Thrust Chamber Modeling Using Navier-Stokes Equations

Code Documentation and Listings - Volume II

by

P. L. Daley and S. F. Owens

December 1988 CHAM 2008/3-A

NAS8-36552

Prepared for

National Aeronautics and Space Administration
Marshall Space Flight Center
Alabama 35812

NASA COR: Klaus W. Gross
Contract Number: NAS8-36552

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose of the Report

This report provides a copy of the PHOENICS [1] input files and FORTRAN code developed for the modeling of thrust chambers. These copies are contained in the Appendices of this report and are described briefly below. The results of the thrust chamber modeling development efforts have been reported in a separate report [2].

1.2 The Listing Provided

The listings are contained in Appendices A through E. Appendix A describes the input statements relevant to thrust chamber modeling as well as the FORTRAN code developed for the Satellite program. Appendix B describes the FORTRAN code developed for the Ground program. Appendices C through E contain copies of the Q1 (input) file, the Satellite program and the Ground program respectively.

---

* Numbers in square brackets refer to references.

2. REFERENCES

1. Rosten, H. I. and Spalding, D. B., "The PHOENICS Reference Manual", CHAM TR/200, October 1987.

2. Daley, P. L. and Owens, S. F., "Thrust Chamber Modeling Using Navier-Stokes Equations - Volume 1", CHAM 2008/3, December 1988.

APPENDIX A

Description of Q1 and SATELLITE

A.1 INTRODUCTION

This appendix provides an explanation of the Q1 and Satellite files. A detailed description of the user-defined variables used in PHOENICS is provided. However, for standard PHOENICS variables a more detailed description can be found in the The PHOENICS Reference Manual [1].

A.2 Q1 File Settings

In this section a group-by-group explanation of the Q1 file is presented. This should be read in conjunction with Appendix C which contains a listing of the Q1 file.

A.2.1 Group 1 - Run Title and Other Preliminaries

This is the largest group in the Q1 file. It is structured in such a way that most of the changes a user will make occur in this area. The input in this group is divided into the following four sections

1. Declaration of Variables,
2. Switches,
3. Grid Specifications, and
4. Properties.

A.2.1.1 Declaration of Variables

In the first section, the integer and real variables used in the Q1 are declared. The variables declared in this section will be discussed further in the text.

A.2.1.2 Switches

In this section, there are 14 switches available for the user. These switches are actually PHOENICS integer and logical arrays. It is through the setting of these array elements that the proper coding sequences are activated in the Satellite and Ground files. These switches control a wide variety of parameters, including grid information, boundary conditions, and print-out.

1. Switch 1 (IG(1)) allows the user to select the type of grid used by PHOENICS. With the selection of option 1, the Satellite will produce an algebraic SSME grid file during execution. Option 2 allows the use of an externally created grid file which is read in during the execution of Satellite through the Readco command.

2. IG(2) is only active when the Satellite produces an algebraic SSME grid. The various options of this switch control how the boundary of the last nozzle section will be characterized. This last section may be specified as a line, a parabola, an arc or through the use of a spline fit. (See Section A.3.2.4).

3. IG(3) is used to specify the type of flow. Choices include; turbulent flow, constant viscosity and inviscid flow.

4. The type of wall function is controlled through IG(4). The options of this section include; built in wall functions, modified wall functions and no wall functions. The modified wall functions account for the strong axial pressure gradients. If the inviscid flow option has been selected (Item 3 above) then the selection of wall function option is ignored; the wall will be treated as a zero-flux boundary.

5. This switch (IG(5)) controls the use of the equilibrium package. If the equilibrium package is not used the chemical composition will consist of reaction product and the nonlimiting reactant (reactant that is remaining after complete reaction).

6. This switch (IG(6)) specifies the type of inlet boundary condition. Choices include a fixed flux or a fixed pressure at the inlet. It is recommended that the fixed flux option be used.

7. IG(7) is used to specify the type of exit boundary condition. Options include an extrapolated flux or a fixed pressure. It is recommended that the extrapolated boundary condition be used as it is more physically realistic and allows for a better prediction of the pressure near the exit.

8. The enthalpy boundary condition at the thrust chamber wall is controlled through IG(8). The first option is for an adiabatic wall. The second option uses a prespecified wall temperature profile to calculate the heat flux at the wall. The final option activates the simplified cooling jacket simulation.

9. This switch (IG(9)) allows the user to activate the two phase options of the code.

10. IG(19) controls the print-out from Satellite. Increasing the integer flag will result in more print-out. This option is used mainly for debugging purposes.

11. IG(20) controls the debug print-out from Ground in a similar manner as switch 10 (IG(19)).

12. LG(1)is only used when the cooling jacket option is exercised. The options of this switch direct the program to the appropriate initial wall temperature profile. The initial wall temperatures for a restart run will come from the profile calculated on the last sweep of the previous run.

13. This switch (LG(2)) allows the user to modify the gas side heat transfer coefficient. When this logical is set to true, the heat transfer coefficient of Equation 24 is adjusted (multiplied) by a user-defined array located in subroutine Darth.

14. This switch (LG(3)) allows the user to modify the liquid side heat transfer coefficient. When this logical is set to true, the coefficient of Equation 25 is adjusted by a user-defined array located in subroutine Darth.

A.2.1.3 Grid Specifications

It is required to input the number of cells (NX, NY, and NZ) in each of the three grid directions. Also required is the number of the cell (NZT) whose foward face in the z-direction is located at the throat.

For single phase flow, the location of the first and last inlet cells in the radial direction (IYBOT & IYTOP) must also be specified. For two phase flow the inlet boundary is set up as a series of jets. Information needed for two phase flow include the number of jets (NJETS) and the location of the jets. The location of the inlet jets is stored in the integer array elements 31 to 50. This limits the number of jets to 20.

If an externally created grid file is used, the user must specify a four character grid file name through the use of the PHOENICS character array. If the algebraic SSME nozzle option is selected, three parameters controlling the grid spacing along the boundary must be specified. The first parameter (PU) controls the grid spacing along the north wall up to the throat. A value of one will provide a uniform grid distribution, while a value greater than one will cluster the points toward the throat. Similarly, the second variable (PD) controls the grid spacing along the wall down from the throat. The last parameter (PR) controls the grid spacing in the radial direction. A value larger than one will cause the grid lines in the radial direction to be clustered toward the wall.

The last variable in the grid specification section is a geometric factor (GPI) which must be specified. The grid in the circumferential direction is wedge shaped and thus represents only a small faction of the total flow area. The geometric factor is the ratio of 2π and θ where θ is the central angle in radians.

