find a rational basis for completely specifying the criteria. Such a basis should take into account the fact that the balance of terms differs markedly in different parts of the boundary layer.

Some features of the final set of equations should be noted. (1) All terms involving models for $u'_{,\ell}^{\ell}$ correlations, as noted above, have dropped out. This is a result, of course, not only of the criteria used, but also of the particular models used for such correlations. (2) All terms involving the second coefficient of viscosity, μ^*, have dropped out. (For the most part, these are terms that also involve $u'_{,\ell}^{\ell}$.) (3) If S and C are both zero (as they are if DIMFL = 0), the equations consisting of the remaining terms are each homogeneous in R. This fact (which is true of the equations before terms are dropped) shows that the change in dimensionality of R as DIMFL changes from one to zero does not violate any principles.

A new equation

BLAM = GYCSZ

has been added to the set and included in Appendix D. Its presence here is merely symbolic. BLAM (Λ) is evaluated as a known function of y and the gross features of the flow, quantified by input parameters (the scale parameters). It is included here and in the program as a pseudo-unknown, satisfying a "differential" equation of zero order which becomes a "difference" equation of zero order. The reason for this treatment is that it is intended to introduce a true differential equation for Λ and incorporate its solution in the program in the near future. The program modification will be much simplified by having Λ already treated as an unknown.

The sixteen differential equations of Appendix D are parabolic in the sense that they have first-order derivatives with respect to x and second-order derivatives (as well as first-order) with respect to y. Thus x is the time-like variable and y is the space-like variable. The dependent variables determined by the equations are referred to as parabolic variables.

The boundary conditions needed for this set consist of the initial conditions, the wall conditions, and the free-stream conditions. The initial conditions consist of profiles of the parabolic variables, that is, their values as functions of y at the x station where computation is to start. The wall conditions on all the turbulence correlations and on $\bar{u}$ are that they are zero there. The wall condition on $\bar{w}$ is that it is equal to the lateral velocity of the cone surface, that is, the product of the angular velocity and the local cone radius. The wall condition on $\bar{h}$ is either its value at the wall or the value of its gradient (a measure of the heat flux) at the wall. Either of these may be functions of x.

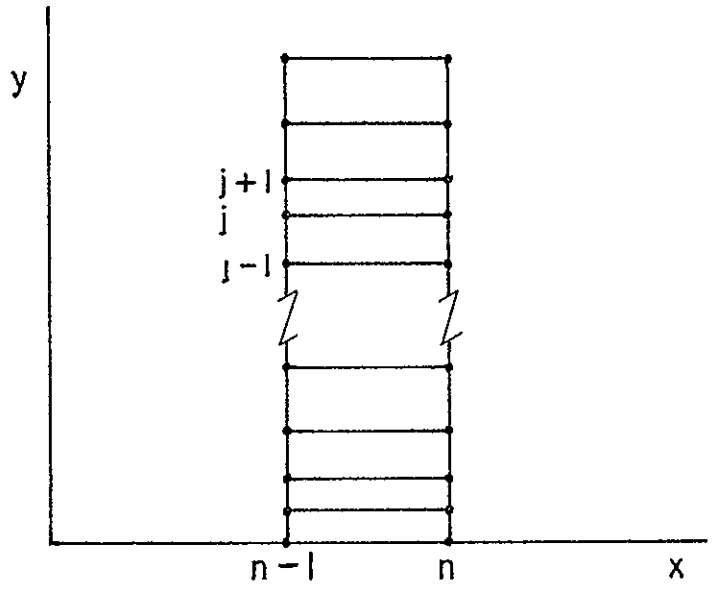
The free-stream conditions, or "edge conditions," are that the parabolic variables approach given values asymptotically. For the turbulence correlations and for $\bar{w}$, the given values are constant in x, but for $\bar{u}$ and $\bar{h}$ they may be functions

of x . In practice the concept "approach asymptotically" is handled as follows. At the largest value of y currently used in the computation, the boundary condition $\frac{\partial f}{\partial y} = 0$, where f stands for any of the unknowns, is used in solving the equations. The resulting value of f at that y is compared with the given value and if they agree within a specified tolerance, the given value is substituted for the calculated value (for the sake of uniformity and neatness) and the solution continues. If the tolerance is not satisfied, for any of the parabolic variables, the maximum value of y is increased by adding another point, and the process is repeated.

These boundary conditions are not the most general that can be posed for the system of equations; rather they are the conditions provided for in the GYC program.

3. NUMERICAL METHOD

A finite-difference mesh is introduced in the x, y plane as shown in the sketch.



The time-like variable, x, is indexed by n and the space-like variable, y, by j. The spacing is not uniform in either direction; furthermore, the spacing in the y direction is adjusted as the solution proceeds in the x direction, as described in Section 5.

At the beginning of each x step, the values of the dependent variables are known at $x = x_{n-1}$ for all y_j. Derivatives with respect to x are approximated by

$$f_x = \frac{f_j^n - f_j^{n-1}}{\Delta x} \qquad (1)$$

where

$$f_j^n = f\left(x_n, y_j\right)$$

$$\Delta x = x_n - x_{n-1}$$

For derivatives with respect to y, some additional notation is convenient:

$$h_+ = y_{j+1} - y_j$$

$$h_- = y_j - y_{j-1}$$

$$h_t = y_{j+1} - y_{j-1}$$

$$H_+ = \frac{h_-}{h_+ h_t}$$

$$H_- = \frac{h_+}{h_- h_t}$$

$$G_+(g) = \frac{g_{j+1}^{n-1} + g_j^{n-1}}{h_+ h_t}$$

$$G_-(g) = \frac{g_j^{n-1} + g_{j-1}^{n-1}}{h_- h_t}$$

Then we have

$$f_y = H_+\left(f_{j+1}^n - f_j^n\right) + H_-\left(f_j^n - f_{j-1}^n\right) \tag{2}$$

$$\left(gf_y\right)_y = G_+(g)\left(f_{j+1}^n - f_j^n\right) - G_-(g)\left(f_j^n - f_{j-1}^n\right) \tag{3}$$

If g is constant, equation (3) can be written

$$f_{yy} = \frac{2}{h_+ h_t}\left(f_{j+1}^n - f_j^n\right) - \frac{2}{h_- h_t}\left(f_j^n - f_{j-1}^n\right) \tag{4}$$

One way of arriving at equations (2) and (4) is by passing a parabola through the three points $\left(y_{j-1}, f^n_{j-1}\right)$, $\left(y_j, f^n_j\right)$, $\left(y_{j+1}, f^n_{j+1}\right)$ and evaluating its derivatives at y_j. Equation (3) is an obvious generalization of equation (4) except that g is evaluated at x_{n-1} instead of at x_n. This is done to linearize the numerical procedure since g, in general, depends on the unknowns. The other nonlinear terms in Appendix D are handled similarly; that is, in any product, at most one factor is evaluated at x_n while the rest of the term is evaluated at x_{n-1}. This artifice has been successfully used in similar equations for several years at A.R.A.P. Of course, it is necessary to keep Δx small enough so that the changes in the dependent variables are small in each step. The linearized system is in the form referred to in the literature as the (complete, or fully) implicit method for the numerical solution of parabolic differential equations.

For terms of the form $\bar{\rho}\bar{v}f_y$, equation (2) is optionally replaced by an upwind differencing formulation, described in Note 4 of the Glossary (Appendix F).

The equations are coupled, but the finite-difference versions can be decoupled to any extent desired by the same device that is used to make them linear. That is, terms or factors which are unknowns if evaluated at x_n become known if evaluated at x_{n-1}. Further, if some unknowns are solved for before others, they become known at both x_{n-1} and x_n for the later solutions. If m equations are solved at once, at each point of the flowfield a matrix of size $m \cdot m$ must be inverted and two matrix multiplications of $m \cdot m$ matrices must be performed. Since these procedures require of the order of m^3 operations each, there is strong motivation for keeping m down. On the other hand, experience at A.R.A.P. has shown that some equations are better solved together in order to have stability and well behaved solutions with reasonable step sizes.

For GYC, these considerations have led to the division of the solution process, for each step in x, into six passes. In the first pass, $\bar{w}$ and $\overline{v'w'}$ are determined so $m = 2$. In the rest of the passes $m = 3$: $\bar{u}$, $\overline{u'v'}$, and $\overline{u'w'}$ are determined in the second; $\overline{u'u'}$, $\overline{v'v'}$, and $\overline{w'w'}$ in the third; $\bar{h}$, $\overline{h'v'}$, and $\overline{h'h'}$ in the fourth; $\overline{\rho'u'}$, $\overline{\rho'v'}$, and $\overline{\rho'w'}$ in the fifth; and $\overline{h'u'}$, $\overline{h'w'}$, and Λ in the last.

In each pass, the m unknowns are thought of as forming a vector, ϕ_j, of m elements at each y_j (for $x = x_n$). Thus in the first pass

$$\phi_j = \begin{pmatrix} \bar{w}_j^n \\ \overline{v'w'}_j^n \end{pmatrix}$$

in the second

$$\phi_j = \begin{pmatrix} \bar{u}_j^n \\ \overline{u'v'}_j^n \\ \overline{u'w'}_j^n \end{pmatrix}$$

etc.

Then the finite-difference equations can be written

$$A_j\phi_{j-1} + B_j\phi_j + C_j\phi_{j+1} = d_j \tag{5}$$

where A_j, B_j, and C_j are square matrices of order m and d_j is a vector of dimension m. The elements are known functions of the dependent variables at x_{n-1} (and in a few instances at x_n for variables determined on previous passes), the independent variables, and, through equations (1)-(4), the mesh spacing.

Equation (5) applies for $2 \le j \le J - 1$ where $J = JTOP$ is the index for the largest y in use and $j = 1$ corresponds to the wall ($y_1 = 0$) . The finite-difference version of the edge boundary condition, that the derivative with respect to y be zero for each variable, can be expressed as

$$\phi_{J-1} - \phi_J = 0 \tag{6}$$

The wall boundary conditions can be written

$$\phi_1 = d_1 \tag{7}$$

except when $\bar{h}$ is involved and its gradient at the wall is specified.

When the gradient of $\bar{h}$ at the wall is specified, the boundary condition is handled this way. It is required that there be no blowing so both components of the mean velocity, as well as all correlations, are zero at the wall. Then, on the wall the differential equation for $\bar{h}$ (the first equation of the fourth pass) reduces to

$$0 = \bar{\mu}\left[\bar{u}_y^2 + (R\bar{w})_y\left(\frac{\bar{w}}{R}\right)_y\right] + \left(\frac{\bar{k}}{c_p}\bar{h}_y\right)_y$$

assuming mechanical units so $GMOMS = 1$. The condition on the gradient is written

$$\bar{h}_y = g$$

where g is the specified value. These two equations are written in finite difference form using three points, the wall, $y_1 = 0$, the first point off the wall, y_2 , and a "phantom" point, y_0 , inside the wall. (Extrapolation is used to evaluate "known" quantities at the phantom point.) Thus there are two equations involving the unknowns h_0^n , h_1^n , h_2^n .

Eliminating h_0^n between these equations leaves one equation which can be written

$$\alpha h_1^n + \beta h_2^n = \gamma \tag{8}$$

where α , β , and γ depend on the known variables and the mesh spacing at the wall.

In any case, we conclude from equations (7) and (8) that the wall boundary conditions for a pass can be written

$$B_1\phi_1 + C_1\phi_2 = d_1 \tag{9}$$

where B_1 is diagonal and C_1 is zero except in the fourth pass when the gradient of $\bar{h}$ is specified, in which case the upper left element of C_1 is the β of equation (8) and the rest of C_1 is zero.

Similarly, equation (6) can be written

$$A_J\phi_{J-1} + B_J\phi_J = d_J \tag{10}$$

where A_J is the unit matrix, B_J is the negative of the unit matrix, and d_J is zero.

Equations (5), (9), and (10) can be written

$$\begin{pmatrix} B_1 & C_1 & & & & & & \\ A_2 & B_2 & C_2 & & & & & \\ & A_3 & B_3 & C_3 & & & & \\ & & \cdot & \cdot & \cdot & & & \\ & & & \cdot & \cdot & \cdot & & \\ & & & & \cdot & \cdot & \cdot & \\ & & & & & A_{J-1} & B_{J-1} & C_{J-1} \\ & & & & & & A_J & B_J \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \cdot \\ \cdot \\ \cdot \\ \phi_{J-1} \\ \phi_J \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \cdot \\ \cdot \\ \cdot \\ d_{J-1} \\ d_J \end{pmatrix}$$

or

$$M\Phi = D$$

where M is a tridiagonal matrix of order J, each element of which is an $m \cdot m$ matrix, and Φ and D are vectors of dimension J, each element of which is a vector of dimension m.

This set is solved by a standard algorithm for tridiagonal systems. The process consists of two parts called the upsweep and the downsweep. In the upsweep the values of auxiliary matrices, Γ_j, and auxiliary vectors, $\bar{\phi}_j$, are computed by

$$\Gamma_1 = B_1^{-1} C_1$$

$$\bar{\phi}_1 = B_1^{-1} d_1$$

$$\Gamma_j = \left(B_j - A_j \Gamma_{j-1}\right)^{-1} C_j \qquad j = 2, 3, \ldots, J-1 \qquad (11)$$

$$\bar{\phi}_j = \left(B_j - A_j \Gamma_{j-1}\right)^{-1} \left(d_j - A_j \bar{\phi}_{j-1}\right) \qquad j = 2, 3, \ldots, J \qquad (12)$$

In the downsweep the solution is obtained by

$$\phi_J = \bar{\phi}_J$$

$$\phi_j = \bar{\phi}_j - \Gamma_j \phi_{j+1} \qquad j = J-1, J-2, \ldots, 1 \qquad (13)$$

(A_1 and C_J do not appear in the system. If we define $A_1 = C_J = 0$, equations (11) and (12) can be used for $j = 1, 2, \ldots, J$; equation (13) can be used for $j = J, J-1, \ldots, 1$; and the special equations for $j = 1$ and $j = J$ are not needed.)

As mentioned in the previous section, the edge boundary condition is really asymptotic. The imposition of equation (6) or (10) at y_J is only tentative. At the end of the upsweep, the values of $\phi_J = \bar{\phi}_J$ are compared with the known free-stream values. If they are not equal within specified tolerances, J is increased by one; the already known solutions are suitably extrapolated; A_j, B_j, C_j, and d_j for $j = J_0$, the old value of J, are recomputed using expressions valid for $j < J$, the upsweep is redone for $j = J_0$ and done for $j = J$; and the test is repeated. When the tolerances are satisfied, the given free-stream values are substituted for the computed values at J (from which they differ by less than the tolerances) and the downsweep is performed.

Consider a regime where $\bar{w} = 0$, the turbulence is negligible, and y derivatives are negligible. The equations for the means reduce to

$$\bar{\rho}\bar{u}\bar{u}_x = -\bar{p}_x \tag{14}$$

$$\bar{\rho}\bar{u}\bar{h}_x = \bar{u}\bar{p}_x \tag{15}$$

(taking GMOMS = 1). If the edge conditions on $\bar{u}$, $\bar{h}$, and $\bar{p}$ satisfy these relations ($\bar{\rho}$ being determined by the equation of state), this handling of the free-stream boundary conditions works well. Otherwise, an unreasonable number of points may be added, satisfying finite-difference versions of equations (14) and (15) at the added points while trying to match conflicting given edge values. In this case, it is probably not right to assume y derivatives are negligible; but if they are not, it isn't clear how the edge of the calculation field should be determined.

If there is free-stream turbulence, another situation arises. Consider the simplest case with $\bar{w} = 0$, the other mean quantities constant, y derivatives negligible, and

$$\overline{u'^i u'_j} = \delta^i_j q^2/3$$

Then it is found that q^2 is governed by

$$\bar{\rho}\bar{u}\left(q^2\right)_x = -\frac{2\bar{\mu}}{\Lambda^2}\left(a + b\frac{\bar{\rho}q\Lambda}{\bar{\mu}}\right)q^2$$

That is, dissipation will make free-stream turbulence decrease downstream unless $\Lambda \to \infty$, according to the equations being used. There is no provision in the program as it stands for the turbulence-level edge condition to vary with x so excessive points will be added if free-stream turbulence is specified unless Λ is made to increase for large y. The scale parameters which control the calculation of Λ allow for such increase.

Excessive adding of points at the edge can always be avoided by increasing the tolerance used to check the matching of edge conditions for the variables giving trouble. This may give results of questionable validity.

There are three dependent variables that are not determined from parabolic equations. These are $\bar{p}$, $\bar{\rho}$, and $\bar{v}$. The mean pressure is calculated at $x = x_n$ during the downsweep of the first pass from the normal momentum equation displayed in Section 2. The integration starts with a given edge value of $\bar{p}$ (which may be a function of x) and proceeds inward to the wall. The mean density is calculated at $x = x_n$ during the downsweep of the fourth pass from the mean equation of state, which can be written

$$\bar{\rho} = \frac{c_p\bar{p}}{R(\bar{h} - \overline{h'h'}/\bar{h})}$$

The normal component of the mean velocity is calculated at $x = x_n$ after the completion of all the passes, using the

continuity equation which is also displayed in Section 2. The integration starts with a given wall value of $\bar{v}$ (which may be a function of x) and proceeds outward to the edge. Trapezoidal integration is used for both $\bar{p}$ and $\bar{v}$.