A.2.1.4 Properties

The following properties and conditions must be specified

1. the propellant mixture ratio (FMIX),
2. the enthalpy for hydrogen (ENTHH2),
3. the enthalpy for oxygen (ENTHO2),
4. the ratio of specific heats (GA),
5. the inlet pressure (PRESIN), and
6. the inlet flow rate (FRATE).

If a fixed pressure is used for the exit boundary condition, two additional inputs are required which include the outlet pressure at the wall (POTOP) and the pressure at the centerline (POBOT). Should these pressures be different, a linear exit pressure profile will be calculated from these two pressures.

A.2.2 Group 6 - Body-fitted Coordinates or Grid Distortion

In the Q1 file, the BFC grid distortion is activated by BFC=T. NONORT=T activates the calculation of non-orthogonal terms is the finite domain equations, while SAVGEO=F will prevent the geometry restart file from being written, thus saving disc space. The execution of the Satellite file is initiated through the SATRUN command.

A.2.3 Group 15 - Termination of Sweeps

In this section the first and last solution sweeps are set. In a restart run the first sweep can be used to continue counting the sweep number.

A.2.4 Group 17 - Under-Relaxation Devices

Linear relaxation is used for P1 and also R1, R2, and RS if the two phase option is exercised. False time step relaxation is used for velocities, turbulence variables (if solved), enthalpy, and concentration.

In certain cases relaxation on enthalpy and concentration is not needed. An estimate (FTS) of the cell residence time is used for the false time step. This estimate is calculated in Satellite. There is a relaxation adjustment factor for both the linear (DLIN) and false time step (DFAL) relaxations. The correct adjustment of these factors requires some user experience. However, in general if the solution is diverging (increase of the dependent variable residuals) these factors need to be lowered. If the solution does not change from sweep to sweep, this could be an indication of too much relaxation.

A.2.5 Group 21 - Print-out of Variables

This section controls the print-out of the initial and final fields.

A.2.6 Group 22 - Spot-Value Print-Out

The location of the spot value printout is controlled by IYMON and IZMON. The residual frequency is regulated by TSTSWP.

A.2.7 Group 23 - Field Print-Out and Plot Control

Various parameters relating to the field print-out and plots are controlled from this section. These are documented in the reference manual [1].

A.2.8 Group 24 - Dumps for Restarts

Should a restart run be required, the variables to be read from a previous solution file and the file name (four characters) must be specified.

A.3 DESCRIPTION OF SATELLITE FORTRAN

The Satellite file contains most of the data-setting statements. This was done for a twofold reason. There are certain coding sequences that cannot be easily implemented in the Q1 file and for ease-of-use only a few selected options and variables were placed in the Q1 file. The listing of this file is provided in Appendix D.

A.3.1 Program Main

The main program is concerned with the allocation of storage for the Satellite. At the top of main are three parameters that can be used to adjust the dimensions of the most commonly changed arrays. The first parameter (NYPAR) is used to set the dimensions of the arrays that contain information oriented in the radial direction. Similarly, the second parameter (NZPAR), can be used to increase or decrease the dimension of the arrays that contain information oriented in the axial direction. The last parameter (NBFPAR) is used to dimension the arrays that contain the BFC information required in Satellite.

The number of PHOENICS variables stored was increased from 25 to 50. Several of the COMMON blocks from LGE1 to RFPL4 needed to be set to a length of 50.

This change was also made in subroutines Sat, Satlit, and Gscale.

A.3.2 Subroutine Satlit

The structure of this subroutine is similar to the Q1 file in that it is divided into 24 groups. Preceding the calls to the 24 groups, several variables have been dimensioned and various data statements have been included. These are documented within the code. It can be noted that most of the variables used in Satellite are documented within the Satellite and will not be covered in great detail in this section. Instead a general discussion of each group will follow.

A.3.2.1 Group 1 - Run Title and Other Preliminaries

This section contains most of the information necessary to create an algebraic SSME grid. The data from this group is used to calculate various radii and nozzle lengths. If an externally created grid is used, most of the information in this section will be ignored. Also included in this section are properties and conditions for the cooling jacket simulation. These data will be ignored if the jacket simulation is not specified.

A.3.2.2 Group 4 - Y-Direction Grid Specifications

This section is only accessed if the algebraic SSME grid option is selected. The y-fractions are calculated in this group. These fractions range between 0.0 and 1.0 and are the normalized y-locations of the grid nodes. The actual values for y will be calculated in Group 6.

A.3.2.3 Group 5 - Z-Direction Grid Specifications

This group is similar to the preceding section. In this group the z-fractions are calculated if required. They are calculated in two sections; one from the injectors to the throat and the other from the throat to the exit plane.

A.3.2.4 Group 6 - Body-fitted Coordinates or Grid Distortion

If an externally created grid is used, the necessary information is read in

from the grid file at this point. Otherwise, the actual y- and z-coordinates based on the z-fractions and other information will be calculated.
The first step for an algebraic grid is to calculate a z-distance based on the z-fractions and the total length. The nozzle length is broken down into 6 regions

1. length up to first bend,
2. first bend,
3. length after first bend,
4. second bend,
5. third bend, and
6. length down from third bend.

Depending on which region the z-distance is located, the corresponding radius is calculated appropriately. There are four ways to define the last region. It can be defined as a line, a parabola, an arc or by a spline fit. The radii for the last section are calculated based on how the last nozzle section has been classified. Based on the y and z-fractions and the calculated radii along with the total nozzle length, the interior y and z grid node locations are calculated. The x-values for the west and east faces are calculated using the geometric factor set in the Q1 file in such a manner as to create a wedge in the x-direction.

Regardless of the type of grid, a geometric subroutine (Geomtx) is then called to calculate various geometric factors. This will be covered later in greater detail. If the two phase option is selected, the available flow area at the inlet is calculated.

A.3.2.5 Group 7 - Variables Stored, Solved and Named

The allocation of variable storage, which variables are solved and any renaming of variables, takes place in this group. While some of the variables are always active, the remainder is controlled through the switches located in the Q1 file. The following table, broken down into variables solved and variables stored, lists the variables of this group.