Another quantity, namely $\bar{u}_x$, is grouped with $\bar{p}$, $\bar{\rho}$, and $\bar{v}$. Since its value for the current step is not available in the first pass, its value saved from the previous step is used there. The contribution of the terms in which it appears is expected to be small, so the error involved in lagging it this way should also be small. The value from the current step is set aside for use in the next step as part of the downsweep of the second pass.

4. AUXILIARY COMPUTATIONS

Some functionals of the dependent variables are calculated at the completion of each x step (as well as during the initialization stage). These calculations serve three main purposes: (1) they provide output which helps the user interpret the results of the program, (2) they provide data used in determining the finite-difference mesh for the next step, and (3) they provide data on which the calculation of the local value of Λ is based. In the last case, the process is not really auxiliary since Λ is basic to the modeling and strongly influences the course of the solution.

Many of the calculations are repeated for each of the parabolic variables. To avoid long descriptions, such procedures are set forth here as if there were only one independent variable, named F.

These quantities include: (1) the "max" which is the maximum over y_j of $|F|$ at x_n, (2) the value of y at which the max occurs, (3) the "global max" which is the maximum over y_j and x_n (up to the current x_n) of $|F|$, (4) the "max change" which is the maximum over y_j of

$$\left|F\left(x_n, y_j\right) - F\left(x_{n-1}, y_j\right)\right|$$

(5) the integral (according to the trapezoidal rule)

$$I(F) = \int_o^{y_J} |F| dy$$

(6) the "integral spread" which is $I(F)$ divided by the max of F, and (7) the "percent spread" which is the value of y for which F is a designated fraction of its max.

In addition, δ_{99}, which is the value of y for which $\bar{u}$ is 99 percent of its free-stream value, is found. A typical length, L_t, is defined as a linear combination of δ_{99} and the percent spread of one of the parabolic variables. The coefficients are inputs as is the choice of which percent spread to use.

The calculation of Λ, the turbulence scale, is given by

$$\Lambda_i = \min\left(cL_t, \Lambda_o + d_i y\right)$$

$$\Lambda = \begin{cases} \Lambda_i & y \le \delta_{99} \\ \Lambda_i + d_o\left(y - \delta_{99}\right) & y > \delta_{99} \end{cases}$$

The quantities c, d_i, d_o, and Λ_o are input parameters. For normal values of these constants, Λ equals Λ_o at the wall and increases with slope d_i until it reaches the value cL_t. It stays at that value the rest of the way out if d_o is zero. Otherwise, it starts increasing again at $y = \delta_{99}$ with slope d_o. The parameter Λ_o is made greater than zero to simulate wall roughness. The parameter d_o is made greater than zero to allow for free-stream turbulence without dissipation there, as mentioned in Section 3. Otherwise, both Λ_o and d_o are normally zero.

The rest of the auxiliary calculations are done to provide informative output only. They have no effect on the course of the program. The output labels and definitions are:

momentum thickness

$$\text{THETA} = \int_0^{y_J} \frac{\bar{\rho}\bar{u}}{\rho_e u_e}\left(1 - \frac{\bar{u}}{u_e}\right)dy$$

displacement thickness

$$\text{DELTA}^* = \int_0^{y_J} \left(1 - \frac{\bar{\rho}\bar{u}}{\rho_e u_e}\right) dy$$

shape factor

$$\text{SHAPE} = \text{DELTA}^*/\text{THETA}$$

kinematic displacement thickness

$$\text{DTI}^* = \int_0^{y_J} \left(1 - \frac{\bar{u}}{u_e}\right) dy$$

wall shear

$$\text{TAUW} = \left[\bar{\mu} \frac{\partial \bar{u}}{\partial y}\right]_{y=0}$$

local skin friction coefficient

$$\text{CF} = \text{TAUW}/\left(\frac{1}{2} \rho_e u_e^2\right)$$

wall heat flux

$$\text{QWALL} = -\left[\frac{\bar{k}}{c_p} \frac{\partial}{\partial y}\left(\bar{h} + \frac{\bar{\mu} c_p}{2\bar{k}} \bar{u}^2\right)\right]_{y=0}$$

total Stanton number

$$\text{STANT} = \text{QWALL}/\left[\rho_e u_e \left(h_w - h_e - \frac{1}{2} u_e^2\right)\right]$$

Reynolds numbers

$$\text{RETHETA} = \text{THETA}\ \frac{\rho_e u_e}{\mu_e}$$

$$\text{REXW} = \text{XW}\ \frac{\rho_e u_e}{\mu_e}$$

laminar stress

$$\text{TAULAM} = \bar{\mu}\,\frac{\partial \bar{u}}{\partial y}$$

total stress

$$\text{TAUTOT} = \bar{\mu}\,\frac{\partial \bar{u}}{\partial y} - \overline{\rho u' v'}$$

total heat flux

$$\text{QTOT} = -\frac{\bar{k}}{c_p}\frac{\partial \bar{h}}{\partial y} + \overline{\rho h' v'}$$

The integrals are done by the trapezoidal rule. The derivatives are evaluated by equation (2) of Section 3 except at the wall where a two-point formulation is used. In this regard it should be pointed out that the $\bar{u}^2$ term in the formula for QWALL is superfluous as a derivative (since $\bar{u} = 0$ at the wall, the derivative of its square is zero there), but in finite-difference terms the form given can be shown to be more accurate, at least if $\bar{w} = 0$ and $\bar{v} = 0$ at the wall.

None of the calculations in this group take into account the possible effects of nonzero $\bar{w}$. Modifications or additional calculations to show such effects could easily be introduced into the program.

5. CONTROL OF THE MESH

In order to provide reasonable accuracy without excessive computing; the mesh is modified by the program as the run progresses. Parameters supplied by the user allow him to control the balance between accuracy and computing time.

The value of Δx for the first step of a run is supplied as an input. Thereafter, a new Δx is computed before each x step. To do this, tentative values of the step size are computed for each of the 17 parabolic variables (that is, the variables determined by parabolic equations), and the smallest of these is selected. Each tentative value is determined by multiplying the previous Δx by the ratio of a criterion established for the particular variable to the max change of that variable for the previous step. The criterion is a nominal change that can be specified either as an absolute quantity or as some factor times the global max of the variable with an optional lower limit. The new Δx , the minimum of the tentative values, is then modified if necessary so that it doesn't exceed either a specified factor times the old Δx or a specified Δx_{max} . On the other hand, if the new Δx is less than a specified fraction of the old, "backup" occurs. That is, some change generated by the previous step is considered unacceptably large so the results of the solution for conditions at x_n just computed are discarded, and a smaller step from the conditions at x_{n-1} is tried, using the new smaller Δx . A minimum Δx may also be specified. If Δx becomes less than the minimum, the run is halted.

A further complication results from the existence of so-called time breaks. These are values of x at which the user requires output. In order to avoid an abnormally small Δx as the last step to such a point, the program looks several steps ahead and modifies the Δx found by the process described above, if need be.

The initial distribution of points in the y (normal) direction is part of the input of the initial conditions. Before the first and before each succeeding x step, a complex process is used to monitor and, if indicated, to adjust the spacing of points in the y direction. The basic idea is to examine each set of three adjacent points, y_{j-1} , y_j , y_{j+1} . If, for all the dependent variables, the values for those points fall almost on a straight line, the mid-point can be dropped. On the other hand, if, for at least one of the dependent variables, there is an excessive departure from a straight line, a point should be inserted in one of the subintervals y_{j-1} , y_j or y_j , y_{j+1} , or in each. (These notions are made quantitative by comparing a nondimensional measure of the departure from a straight line, for each dependent variable, with input parameters. The measure is the absolute value of the second derivative, as given by equation (4), divided by the global max and multiplied by the square of the length of the appropriate interval.) However, it is also required that the ratio of the lengths of adjacent intervals be less than a specified value. This requirement can prevent a point from being dropped, that otherwise would be, but does not prevent an insertion. Instead, insertions are made in other intervals until the ratio requirement is satisfied everywhere.

There are also means of specifying maximum and minimum interval lengths. The maximum prevents the dropping of a point which would cause it to be exceeded. If the insertion of a point creates an interval length smaller than the minimum, a warning message is printed.

Furthermore, if the insertion of points would make J = JTOP bigger than its upper limit, JMAX (an input) , the user has the option of allowing the relaxation of the criteria governing the dropping and insertion of points. If the relaxation is not allowed or is unsuccessful in keeping $J \leq JMAX$, the run halts.

When a point is inserted, it is not necessarily put at the mid-point of the interval. Rather it is weighted toward the point with the largest (in absolute value) normalized second derivative considering all the dependent variables. The restriction on ratio between the lengths of adjacent intervals is observed, of course.

The values of the dependent variables that are assumed for an inserted point are obtained by linear interpolation. This seems like an anticlimax, but it is the only simple scheme that has been found here that does not lead to undesirable results in some cases.

6. PROGRAM STRUCTURE

The principles of invariant modeling have been applied at A.R.A.P. to several different types of turbulent flows that can be described by parabolic equations. The technique developed for solving those problems has been used for the GYC program; and, to a large extent, the actual subprograms have been carried over, with varying amounts of modification.

What has been referred to as the GYC program is actually a main program, GYCON, and a number of subprograms. GYCON itself is quite simple. Basically, it consists of a call to GYCIN, which handles input and initialization, followed by a loop consisting of calls to: (1) GYCAL, which handles the auxiliary computations, the determination of the new Δx , and output; (2) GYCAA, which handles the distribution of the points y_j ; (3) GYCSS, which does the actual solution for one x step; and (4) GYCRV, which determines $\bar{v}$. The call to GYCRV is followed by a return to the call to GYCAA. This basic flow of the program is modified by certain special conditions; for example, if backup is required (see Section 5), the call to GYCAA is skipped. When the run is halted, GYCIN is called again. Depending on the input cards supplied, this can result in a restart of the same run, the start of a new run, or job termination.

Before describing the operation of the principal subprograms mentioned above, an aspect of data handling should be mentioned. Some of the other programs from which this one evolved needed more data storage during the solution process than was available in core (especially when the A.R.A.P. computer had only 8,192 words of 16 bits each in core). Such programs used external (disk) storage; and the GYC program, although contained entirely in core, mimics that arrangement by keeping a good part of the working data in a two-dimensional

array called STOR. The elements of STOR are never addressed directly; instead, parts of STOR are copied to and from other arrays in core, much as if STOR were an external device.

STOR consists of 52 elements for each j, that is, for each normal mesh point. The first element is the value of y_j itself. The next 17 elements are the old values of the parabolic variables, that is, the values evaluated at (x_{n-1}, y_j) in the order they are listed in Table 5. Then come the old values of $\bar{p}$, $\bar{\rho}$, $\bar{v}$, and $\bar{u}_x$. The following 17 elements are used for the new values of the parabolic variables and also for the values of $\bar{\phi}_j$, in the notation of equation (12), while generating the new values. The next four elements are the new values of $\bar{p}$, $\bar{\rho}$, $\bar{v}$, and $\bar{u}_x$. The rest of the 52 elements are used to hold the matrix Γ_j.

The copying or moving in and out of STOR is done by a subprogram, SFVMV. With parameters A, B, and N, SFVMV is equivalent to the simple DO loop:

```
      DO  8  I = 1,N
    8 B(I) = A(I)
```

It exists as a subroutine because it is used so often; at A.R.A.P. it is implemented in firmware to speed its operation. A companion subprogram, SFVFL, fills N elements of the vector B with the value of the variable A, when the parameters are A, B, and N.

Of the principal subprograms, GYCIN is the most straightforward. It initializes certain quantities, it reads the first data card, it calls GYCRI to read the rest of the data cards if the first card indicates that there are more, it calls GYCIC to do more initialization if a new run is being started, and finally it calls GYCPI to print the title pages of the output.

GYCAL first calls GYCQD and GYCAQ which do the auxiliary computations described in Section 4 and which also compute the new Δx as described in the first part of Section 5, except for the modification due to time breaks which is done by a call to GYCTB. The bulk of GYCAL is concerned with calls to GYCOI and GYCOT if the various criteria so indicate. GYCOI produces a minor printout, whereas GYCOT produces a major printout including tables of the dependent variables as functions of y.

GYCAA is the most complex of the subprograms. It is also the least changed from versions in use for other problems. Its function is to monitor and, if indicated, to adjust the spacing of points in the y direction, as described in Section 5. Suffice it to say that it works as advertised.

GYCSS, together with its subprograms, forms the heart of the system. It performs the upsweep and the downsweep, repeating the process for each of the passes described in Section 3. Since three-point differencing is used in the y direction, a section of COMMON is used for values of variables at $j-1$, that is, variable names ending with the letters M and I (compare the sketches at the beginning of Section 3 and the beginning of the Glossary, Appendix F); another section of COMMON is used for values of variables at j, with names ending with the letters Z and J; and a third section is used for values of variables at $j+1$, with names ending with the letters P and K.

As j is incremented at the beginning of a stage of the upsweep of a typical pass, the values in the Z-J section are shifted to the M-I section, and those of the P-K section to the Z-J section. Then GYCFP moves to the P-K section from STOR the values that are available there and calculates the values of others. Obviously, special provisions are made for $j = 1$. The values of the elements of A_j, B_j, C_j, and d_j are calculated by the subroutines GYCBI (for $j = 1$), GYCMC (for $2 \leq j \leq J - 1$), and GYCBO (for $j = J$). After making some

calculations common to all passes, GYCMC in turn calls separate subroutines for each pass. (Due to some confusion in the early stages of the development, the subprogram for the first pass is named GYCP2 and that for the second pass is named GYCP1.) It is in these pass subprograms that the actual equations being solved (Appendix D) make themselves felt. They were automatically generated, for the most part, by DIFFR, another A.R.A.P. software package.

DIFFR took the equations of Appendix D, one pass at a time, along with some information about which symbols to treat as unknowns, as known variables, or as constants, and generated Fortran statements evaluating the elements of A_j , B_j , C_j , and d_j . These statements, after some minor manual editing, constitute the bulk of the pass subprograms.

Returning to the description of a stage of the upsweep, Γ_j and $\bar{\phi}_j$ (equations (11) and (12) of Section 3) are evaluated using the special matrix subroutines GYCMP and GYCMI. Finally Γ_j and $\bar{\phi}_j$ are moved to STOR, and j is incremented again.

The handling of the free-stream boundary conditions has already been described. When they are satisfied, the downsweep according to equation (13) of Section 3 is trivial. It is slightly complicated in the first, second, and fourth passes by the incorporation of the calculation of $\bar{p}$, $\bar{u}_x$, and $\bar{\rho}$, respectively.

There are several other subprograms which have not been mentioned here. All are included in the Glossary (Appendix F) with brief descriptions. At the beginning of the listing of each subprogram are comments giving another view of its role.

Three ficticious subprograms are included in the Glossary, and places for calls to them are suggested by comments in the program. These provide for the possibility that it may be

desirable to create files in external storage for later plotting. Two types of files are anticipated. One, the "running file," would be of quantities that depend on x only, such as skin friction and momentum thickness. The other, the "profile file," would be of quantities that are functions of both x and y, such as any of the dependent variables. The three subprograms are GYCPS which would initiate both files, GYCRF which would write to the running file, and GYCPF which would write to the profile file.

7. INPUT AND OUTPUT

Table 1 lists the variables read on each input card along with the format for each. Definitions of the variables are included in the Glossary (Appendix F).

There must be at least two and no more than two hundred Cards 16.n (see Table 2). They produce an internal table specifying the velocity, enthalpy, and pressure at the edge of the boundary layer and the wall blowing velocity and either the enthalpy or enthalpy gradient (heat transfer) at the wall, all as functions of downstream distance x. The program uses linear interpolation in this table. For values of x outside the range of the table, it extrapolates using the first two or last two entries. The input values of x must be in ascending order. The end of the table is flagged by a nonincreasing value for x or by n reaching 200. In the former case, the values on the terminating card are not used. If all three quantities are constant with x, only two cards, including the terminating card, need be used; the program will automatically generate another point to store in its table for interpolation. As noted at the foot of Table 2, h_e and p_e need not be input. (An exception occurs for p_e when GMOMS = 0. In this case, the values supplied for p_e will be used whether positive, zero, or negative.) When the program needs them at a particular x, the interpolated values from the internal table are first determined. If these are zero (or negative), values computed from the homentropic relations for a perfect gas are substituted. (If the inputs HSTAG = h_s and PSTAG = p_s are not needed for these calculations, they are not used at all.)

There must be at least five and no more than JMAX Cards 31.j (see Table 3). The values of y must be in ascending order. The end of the set is flagged by a nonincreasing value of y or by j reaching JMAX. In the former case, the values on the terminating card are not used.

Provision is made for specifying the initial profile of Λ on Cards 31.j. This is intended chiefly for use when Λ is solved for by a differential equation in the future. If zero is supplied for Λ, the values will be determined in the usual way from the scale parameters (see Note 1 of the Glossary, Appendix F).

Cards 32 and 33.j are not read at all if the flag terminating the set of Cards 31.j is negative. The corresponding dependent variables (see Table 4) are initialized to zero for all values of y in that case. If the flag terminating Cards 31.j is zero or positive, then Cards 32 and 33.j are read. The end of the set is flagged by the first value of Y scaled by GV(1) which fails to equal the corresponding value of Y scaled by FV(1) in Cards 31.j, taken in order, or by j reaching the total number of cards in that set. In the former case, the corresponding dependent variables are initialized to zero for the remaining values of y.

As indicated in Tables 2, 3, and 4, the card sets designated 16.n, 31.j, and 33.j are multiplied, or scaled, by elements of the vectors EV, FV, and GV, respectively. It is recommended that the beginner set all values of these vectors equal to one and forget them until a need is felt for them. Two typical situations where they would be useful are: (1) The initial conditions are supplied as a function of y in, say, inches, and it is desired to use feet in the program. Then FV(1) (and GV(1) if needed) could be set at .0833333 and y values in inches punched in Cards 31.j (and 33.j if needed). (2) After one run starting with a certain level of turbulence, it is decided to make another under the same conditions except that the initial turbulence level is different, or zero. Then it is only necessary to change one card to prepare the input to the new run.

Input may be in any consistent system of units; outputs will be the same units. Nondimensional inputs may be used, producing nondimensional output, but they must be consistent. For example, if BLAM0 is made nondimensional with respect to a reference length, L_r , then all other input lengths (DELTI, DTMAX, etc.) must be supplied as ratios to L_r and all output lengths (X, Y, R, etc.) will be referred to L_r . The introduction to the Glossary contains additional information on dimensions and nondimensionalization.

A deck for one job might be set up as follows:

Card 1 with INFLG = 1

Cards 2-31.j, input for first run

Card 1 with INFLG = 2

Cards 2-29, restart input for first run

Card 1 with INFLG = 1

Cards 2-31.j, input for second run

Card 1 with INFLG = 0

This assumes a negative flag terminating the profile input is included in Cards 31.j. The above job, then, consists of two completely separate runs, the first being restarted after its first halt. The restart could be used to introduce a different cone angle, for example. If a run which halts on an error condition is set up for a restart, the restart probably won't be what's intended but the error condition is likely to recur soon. On the other hand, it's possible to design the changes introduced on restart precisely to overcome an anticipated error condition. See Note 7 of the Glossary (Appendix F) for some specific information on restarts.

Notes 5 and 6 of the Glossary contain information on inputs for special modes of operation.

Output consists of title pages, minor printouts, and major printouts. The title pages display the information input on Cards 1-29, for the most part in the same order. The labels are included in the Glossary, but one item near the top of the second title page is not labeled. Normally this is F but if the option of zeroing out correlations involving ρ' and h' has been selected, a T appears there.

A minor printout gives basic information on the progress of the run and includes most of the auxiliary quantities described in Section 4. Two unlabeled items appear in the top line. On the left, there appears the name of the parabolic variable for which the smallest tentative Δx for the next step was computed, as described in Section 5. The letter Y appears there if such a calculation has not just been made, e.g., at the start of a run. On the right, there appears the name of a variable which did not satisfy its edge tolerance until a point was added, as described in Section 3. It is not necessarily the only such variable. If no points have been added at the outer edge since the last minor printout, the letter Y appears there.

A major printout consists of a minor printout plus profiles of the dependent variables and of the auxiliary quantities which are functions of y.

A minor printout occurs alone only if NSTEP is a multiple of NIOLP. A major printout occurs: (1) if NSTEP is a multiple of NFOLP, (2) at time breaks (TBRKV), (3) if the maximum or the percent spread has changed by more than PCFMX or PCFSP times the value at the last major printout, (4) if NSTEP reaches NSTMX, or (5) if an error condition occurs. If NFOLP is one (a major printout at every step is specified), two extras are included: (1) profiles are printed after points have been inserted in or deleted from the y mesh by GYCAA and (2) a major printout shows the discarded results generated before each backup.

A run halts when the absolute value of a negative time break is reached, when NSTEP reaches NSTMX, or when an error condition occurs.

Error conditions produce messages indicating the nature of the problem. A.R.A.P. should be consulted if a cure doesn't suggest itself.

8. FINAL REMARKS

Along with the GYC program, an "SSF" listing of it was delivered to ARC. The SSF listing (named after the A.R.A.P. program that produced it) is an alphabetical index of variable names and subprogram names, locating every reference to each by subprogram and position within it, and printing the lines so located. Since GYC makes much use of COMMON, the SSF listing is a valuable aid in tracing the modification and uses of variables.

Anyone who wants to understand the details of the operation of GYC must turn to the program listings. Considerable effort has been expended (with varying degrees of success) to reduce clutter and to make the listings readable. With rare exceptions, the program flow is from top to bottom. Statement numbers appear in order according to an obvious scheme. DO loops are indented and occasionally indentation is used to set off other statements. Natural breaks in the program are signalled by blank lines. (Rows of asterisks or other symbols are not used. It has been realized for several hundred years that decorated pages are not easy to read, attractive though they may be.)

With the descriptions of Section 6 as a start and with the aid of the SSF listing, those familiar with Fortran should be able to determine what is going on throughout GYC.

9. REFERENCES

1. W. S. Lewellen and M. Teske, "Turbulence Modeling and Its Application to Atmospheric Diffusion," Part II, EPA-600/4-75-016b, December 1975.

2. Ashok K. Varma, "Second-Order Closure of Turbulent Reacting Shear Flows," A.R.A.P. Report No. 235, February 1975.

APPENDIX A.- THE EQUATIONS BEFORE MODELING

$$\left(\bar{\rho}\bar{u}^{\ell}\right)_{,\ell} + \left(\overline{\rho' u'^{\ell}}\right)_{,\ell} = 0$$

$$\bar{\rho}\bar{u}^{\ell}\bar{u}_{i,\ell} + \overline{\rho' u'^{\ell}}\,\bar{u}_{i,\ell} + \left(\bar{\rho}\overline{u'^{\ell} u'_i} + \bar{u}^{\ell}\overline{\rho' u'_i} + \overline{\rho' u'^{\ell} u'_i}\right)_{,\ell} =$$

$$= -\bar{p}_{,i} + \left(\bar{\mu}\bar{u}^{\ell}_{,i} + \bar{\mu}\bar{u}^{\ell}_{,i} + \overline{\mu' u'^{\ell}_{,i}} + \overline{\mu' u'_i{}^{,\ell}}\right)_{,\ell} +$$

$$+ \left(\bar{\mu}^{*}\bar{u}^{k}_{,k} + \overline{\mu^{*\prime} u'^{k}_{,k}}\right)_{,i}$$

$$\bar{\rho}\bar{u}^{\ell}\bar{h}_{,\ell} + \overline{\rho' u'^{\ell}}\,\bar{h}_{,\ell} + \left(\bar{\rho}\overline{h' u'^{\ell}} + \bar{u}^{\ell}\overline{\rho' h'} + \overline{\rho' h' u'^{\ell}}\right)_{,\ell} =$$

$$= \bar{u}^{\ell}\bar{p}_{,\ell} + \overline{u'^{\ell} p'_{,\ell}} + \left(\bar{\mu}\bar{u}^{\ell}_{,m} + \bar{\mu}\bar{u}^{\ell}_{,m} + \overline{\mu' u'^{\ell}_{,m}} + \overline{\mu' u'_m{}^{,\ell}}\right)\bar{u}^{m}_{,\ell} +$$

$$+ \left(\bar{\mu}^{*}\bar{u}^{k}_{,k} + \overline{\mu^{*\prime} u'^{k}_{,k}}\right)\bar{u}^{\ell}_{,\ell} + \bar{\mu}\,\overline{u'^{\ell}_{,m} u'^{,m}_{\ell}} +$$

$$+ \bar{\mu}\left[\left(\overline{u'^{\ell} u'^{m}}\right)_{,\ell m} - 2\left(\overline{u'^{m} u'^{\ell}_{,\ell}}\right)_{,m} + \overline{u'^{\ell}_{,\ell} u'^{m}_{,m}}\right] +$$

$$+ \left(\bar{u}^{,m}_{\ell} + \bar{u}^{m}_{,\ell}\right)\overline{\mu' u'^{\ell}_{,m}} + \bar{\mu}^{*}\,\overline{u'^{m}_{,m} u'^{\ell}_{,\ell}} + \bar{u}^{m}_{,m}\,\overline{\mu^{*\prime} u'^{\ell}_{,\ell}} +$$

$$+ \left(\bar{k}\,\bar{T}_{,\ell}\right)^{,\ell} + \left(\overline{k' T'_{,\ell}}\right)^{,\ell}$$

$$\bar{\rho}\bar{u}^{\ell}\left(\overline{u_i'u_j'}\right)_{,\ell} + \left(\bar{\rho}\overline{u'^{\ell}u_i'u_j'} + \bar{u}^{\ell}\overline{\rho' u_i'u_j'}\right)_{,\ell} - \left(\overline{\rho'u'^{\ell}}\right)_{,\ell}\overline{u_i'u_j'} +$$

$$+ \overline{\rho'u_i'}\,\bar{u}^{\ell}\bar{u}_{j,\ell} + \overline{\rho'u_j'}\,\bar{u}^{\ell}\bar{u}_{i,\ell} + \bar{\rho}\overline{u'^{\ell}u_i'}\,\bar{u}_{j,\ell} +$$

$$+ \bar{\rho}\overline{u'^{\ell}u_j'}\,\bar{u}_{i,\ell} + \overline{\rho'u'^{\ell}u_i'}\,\bar{u}_{j,\ell} + \overline{\rho'u'^{\ell}u_j'}\,\bar{u}_{i,\ell} =$$

$$= -\overline{u_i'p'_{,j}} - \overline{u_j'p'_{,i}} +$$

$$+\bar{\mu}\left(\left(\overline{u_i'u_j'}\right)_{,\ell}^{,\ell} - 2\overline{u_{i,\ell}'u_j'^{,\ell}} + \overline{u_i'u'^{\ell}_{,j\ell}} + \overline{u_j'u'^{\ell}_{,i\ell}}\right) +$$

$$+\bar{\mu}_{,\ell}\left(\left(\overline{u_i'u_j'}\right)^{,\ell} + \overline{u_i'u'^{\ell}_{,j}} + \overline{u_j'u'^{\ell}_{,i}}\right) + \left(\bar{u}_{i,\ell} + \bar{u}_{\ell,i}\right)\overline{u_j'\mu'^{,\ell}} +$$

$$+ \left(\bar{u}_{j,\ell} + \bar{u}_{\ell,j}\right)\overline{u_i'\mu'^{,\ell}} + \left(\bar{u}_{i,\ell}^{,\ell} + \bar{u}_{\ell,i}^{,\ell}\right)\overline{\mu'u_j'} +$$

$$+ \left(\bar{u}_{j,\ell}^{,\ell} + \bar{u}_{\ell,j}^{,\ell}\right)\overline{\mu'u_i'} + \bar{u}^{m}_{,m}\left(\overline{u_i'\mu^{*'}_{,j}} + \overline{u_j'\mu^{*'}_{,i}}\right) +$$

$$+ \bar{u}^{m}_{,mi}\overline{\mu^{*'}u_j'} + \bar{u}^{m}_{,mj}\overline{\mu^{*'}u_i'} + \bar{\mu}^{*}\left(\overline{u_i'u'^{m}_{,mj}} + \overline{u_j'u'^{m}_{,mi}}\right) +$$

$$+ \bar{\mu}^{*}_{,i}\overline{u_j'u'^{m}_{,m}} + \bar{\mu}^{*}_{,j}\overline{u_i'u'^{m}_{,m}}$$

$$\bar{\rho}\bar{u}^{\ell}\left(\overline{h'u_i'}\right)_{,\ell} + \left(\bar{\rho}\overline{h'u'^{\ell}u_i'} + \bar{u}^{\ell}\overline{\rho'h'u_i'}\right)_{,\ell} - \left(\overline{\rho'u'^{\ell}}\right)_{,\ell}\overline{h'u_i'} +$$

$$+ \overline{\rho'u_i'}\bar{u}^{\ell}\bar{h}_{,\ell} + \overline{\rho'h'}\bar{u}^{\ell}\bar{u}_{i,\ell} + \bar{\rho}\left(\overline{u'^{\ell}u_i'}\bar{h}_{,\ell} + \overline{h'u'^{\ell}}\bar{u}_{i,\ell}\right) +$$

$$+ \overline{\rho'u'^{\ell}u_i'}\bar{h}_{,\ell} + \overline{\rho'h'u'^{\ell}}\bar{u}_{i,\ell} =$$

$$= \overline{u_i'p_t'} + \bar{u}^{\ell}\overline{u_i'p'}_{,\ell} + \bar{p}_{,\ell}\overline{u'^{\ell}u_i'} + \overline{u'^{\ell}u_i'p'}_{,\ell} - \overline{h'p'}_{,i} +$$

$$+ \left(\bar{\mu}\bar{u}_{m}^{,\ell} + \bar{\mu}\bar{u}_{,m}^{\ell} + \overline{\mu'u_m'^{,\ell}} + \overline{\mu'u'^{\ell}_{,m}}\right)\overline{u_i'u'^{m}_{,\ell}} +$$

$$+ \left(\bar{\mu}^{*}\bar{u}^{k}_{,k} + \overline{\mu^{*'}u'^{k}_{,k}}\right)\overline{u_i'u'^{\ell}_{,\ell}} + \bar{\mu}\bar{u}^{\ell}_{,m}\left(\overline{u_i'u_{\ell}'^{,m}} + \overline{u_i'u'^{m}_{,\ell}}\right) +$$

$$+ \bar{u}^{\ell}_{,m}\left(\bar{u}_{\ell}^{,m} + \bar{u}^{m}_{,\ell}\right)\overline{\mu'u_i'} + \bar{u}^{\ell}_{,\ell}\bar{u}^{m}_{,m}\,\overline{\mu^{*'}u_i'} + \bar{\mu}^{*}\bar{u}^{\ell}_{,\ell}\,\overline{u_i'u'^{m}_{,m}} +$$

$$+ \bar{k}\left(\left(\overline{T'u_i'}\right)^{,\ell}_{,\ell} - \overline{T'^{,\ell}u_{i,\ell}'} - \left(\overline{T'u_i'^{,\ell}}\right)_{,\ell}\right) + \bar{k}^{,\ell}\left(\overline{T'u_i'}\right)_{,\ell} +$$

$$- \bar{k}_{,\ell}\overline{T'u_i'^{,\ell}} + \bar{T}^{,\ell}_{,\ell}\overline{k'u_i'} + \bar{T}_{,\ell}\overline{u_i'k'^{,\ell}} +$$

$$+ \bar{\mu}\left(\overline{h'u_{i,\ell}'^{,\ell}} + \overline{h'u'^{\ell}_{,i\ell}}\right) + \bar{\mu}_{,\ell}\left(\overline{h'u_i'^{,\ell}} + \overline{h'u'^{\ell}_{,i}}\right) +$$

$$+ \left(\bar{u}_{i,\ell} + \bar{u}_{\ell,i}\right)\overline{h'\mu'^{,\ell}} + \left(\bar{u}_{i,\ell} + \bar{u}_{\ell,i}\right)^{,\ell}\overline{\mu'h'} +$$

$$+ \bar{u}^{m}_{,m}\overline{h'\mu^{*'}_{,i}} + \bar{u}^{m}_{,mi}\overline{\mu^{*'}h'} + \bar{\mu}^{*}\overline{h'u'^{m}_{,mi}} + \bar{\mu}^{*}_{,i}\overline{h'u'^{m}_{,m}}$$

$$\bar{\rho}\bar{u}^{\ell}\left(\overline{h'h'}\right)_{,\ell} + \left(\bar{\rho}\overline{u'^{\ell}h'h'} + \bar{u}^{\ell}\overline{\rho'h'h'}\right)_{,\ell} - \left(\overline{\rho'u'^{\ell}}\right)_{,\ell}\overline{h'h'} +$$

$$+ 2\overline{\rho'h'}\bar{u}^{\ell}\bar{h}_{,\ell} + 2\bar{\rho}\bar{h}_{,\ell}\overline{u'^{\ell}h'} + 2\bar{h}_{,\ell}\overline{\rho'u'^{\ell}h'} =$$

$$= 2\overline{h'p'_t} + 2\bar{u}^{\ell}\overline{h'p'_{,\ell}} + 2\bar{p}_{,\ell}\overline{h'u'^{\ell}} + 2\overline{h'u'^{\ell}p'_{,\ell}} +$$

$$+ 2\left(\bar{\mu}\bar{u}^{\ell}_{,m} + \bar{\mu}\bar{u}^{\ell}_{,m} + \overline{\mu'u'^{\ell}_{,m}} + \overline{\mu'u'^{\ell}_{,m}}\right)\overline{h'u'^{m}_{,\ell}} +$$

$$+ 2\left(\bar{\mu}^{*}\bar{u}^{k}_{,k} + \overline{\mu^{*'}u'^{k}_{,k}}\right)\overline{h'u'^{\ell}_{,\ell}} + 2\bar{\mu}\bar{u}^{\ell}_{,m}\left(\overline{h'u'^{m}_{\ell}} + \overline{h'u'^{m}_{,\ell}}\right) +$$

$$+ 2\bar{u}^{\ell}_{,m}\left(\bar{u}^{m}_{\ell} + \bar{u}^{m}_{,\ell}\right)\overline{\mu'h'} + 2\bar{u}^{\ell}_{,\ell}\bar{u}^{m}_{,m}\overline{\mu^{*'}h'} + 2\bar{\mu}^{*}\bar{u}^{\ell}_{,\ell}\overline{h'u'^{m}_{,m}} +$$

$$+ 2\bar{k}\overline{h'T'^{,\ell}_{,\ell}} + 2\bar{k}_{,\ell}\overline{h'T'^{,\ell}} + 2\bar{T}^{,\ell}_{,\ell}\overline{k'h'} + 2\bar{T}_{,\ell}\overline{h'k'^{,\ell}}$$

$$\bar{\rho}\bar{u}^{\ell}\left(\overline{\rho'u'_i}\right)_{,\ell} + \left(\bar{\rho}\overline{\rho'u'^{\ell}u'_i} + \bar{u}^{\ell}\overline{\rho'\rho'u'_i}\right)_{,\ell} - \left(\overline{\rho'u'^{\ell}}\right)_{,\ell}\overline{\rho'u'_i} +$$

$$+ \overline{\rho'\rho'}\,\bar{u}^{\ell}\bar{u}_{i,\ell} + \overline{\rho'u'_i}\,\bar{u}^{\ell}\bar{\rho}_{,\ell} + \overline{\rho'u'^{\ell}}\,\bar{\rho}\bar{u}_{i,\ell} + \overline{u'^{\ell}u'_i}\,\bar{\rho}\bar{\rho}_{,\ell} +$$

$$+ \overline{\rho'\rho'u'^{\ell}}\bar{u}_{i,\ell} + \overline{\rho'u'^{\ell}u'_i}\,\bar{\rho}_{,\ell} =$$

$$= -\overline{\rho'p'_{,i}} - \bar{\rho}\bar{\rho}\overline{u'_i u'^{\ell}_{,\ell}} - 2\bar{\rho}\left(\bar{u}^{\ell}_{,\ell}\overline{\rho'u'_i} + \overline{\rho'u'_i u'^{\ell}_{,\ell}}\right) - \bar{u}^{\ell}_{,\ell}\overline{\rho'\rho'u'_i} +$$

$$+ \bar{\mu}\left(\overline{\rho'u'^{,\ell}_{i,\ell}} + \overline{\rho'u'^{\ell}_{,i\ell}}\right) + \bar{\mu}_{,\ell}\left(\overline{\rho'u'^{,\ell}_{i}} + \overline{\rho'u'^{\ell}_{,i}}\right) +$$

$$+ \left(\bar{u}_{i,\ell} + \bar{u}_{\ell,i}\right)\overline{\rho'\mu'^{,\ell}} + \left(\bar{u}_{i,\ell} + \bar{u}_{\ell,i}\right)^{,\ell}\overline{\mu'\rho'} +$$

$$+ \bar{u}^{m}_{,m}\,\overline{\rho'\mu^{*}_{,i}} + \bar{u}^{m}_{,mi}\overline{\mu^{*'}\rho'} + \bar{\mu}^{*}\overline{\rho'u'^{m}_{,mi}} + \bar{\mu}^{*}_{,i}\overline{\rho'u'^{m}_{,m}}$$