Index	Name	Dependent Variables	Switch
1	P1	First phase pressure	–
5	V1	First phase radial velocity	–
6	V2	Second phase radial velocity	9
7	W1	First phase axial velocity	–
8	W2	Second phase axial velocity	9
9	R1	First phase volume fraction	9
10	R2	Second phase volume fraction	9
11	RS	Second phase shadow volume fraction	9
12	KE	Turbulent kinetic energy	3
13	EP	Rate of dissipation of turbulent kinetic energy	3
14	H1	Enthalpy	–
16	C1	Concentration of total hydrogen	–

Index	Name	Auxiliary Variables	Switch
17	HH	Concentration of molecular hydrogen	–
18	02	Concentration of molecular oxygen	–
19	H20	Concentration of water	–
20	O	Concentration of atomic O	–
21	H	Concentration of atomic H	–
22	OH	Concentration of radical OH	–
23	HO2	Concentration of HO_2	–
24	ENUT	Turbulent viscosity	–
25	RHO1	First phase density	–
26	TEMP	First phase temperature	–
27	ETPY	Entropy	–
28	GAMA	Ratio of specific heats	–
29	MACH	Mach number	–
30	PSIA	Pressure in psia	–
31	AMDT	Interphase mass transfer	–
32	LTEM	Wall temp from cooling jacket	8
45	YCOR	Y-direction cell centers	–
46	ZCOR	Z-direction cell centers	–
47	V2CR	Second phase y-Cartesian velocity resolute	9
48	W2CR	Second phase z-Cartesian velocity resolute	9

49	VCRT	First phase y-Cartesian velocity resolute	-
50	WCRT	First phase z-Cartesian velocity resolute	-

A.3.2.6 <u>Group 8 - Terms (In Differential Equations) and Devices</u>

In order to save computational time, three logical variables (NEWENL, NEWENT, and NEWRH1) have been set to false. This prevents the recalculation of the laminar and turbulent viscosity and density at the current slab. In this section, the upwind differencing scheme has been activated with the DIFCUT command.

A.3.2.7 <u>Group 9 - Properties of the Medium (or Media)</u>

In this section the physical properties are used to calculate the boundary and initial conditions used in other sections in Satellite and Ground. For one phase flow, the inlet conditions are based on the assumption that the inlet materials have fully reacted. Using the propellant mixture ratio and the above assumption, the inlet mass fractions are calculated. With this information along with the enthalpy of the mixture, an inlet temperature is computed. Using the inlet pressure, temperature, and area, the density and incoming velocity are calculated.

For two phase calculations, the flow is split into two parts; partially reacted hydrogen and unreacted oxygen. It is assumed that a fraction of the total oxygen has been allowed to react with the inlet hydrogen. Based on this assumption, calculations similar to the ones listed above are performed. Also in this section, the mean drop diameter for material stripped from the jets is calculated. The diameter can be calculated from Equation (27) or from a fraction of the oxidizer injection element diameter.

A.3.2.8 <u>Group 10 - Inter-Phase-Transfer Processes and Properties</u>

In this section, it is indicated that during two-phase flow calculations, the interphase friction and mass transfer rate will be calculated in Ground.

A.3.2.9 Group 11 - Initialization of Variables or Porosity Fields

The following variables are initialized in Ground; pressure, w-velocity, enthalpy, density, and temperature. Their initialization is based on a one-dimensional isentropic flow relationship for nozzles. For two-phase flow the initial values of the volume fractions are provided in two different zones. In the first area (roughly related to half the distance to the throat), the volume fraction of the second phase is initialized to be 2 percent. Over the remaining area, the volume fraction of the second phase is set equal to zero.

The total hydrogen fraction (C1) is set equal to the inlet mass fraction of total hydrogen. Total hydrogen is defined as hydrogen of any form (H_2, H_2O, etc.). The initial values of hydrogen and water are calculated based on total combustion of oxygen. Initial values for KE and EP are estimated on the basis of an inlet turbulence intensity using 10 percent of the inlet velocity and a length scale of 1 percent of the inlet chamber diameter. For a flow with a high Reynolds number, interior values of KE and EP are not expected to be sensitive to the inlet values.

A.3.2.10 Group 13 - Boundary Conditions and Special Sources

Boundary conditions are set up in this section through the use of PATCH and COVAL statements. The PATCH statements are used to define regions of space and the COVAL statements are used for setting sources of the dependent variables over these areas. The patches used in this section fall in six different categories

1. Inlets,
2. Outlets,
3. Walls,
4. Two-phase sources,
5. KE & EP sources, and
6. Corrections.

The first five of these are controlled by various switches in the Q1 file and thus have already been discussed. One patch was added to correct a problem with compressible flow. This correction was added to prevent the downstream density from influencing the upstream mass flux.

A.3.2.11 Group 16 - Termination of Iterations

The maximum number of iterations performed by the linear-equation solver for pressure was set to 25. The iteration number may need further increase for problems where the total number of grid cells exceeds 5000. The iteration-termination criterion for P1 was set to 0.01.

A.3.2.12 Group 18 - Limits on Variables or Increments to Them

An upper limit was placed upon three variables (P1, KE, & EP) in this group. The maximum value for KE and EP was increased over the default value because in early studies it was found that the values of these variables could exceed the original upper limit. The maximum value of P1 is limited to 150 percent of the inlet value as a precaution against large pressure variations during early sweeps.

A.3.2.13 Group 19 - Data Communicated by Satellite to GROUND

To insure the inclusion of all spatial derivatives of velocity in the generation function, GENK has been set to true.

A.3.3 Subroutine Enthal

This subroutine is used to calculate mixture specific heat capacity and mixture enthalpy. Inputs into this subroutine include; the temperature, molar concentrations, the number of species, and an information flag. Using a polynomial fit, the thermodynamic properties are calculated as a function of temperature as follows

$$C_p/R = Z_1 + Z_2T + Z_3T^2 + Z_4T^3 + Z_5T^4$$

$$CPSUM = \sum_{i=1}^{NS} \left[\frac{C_{pi}}{R}\right] \sigma_i$$

$$h/RT = Z_1 + \frac{Z_2T}{2} + \frac{Z_3T^2}{3} + \frac{Z_4T^3}{4} + \frac{Z_5T^4}{5} + \frac{Z_6}{T}$$

$$HSUM = \sum_{i=1}^{NS} [\frac{h_i}{RT}] \sigma_i$$

The coefficients for H_2, O_2, and H_2O are supplied through DATA statements and are valid over a temperature range of 300 to 5000 K.

A.3.4 Subroutine Temper

This subroutine is used to calculate a temperature for a given static enthalpy and molar concentrations of H_2, O_2, and H_2O. Inputs into this subroutine include; the static enthalpy, a guess for final temperature, the gas constant, molar concentrations, the number of species, and an information flag used to control debug print-out.

This is an iterative procedure in which several calls to subroutine Enthal are made. The initial temperature is passed on to subroutine Enthal. The returned enthalpy is checked against the given enthalpy. Should the difference fall outside a given tolerance, the temperature is adjusted and again passed on to subroutine Enthal.

A.3.5 Subroutine Xslp

This subroutine computes slopes for a cubic spline fit to a planar set of data. Input to this subroutine include the number of data points and their coordinates. The slopes are then used in Group 6 to calculate a y-value (located on the wall) for a given z-location.

A.3.8 Subroutine Geomtx

Geometric quantities along with a false time step are calculated in this subroutine. Inputs to this subroutine include: the x-, y-, and z-grid node locations, the total number of grid nodes, the z-direction cell located at the throat, the first and last radial inlet cells, the number of cells in the y- and z-directions, an average velocity, and the geometric factor needed to obtain the total flow area. Using this information the following geometric factors are computed; the open inlet radius, the radius at the throat, the cross sectional inlet area, and the low face area of the inlet cells.

APPENDIX B

Description of GROUND FORTRAN

ORIGINAL PAGE IS
OF POOR QUALITY

B.1 INTRODUCTION

This appendix provides an explanation of the GROUND file. A description of each of the relevant groups along with the subroutines is provided.

B.2 Program Main

The main program deals with the allocation of storage for the Earth Program. At the beginning of main, there are seven parameters that can be used to adjust the dimensions of the most commonly changed arrays. The first parameter (NFPAR) is used to set the dimension for the computer memory required for storage of the main and auxiliary variables. The next six parameters are used to set the dimensions of arrays that contain information oriented in one of the three grid directions.

The number of PHOENICS variables stored was increased from 25 to 50. Several of the COMMON blocks from LGE1 to RFPL4 needed to be set to a length of 50. This change was also made in subroutines Grosta, Ground, Darth, Twcool and Cjprnt. Additionally, COMMON/F01/, has been set to I9(200).

B.3 Subroutine Grosta

This is the junction box which enables different Ground subroutines to be called. Two subroutines (Grexl and Ground) are called in this section.

B.4 Subroutine Ground

The structure of this subroutine is similar to the Q1 file in that it is divided into 24 groups. Preceding the calls to the 24 groups, several arrays have been dimensioned and various data statements have been included. Variables transferred in from Q1 and Satellite are equivalenced in this location. A complete documentation of the variables is not provided as was the case with the Satellite (see Section A.3.2). Instead comments are provided for individual blocks of coding.

B.4.1 Group 1 - Run Title and Other Preliminaries

In this section several preliminary calculations are performed. Comments on these calculations are provided in the listing included in Appendix E.

A call to subroutine Chemic is made even if the equilibrium package is not used. This allows for the calculation of the molecular weights of the individual species.

B.4.2 Group 9 - Properties of the Medium (or Media)

The density of the mixture is calculated in Section 1 of this group. The mixture density is calculated from the ideal gas law

$$\rho = \frac{P\ MW}{RT}$$

where

P - is the pressure,

MW - is the molecular weight of the mixture,

R - is the universal gas constant, and

T - is the mixture temperature.

As a first step, the molecular weight of the mixture and the local temperature are calculated. There are two methods by which these variables may be calculated. The first method assumes total combustion and three chemical species (H_2, O_2, & H_2O). The local molar concentrations, enthalpy, and a guess for temperature are passed into subroutine Temper. From this subroutine, the local temperature is returned. The second method assumes that 7 chemical species (H_2, O_2, H_2O, OH, O, H, & HO_2) are in equilibrium. With this method the local pressure, enthalpy, and molar concentrations are passed into subroutine Chemic. This subroutine returns the equilibrium temperature, molar concentrations, and mixture molecular weight.

Following the density calculations, the various items calculated in this section are stored for further use. On the last sweep, the entropy and ratio of specific heats are calculated from

$$S/R = Z_1 \log T + Z_2 T + \frac{Z_3 T^3}{2} + \frac{Z_4 T^3}{3} + \frac{Z_5 T^4}{4} + Z_7$$

$$GAMMA = \frac{CPSUM}{(CPSUM - R/MW)}$$

If a non-adiabatic wall option is selected, the wall enthalpy used for the boundary condition will be calculated from the gas side wall temperature.

In Section 6 of this group, the molecular kinematic viscosity is calculated from the dynamic viscosity and the density according to

$$\nu = \frac{\mu}{\rho}$$

where

μ - has the constant value of $4.3e^{-5}$ kg/m-s.

B.4.3 Group 10 - Inter-Phase-Transfer Processes and Properties

The inter-phase friction coefficient is calculated in the first section of this group. This coefficient is taken as the product of the drag on a spherical object and the density of the object as given by Equation (28). In Section 2, the inter-phase mass transfer rate is calculated from Equation (31).

B.4.4 Group 11 - Initialization of Variable or Porosity Fields

The initial fields for pressure, enthalpy, w-velocity, temperature and density are calculated in this group. These calculations are based on 1-D isentropic relationships. A local Mach number is calculated in subroutine Msolv along with other isentropic terms which are used to calculate the static pressure and temperature and local w-velocity.

B.4.5 Group 13 - Boundary Conditions and Special Sources

In the first section, the coefficient for V1 and W1 is modified to insure the downstream density does not influence the upstream mass flux. Following this is the coding for the modified wall functions. In this section the coefficient for W1 is calculated along with the coefficient and values for KE and EP. These values and coefficients are calculated in subroutine Waldp.

The coding in Section 3 is activated for two-phase flow. The gas side heat transfer coefficient is calculated here for later use. If necessary the coefficient can be modified at this location.

In Section 8, the coefficient for the concentration is set. It is set to the value of the mass transfer rate.

The value for P1 at the exit is calculated in Section 12. For a fixed flux boundary condition the exit flux is extrapolated from the previous cell and is calculated as the product of the in-cell density and the upstream velocity. For two-phase flow the second phase flux is calculated as the product of the in-cell second phase density and the upstream second phase velocity. In this case the first and second phase flux is multiplied by the appropriate volume fraction.

For a fixed exit pressure the value is taken to be the actual exit pressure. With this type of boundary condition the value of the exit pressure at the centerline and at the wall must be specified. Should these values be different, a linear outlet pressure profile will be calculated. For two-phase flow, the exit pressure for both phases will be the same.

In Section 19, the enthalpy brought in at every cell by the second phase into the first phase is calculated.

In Section 20, the amount of mass stripped from a jet is calculated. The value is calculated from Equation (26). The diameter of the jet is recalculated as mass is stripped away. The amount of mass stripped away is checked against the total incoming mass to insure that the total mass stripped away does not exceed that which enters the calculation domain.