APPENDIX B.- THE MODELS

$$\overline{u_i' u_j' u_k'} = -VUUq\Lambda\left(\left(\overline{u_i' u_j'}\right)_{,k} + \left(\overline{u_j' u_k'}\right)_{,i} + \left(\overline{u_k' u_i'}\right)_{,j}\right)$$

$$\overline{u_i' u_j' \rho'} = -VRUq\Lambda\left(\left(\overline{\rho' u_i'}\right)_{,j} + \left(\overline{\rho' u_j'}\right)_{,i}\right)$$

$$\overline{u_i' u_j' h'} = -VUHq\Lambda\left(\left(\overline{h' u_i'}\right)_{,j} + \left(\overline{h' u_j'}\right)_{,i}\right)$$

$$\overline{u_i' \rho' \rho'} = -VRRq\Lambda\left(\overline{\rho' \rho'}\right)_{,i}$$

$$\overline{u_i' \rho' h'} = -VRHq\Lambda\left(\overline{\rho' h'}\right)_{,i}$$

$$\overline{u_i' h' h'} = -VHHq\Lambda\left(\overline{h' h'}\right)_{,i}$$

$$\overline{u_i'^{,m} u_{j,m}'} = \frac{a}{\Lambda^2}\overline{u_i' u_j'} + b\,\frac{\bar{\rho} q}{\Lambda \bar{\mu}}\left(\beta \overline{u_i' u_j'} + (1-\beta) g_{ij}\,\frac{\overline{u'^{\ell} u_{\ell}'}}{3}\right),$$

$$\overline{u_i'^{,m} h'_{,m}} = \left(\frac{AHU}{\Lambda^2} + BHU\,\frac{\bar{\rho} q}{\Lambda \bar{\mu}}\right)\overline{h' u_i'}$$

$$\overline{u_i'^{,m} \rho'_{,m}} = \left(\frac{ARU}{\Lambda^2} + BRU\,\frac{\bar{\rho} q}{\Lambda \bar{\mu}}\right)\overline{\rho' u_i'}$$

$$\overline{h'^{,m} h'_{,m}} = \left(\frac{AHH}{\Lambda^2} + BHH\,\frac{\bar{\rho} q}{\Lambda \bar{\mu}}\right)\overline{h' h'}$$

$$\overline{u_i' u'^{\ell}_{,\ell}} = \pi_{\ell}\overline{u_i' u'^{\ell}} \qquad \text{where} \quad \pi_{\ell} = -BDV\,\frac{\bar{p}_{,\ell}}{\gamma \bar{p}}$$

$$\overline{h' u'^{\ell}_{,\ell}} = \pi_{\ell}\overline{h' u'^{\ell}}$$

$$\overline{\rho' u'^{\ell}_{,\ell}} = \pi_{\ell} \overline{\rho' u'^{\ell}}$$

$$\overline{\rho' u_i' u'^{\ell}_{,\ell}} = \text{VBS} \pi_{\ell} \overline{\rho' u_i' u'^{\ell}}$$

$$\overline{u'^{k}_{,k} u'^{\ell}_{,\ell}} = \pi_k \pi_{\ell} \overline{u'^{k} u'^{\ell}}$$

$$\overline{h' u'^{\ell}_{,\ell i}} = \frac{1}{2} \pi_{\ell} \left(\overline{h' u'^{\ell}} \right)_{,i}$$

$$\overline{\rho' u'^{\ell}_{,\ell i}} = \frac{1}{2} \pi_{\ell} \left(\overline{\rho' u'^{\ell}} \right)_{,i}$$

$$\overline{\rho' h'_{,i}} = \frac{1}{2} \left(\overline{\rho' h'} \right)_{,i}$$

$$\overline{u_i' u_j'_{,k}} = \frac{1}{2} \left(\overline{u_i' u_j'} \right)_{,k} - \frac{1}{4} \left[g_{jk} \left(\overline{u_i' u'^{\ell}} \right)_{,\ell} - g_{ik} \left(\overline{u_j' u'^{\ell}} \right)_{,\ell} \right] +$$

$$+ \frac{1}{2} \left(g_{jk} \overline{u_i' u'^{\ell}_{,\ell}} - g_{ik} \overline{u_j' u'^{\ell}_{,\ell}} \right)$$

$$\overline{h' u_i'_{,j}} = \text{WMH} \left[\left(\overline{h' u_i'} \right)_{,j} - \frac{1}{3} g_{ij} \left(\overline{h' u'^{\ell}} \right)_{,\ell} \right] + \frac{1}{3} g_{ij} \overline{h' u'^{\ell}_{,\ell}}$$

$$\overline{\rho' u_i'_{,j}} = \text{WMR} \left[\left(\overline{\rho' u_i'} \right)_{,j} - \frac{1}{3} g_{ij} \left(\overline{\rho' u'^{\ell}} \right)_{,\ell} \right] + \frac{1}{3} g_{ij} \overline{\rho' u'^{\ell}_{,\ell}}$$

$$\overline{u_i'_{,j} u'^{\ell}_{,\ell}} = \text{WMD} \left[\left(\overline{u_i' u'^{\ell}_{,\ell}} \right)_{,j} - \frac{1}{3} g_{ij} \left(\overline{u'^{k} u'^{\ell}_{,\ell}} \right)_{,k} \right] + \frac{1}{3} g_{ij} \overline{u'^{k}_{,k} u'^{\ell}_{,\ell}}$$

$$\overline{u_i'_{,j} u'_{n,m}} = \left(\text{WGU}/\Lambda^2 \right) \left[g_{mj} \overline{u_i' u_n'} - \frac{1}{3} g_{nm} \overline{u_i' u_j'} + \right.$$

$$\left. - \frac{1}{3} g_{ij} \overline{u_n' u_m'} + \frac{1}{9} g_{ij} g_{nm} q^2 \right]$$

$$\overline{u'_{i,j}u'^{m}_{,n}u'^{n}_{,m}} = \left(WWGUq/\Lambda^3\right)\left(\overline{u'_iu'_j} - \frac{1}{3}\,g_{ij}q^2\right)$$

$$\overline{u'_iu'^{m}_{,n}u'^{n}_{,m}} = (q/\Lambda)\left[WWU1\left(\overline{u'_iu'^{\ell}}\right)_{,\ell} + WWU2\ \overline{u'_iu'^{\ell}_{,\ell}}\right]$$

$$\overline{u'^{m}_{,n}u'^{n}_{,m}u'^{\ell}_{,\ell}} = (q/\Lambda)\left[WWD1\left(\overline{u'^{k}u'^{\ell}_{,\ell}}\right)_{,k} + WWD2\ \overline{u'^{k}_{,k}u'^{\ell}_{,\ell}}\right]$$

$$\overline{h'u'^{m}_{,n}u'^{n}_{,m}} = (q/\Lambda)\left[WWH1\left(\overline{h'u'^{\ell}}\right)_{,\ell} + WWH2\ \overline{h'u'^{\ell}_{,\ell}}\right]$$

$$\overline{\rho'u'^{m}_{,n}u'^{n}_{,m}} = (q/\Lambda)\left[WWR1\left(\overline{\rho'u'^{\ell}}\right)_{,\ell} + WWR2\ \overline{\rho'u'^{\ell}_{,\ell}}\right]$$

$$\overline{p'u'_i} = \bar{\rho}\Lambda^2\left[PMU\bar{u}^{m}_{,n}\ \overline{u'_iu'^{n}_{,m}} + PMU2\ \overline{u'_iu'^{m}_{,n}u'^{n}_{,m}}\right]$$

$$\overline{p'h'} = \bar{\rho}\Lambda^2\left[PMH\bar{u}^{m}_{,n}\ \overline{h'u'^{n}_{,m}} + PMH2\ \overline{h'u'^{m}_{,n}u'^{n}_{,m}}\right]$$

$$\overline{p'\rho'} = \bar{\rho}\Lambda^2\left[PMR\bar{u}^{m}_{,n}\ \overline{\rho'u'^{n}_{,m}} + PMR2\ \overline{\rho'u'^{m}_{,n}u'^{n}_{,m}}\right]$$

$$\overline{p'u'_{i,j}} = \bar{\rho}\Lambda^2\left[PGU\bar{u}^{m}_{,n}\ \overline{u'_{i,j}u'^{n}_{,m}} + PGU2\ \overline{u'_{i,j}u'^{m}_{,n}u'^{n}_{,m}}\right]$$

$$\overline{p'h'_{,i}} = PGH\bar{\rho}\left[WGH1\bar{u}^{m}_{,m}\ \overline{h'u'_i} + WGH2\bar{u}^{m}_{,i}\ \overline{h'u'_m} + WGH3\bar{u}_{i,m}\ \overline{h'u'^{m}}\right] +$$

$$+ PGH2*WWGH\ \frac{\bar{\rho}q}{\Lambda}\ \overline{h'u'_i}$$

$$\overline{p'\rho'_{,i}} = PGR\bar{\rho}\left[WGR1\bar{u}^{m}_{,m}\ \overline{\rho'u'_i} + WGR2\bar{u}^{m}_{,i}\ \overline{\rho'u'_m} + WGR3\bar{u}_{i,m}\ \overline{\rho'u'^{m}}\right] +$$

$$+ PGR2*WWGR\ \frac{\bar{\rho}q}{\Lambda}\ \overline{\rho'u'_i}$$

$$\overline{u_i'p'}_t + \bar{u}^{\ell}\overline{u_i'p'}_{,\ell} = \text{PTUM}\left((\overline{u_i'p'})_t + \bar{u}^{\ell}(\overline{u_i'p'})_{,\ell}\right)$$

$$\overline{h'p'}_t + \bar{u}^{\ell}\overline{h'p'}_{,\ell} = \text{PTHM}\left((\overline{h'p'})_t + \bar{u}^{\ell}(\overline{h'p'})_{,\ell}\right)$$

$$\overline{\rho'h'} = -\overline{h'h'}\ \bar{\rho}/\bar{h}$$

$$\overline{\rho'\rho'} = -\overline{\rho'h'}\ \bar{\rho}/\bar{h}$$

$$\overline{\rho'h'h'} = 0$$

$$\overline{u'^{\ell}u_i'p'}_{,\ell} = 0$$

$$\overline{h'u'^{\ell}p'}_{,\ell} = 0$$

The following constraints apply:

$$3*\text{WGH1} + \text{WGH2} + \text{WGH3} = 0$$

$$3*\text{WGR1} + \text{WGR2} + \text{WGR3} = 0 \quad .$$

APPENDIX C.- THE EQUATIONS AFTER MODELING

```
#                                          MEAN CONTINUITY

  (RS*US"L),L + RU"L,L = 0


#                                          MEAN MOMENTUM

  RS*US"L*US'I,L + RU"L*US'I,L
    + (RS*UU"L'I + US"L*RU'I),L
    - VRU*(QL*(RU'I!L + RU"L,I)),L
    = - PS,I + (EMU*US'I!L + EMU*US"L,I),L
      + EMUT*WMT*(TU'I!L + TU"L,I),L
      - 2/3*EMUT*(WMT*TU"M,M - BDGP*TU"M*PS,M),I
      + (EMS*US"M,M + EMST*BDGP*TU"M*PS,M),I


#                                          MEAN ENTHALPY

RS*US"L*HS,L + RU"L*HS,L + (RS*HU"L + US"L*RH),L
  - VRH*(QL*RH,L)!L
   = GMOMS*(    US"L*PS,L
     + (RS*BLAM*BLAM*(  PMU*US"M,N*( 1/2*UU"LN,M
        - 1/4*(&"N'M*UU"LK,K - &"L'M*UU"NK,K)
        + 1/2*(&"N'M*BDGP*UU"LK*PS,K - &"L'M*BDGP*UU"NK*PS,K) )
       + PMU2*Q/BLAM*(WWU1*UU"KL,K + WWU2*BDGP*UU"KL*PS,K)  )),L
     - RS*BLAM*BLAM*(  PGU*US"M,N*(WMD*( (BDGP*UU"NL*PS,L),M
        - 1/3*&"N'M*(BDGP*UU"KL*PS,L),K )
       + 1/3*&"N'M*BDGP*BDGP*UU"KL*PS,K*PS,L)
      + PGU2*Q/BLAM*(WWD1*(BDGP*UU"KL*PS,K),L
       + WWD2*BDGP*BDGP*UU"KL*PS,K*PS,L)  )
     + (EMU*(US'M!L + US"L,M)
      + EMUT*( WMT*(TU'M!L + TU"L,M - 2/3*&"L'M*TU"K,K)
       + 2/3*&"L'M*BDGP*TU"K*PS,K ))*US"M,L
     + (EMS*US"K,K + EMST*BDGP*TU"K*PS,K)*US"L,L
     + EMU*A/BLAM/BLAM*UU"L'L + B/BLAM*RS*Q*UU"L'L
     + EMU*(UU"LM,LM - 2*(BDGP*UU"ML*PS,L),M
      + BDGP*BDGP*UU"ML*PS,M*PS,L)
     + EMUT*(US'L!M + US"M,L)*(WMT*(TU"L,M - 1/3*&"L'M*TU"N,N)
      + 1/3*&"L'M*BDGP*TU"N*PS,N)
     + EMS*BDGP*BDGP*UU"ML*PS,M*PS,L
     + EMST*US"M,M*BDGP*TU"L*PS,L    )
    + (CAP*HS,L)!L  + 1/2*CAYT*TT,L!L
```

```
RS*US"L*UU'IJ,L - (VUU*RQL*(UU'IJ!L + UU"L'J,I + UU"L'I,J)
   + VRU*QL*US"L*(RU'I,J + RU'J,I)),L
  - RU"L,L*UU'IJ + RU'I*US"L*US'J,L + RU'J*US"L*US'I,L
  + RS*(UU"L'I*US'J,L + UU"L'J*US'I,L)
  - VRU*QL*(US'J,L*(RU'I!L + RU"L,I) + US'I,L*(RU'J!L + RU"L,J))
   = -(   RS*BLAM*BLAM*(PMU*US"M,N*(  1/2*UU"N'I,M
       - 1/4*(&"N'M*UU"L'I,L - @'IM*UU"LN,L)
       + 1/2*(&"N'M*BDGP*UU"L'I*PS,L - @'IM*BDGP*UU"LN*PS,L)  )
     + PMU2*Q/BLAM*(WWU1*UU"L'I,L + WWU2*BDGP*UU"L'I*PS,L))  ),J
   - (   RS*BLAM*BLAM*(PMU*US"M,N*(  1/2*UU"N'J,M
       - 1/4*(&"N'M*UU"L'J,L - @'JM*UU"LN,L)
       + 1/2*(&"N'M*BDGP*UU"L'J*PS,L - @'JM*BDGP*UU"LN*PS,L)  )
     + PMU2*Q/BLAM*(WWU1*UU"L'J,L + WWU2*BDGP*UU"L'J*PS,L))  ),I
   + RS*( PGU*WGU*US"M,N*(@'MJ*UU"N'I + @'MI*UU"N'J
     - 2/3*&"N'M*UU'IJ - 2/3*@'IJ*UU"N'M + 2/9*@'IJ*&"N'M*UU"L'L)
     - Q/BLAM*(UU'IJ - 1/3*@'IJ*UU"L'L) )
   + (EMU*UU'IJ,L)!L
   - 2*EMU*A/BLAM/BLAM*UU'IJ
   - 2*B/BLAM*RS*Q*(BETA*UU'IJ + OMBET*@'IJ/3*UU"L'L)
   + (EMU + EMS)*( (BDGP*UU"L'I*PS,L),J + (BDGP*UU"L'J*PS,L),I
     - WMD*((BDGP*UU"L'I*PS,L),J + (BDGP*UU"L'J*PS,L),I
       - 2/3*@'IJ*(BDGP*UU"LK*PS,L),K)
     + 2/3*@'IJ*BDGP*BDGP*UU"LK*PS,L*PS,K )
   + EMU,L*( 1/2*UU"L'I,J - 1/4*(&"L'J*UU"M'I,M - @'IJ*UU"LM,M)
     + 1/2*(&"L'J*BDGP*UU"M'I*PS,M - @'IJ*BDGP*UU"LM*PS,M)
           + 1/2*UU"L'J,I - 1/4*(&"L'I*UU"M'J,M - @'IJ*UU"LM,M)
     + 1/2*(&"L'I*BDGP*UU"M'J*PS,M - @'IJ*BDGP*UU"LM*PS,M) )
   + EMUT*( (US'I,L + US'L,I)*(TU'J!L
       - WMT*(TU'J!L - 1/3*&"L'J*TU"M,M)
       - 1/3*&"L'J*BDGP*TU"M*PS,M)
           + (US'J,L + US'L,J)*(TU'I!L
       - WMT*(TU'I!L - 1/3*&"L'I*TU"M,M)
       - 1/3*&"L'I*BDGP*TU"M*PS,M)
     + TU'J*(US'I,L!L + US'L,I!L) + TU'I*(US'J,L!L + US'L,J!L) )
   + EMST*( US"M,M*(TU'I,J + TU'J,I - WMT*(TU'I,J + TU'J,I
       - 2/3*@'IJ*TU"L,L) + 2/3*@'IJ*BDGP*TU"L*PS,L)
     + TU'J*US"M,MI + TU'I*US"M,MJ )
   + EMS,I*BDGP*UU"M'J*PS,M + EMS,J*BDGP*UU"M'I*PS,M
```

```
RS*US"L*HU'I,L - (VUH*RQL*(HU'I!L + HU"L,I) + VRH*QL*US"L*RH,I),L
  - RU"L,L*HU'I + RU'I*US"L*HS,L + RH*US"L*US'I,L
  + RS*(UU"L'I*HS,L + US'I,L*HU"L)
  - VRU*QL*(RU'I!L + RU"L,I)*HS,L - VRH*QL*RH!L*US'I,L
   = GMOMS*(   PTUM*US"L* (  RS*BLAM*BLAM*(PMU*US"M,N*1/2*UU"N'I,M
         - 1/4*PMU*(US"M,M*UU"K'I,K - US'I,N*UU"NK,K)
         + 1/2*PMU*(US"M,M*BDGP*UU"K'I*PS,K
          - US'I,N*BDGP*UU"KN*PS,K))
        + PMU2*RQL*(WWU1*UU"K'I,K + WWU2*BDGP*UU"K'I*PS,K) ),L
     + UU"L'I*PS,L    )
    - (  RS*BLAM*BLAM*(PMH*US"M,N*( WMH*(HU"N,M
         - 1/3*&"N'M*HU"L,L) + 1/3*&"N'M*BDGP*HU"L*PS,L )
       + PMH2*Q/BLAM*(WWH1*HU"L,L + WWH2*BDGP*HU"L*PS,L))  ),I
    + PGH*RS*(WGH1*US"M,M*HU'I + WGH2*US"M,I*HU'M
     + WGH3*US'I,M*HU"M) + PGH2*WWGH*RS*Q/BLAM*HU'I
    + GMOMS*(  (EMU*(US'M!L + US"L,M) + EMUT/CP*( WMT*(HU'M!L
         + HU"L,M - 2/3*&"L'M*HU"K,K)
        + 2/3*&"L'M*BDGP*HU"K*PS,K ))*(1/2*UU"M'I,L
      - 1/4*(&"M'L*UU"N'I,N - @'IL*UU"MN,N)
      + 1/2*(&"M'L*BDGP*UU"N'I*PS,N - @'IL*BDGP*UU"MN*PS,N))
     + (EMS*US"K,K + EMST/CP*BDGP*HU"K*PS,K)*BDGP*UU"L'I*PS,L
     + EMU*US"L,M*(1/2*(UU'IL!M + UU"M'I,L)
      - 1/4*(2*&"M'L*UU"K'I,K - &"M'I*UU"K'L,K - @'IL*UU"KM,K)
      + 1/2*(2*&"M'L*BDGP*UU"K'I*PS,K - &"M'I*BDGP*UU"K'L*PS,K
       - @'IL*BDGP*UU"KM*PS,K))
     + EMUT/CP*US"L,M*(US'L!M + US"M,L)*HU'I
     + EMST/CP*US"L,L*US"M,M*HU'I + EMS*US"L,L*BDGP*UU"M'I*PS,M )
    + (CAP*HU'I,L)!L - CAP*HUDMF*HU'I
    - (CAP*(WMT*(HU'I!L - 1/3*&"L'I*HU"K,K)
      + 1/3*&"L'I*BDGP*HU"K*PS,K)),L
    + CAYT/CP/CP*((HU'I*HS,L)!L - HS,L*( WMT*(HU'I!L
       - 1/3*&"L'I*HU"K,K) + 1/3*&"L'I*BDGP*HU"K*PS,K ))
    + EMU*(WMH*(HU'I,L - 1/3*@'IL*HU"K,K)
     + 1/3*@'IL*BDGP*HU"K*PS,K)!L
    - EMU*HUDMF*HU'I + 1/2*(EMU + EMS)*(BDGP*HU"K*PS,K),I
    + EMU,L*(WMH*(HU'I!L + HU"L,I - 2/3*&"L'I*HU"K,K)
     + 2/3*&"L'I*BDGP*HU"K*PS,K)
    + EMUT/CP*((US'I,L + US'L,I)*1/2*HH!L
     + (US'I,L + US'L,I)!L*HH)
    + EMST/CP*(US"M,M*1/2*HH,I + US"M,MI*HH)
    + EMS,I*BDGP*HU"M*PS,M
```

```
#                                                    HH

RS*US"L*HH,L - VHH*(RQL*HH,L)!L - RU"L,L*HH
  + 2*RH*US"L*HS,L + 2*RS*HS,L*HU"L - 2*VRH*QL*HS,L*RH!L
   = 2*GMOMS*(   PTHM*US"L* (  RS*BLAM*BLAM*(PMH*US"M,N*( WMH*
         (HU"N,M - 1/3*&"N'M*HU"K,K)
        + 1/3*&"N'M*BDGP*HU"K*PS,K )
       + PMH2*Q/BLAM*( WWH1*HU"K,K + WWH2*BDGP*HU"K*PS,K ))  ),L
     + PS,L*HU"L
     + (EMU*( US'M!L + US"L,M )
      + EMUT*( WMT*(TU'M!L + TU"L,M - 2/3*&"L'M*TU"K,K)
       + 2/3*&"L'M*BDGP*TU"K*PS,K ))*(WMH*( HU"M,L
       - 1/3*&"M'L*HU"N,N ) + 1/3*&"M'L*BDGP*HU"N*PS,N)
     + (EMS*US"K,K + EMST*BDGP*TU"K*PS,K)*BDGP*HU"L*PS,L
     + EMU*US"L,M*(WMH*( HU'L!M + HU"M,L - 2/3*&"M'L*HU"K,K )
      + 2/3*&"M'L*BDGP*HU"K*PS,K)
     + EMUT/CP*US"L,M*(US'L!M + US"M,L)*HH
     + EMST/CP*US"L,L*US"M,M*HH + EMS*US"L,L*BDGP*HU"M*PS,M   )
    + (CAP*HH,L)!L - 2*CAP*HHDMF*HH
    + CAYT/CP/CP*(2*HS,L!L*HH + HS,L*HH!L)



#                                                    RU

RS*US"L*RU'I,L - (VRU*RQL*(RU'I!L + RU"L,I)
   + VRR*QL*US"L*RR,I),L - RU"L,L*RU'I
  + RR*US"L*US'1,L + US"L*RS,L*RU'I
  + RS*US'I,L*RU"L + RS*RS,L*UU"L'I
  - VRR*QL*RR!L*US'I,L - VRU*QL*RS,L*(RU'I!L + RU"L,1)
   = -(RS*BLAM*BLAM*(PMR*(WMR*(US"M,N*RU"N,M - 1/3*US"M,M*RU"L,L)
       + 1/3*US"M,M*BDGP*RU"L*PS,L)
      + PMR2*Q/BLAM*(WWR1*RU"L,L + WWR2*BDGP*RU"L*PS,L))),I
    + PGR*RS*(WGR1*US"M,M*RU'I + WGR2*US"M,I*RU'M
     + WGR3*US'1,M*RU"M) + PGR2*WWGR*RS*Q/BLAM*RU'I
    - RS*RS*BDGP*UU"L'I*PS,L - 2*RS*US"L,L*RU'I
    + 2*VBS*VRU*RQL*BDGP*PS,L*(RU'I!L + RU"L,I)
    + VRR*QL*US"L,L*RR,I
    + EMU*(WMR*(RU'I,L - 1/3*@'IL*RU"K,K)
     + 1/3*@'IL*BDGP*RU"K*PS,K)!L - EMU*RUDMF*RU'I
    + 1/2*(EMU + EMS)*(BDGP*RU"L*PS,L),I
    + EMU,L*(WMR*(RU'I!L + RU"L,I- 2/3*&"L'I*RU"K,K)
     + 2/3*&"L'I*BDGP*RU"K*PS,K)
    + EMUT*((US'I,L + US'L,I)*1/2*RT!L + (US'I,L + US'L,I)!L*RT)
    + EMST*(US"M,M*1/2*RT,I + US"M,MI*RT)
    + EMS,I*BDGP*RU"M*PS,M
```