B.4.6 Group 19 - Special Calls to GROUND from EARTH

At the beginning of a run a wall temperature profile is calculated based on grid distance and a tabular set of wall temperature data provided in subroutine Twall. For a restart run with the simplified cooling jacket simulation, the wall temperature profile will be read in from the restart file.

In Section 3 there is a check on the cell volumes to insure the values are not below 1.E-10. On the last sweep the pressure is converted into psia and is stored for plotting purposes. Also, the x- and y-grid cell centers are stored. This gives the user all of the necessary information in the results file needed to plot results with another plotting package. The translation of this data to another format is left entirely to the user.

There are many auxiliary calculations performed at the end of each IZ slab. When the cooling jacket is simulated, a new wall temperature profile is calculated. If the two-phase option is used, the mass stripped from the jet is summed. On the last sweep the Mach number is calculated by

$$Ma = \frac{(V)^{.5}}{S}$$

where

V - is the velocity and

S - is the local speed of sound

The speed of sound is calculated from

$$S = \left(\frac{\gamma P}{\rho}\right)^{.5}$$

where

γ - is the ratio of specific heats.

The thrust (F) is calculated from

$$F = \dot{m}V + (P - P_1)A$$

where

$\dot{m}$ - is the mass flow rate,
P - is the exit pressure,
P_1 - is the atmospheric pressure, and

A - is the exit area.

The specific impulse (I_{sp}) is calculated by

$$I_{sp} = \frac{F}{\dot{m}g}$$

where

g - is the acceleration due to gravity.

Just before the last sweep the inlet and outlet fluxes are calculated. These values are used on the last sweep in the specific impulse calculations. One criterion for a fully converged run is that these two quantities must be equal.

B.5 Subroutines Enthal and Temper

These subroutines are the same as those described in Appendix A.

B.6 Subroutine Msolv

This subroutine is used in the initialization of the flow field. Inputs to this subroutine include: ratio of specific heats, a flag to indicate subsonic or supersonic flow, the local throat area ratio, the gas constant, and a guess for the Mach number. The results of local calculations are the correct Mach number, and the following quantities

$$VRT = Ma \frac{\gamma R}{T_t/T}$$

$$QRT = Ma \frac{\gamma}{R} \left(\frac{T}{T_t}\right)^{\frac{\gamma + 1}{2(\gamma-1)}}$$

$$PTP = 1 + \frac{\gamma - 1}{2} M_a^2$$

$$TTT = T_t/T$$

where

T_t - is the inlet temperature and

T - is the local temperature.

These terms are used in Group 11 to calculate the pressure, temperature, and velocity.

B.7 Subroutine Twalbc

The wall temperature profile is calculated in this subroutine. Inputs to this subroutine include: the current slab number, an array containing the z-cell midpoints, the z-distance at the throat, and the radius at the throat. With this information and using the local z-distance temperature profile, a linear interpolation is employed to calculate a wall temperature for each grid cell.

B.8 Subroutine Waldp

This subroutine calculates the wall functions for flows with significant axial pressure gradients. Inputs to this subroutine include: the current slab number, the current sweep, the last sweep, a variable flag, the wall to cell node distance, the absolute viscosity, the resultant velocity, the resultant density, and an information flag. The value and coefficients for the near wall source terms for W1, KE, and EP calculated.

B.9 Subroutine Chemic

This subroutine is the junction box for the equilibrium package. It is called in Group 1 for initialization purposes. During this first call the equilibrium data is read in from a data file and the molecular weight for each species is calculated. It is called for each cell in Group 9 and returns a temperature used in the density calculations. Inputs into this subroutine include: the call type, an information flag, an equilibrium flag, the number of species, the number of elements, a guess for the temperature, the enthalpy, an enthalpy reference, the gas constant, the incoming molar concentrations, a guess for the final molar concentrations, the atomic symbols, and the atomic weights. This subroutine will return the molecular weights, the density, the average molecular weight, the nondimensional enthalpy, and the nondimensional entropy.

B.10 Subroutine Chemin

This subroutine is called during the initialization process to read in input data. Because there are so many parameters passed in and out of this subroutine, they will not be individually elaborated.

B.11 Subroutine Chemso

This routine calls subroutine Compch to compute the corrections to the chemical species and temperature. This subroutine determines the under-relaxation prior to the application of the corrections and also checks for convergence.

B.12 Subroutine Compch

In this subroutine the Newton-Raphson derivative matrix is constructed. This matrix is then solved by pivotal Gaussian reduction.

B.13 Subroutine Hcps

This subroutine calculates the nondimensional values of enthalpy, specific heat, and entropy. This subroutine is also called in Group 9 to calculate entropy and the ratio of specific heats. Inputs include: a call type, an information flag, the temperature, the log temperature, the number of species, and the molar concentrations. Outputs include: the individual enthalpies and entropies along with the mixture enthalpy, specific heat, and entropy.

B.14 Subroutine Darth

This subroutine initializes the cooling jacket geometric data and interpolates for the computational grid. Inputs into Darth include: the z-distance of cell centers, the radii of the grid cells, the z-distance at the throat, the number of channels, the number of tubes, the number of data points in the combustor, the number of data points in the nozzle, the coolant flow rate in the combustor, the coolant flow rate in the nozzle, the z-cell number at the junction of the combustor and nozzle, an information flag and a geometric factor. The outputs from this routine are; the available coolant flow area, the wall thickness, the hydrodynamic diameter for the coolant duct, the parametric adjustment factors for the heat transfer coefficients, and the distance from the throat.

B.15 Subroutine Twcool

The heat flux across the wall, the heat transfer coefficients and various temperatures are calculated in this subroutine. Inputs into this subroutine include; the gas side temperature, the location of the cooling jacket split, the temperature of hydrogen at the two inlets, the energy rate at both inlets, the thermal conductivity of copper and steel, the dynamic viscosity of hydrogen, the Prandtl number of hydrogen, the mass flow rates for both inlets, a logical variable for coefficient adjustment, an information flag, and a geometric factor. Outputs include; the wall temperature on the gas side, the wall temperature on the coolant side, and the coolant temperature. The mathematical formulation used in this subroutine is located in Section 4.5.2.

B.16 Subroutine Cjprnt

This is the routine that controls the final print-out of the cooling jacket information. Various items including heat fluxes, temperatures, and heat transfer coefficients will be printed out on the final sweep.

APPENDIX C

Q1 File

```
TALK=F;RUN( 1, 1);VDU= 0
    GROUP 1. Run title and other preliminaries
TEXT(NOZZLE FLOW)

  ********  NOTE:  COMMENTS START AFTER THE SECOND COLUMN     ********

  ********************************************************************
  ******** DECLARE REAL AND INTEGER VARIABLES IN THIS SECTION ********
  ***                                                              ***
REAL(PU,PD,PR,FMIX,ENTHH2,ENTHO2,GA,PRESIN,FRATE,POTOP,POBOT)
REAL(GPI,FTS,DFAL,DLIN)
INTEGER(NZT,IYTOP,IYBOT,NJETS)
INTEGER(NCHA,NTUB,NCOM,NNOZ)
  ***                                                              ***
  ********************************************************************

  **************************** SWITCHS *******************************
  ***     THE FOLLOWING OPTIONS ARE AVAILABLE BY SETTING THE       ***
  ***     THE INDICATOR FLAGS IN THE REQUIRED MANNER               ***
  ***                                                              ***
  ***     IG(1)  = 1  FOR ALGEBRAIC GRID                           ***
  ***            = 2  FOR GRID GENERATED GRID                      ***
IG(1)=2
  ***                                                              ***
  ***     IG(2)  = 1  FOR CONE SHAPE (after last radius)           ***
  ***            = 2  FOR PARABOLIC SHAPE                          ***
  ***            = 3  FOR ARC SHAPE                                ***
  ***            = 4  FOR A SPLINE FIT                             ***
IG(2)=4
  ***                                                              ***
  ***     IG(3)  = 1  FOR K-E MODEL                                ***
  ***            = 2  FOR LAMINAR FLOW                             ***
  ***            = 3  FOR INVISCID FLOW                            ***
IG(3)=3
  ***                                                              ***
  ***     IG(4)  = 1  FOR PH84 WALL FUNCTIONS                      ***
  ***            = 2  FOR MODIFIED WALL FUNCTIONS                  ***
  ***            = 3  FOR NO WALL FUNCTIONS                        ***
IG(4)=3
  ***                                                              ***
  ***     IG(5)  = 1  FOR REACTIVE CASE (EQUILBRIUM PACKAGE)       ***
  ***            = 2  FOR NON-REACTIVE CASE(W/ COMBUSTION PRODUCTS) ***
IG(5)=1
  ***                                                              ***
  ***     IG(6)  = 1  FOR FIXED FLUX INLET BOUNDARY CONDITION      ***
  ***            = 2  FOR FIXED PRESSURE INLET BOUNDARY CONDITION  ***
IG(6)=1
  ***                                                              ***
  ***     IG(7)  = 1  FOR EXTRAPLATED OUTLET BOUNDARY CONDITION    ***
  ***            = 2  FOR FIXED PRESSURE OUTLET BOUNDARY CONDITION ***
```

```
IG(7)=1
  ***                                                                  ***
  ***     IG(8)   = 1  FOR ADIBATIC WALLS                              ***
  ***             = 2  FOR SSME WALL TEMPERATURE PROFILE               ***
  ***             = 3  FOR SSME COOLING JACKET                         ***
IG(8)=1
  ***                                                                  ***
  ***     IG(9)   = 1  FOR SINGLE PHASE                                ***
  ***             = 2  FOR TWO PHASE FLOW                              ***
IG(9)=1
  ***                                                                  ***
  ***     IG(19) = 0-4  THIS IS A INFO FLAG - HIGHER VALUES RESULT     ***
  ***                   IN MORE PRINTOUT OBTAINED FROM SATELLITE       ***
IG(19)=1
  ***                                                                  ***
  ***     IG(20) = 0-5  THIS IS A INFO FLAG - HIGHER VALUES RESULT     ***
  ***                   IN MORE PRINTOUT OBTAINED FROM GROUND          ***
IG(20)=1
  ***                                                                  ***
  ***     LG(1)   = F  FOR COOLING JACKET SCRATCH RUN                  ***
  ***               T  FOR COOLING JACKET RESTART RUN                  ***
LG(1)=T
  ***                                                                  ***
  ***     LG(2)   = F  FOR NORMAL GAS FILM COEFF                       ***
  ***               T  FOR USER ENHANCEMENT OF GAS FILM COEFF          ***
LG(2)=F
  ***                                                                  ***
  ***     LG(3)   = F  FOR NORMAL LIQUID FILM COEFF                    ***
  ***               T  FOR USER ENHANCEMENT OF LIQUID FILM COEFF       ***
LG(3)=F
  ***                                                                  ***
  ***       Note: no wall function used for inviscid case              ***
  ***             option 2 ignored if option 1 set to 2                ***
  ***             logical options ingnored if opt 8 not eq 3           ***
  ***                                                                  ***
  ************************************************************************

  *********************** GRID  SPECIFICATIONS **************************
  ***                                                                  ***
  ***     ENTER NUMBER OF X Y & Z CELLS                (NX NY & NZ)    ***
NX=1;NY=40;NZ=99
  ***                                                                  ***
  ***     ENTER THROAT LOCATION                        (NZT)           ***
NZT=34; IG(21)=NZT
  ***                                                                  ***
  ***     ENTER FIRST & LAST IY LOCATIONS FOR INLET (IYBOT & IYTOP) ***
IYBOT=1;  IG(22)=IYBOT
IYTOP=NY; IG(23)=IYTOP
  ***                                                                  ***
  ***     ENTER NUMBER OF INJECTORS                    (NJETS)         ***
NJETS=10; IG(30)=NJETS
  ***                                                                  ***
  ***     ENTER LOCATIONS OF JETS                                      ***
IG(31)=4
IG(32)=8
```

```
IG(33)=12
IG(34)=16
IG(35)=20
IG(36)=24
IG(37)=28
IG(38)=31
IG(39)=34
IG(40)=37
  ***                                                                   ***
  ***     ENTER GRID FILE NAME                            (CSG1)        ***
CSG1=SSME
  ***                                                                   ***
  ***     ENTER POWER UPTO THROAT                         (PU)          ***
PU=1.5; RG(1)=PU
  ***                                                                   ***
  ***     ENTER POWER DOWN FROM THROAT                    (PD)          ***
PD=2.0; RG(2)=PD
  ***                                                                   ***
  ***     ENTER POWER TO WALL                             (PR)          ***
PR=1.5; RG(3)=PR
  ***                                                                   ***
  ***     ENTER GEOMETRIC FACTOR                          (GPI)         ***
GPI=60.;RG(12)=GPI
  ***                                                                   ***
  ***           NOTE:  JET INFO IGNORNED IF ONE PHASE                   ***
  ***                  GRID FILE IGNORED FOR ALGBRAIC GRID              ***
  ***                  POWERS IGNORED IF GGP IS USED                    ***
  ***                                                                   ***
  *********************************************************************

  ********************** PROPERTIES & CONDITIONS ********************
  ***                                                                   ***
  ***     ENTER FUEL MIXTURE RATIO                        (FMIX)        ***
FMIX   = 6.054851     ;RG(4)=FMIX