APPENDIX D.- THE FINAL EQUATIONS

```
#          FIRST PASS


 RSUS*(R*WS):1
   + RSVS*(R*WS):2
   + RV*(R*WS):2
   + (RS*R*VW):2
    = (FMU*(R*WS):2):2


RSUS*(R*VW):1
  + RSVS*(R*VW):2
  - 2*C*RS*WW*WS
  - 2*(VUU*RQL*(R*VW):2):2
  - RV2*R*VW
  + US*RV*(R*WS):1
  + VS*RV*(R*WS):2
  - C*WS*RW*WS
  + RS*UV*(R*WS):1
  + RS*VV*(R*WS):2
  + RS*R*VW*VS2
  - 2*VRU*QL*RV2*(R*WS):2
   = -1/4*PMU*R*R*VV2*(RBLS*(1/R*WS):2):2
     - 1/4*PMU*RBLS*WSR2*(R*R*(VV):2):2
     - (PMU2*WWU1*RQL*(R*VW):2):2
     + PGU*WGU*RS*R*R*UV*(1/R*WS):1
     + PGU*WGU*RS*R*R*VV*(1/R*WS):2
     + 1/3*S*PGU*WGU*RSUS*VW
     + S*PGU*WGU*RS*UV*WS
     + C*PGU*WGU*RS*VV*WS
     + 1/3*PGU*WGU*RS*VS2*R*VW
     - C*PGU*WGU*RS*WS*WW
     - 2/3*PGU*WGU*RS*US1*R*VW
     - RQOL*R*VW
     + (EMU*(R*VW):2):2
     - 2*A*EMU/BLAM/BLAM*R*VW
     - 2*B*BETA*RQOL*R*VW
     + 1/4*EMU2*(R*VW):2
     + (1 - WMT/3)*EMUT*TV2*(R*WS):2
     + EMUT*TV*((R*WS):2):2
```

```
#          SECOND PASS


 RSUS*(US):1
    + RSVS*(US):2
    - S*RS/R*WS*WS
    + RV*(US):2
    + (RS*UV):2
     = -DPDX
       + (EMU*(US):2):2


RSUS*(UV):1
   + RSVS*(UV):2
   - 2*S*RS/R*WS*VW
   - 2*C*RS/R*WS*UW
   - 2*(VUU*RQL*(UV):2):2
   - RV2*UV
   - C/R*WS*WS*RU
   + US*RV*(US):1
   + VS*RV*(US):2
   - S/R*WS*WS*RV
   + RS*VS2*UV
   + RS*US1*UV
   + RS*VV*(US):2
   - 2*VRU*QL*RV2*(US):2
    = -1/4*PMU*VV2*(RSLS*(US):2):2
      - 1/4*PMU*RSLS*US2*((VV):2):2
      - (PMU2*WWU1*RQL*(UV):2):2
      + 1/3*PGU*WGU*RS*VS2*UV
      - C*PGU*WGU*RS/R*WS*UW
      + 1/3*PGU*WGU*RS*US1*UV
      + PGU*WGU*RS*VV*(US):2
      - S*PGU*WGU*RS/R*WS*VW
      - 2/3*S*PGU*WGU*RS/R*US*UV
      - RQOL*UV
      + (EMU*(UV):2):2
      - 2*A*EMU/BLAM/BLAM*UV
      - 2*B*BETA*RQOL*UV
      + 1/4*EMU2*(UV):2
      + (1 - WMT/3)*EMUT*TV2*(US):2
      + EMUT*TV*((US):2):2


RSUS*(R*UW):1
   + RSVS*(R*UW):2
   - 2*S*RS*WS*WW
   - (VUU*RQL*(R*UW):2):2
   - RV2*R*UW
```

```
+ US*RU*RWS1
+ VS*RU*RWS2
+ R*US*RW*(US):1
+ R*VS*RW*(US):2
- S*WS*WS*RW
+ RS*UU*RWS1
+ RS*UV*RWS2
+ RS*R*US1*UW
+ RS*R*VW*(US):2
- VRU*QL*RWS2*(RU):2
- VRU*QL*US2*(R*RW):2
 = PGU*WGU*RS*WSR1*R*R*UU
   + PGU*WGU*RS*WSR2*R*R*UV
   + 1/3*S*PGU*WGU*RSUS*UW
   + S*PGU*WGU*RS*WS*UU
   + C*PGU*WGU*RS*WS*UV
   + 1/3*PGU*WGU*RS*US1*R*UW
   + PGU*WGU*RS*US2*R*VW
   - S*PGU*WGU*RS*WS*WW
   - 2/3*PGU*WGU*RS*VS2*R*UW
   - RQOL*R*UW
   + (EMU*(R*UW):2):2
   - 2*A*EMU/BLAM/BLAM*R*UW
   - 2*R*BETA*RQOL*R*UW
   + (1 - WMT)*EMUT*RTW2*(US):2
   + (1 - WMT)*EMUT*RWS2*(TU):2
   + EMUT*R*TW*((US):2):2
   + EMUT*TU*RWS22
```

```
#          THIRD PASS


RSUS*(UU):1
  + RSVS*(UU):2
  - 4*S*RS/R*WS*UWJ
  - (VUU*RQL*(UU):2):2
  - RV2*UU
  + 2*US*RU*US1
  + 2*VS*RU*US2
  - 2*S/R*WS*WS*RU
  + 2*RS*US1*UU
  + 2*RS*US2*UVJ
  - 2*VRU*QL*US2*(RU):2
   = 8/9*PGU*WGU*RS*US1*UU
     + 4/3*PGU*WGU*RS*US2*UVJ
     - 2*S*PGU*WGU*RS/R*WS*UWJ
     - 4/9*PGU*WGU*RS*VS2*UU
     - 4/9*S*PGU*WGU*RS/R*US*UU
     - 4/9*PGU*WGU*RS*VS2*VV
     - 2/3*PGU*WGU*RS*WSR1*R*UWJ
     - 2/3*PGU*WGU*RS*WSR2*R*VWJ
     - 4/9*S*PGU*WGU*RS/R*US*WW
     + 2/9*PGU*WGU*RS*US1*VV
     + 2/9*PGU*WGU*RS*US1*WW
     + 2/9*PGU*WGU*RS*VS2*WW
     + 2/9*S*PGU*WGU*RS/R*US*VV
     - 2/3*RQOL*UU
     + 1/3*RQOL*VV
     + 1/3*RQOL*WW
     + (EMU*(UU):2):2
     - 2*A*EMU/BLAM/BLAM*UU
     - 2*B*BETA*RQOL*UU
     - 2/3*B*OMBET*RQOL*UU
     - 2/3*B*OMBET*RQOL*VV
     - 2/3*B*OMBET*RQOL*WW
     + 1/2*EMU2*(VV):2
     + 2*(1 - WMT)*EMUT*US2*(TU):2
     + 2*EMUT*TU*US22.


RSUS*(VV):1
  + RSVS*(VV):2
  - 4*C*RS/R*WS*VWJ
  - 3*(VUU*RQL*(VV):2):2
  - RV2*VV
  - 2*C/R*WS*WS*RV
  + 2*RS*VS2*VV
   = -2*(PMU2*WWU1*RQL*(VV):2):2
```

```
      + 8/9*PGU*WGU*RS*VS2*VV
      - 2*C*PGU*WGU*RS/R*WS*VWJ
      - 4/9*PGU*WGU*RS*US1*VV
      - 4/9*S*PGU*WGU*RS/R*US*VV
      - 4/9*PGU*WGU*RS*US1*UU
      - 2/3*PGU*WGU*RS*US2*UVJ
      - 2/3*PGU*WGU*RS*WSR1*R*UWJ
      - 2/3*PGU*WGU*RS*WSR2*R*VWJ
      - 4/9*S*PGU*WGU*RS/R*US*WW
      + 2/9*PGU*WGU*RS*US1*WW
      + 2/9*PGU*WGU*RS*VS2*UU
      + 2/9*PGU*WGU*RS*VS2*WW
      + 2/9*S*PGU*WGU*RS/R*US*UU
      - 2/3*RQOL*VV
      + 1/3*RQOL*UU
      + 1/3*RQOL*WW
      + (EMU*(VV):2):2
      - 2*A*EMU/BLAM/BLAM*VV
      - 2*B*BETA*RQOL*VV
      - 2/3*B*OMBET*RQOL*UU
      - 2/3*B*OMBET*RQOL*VV
      - 2/3*B*OMBET*RQOL*WW
      + EMU2*(VV):2


RSUS*(R*R*WW):1
   + RSVS*(R*R*WW):2
   - (VUU*RQL*(R*R*WW):2):2
   - RV2*R*R*WW
   + 2*R*US*RW*RWS1
   + 2*R*VS*RW*RWS2
   + 2*RS*R*RWS1*UWU
   + 2*RS*R*RWS2*VWU
   - 2*VRU*QL*RWS2*(R*RW):2
    = 4/3*PGU*WGU*RS*WSR1*R*R*R*UWJ
      + 4/3*PGU*WGU*RS*WSR2*R*R*R*VWJ
      + 8/9*S*PGU*WGU*RS*R*US*WW
      + 2*S*PGU*WGU*RS*R*WS*UWJ
      + 2*C*PGU*WGU*RS*R*WS*VWJ
      - 4/9*PGU*WGU*RS*US1*R*R*WW
      - 4/9*PGU*WGU*RS*VS2*R*R*WW
      - 4/9*PGU*WGU*RS*US1*R*R*UU
      - 2/3*PGU*WGU*RS*US2*R*R*UV
      - 4/9*PGU*WGU*RS*VS2*R*R*VV
      + 2/9*PGU*WGU*RS*US1*R*R*VV
      + 2/9*PGU*WGU*RS*VS2*R*R*UU
      + 2/9*S*PGU*WGU*RS*R*US*UU
      + 2/9*S*PGU*WGU*RS*R*US*VV
      - 2/3*RQOL*R*R*WW
```

```
     + 1/3*RQOL*R*R*UU
     + 1/3*RQOL*R*R*VV
     + (EMU*(R*R*WW):2):2
     - 2*A*EMU/BLAM/BLAM*R*R*WW
     - 2*B*BETA*RQOL*R*R*WW
     - 2/3*B*OMBET*RQOL*R*R*UU
     - 2/3*B*OMBET*RQOL*R*R*VV
     - 2/3*B*OMBET*RQOL*R*R*WW
     + 1/2*EMU2*R*R*(VV):2
     + 2*(1 - WMT)*EMUT*RWS2*(R*TW):2
     + 2*EMUT*R*TW*RWS22
```

```
#          FOURTH PASS
 RSUS*(HS):1
   + RSVS*(HS):2
   + RV*(HS):2
   + (RS*HV):2
    = GMOMS*US*DPDX
      + GMOMS*EMU*US2*USJ2
      + GMOMS*EMU*RWS2*WSRJ2
      + A*GMOMS*EMU/BLAM/BLAM*QQJ
      + B*GMOMS*RQOL*QQJ
      + (CAP*(HS):2):2


RSUS*(HV):1
  + RSVS*(HV):2
  - 2*C*RS/R*WS*HW
  - 2*(VUH*RQL*(HV):2):2
  - RV2*HV
  + US*RV*(HS):1
  + VS*RV*(HS):2
  - C/R*WS*WS*RH
  + RS*VV*(HS):2
  + RS*VS2*HV
  - 2*VRU*QL*RV2*(HS):2
   = GMOMS*UV*DPDX
     + GMOMS*VV*DPDY
     - (PMH2*WWH1*RQL*(HV):2):2
     + PGH*WGH1*RS*US1*HV
     + PGH*WGH1*RS*VS2*HV
     + S*PGH*WGH1*RS/R*US*HV
     + PGH*WGH2*RS*US2*HU
     + PGH*WGH2*RS*VS2*HV
     + PGH*WGH2*RS*WSR2*R*HW
     + C*PGH*WGH2*RS/R*WS*HW
     + PGH*WGH3*RS*VS2*HV
     - C*PGH*WGH3*RS/R*WS*HW
     + PGH2*WWGH*RQOL*HV
     + 3/2*GMOMS*EMU*US2*UVJ2
     + 3/4*GMOMS*EMU*RWS2*VWRJ2
     + 3/4*GMOMS*EMU*WSR2*RVWJ2
     + GMOMS*EMUT/CP*US2*USJ2*HV
     + GMOMS*EMUT/CP*WSR2*RWSJ2*HV
     + (1 - 2*WMT/3)*(CAP*(HV):2):2
     - CAP*HUDMF*HV
     + CAYT/CP/CP*(HV*(HS):2):2
     - 2/3*CAYT/CP/CP*HS2*WMT*(HV):2
     + 2/3*EMU*(WMH*(HV):2):2
     - EMU*HUDMF*HV
     + 4/3*EMU2*WMH*(HV):2


RSUS*(HH):1
  + RSVS*(HH):2
  - VHH*(RQL*(HH):2):2
  - RV2*HH
```

```
+ 2*US*RH*(HS):1
+ 2*VS*RH*(HS):2
+ 2*RS*(HS):2*HV
- 2*VRH*QL*HS2*(RH):2
 = 2*GMOMS*DPDX*HU
  + 2*GMOMS*DPDY*HV
  + 4*GMOMS*EMU*US2*WMH*(HU):2
  + 2*GMOMS*EMU*RWS2*WMH*(1/R*HW):2
  + 2*GMOMS*EMU*WSR2*WMH*(R*HW):2
  + 2*GMOMS*EMUT/CP*US2*USJ2*HH
  + 2*GMOMS*EMUT/CP*WSR2*RWSJ2*HH
  + (CAP*(HH):2):2
  - 2*CAP*HHDMF*HH
  + 2*CAYT/CP/CP*HH*((HS):2):2
  + CAYT/CP/CP*HS2*(HH):2
```

```
#          FIFTH PASS


RSUS*(RU):1
   + RSVS*(RU):2
   - 2*S*RS/R*WS*RW
   - (VRU*RQL*(RU):2):2
   - RV2*RU
   + US*RR*US1
   + VS*RR*US2
   - S/R*WS*WS*RR
   + US*RS1*RU
   + VS*RS2*RU
   + 3*RS*US1*RU
   + RS*US2*RV
   + RS*RS2*UVJ
   - VRR*QL*RR2*US2
   - VRU*QL*RS2*(RU):2
    = PGR*WGR1*RS*US1*RU
      + PGR*WGR1*RS*VS2*RU
      + S*PGR*WGR1*RS/R*US*RU
      + PGR*WGR2*RS*US1*RU
      + PGR*WGR2*RS*WSR1*R*RW
      + S*PGR*WGR2*RS/R*WS*RW
      + PGR*WGR3*RS*US1*RU
      + PGR*WGR3*RS*US2*RV
      - S*PGR*WGR3*RS/R*WS*RW
      + PGR2*WWGR*RQOL*RU
      - 2*RS*VS2*RU
      - 2*S*RS/R*US*RU
      + EMU*(WMR*(RU):2):2
      - EMU*RUDMF*RU
      + EMU2*WMR*(RU):2
      + 1/2*EMUT*US2*(RT):2
      + EMUT*US22*RT


RSUS*(RV):1
   + RSVS*(RV):2
   - 2*C*RS/R*WS*RW
   - 2*(VRU*RQL*(RV):2):2
   - RV2*RV
   - C/R*WS*WS*RR
   + US*RS1*RV
   + VS*RS2*RV
   + 3*RS*VS2*RV
   + RS*RS2*VV
   - 2*VRU*QL*RS2*(RV):2
    = -(PMR2*WWR1*RQL*(RV):2):2
```

```
      + PGR*WGR1*RS*US1*RV
      + PGR*WGR1*RS*VS2*RV
      + S*PGR*WGR1*RS/R*US*RV
      + PGR*WGR2*RS*US2*RU
      + PGR*WGR2*RS*VS2*RV
      + PGR*WGR2*RS*WSR2*R*RW
      + C*PGR*WGR2*RS/R*WS*RW
      + PGR*WGR3*RS*VS2*RV
      - C*PGR*WGR3*RS/R*WS*RW
      + PGR2*WWGR*RQOL*RV
      - 2*RS*US1*RV
      - 2*S*RS/R*US*RV
      + 2/3*EMU*(WMR*(RV):2):2
      - EMU*RUDMF*RV
      + 4/3*EMU2*WMR*(RV):2


RSUS*(R*RW):1
   + RSVS*(R*RW):2
   + 2*S*RSUS*RW
   - (VRU*RQL*(R*RW):2):2
   - RV2*R*RW
   + US*RR*RWS1
   + VS*RR*RWS2
   + US*RS1*R*RW
   + VS*RS2*R*RW
   + RS*RWS1*RU
   + RS*RWS2*RV
   + RS*RS2*R*VW
   - VRR*QL*RR2*RWS2
   - VRU*QL*RS2*(R*RW):2
    = PGR*WGR1*RS*US1*R*RW
      + PGR*WGR1*RS*VS2*R*RW
      + S*PGR*WGR1*RSUS*RW
      - S*PGR*WGR2*RS*WS*RU
      - C*PGR*WGR2*RS*WS*RV
      + S*PGR*WGR2*RSUS*RW
      + PGR*WGR3*RS*RWS1*RU
      + PGR*WGR3*RS*RWS2*RV
      + S*PGR*WGR3*RSUS*RW
      - S*PGR*WGR3*RS*WS*RU
      - C*PGR*WGR3*RS*WS*RV
      + PGR2*WWGR*RQOL*R*RW
      - 2*RS*US1*R*RW
      - 2*RS*VS2*R*RW
      + EMU*(WMR*(R*RW):2):2
      - EMU*RUDMF*R*RW
      + EMU2*WMR*(R*RW):2
      + 1/2*EMUT*RWS2*(RT):2
      + EMUT*RWS22*RT
```

```
#          SIXTH PASS
RSUS*(HU):1
  + RSVS*(HU):2
  - 2*S*RS/R*WS*HW
  - (VUH*RQL*(HU):2):2
  - RV2*HU
  + US*PU*(HS):1
  + VS*RU*(HS):2
  + US*RH*US1
  + VS*RH*US2
  - S/R*WS*WS*RH
  + RS*UV*(HS):2
  + RS*US1*HU
  + RS*US2*HVJ
  - VRU*QL*RU2*HSJ2
  - VRH*QL*RH2*USJ2
   = GMOMS*UU*DPDX
    + GMOMS*UV*DPDY
    + PGH*WGH1*RS*US1*HU
    + PGH*WGH1*RS*VS2*HU
    + S*PGH*WGH1*RS/R*US*HU
    + PGH*WGH2*RS*US1*HU
    + PGH*WGH2*RS*WSR1*R*HW
    + S*PGH*WGH2*RS/R*WS*HW
    + PGH*WGH3*RS*US1*HU
    + PGH*WGH3*RS*US2*HVJ
    - S*PGH*WGH3*RS/R*WS*HW
    + PGH2*WWGH*RQOL*HU
    + GMOMS*EMU*US2*UUJ2
    + 1/2*GMOMS*EMU*RWS2*UWRJ2
    + 1/2*GMOMS*EMU*US2*VVJ2
    + 1/2*GMOMS*EMU*WSR2*RUWJ2
    + GMOMS*EMUT/CP*US2*USJ2*HU
    + GMOMS*EMUT/CP*WSR2*RWSJ2*HU
    + (CAP*(HU):2):2
    - CAP*HUDMF*HU
    - (WMT*CAP*(HU):2):2
    + CAYT/CP/CP*(HU*(HS):2):2
    - CAYT/CP/CP*HS2*WMT*(HU):2
    + EMU*(WMH*(HU):2):2
    - EMU*HUDMF*HU
    + EMU2*WMH*(HU):2
    + 1/2*EMUT/CP*US2*(HH):2
    + EMUT/CP*US22*HH


RSUS*(R*HW):1
  + RSVS*(R*HW):2
  - (VUH*RQL*(R*HW):2):2
  - RV2*R*HW
  + R*US*RW*(HS):1
  + R*VS*RW*(HS):2
  + US*RH*RWS1
  + VS*RH*RWS2
  + RS*R*VW*(HS):2
```

```
  + RS*RWS1*HU
  + RS*RWS2*HVJ
  - VRU*QL*RRW2*HSJ2
  - VRH*QL*RH2*RWSJ2
   = GMOMS*R*UW*DPDX
     + GMOMS*R*VW*DPDY
     + PGH*WGH1*RS*US1*R*HW
     + PGH*WGH1*RS*VS2*R*HW
     + S*PGH*WGH1*RSUS*HW
     - S*PGH*WGH2*RS*WS*HU
     - C*PGH*WGH2*RS*WS*HVJ
     + S*PGH*WGH2*RSUS*HW
     + PGH*WGH3*RS*RWS1*HU
     + PGH*WGH3*RS*RWS2*HVJ
     + S*PGH*WGH3*RSUS*HW
     - S*PGH*WGH3*RS*WS*HU
     - C*PGH*WGH3*RS*WS*HVJ
     + PGH2*WWGH*RQOL*R*HW
     + GMOMS*EMU*US2*RUWJ2
     + 1/2*GMOMS*EMU*RWS2*WWJ2
     + 1/2*GMOMS*EMU*WSR2*R*R*VVJ2
     + 1/2*GMOMS*EMU*WSR2*RWRJ2
     + GMOMS*EMUT/CP*US2*USJ2*R*HW
     + GMOMS*EMUT/CP*WSR2*RWSJ2*R*HW
     + (CAP*(R*HW):2):2
     - CAP*HUDMF*R*HW
     - (WMT*CAP*(R*HW):2):2
     + CAYT/CP/CP*(R*HW*(HS):2):2
     - CAYT/CP/CP*HS2*WMT*(R*HW):2
     + EMU*(WMH*(R*HW):2):2
     - EMU*HUDMF*R*HW
     + EMU2*WMH*(R*HW):2
     + 1/2*EMUT/CP*RWS2*(HH):2
     + EMUT/CP*RWS22*HH

BLAM = GYCSZ
```

APPENDIX E.- TENSR NOTATION EXAMPLES

; is used as a separator in these examples.

Common Notation	TENSR Notation
Scalars	
A ; ϕ ; t_2 ; ν	A ; PHI ; TIME2 ; NU
Covariant Vectors	
A_i ; ϕ_r ; V_α	A'I ; PHI'R ; V'A
Contravariant Vectors	
A^i ; ϕ^r ; V^α	A"I ; PHI"R ; V"A
Various Tensors	
A^{ij}	A"IJ or A"I"J
A_{ij}	A'IJ or A'I'J
$A^i{}_j$; $A_i{}^j$	A"I'J ; A'I"J
$\sum^{ijm}_{km}$	CAPSIG"IJM'KM
Covariant Derivative	
$\phi_{,i}$	PHI,I
Partial Derivative	
$\frac{\partial \phi}{\partial x^i}$	PHI:I

Common Notation	TENSR Notation

Metric (or Fundamental) Tensor

g_{ij} ; g^{ij}	@'IJ ; @"IJ

Kronecker Delta

$\delta^i_j = g^i_j$	&"I'J = @"I'J
δ_{ij} ; δ^{ij}	&'IJ ; &"IJ

Covariant Derivative with Index Raised
(or "Contravariant Derivative")

$g^{im}\phi_{,m} = \phi^{,i} = \phi_{,}{}^{i}$	@"IM*PHI,M = PHI!I

APPENDIX F.- GLOSSARY

This glossary includes:

1) The names of the variables appearing in the tensor equations of Appendix C and in the final equations of Appendix D.

2) The names of all the subprograms of the GYC system.

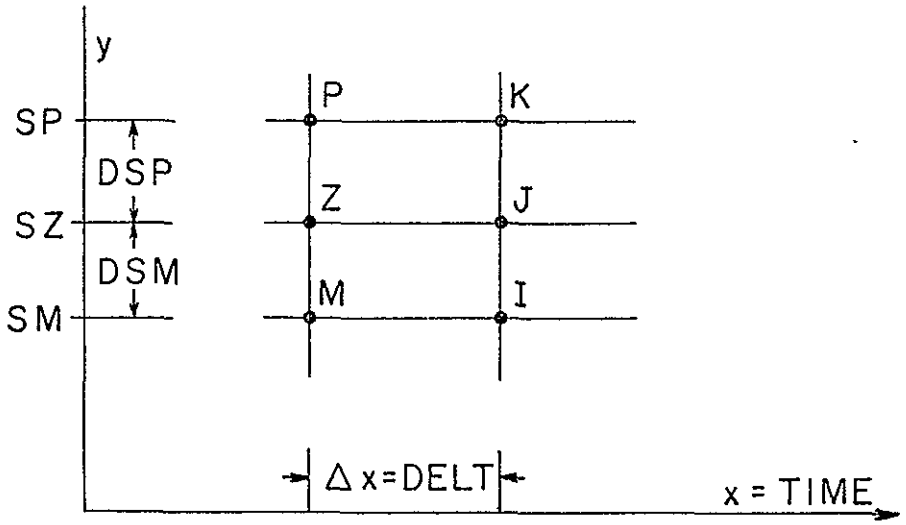
3) The Fortran names of all the input variables.

4) The Fortran names of some of the important variables of the programs.

5) The labels used on the program output.

Many of the entries fit in several of the above categories.

In the list, several of the names end with __. This is used to indicate that the name may appear with any of the following appearing in place of __: M, Z, P, I, J, K, 1, 2, J2, 22, or none of them. The letters refer to the six-point finite-difference mesh used in the main solution process as indicated in the sketch.



The values denoted by names ending in M, Z, and P, at the old x, are known. Those ending in I, J, and K may be known or unknown depending on the stage of the computation. The suffix 1 indicates the first derivative with respect to x between Z and J, 2 the first derivative with respect to y at Z, J2 the first derivative with respect to y at J, and 22 the second derivative with respect to y at Z. A name that has a different meaning without any such suffix is listed both with and without __, e.g., S. On the other hand, a variable that carries the same basic meaning without a suffix is listed once. For example, HH appears as such in Appendices C and D, and as an input variable, as well as in the form HHM, HHZ, etc. In any case it signifies $\overline{h'h'}$, so it appears in the list only as HH__.

The column labeled Dimensions gives the dimensions of the quantity in terms of a density, D, a velocity, V, a length, L, a temperature, θ, and an enthalpy per unit mass, H. (If mechanical units are used for heat, then $H = V^2$. The substitutions $D = ML^{-3}$ and $V = LT^{-1}$ will transform this set into a more familiar one.) Absence of any indication means the quantity is nondimensional. The symbol W means the elements of the vector have the same dimensions as the corresponding elements of the vector XV as given in Table 5. The combination W,N used with EPSBV and EPSTV means W if the element is input as a positive number and nondimensional if it is input as a negative number. Finally, X indicates some other mixture of dimensions for different elements.

At the end of several of the definitions there are expressions in square brackets. These are evaluations in terms of Mach number, Reynolds number, etc. of the variables when they are interpreted as nondimensional quantities under the following conditions:

1) The nondimensionalization is done with respect to a length, L_r , and with respect to a reference density, ρ_r , velocity, u_r , temperature, T_r , and enthalpy per unit mass, h_r , that all refer to the same condition or point in the flow.

2) The reference state for the transport coefficients is that same condition, i.e., $T_{rf} = T_r$.

An example of how they are derived is given for the case of GMOMS in Section 2.

Notes to the Glossary appear at the end.

Name	Dimensions	Definition
A		Modeling parameter, a . See Note 2.
AHH		Modeling parameter. See Note 2.
AHU		Modeling parameter. See Note 2.
AMAT	X	Matrix A (Section 3).
ARU		Modeling parameter. See Note 2.
AV_	W	Vectors of the unknowns at the new x . AVM, AVZ, and AVP would more logically be called XVI, XVJ, and XVK.
B		Modeling parameter, b . See Note 2.
BDGP	$1/(DV^2)$	$-BDV/(\gamma\bar{p})$
BDV		Modeling parameter. See Note 2.
BETA		Modeling parameter, β . See Note 2.
BHH		Modeling parameter. See Note 2.
BHU		Modeling parameter. See Note 2.

Name	Dimensions	Definition
BLAM_	L	Turbulence scale length. See Tables 3 and 5 and Note 1.
BLAM0	L	Scale parameter. See Note 1.
BMAT	X	Matrix B (Section 3).
BREAK POINTS	L	TBRKV
BRU		Modeling parameter. See Note 2.
BUFAC		Maximum ratio by which EPSTV criteria may be exceeded without causing backup.
C		DIMFL* cos θ_c
CAP_	DVL	k/c_p
CAY_	DHVL/θ	Heat conductivity, k .
CAYRF	DHVL/θ	Reference value of k : its value at TMRF. See Note 3. [$1/(Re_r Pr)$]
CAYT_	$DHVL/\theta^2$	Derivative of k with respect to temperature.
CF		Skin friction coefficient.
CHANGE PER STEP	W,N	EPSTV
CLAMB		Scale parameter. See Note 1.
CMAT	X	Matrix C (Section 3).
CMNTS		Up to 76 characters of comments to be printed on the title pages.
CONGL		Cone half-angle, θ_c , in degrees. See Note 5.
CP	H/θ	Specific heat at constant pressure, c_p , assumed constant. [1.]

Name	Dimensions	Definition
CSUTH	θ	Constant in the viscosity law. See Note 3.
DELT	L	Step size in the x direction, Δx .
DELTA*	L	Displacement thickness.
DELTI	L	Input value of DELT. See Note 7.
DIMFL		Important: see Note 5. Dimension flag. A value of zero signifies flat-plate mode; a value of one signifies axisymmetric mode. Any other input value is converted to one.
DIN		Scale parameter. See Note 1.
DH/DY	H/L	The wall value of $\frac{\partial \bar{h}}{\partial y}$.
DLT99	L	Value of y for which $\bar{u}$ is 99 percent of its free stream value.
DOUT		Scale parameter. See Note 1.
DPDX	DV^2/L	Partial derivative of $\bar{p}$ with respect to x .
DPDY	DV^2/L	Partial derivative of $\bar{p}$ with respect to y .
DSM	L	SZ - SM = h_- (Section 3).
DSP	L	SP - SZ = h_+ (Section 3).
DST	L	SP - SM = h_t (Section 3).
DTFMX		Maximum ratio by which Δx may increase from one step to the next.
DTI*	L	Velocity thickness.
DTMAX	L	Maximum Δx .
DTMIN	L	Minimum Δx .

Name	Dimensions	Definition
DU/DX	V/L	$\bar{u}_x$
DVEC	X	Vector D (Section 3).
DZFMN		DZFMN*TYPL is the minimum Δy .
DZFMX		DZFMX*TYPL is the maximum Δy .
DZRMX		Maximum ratio of two adjacent values of Δy .
ECMN		Governs dropping of points from the profile.
ECMNI		Initial value of ECMN at each step.
ECMX		Governs adding points in the profile.
ECMXI		Initial value of ECMXI at each step.
EDGE TOLERANCE	W,N	EPSBV
EDGEV	W	Free stream values of dependent variables.
EMS	DVL	Second coefficient of viscosity, μ^* .
EMST	DVL/θ	Derivative of μ^* with respect to temperature.
EMU_	DVL	First coefficient of viscosity, μ .
EMURF	DVL	Reference value of μ : its value at TMRF. See Note 3. $[1/Re_r]$
EMUT_	DVL/θ	Derivative of μ with respect to temperature.
EPSBV	W,N	The solution is extended in the radial direction until the difference between the calculated value of each parabolic variable and its asymptotic value is less than EPSBV(I) or -EPSBV(I)*GMAXV(I), whichever is positive.

Name	Dimensions	Definition
EPSNV	W	When ABS(GMAXV(I)) is less than ABS(EPSNV(I)), the corresponding variable is not checked in the determination of DELT, nor in considering where to insert or drop points, nor in checking the outer boundary conditions if EPSBV(I) is negative. Furthermore, in the determination of Δx , if EPSNV(I) is negative, the change in the corresponding variable is compared with the larger of -EPSNV(I) and the criterion derived from EPSTV(I).
EPSTV	W,N	Δx is controlled by attempting to keep the change, TCMXV(I), in each of the parabolic variables (in the axial direction) below EPSTV(I) or -EPSTV* GMAXV(I), whichever is positive.
EV		See Table 2.
EXPMU		Exponent in the viscosity law. See Note 3.
FAC99		See Note 1.
FCMX		Maximum factor by which ECMNI and ECMXI may be increased through repeated applications of FCURI.
FCURI		Factor by which ECMN and ECMX are increased when JMAX impedes adding points in the profile.
FHM	1/L	$-H_-$ (Section 3).
FHP	1/L	H_+ (Section 3).
FHZ	1/L	$H_- - H_+$ (Section 3).
FINTV	L	Spreads of the dependent variables based on their integrals (output label: INT SPREAD).

Name	Dimensions	Definition
FMAXV	W	Maxima over y of the dependent variables (output label: MAXIMUM).
FUP		Upwind differencing parameter. See Note 4.
FUPV		Upwind differencing parameter. See Note 4.
FV		See Table 3.
FZMXV	L	Locations of the maxima (output label: Y AT MAX).
GLOBAL MAX	W	GMAXV
GMAT	X	Matrix Γ (Section 3).
GMAXV	W	Maxima over y and x of the dependent variables (output label: GLOBAL MAX).
GMOMS	H/V^2	Important: see Note 6. The reciprocal of the mechanical equivalent of heat. $\left[(\gamma-1)M_r^2\right]$
GV		See Table 4.
GYCAA		Subroutine which adjusts the spacing of points in the y direction.
GYCAL		Subroutine which controls output and auxiliary computations.
GYCAQ		Subroutine which computes spreads and the next step size and which initializes for the next step.
GYCBI		Subroutine which implements the wall boundary conditions.
GYCBO		Subroutine which implements the outer boundary conditions.

Name	Dimensions	Definition
GYCCM		COMMON file.
GYCCP		Subroutine which calculates the pressure distribution and the density distribution when they need to be determined separately.
GYCDZ		Subroutine which determines the position in y of a new point which is to be inserted.
GYCEW		Subroutine which interpolates to determine values of the specified free-stream variables at the current x .
GYCFP		Subroutine which transfers from STOR, or calculates, the values in the section of COMMON from QP to USK1.
GYCIC		Subroutine which handles some of the initialization for a new run.
GYCIN		Subroutine which controls start or restart procedures.
GYCIP		Subroutine which determines values of the dependent variables at a point inserted in the profile.
GYCLI		Subroutine for linear interpolation.
GYCMC		Subroutine which controls the computation of AMAT, BMAT, CMAT, and DVEC.
GYCMI		Subroutine which inverts 2 · 2 and 3 · 3 matrices.
GYCMP		Subroutine which computes a matrix product and optionally subtracts the product from a third matrix.
GYCNS		Subroutine which flags certain unlikely errors.

Name	Dimensions	Definition
GYCOI		Subroutine which produces a minor printout and the first part of a major printout.
GYCON		Main program of the GYC system.
GYCOT		Subroutine which produces the three profile tables of a major printout and also prints any error messages.
GYCPF		Ficticious subroutine which would write to a file for later plotting of profiles.
GYCPI		Subroutine which prints the first two pages of a new or restarted run.
GYCPS		Ficticious subroutine which would set up plot files.
GYCP1		Subroutine to GYCMC for the second pass.
GYCP2		Subroutine to GYCMC for the first pass.
GYCP3		Subroutine to GYCMC for the third pass.
GYCP4		Subroutine to GYCMC for the fourth pass.
GYCP5		Subroutine to GYCMC for the fifth pass.
GYCP6		Subroutine to GYCMC for the sixth pass.
GYCQD		Subroutine which calculates various integrals and determines maxima.
GYCRI		Subroutine which reads all but the first input card.
GYCRF		Ficticious subroutine which would write to a file for later plotting of "running variables" as a function of x .
GYCRV		Subroutine which determines the normal component of the mean velocity.

Name	Dimensions	Definition
GYCSC	L	Function which determines the value of the turbulent scale. See Note 1.
GYCSS		Subroutine which controls the main solution process.
GYCSZ	L	GYCSC(SZ)
GYCTB		Subroutine which makes a final adjustment to the step size, Δx , if needed to hit a break point.
GYCTC		Subroutine which evaluates k and μ and their derivatives with respect to temperature. See Note 3.
H	H	$\bar{h}$. See Tables 3 and 5.
HE	H	Free-stream value of $\bar{h}$. See Table 2.
HG	H/L	$\bar{h}_y$ at the wall.
HH_	H^2	$\overline{h'h'}$. See Tables 4 and 5.
HHDMF	$1/L^2$	(AHH + BHH*REL)/BLAMZ/BLAMZ
HS_	H	$\bar{h}$. See Table 5.
HSTAG	H	The stagnation enthalpy is ABS(HSTAG). A positive value indicates that the wall value of the mean enthalpy is specified on Cards 16.n. A negative value indicates that the wall value of the gradient of the mean enthalpy is specified on Cards 16.n. $\left[1 + \frac{\gamma-1}{2} M_r^2\right]$
HSW	H,H/L	The wall value of $\bar{h}$ or of $\frac{\partial \bar{h}}{\partial y}$. See Table 2.
HU_	HV	$\overline{h'u'}$. See Tables 4 and 5.
HUDMF	$1/L^2$	(AHU + BHU*REL)/BLAMZ/BLAMZ
HV_	HV	$\overline{h'v'}$. See Tables 4 and 5.

Name	Dimensions	Definition
HW_	HV	$\overline{h'w'}$. See Tables 4 and 5.
I		Index almost always used to range over the dependent variables.
INFLG		1 flags a new run (all cards are read); 2 flags a restart with changes (Cards 1 through 29 are read); 0 flags end of job. See Note 7.
INT SPREAD	L	FINTV
IOFAC		Scale parameter. See Note 1.
ITEST		Index of variable to which PCFMX and PCFSP apply.
J		Index used to range across the profile. J=1 corresponds to the wall; J=JTOP corresponds to the edge of the boundary layer (free stream).
JMAX		Maximum number of points to be allowed across the profile, i.e., the upper limit on JTOP.
JMXMX		Maximum permissible value of JMAX.
JTOP		Current number of points across the profile.
LAST DX	L	Δx for the step just completed.
LBURF		True indicates that the step is to be tried again with a smaller Δx .
LDELT		True indicates that Δx is less than its minimum.
LDETZ		True indicates that GYCMI has tried to invert a singular, or nearly singular, matrix.
LFOLF		True indicates that full output is to occur.

Name	Dimensions	Definition
LFOPF		True indicates that output to a profile plot file would occur.
LHMWK		True indicates that the wall value of enthalpy is specified; false that its gradient is specified.
LIOLF		True indicates that a minor printout is to occur.
LIOPF		True indicates that output to a running plot file would occur.
LJMXA		True indicates that more than JMAX points are needed to satisfy the curvature requirements.
LNWPG		True indicates that the next printout should start on a new page.
LPKRF		True indicates that more than JMAX points are needed to satisfy the outer boundary conditions.
LPOSV		True indicates a non-negative dependent variable.
LPPFF		True would indicate that a profile file was full.
LRHOF		True indicates a problem in the calculation of $\bar{\rho}$: either $\overline{h'^2} \geq \bar{h}^2$ or too many iterations in GYCCP.
LRORG		True indicates that points have been added or deleted in the current step.
LRPFF		True would indicate that a running file was full.
LRSTF		True indicates that a restart is in progress.

Name	Dimensions	Definition
LSTFL		True indicates that a new run is starting.
LTRNF		True indicates that the run is terminating.
LZHFF		True indicates that y has reached YMAX.
LZHRP		True indicates that correlations involving ρ' and h' are to be set to zero.
MAX CHANGE	W	TCMXV
MAXIMUM	W	FMAXV
NEXT DX	L	Δx for the next step.
NFOLP		Steps between major printouts.
NFOPP		Steps between fictitious writes to a profile plot file.
NIOLP		Steps between minor printouts.
NIOPP		Steps between fictitious writes to a running plot file.
NOISE THRESHOLD	W	EPSNV
NPASS		Number of the pass through the procedure of GYCSS.
NRUN		Run number.
NSKIP		BLYAA is skipped for the first NSKIP steps whether starting or restarting.
NSTEP		Number of the current step in the x direction.
NSTMX		The run is halted if the number of steps reaches NSTMX. See Note 7.

Name	Dimensions	Definition
NSVRT		NVART plus one for the dependent variable y ($S_{\_}$); 22.
NVARP		Number of parabolic variables; 17.
NVART		Total number of dependent variables saved for the old and the new value of x ; 21.
NWQRD		Number of words in COMMON representing the M-I, the Z-J, or the P-K conditions; 59.
NWSAV		Total number of variables saved; 43.
NWSGM		NWSAV plus the size of GMAT; 52.
NWSRD		NWSAV; 43.
NWVEC		Number of unknowns in a pass; 2 in the first pass, 3 in the others.
OFAC		Scale parameter. See Note 1.
OMBET		1 - BETA.
OUTPUT CONTROLS		Parameters input on Card 17.
P	DV^2	$\bar{p}$. See Note 6.
PCFMX		If FMAXV(ITEST) has changed more than PCFMX times the value it had the last time both BLYOT and BLYPF were called, a call to each is forced.
PCFSP		If ZPSM(ITEST) has changed more than PCFSP times the value it had the last time both BLYOT and BLYPF were called, a call to each is forced.
PCTMV		ZPSM(I) is the largest value of y for which XV(I) = PCTMV(I)*FMAXV(I).