  ***                                                                   ***
  ***     ENTER ENTHALPY FOR HYDROGEN -- CAL/MOLE         (ENTHH2)      ***
ENTHH2 = -1837.66     ;RG(5)=ENTHH2
  ***                                                                   ***
  ***     ENTER ENTHALPY FOR OXYGEN   -- CAL/MOLE         (ENTHO2)      ***
ENTHO2 = -2884.385    ;RG(6)=ENTHO2
  ***                                                                   ***
  ***     ENTER GAMMA -- USE AS INITAL GUESS              (GA)          ***
GA     = 1.3          ;RG(7)=GA
  ***                                                                   ***
  ***     ENTER INLET PRESSURE -- PSI                     (PRESIN)      ***
PRESIN = 2935.7       ;RG(8)=PRESIN
  ***                                                                   ***
  ***     ENTER FLOW RATE -- LB/SEC                       (FRATE)       ***
FRATE  = 1036.6       ;RG(9)=FRATE
  ***                                                                   ***
  ***     ENTER OUTLET WALL PRESSURE -- PSI               (POTOP)       ***
POTOP  = 1.0          ;RG(10)=POTOP
  ***                                                                   ***
  ***     ENTER OUTLET CENTERLINE PRESSURE -- PSI         (POBOT)       ***
POBOT  = 1.0          ;RG(11)=POBOT
```

```
 ***                                                                     ***
 ***         NOTE:  OUTLET PRESSURE IGNORED IF IG(7)=1                   ***
 ***                                                                     ***
 ****************************************************************************

    GROUP 2. Transience; time-step specification
    GROUP 3. X-direction grid specification
    GROUP 4. Y-direction grid specification
    GROUP 5. Z-direction grid specification
    GROUP 6. Body-fitted coordinates or grid distortion
BFC=T
NONORT=T
SATRUN(NOZ)
SAVGEO=F
    GROUP 7. Variables stored, solved & named
    GROUP 8. Terms (in differential equations) & devices
    GROUP 9. Properties of the medium (or media)
    GROUP 10. Inter-phase-transfer processes and properties
    GROUP 11. Initialization of variable or porosity fields
    GROUP 12. Convection and diffusion adjustments
    GROUP 13. Boundary conditions and special sources
    GROUP 14. Downstream pressure for PARAB=.TRUE.
    GROUP 15. Termination of sweeps
FSWEEP=1
LSWEEP=800
    GROUP 16. Termination of iterations
    GROUP 17. Under-relaxation devices
FTS=RG(31)
DFAL=1.
DLIN=.3
RELAX(P1,LINRLX,1.0*DLIN)
RELAX(R1,LINRLX,1.0*DLIN)
RELAX(R2,LINRLX,1.0*DLIN)
RELAX(RS,LINRLX,1.0*DLIN)
RELAX(W1,FALSDT,FTS*DFAL)
RELAX(W2,FALSDT,FTS*DFAL)
RELAX(V1,FALSDT,FTS*DFAL)
RELAX(V2,FALSDT,FTS*DFAL)
RELAX(KE,FALSDT,FTS*DFAL)
RELAX(EP,FALSDT,FTS*DFAL)
RELAX(H1,FALSDT,FTS*DFAL)
RELAX(C1,FALSDT,FTS*DFAL)
    GROUP 18. Limits on variables or increments to them
    GROUP 19. Data communicated by satellite to GROUND
    GROUP 20. Preliminary print-out
    GROUP 21. Print-out of variables
INIFLD=F
OUTPUT(P1  ,Y,N,N,Y,Y,Y);   OUTPUT(ETPY,N,N,N,N,N,N)
OUTPUT(GAMA,N,N,N,N,N,N);   OUTPUT(VCRT,N,N,N,N,N,N)
OUTPUT(V1  ,Y,N,N,Y,Y,Y);   OUTPUT(WCRT,N,N,N,N,N,N)
OUTPUT(W1  ,Y,N,N,Y,Y,Y);   OUTPUT(MACH,N,N,N,N,N,N)
OUTPUT(KE  ,N,N,N,Y,Y,Y);   OUTPUT(EP  ,N,N,N,Y,Y,Y)
OUTPUT(H1  ,N,N,N,Y,Y,Y);   OUTPUT(ENUT,N,N,N,N,N,N)
OUTPUT(HH  ,N,N,N,N,N,N);   OUTPUT(TEMP,Y,N,N,N,Y,N)
OUTPUT(RHO1,Y,N,N,N,N,N);   OUTPUT(O2  ,N,N,N,N,N,N)
```

```
OUTPUT(H2O ,N,N,N,N,N,N);   OUTPUT(O   ,N,N,N,N,N,N)
OUTPUT(H   ,N,N,N,N,N,N);   OUTPUT(OH  ,N,N,N,N,N,N)
OUTPUT(HO2 ,N,N,N,N,N,N);   OUTPUT(PSIA,N,N,N,N,N,N)
OUTPUT(YCOR,N,N,N,N,N,N);   OUTPUT(ZCOR,N,N,N,N,N,N)
OUTPUT(C1  ,N,N,N,Y,Y,Y);   OUTPUT(R1  ,N,N,N,Y,Y,Y)
OUTPUT(R2  ,N,N,N,Y,Y,Y);   OUTPUT(RS  ,N,N,N,Y,Y,Y)
OUTPUT(W2  ,N,N,N,Y,Y,Y);   OUTPUT(W2CR,N,N,N,N,N,N)
OUTPUT(V2  ,N,N,N,Y,Y,Y);   OUTPUT(V2CR,N,N,N,N,N,N)
OUTPUT(AMDT,Y,N,N,Y,Y,Y)
OUTPUT(LTEM,Y,N,N,Y,Y,Y)
    GROUP 22. Spot-value print-out
IYMON=4;IZMON=80
TSTSWP=50;          NPRMON=TSTSWP
    GROUP 23. Field print-out and plot control
NPRINT=LSWEEP;   IPLTL=LSWEEP;   ITABL=3
ABSIZ=.8;        ORSIZ=.8;       NUMCLS=10
NPLT=TSTSWP;     LUPR3=6
    GROUP 24. Dumps for restarts
  RESTRT(ALL);NAMFI=INXS
STOP
```

```
      CALL TEMPER(ENTMX2,TGUESS,XTEMP,CPDR,RGAS,SC,3,NFO)
C-pd---Calculate density and flow rates------------------------------
      CMW2=(SC(1)*2.*GMWH+SC(2)*2.*GMWO+SC(3)*(2.*GMWH+GMWO))/
     &(SC(1)+SC(2)+SC(3))
      RHOIN=PRESIN*CMW2/(RGAS*XTEMP)
      IF(NFO.GE.1) WRITE(6,955) RHOIN,PRESIN,CMW2,RGAS,XTEMP
      XMDHOT=(XMDOTH+XMDCOM)/TNOJET
      XMDCLD=(XMDOTO-XMDCOM)/TNOJET
      VELH2=XMDHOT/RHOIN/AREAJ2
      VELO2=XMDCLD/RHO2/AREAJ1
      XMDHOT=(XMDOTH+XMDCOM)/GPI/ASUM
      XMDCLD=(XMDOTO-XMDCOM)/GPI/ASUM
      DO 970 IY=1,NY
      DO 970 JJ=31,50
      IF(IG(JJ).EQ.IY) THEN
        RG(JJ+50)=XMDCLD*ACELL(IY)
      ENDIF
  970 CONTINUE
C-pd---Physical properties used to calculate a drop diameter----------
      STEN=.001
      VISXY=3.E-4
      CABS=.037854
      DSC=3.0553
      TERM1=VISXY*((STEN/RHO2)**.5)
      TERM2=RHOIN*((VELH2-VELO2)**2)
      DROPDI=DSC*((TERM1/TERM2)**.6666666)
      DROPDI=DIAJ1/20.
      ENDIF
C*********************************************************************
C-pd---This information is passed into ground from satellite ----------
C---       PRESIN --> total pressure                          (n/sq m)
C---       FRATE  --> flow rate                                (kg/s)
C---       ENTHMX --> enthalpy in                              (j/kg)
C---       CTEMP  --> combustion temperature                      (k)
C---       RGAS   --> gas constant               (n-m/(deg K-kg mole))
C---       RT     --> radius throat                               (m)
C---       CMW    --> combusted mixture molecular weight
C---       AVISC  --> viscosity                          (kg/(m-sec))
C---       CONST2 --> converts psi to N/sq m
C---       FTS    --> false time step                           (sec)
C---       DROPDI --> two phase droplet diameter                  (m)
C---       VELO2  --> velocity of oxygen                      (m/sec)
C---       XMDCLD --> O2 flow rate in one jet                (kg/sec)
C---       STEN   --> surface tension of oxygen                 (N/m)
C---       VISXY  --> viscosity of liquid oxygen         (kg/(m-sec))
C---       CABS   --> factor used in stripping rate
C---       DIAJ1  --> oxygen injection element diameter           (m)
C---       EHTHO2 --> enthalpy of oxygen                       (j/kg)
C---       CONCOP --> Thermal conductivity of copper          (W/K-m)
C---       CONSTE --> Thermal conductivity of steel           (W/K-m)
C---       FLXINL --> Energy rate at inlet for lower            (J/s)
C---                 cooling jacket
C---       FLXINU --> Energy rate at inlet for upper            (J/s)
C---                 cooling jacket
C---       PRHYD  --> Prandtl No. for liquid hydrogen
```