Name	Dimensions	Definition
PCT SPREAD	L	ZPSM
PE	DV^2	Free-stream value of $\bar{p}$. See Table 2 and Note 7.
PGH		Modeling parameter. See Note 2.
PGH2		Modeling parameter. See Note 2.
PGR		Modeling parameter. See Note 2.
PGR2		Modeling parameter. See Note 2.
PGU		Modeling parameter. See Note 2.
PGU2		Modeling parameter. See Note 2.
PMH		Modeling parameter. See Note 2.
PMH2		Modeling parameter. See Note 2.
PMR		Modeling parameter. See Note 2.
PMR2		Modeling parameter. See Note 2.
PMU		Modeling parameter. See Note 2.
PMU2		Modeling parameter. See Note 2.
PS_	DV^2	$\bar{p}$. See Note 6.
PSTAG	DV^2	Important: see Note 6. The stagnation pressure is ABS(PSTAG). A positive value indicates a normal run; a negative value indicates that all correlations involving ρ' and h' are zeroed out at each step. (This is intended as a debugging device.) $\left[\left(1 + \frac{\gamma-1}{2} M_r^2\right)^{\gamma/(\gamma-1)} \Big/ \left(\gamma M_r^2\right)\right]$
PTHM		Modeling parameter. See Note 2.
PTS		JTOP

Name	Dimensions	Definition
PTUM		Modeling parameter. See Note 2.
Q_	V	$q = \sqrt{\overline{u'u'} + \overline{v'v'} + \overline{w'w'}}$
QL_	VV	$q\Lambda$
QQ	V^2	UUZ + VVZ + WWZ
QQJ	V^2	UUJ + VVJ + WWJ
QTOT	DHV	Total heat transfer rate in the y direction.
QWALL	DHV	Heat transfer rate out of the wall.
R_	L	Distance to the axis of the cone. See Note 5.
RBLS_	DL^2	$\tilde{\rho}\Lambda^2$
RDT	1/L	$1/\Delta x$
REL		$\tilde{\rho}q\Lambda/\tilde{\mu}$
RETHETA		Reynolds number based on momentum thickness.
REXW		Reynolds number based on wetted length.
RGAS	V^2/θ	Important: see Note 6. Gas constant. $\left[1/\left(\gamma M_r^2\right)\right]$
RH_	DH	$\overline{\rho'h'}$
RHO	D	$\tilde{\rho}$
RNOSE	L	Origin radius, r_0 . See Note 5.
ROTAT	V/L	Important: see Note 5. Rate of rotation of the cone in radians per unit time.

Name	Dimensions	Definition
RQL_	DVL	$\bar{\rho} q \Lambda$
RQOL	DV/L	$\bar{\rho} q / \Lambda$
RR_	D^2	$\overline{\rho' \rho'}$
RRW2	D^2V/L	$(\bar{\rho} \overline{\rho' w'})_y$
RS_	D	$\bar{\rho}$
RSUS	DV	RS*US
RSVS	DV	RS*VS
RT_	Dθ	$\overline{\rho' T'}$
RTW2	DVθ/L	$(\bar{\rho} \overline{T' w'})_y$
RU_	DV	$\overline{\rho' u'}$. See Tables 4 and 5.
RUDMF	$1/L^2$	(ARU + BRU*REL)/BLAMZ/BLAMZ
RUN		NRUN
RUWJ2	V^2	$(R \overline{u' w'})_y$ at the new value of x .
RV_	DV	$\overline{\rho' v'}$. See Tables 4 and 5.
RVWJ2	V^2	$(R \overline{v' w'})_y$ at the new value of x .
RW_	DV	$\overline{\rho' w'}$. See Tables 4 and 5.
RWRJ2	V^2L	$(R^2 \overline{w' w'})_y$ at the new value of x .
RWS_	DV	$R \bar{w}$
S		DIMFL* sin θ_c
S_	L	Normal independent variable, y .
SFVFL		Subroutine which fills a vector.
SFVMV		Subroutine which moves the contents of one vector to another.

Name	Dimensions	Definition
SHAPE		Displacement thickness divided by momentum thickness.
SHM	$1/L^2$	$1/(h\_h_t)$ (Section 3).
SHP	$1/L^2$	$1/(h_+h_t)$ (Section 3).
SPREAD PARAMETER		PCTMV
SPRO	L	ZPSM(IOFAC). See Note 1.
STANT		Stanton number.
STEP		NSTEP
STOR	X	Storage array mimicking an external device.
TAULAM	DV^2	Laminar stress.
TAUTOT	DV^2	Total stress.
TAUW	DV^2	Wall stress.
TBRKV	L	Time break vector. When TIME reaches the absolute value of one of the elements of TBRKV, calls to GYCOT and GYCPF are forced. If the element is negative, the run is halted. See Note 7.
TCMXV	W	Maxima over y of the changes in the dependent variables during the last step (output label: MAX CHANGE).
THETA	L	Momentum thickness.
TIME	L	Streamwise independent variable, x. See Note 7.
TMRF	θ	Reference temperature for μ and k. See Note 3. [1.]

Name	Dimensions	Definition
TT	θ^2	$\overline{T'T'}$
TU_	$V\theta$	$\overline{T'u'}$
TV_	$V\theta$	$\overline{T'v'}$
TW_	$V\theta$	$\overline{T'w'}$
TYPL	L	Typical length. See Note 1.
U	V	$\bar{u}$. See Tables 3 and 5.
UE	V	Free-stream value of $\bar{u}$. See Table 2.
US_	V	$\bar{u}$. See Table 5.
UU_	V^2	$\overline{u'u'}$. See Tables 3 and 5.
UV_	V^2	$\overline{u'v'}$. See Tables 3 and 5.
UW_	V^2	$\overline{u'w'}$. See Tables 4 and 5.
UWRJ2	V^2/L^2	$(\overline{u'w'}/R)_y$ at the new value of x .
V	V	$\bar{v}$. See Table 3.
VARNM		Vector of variable names.
VBS		Modeling parameter. See Note 2.
VHH		Modeling parameter. See Note 2.
VRH		Modeling parameter. See Note 2.
VRR		Modeling parameter. See Note 2.
VRU		Modeling parameter. See Note 2.
VS_	V	$\bar{v}$
VSW	V	The wall value of $\bar{v}$. See Table 2.
VUH		Modeling parameter. See Note 2.

Name	Dimensions	Definition
VUU		Modeling parameter. See Note 2.
VV_	V^2	$\overline{v'v'}$. See Tables 3 and 5.
VW_	V^2	$\overline{v'w'}$. See Tables 4 and 5.
VWRJ2	V^2/L^2	$(\overline{v'w'}/R)_y$ at the new value of x .
W	V	$\bar{w}$. See Tables 3 and 5.
WGH1		Modeling parameter computed from constraint. See Note 2.
WGH2		Modeling parameter. See Note 2.
WGH3		Modeling parameter. See Note 2.
WGR1		Modeling parameter computed from constraint. See Note 2.
WGR2		Modeling parameter. See Note 2.
WGR3		Modeling parameter. See Note 2.
WGU		Modeling parameter. See Note 2.
WMD		Modeling parameter. See Note 2.
WMH		Modeling parameter. See Note 2.
WMR		Modeling parameter. See Note 2.
WMT		Modeling parameter. See Note 2.
WS_	V	$\bar{w}$. See Table 5.
WSR_	V/L	$\bar{w}/R$
WW_	V^2	$\overline{w'w'}$. See Tables 3 and 5.
WWD1		Modeling parameter. See Note 2.
WWD2		Modeling parameter. See Note 2.

Name	Dimensions	Definition
WWGH		Modeling parameter. See Note 2.
WWGR		Modeling parameter. See Note 2.
WWGU		Modeling parameter. See Note 2.
WWH1		Modeling parameter. See Note 2.
WWH2		Modeling parameter. See Note 2.
WWR1		Modeling parameter. See Note 2.
WWR2		Modeling parameter. See Note 2.
WWU1		Modeling parameter. See Note 2.
WWU2		Modeling parameter. See Note 2.
X	L	x . See Table 2.
XSTRT	L	Initial value of XWET. See Note 7.
XV_	W	Vectors of the unknowns at the old x . Compare AV. See Table 5.
XW	L	Wetted length.
XWET	L	Wetted length (output label: XW).
Y	L	y . See Tables 3 and 4.
Y AT MAX	L	FZMXV
YMAX	L	The maximum value of y to which the solution may spread.
ZPSM	L	Spreads of the dependent variables based on PCTMV (output label: PCT SPREAD).

Note 1.

The turbulence scale Λ that enters in many of the models is computed as follows. First, a typical length is computed from

$$\text{TYPL} = \text{FAC99}*\text{DLT99} + \text{OFAC}*\text{ZPSM(IOFAC)}$$

where DLT99 is the value of y where $\bar{u}$ is 99 percent of the free-stream value; ZPSM is the vector of spreads of the dependent variables based on PCTMV; and FAC99, OFAC, and IOFAC are inputs. (ZPSM(IOFAC) is included in the output with the label SPRO.)

The value of Λ is made equal to BLAM0 at the wall. This parameter is ordinarily zero but may be made positive to simulate roughness. Off the wall, Λ increases with slope DIN until it reaches a maximum of CLAMB*TYPL. For $y >$ DLT99, Λ increases again with slope DOUT. This parameter is ordinarily zero but should be positive if there is free-stream turbulence.

All the parameters determining Λ are nondimensional except BLAM0.

Note 2.

The modeling parameters are defined by their appearance in the models, as given in Appendix B. They are all nondimensional. Two of them, WGH1 and WGR1, are not input; instead, they are computed from the constraints given at the end of Appendix B.

Note 3.

The viscosity is determined from

$$\bar{\mu} = \mu_{rf}\left(\frac{T}{T_{rf}}\right)^{\omega} \frac{1 + C/T_{rf}}{1 + C/T}$$

where μ_{rf} = EMURF , T_{rf} = TMRF , ω = EXPMU , and C = CSUTH . With $\omega = 1/2$, this is the Sutherland relation. With $C = 0$, it is the power-law relation. With both $\omega = 0$ and $C = 0$, it gives constant $\bar{\mu}$.

The heat conductivity is determined from

$$\bar{k} = k_{rf} \frac{\bar{\mu}}{\mu_{rf}}$$

Since c_p is constant, this is equivalent to assuming that the Prandtl number is constant.

Note 4.

Upwind differencing is suppressed when FUP = 0 , used when FUP = 1 . In this case, the coefficient, $\bar{\rho}\bar{v}$, is evaluated as (1 - FUPV) times the central-point value and FUPV times the upwind-point value. If FUP = 0 , the value of FUPV is immaterial. Values of FUP other than 0 and 1 produce a mixture of standard and upwind differencing.

Note 5.

If the dimension flag, DIMFL, is input as zero, the program operates in flat-plate mode. In this case, RNOSE and CONGL are not used and the following acquire different meanings:

R_ (Dimensionless) is the number 1.

ROTAT (Dimension: V) is the sideslip velocity of the surface.

Thus, the value of $\bar{w}$ at the surface is

ROTAT*R

(where R is evaluated at the surface) for either value of DIMFL.

Note 6.

If GMOMS is input as zero, the program operates in Mach number zero mode. In this case, the following acquire different meanings:

GMOMS (Dimensionless) is the number 0. The units for expressing heat quantities may still be chosen at will. A conversion factor between mechanical and heat units is not needed in this case.

P, PE, and PS_ (Dimensions: DV^2) all represent pressure differences instead of absolute pressures. The datum for zero values for the outputs is whatever is used for the input (PE). If the free-stream value is constant, it would ordinarily be input as zero, and the output pressures would represent departures therefrom. If PE is input as the absolute free-stream pressure, the output pressures are absolute too.

PSTAG (Dimensionless) is one of the numbers ± 1. This is used only to flag runs for which correlations involving ρ' and h' are to be zeroed out.

RGAS (Dimension: D) is the density corresponding to HSTAG.

Note 7.

On restart (INFLG = 2) two inputs, DELTI and XSTRT, are ignored if input as zero; the current values of DELT and XWET in the halted run are used instead. On the other hand, the input values will be used if they are not zero. The input value of TIME (x) is always used, so if it is desired to have it run continuously the final value from the halted run must be supplied as input. The time break values, TBRKV, are interpreted in terms of TIME, so they may need to be altered on restart. If the run was halted by the use of a negative time break, that should probably be changed for restart. If the run was halted by NSTMX, that should be changed for restart.

Table 1. Input Cards

Card 1	INFLG, NSKIP	(I1,3X,I4)
Card 2	CMNTS, NRUN	(19A4,I4)
Card 3	DIMFL, RNOSE, CONGL, ROTAT, TIME	(5F8.0)
Card 4	RGAS, CP, HSTAG, PSTAG, GMOMS, XSTRT	(6F8.0)
Card 5	TMRF, EMURF, CAYRF, CSUTH, EXPMU	(5F8.0)
Card 6	CLAMB, DIN, DOUT, BLAMO, FAC99, OFAC, IOFAC	(6F8.0,I4)
Card 7	A, B, BETA	(3F8.0)
Card 8	AHH, AHU, ARU, BHH, BHU, BRU	(6F8.0)
Card 9	VUU, VUH, VHH, VRU, VRH, VRR	(6F8.0)
Card 10	PMU, PMU2, PMH2, PMR2, PTHM, PTUM	(6F8.0)
Card 11	PGH, PGH2, PGR, PGR2, PGU, PGU2	(6F8.0)
Card 12	WGU, WMH, WMR, WMT	(4F8.0)
Card 13	WGH2, WGH3, WGR2, WGR3	(4F8.0)
Card 14	WWU1, WWR1, WWH1, WWGU, WWGR, WWGH	(6F8.0)
Card 15	EV	(6F8.0)
Cards 16.n	X, UE, HE, PE, VSW, HSW	(6F8.0)
Card 17	NIOLP, NIOPP, NFOLP, NFOPP, PCFMX, PCFSP, ITEST	(4I4,2F8.0,I4)
Card 18	TBRKV	(10F8.0)
Card 19	NSTMX, JMAX, YMAX, FUP, FUPV	(2I4,3F8.0)
Card 20	ECMNI, ECMXI, FCURI, FCMX, DZFMN, DZFMX, DZRMX	(7F8.0)
Card 21	DELTI, DTMIN, DTMAX, DTFMX, BUFAC	(5F8.0)

C. 2

Table 1. Input Cards (Continued)

Cards 22/23	PCTMV	(8F8.0/9F8.0)
Cards 24/25	EPSTV	(8F8.0/9F8.0)
Cards 26/27	EPSBV	(8F8.0/9F8.0)
Cards 28/29	EPSNV	(8F8.0/9F8.0)
Card 30	FV	(10F8.0)
Cards 31.j	Y, U, V, W, H, UU, VV, WW, UV, BLAM	(10F8.0)
Card 32	GV	(10F8.0)
Cards 33.j	Y, UW, VW, HU, HV, HW, RU, RV, RW, HH	(10F8.0)

Table 2. Edge Values and Wall Values

Name	Multiplied By	Gives	Dimensions
X	EV(1)	x	(L)
UE	EV(2)	u_e	(V)
HE	EV(3)	h_e	(H)
PE	EV(4)	p_e	(DV^2)
VSW	EV(5)	v_w	(V)
HSW	EV(6)	h_w	(H)
		$\left(\frac{\partial \bar{h}}{\partial y}\right)_w$	(H/L)

If h_e is input as zero, it is calculated from $h_e = h_s - \frac{1}{2}u_e^2$ where h_s is the stagnation enthalpy.

If GMOMS > 0 and p_e is input as zero, it is calculated from $p_e = p_s\left(h_e/h_s\right)^{\gamma/(\gamma-1)}$ where p_s is the stagnation pressure.

Table 3. Initial Conditions, First Set

Name	Multiplied By	Gives	Dimensions
Y	FV(1)	y	(L)
U	FV(2)	$\bar{u}$	(V)
V	FV(3)	$\bar{v}$	(V)
W	FV(4)	$\bar{w}$	(V)
H	FV(5)	$\bar{h}$	(H)
UU	FV(6)	$\overline{u'u'}$	(V^2)
VV	FV(7)	$\overline{v'v'}$	(V^2)
WW	FV(8)	$\overline{w'w'}$	(V^2)
UV	FV(9)	$\overline{u'v'}$	(V^2)
BLAM	FV(10)	Λ	(L)

Table 4. Initial Conditions, Second Set

Name	Multiplied By	Gives	Dimensions
Y	GV(1)	y	(L)
UW	GV(2)	$\overline{u'w'}$	(V^2)
VW	GV(3)	$\overline{v'w'}$	(V^2)
HU	GV(4)	$\overline{h'u'}$	(HV)
HV	GV(5)	$\overline{h'v'}$	(HV)
HW	GV(6)	$\overline{h'w'}$	(HV)
RU	GV(7)	$\overline{\rho'u'}$	(DV)
RV	GV(8)	$\overline{\rho'v'}$	(DV)
RW	GV(9)	$\overline{\rho'w'}$	(DV)
HH	GV(10)	$\overline{h'h'}$	(H^2)

Table 5. Dependent Variables

Name	Dimensions	Output Label	Meaning
WS = XV(1)	(V)	W	$\bar{w}$
VW = XV(2)	(V^2)	VW	$\overline{v'w'}$
US = XV(3)	(V)	U	$\bar{u}$
UV = XV(4)	(V^2)	UV	$\overline{u'v'}$
UW = XV(5)	(V^2)	UW	$\overline{u'w'}$
UU = XV(6)	(V^2)	UU	$\overline{u'u'}$
VV = XV(7)	(V^2)	VV	$\overline{v'v'}$
WW = XV(8)	(V^2)	WW	$\overline{w'w'}$
HS = XV(9)	(H)	H	$\bar{h}$
HV = XV(10)	(HV)	HV	$\overline{h'v'}$
HH = XV(11)	(H^2)	HH	$\overline{h'h'}$
RU = XV(12)	(DV)	RU	$\overline{\rho'u'}$
RV = XV(13)	(DV)	RV	$\overline{\rho'v'}$
RW = XV(14)	(DV)	RW	$\overline{\rho'w'}$
HU = XV(15)	(HV)	HU	$\overline{h'u'}$
HW = XV(16)	(HV)	HW	$\overline{h'w'}$
BLAM = XV(17)	(L)	BLAM	Λ