APPENDIX D

SATELLITE Program

LISTING D.1 SATELLITE FILE

```
      PROGRAM MAIN
C-pd---The following parameters have size limitations and may need ---
C---   increasing as your grid becomes larger.  The number in --------
C---   () is the number of times that parameter occurs in satellite --
C---          NYPAR   -->  now set at 100      (2)
C---          NZPAR   -->  now set at 300      (2)
C---          NBFPAR  -->  now set at 100000   (2)
C---
C-----------------------------------------------------------------------
C THIS IS THE MAIN PROGRAM OF THE SATELLITE
C FILE NAME SATLIT.FTN --- 16 July 1986
C
C   (C) COPYRIGHT 1984, LAST REVISION 1986.
C   CONCENTRATION HEAT AND MOMENTUM LTD. ALL RIGHTS RESERVED.
C   This subroutine and the remainder of the PHOENICS code are
C   proprietary software owned by Concentration Heat and Momentum
C   Limited, 40 High Street, Wimbledon, London SW19 5AU, England.
C
      LOGICAL TALK,RUN,DBGFIL,LVAL
      EXTERNAL WAYOUT
C
C 1   Set dimension of blank common arrays, patch-name array &
C     the instruction-stack store here. The dimension of NLN must
C     equal that of STACK; the dimension of STACK must not be less
C     than 250.
      PARAMETER (NYPAR=100,NZPAR=300,NBFPAR=100000)
      COMMON TCVDA(2500),XFRAC(100),YFRAC(NYPAR),ZFRAC(NZPAR),
     1TFRAC(100),BFCS(NBFPAR)
      COMMON/NPAT/NAMPAT(100)/NSTCK/STACK(250)/LINENO/NLN(250)
      CHARACTER NAMPAT*8,STACK*70
C
C 2   Set dimension of run array to MAXRUN.
      COMMON/RUNS/RUN(200)
      COMMON/DISC/DBGFIL
C
C 3   Set dimensions of data-for-GROUND arrays here.
      COMMON/LGRND/LG(20)/IGRND/IG(50)/RGRND/RG(100)/CGRND/CG(10)
      LOGICAL LG
      CHARACTER*4 CG
C
C 4   Set dimensions of data-for-GREX1 arrays here.
      COMMON/LSG/LSGD(20)/ISG/ISGD(20)/RSG/RSGD(100)/CSG/CSGD(10)
      LOGICAL LSGD
      CHARACTER*4 CSGD
C
C 5   Set dimensions for user-declared PIL variables here.
      COMMON/NIDEC/INDEC(45)/IDEC/INVAL(45)
      COMMON/NRDEC/REDEC(45)/RDEC/REVAL(45)
      CHARACTER REDEC*6,INDEC*6
C
```

```
C 6   For more than 25 variables, increase following dimensions.
C     (also, see MAIN of EARTH)
      COMMON/LDB1/DBGPHI(50)/IDA1/ITERMS(50)/IDA2/LITER(50)
     1/IDA3/IORCVF(50)/IDA4/IORCVL(50)/IDA5/ISLN(50)/IDA6/IPRN(50)
     1/HDA1/NAME(50)/RDA1/DTFALS(50)/RDA2/RESREF(50)
     1/RDA3/PRNDTL(50)/RDA4/PRT(50)/RDA5/ENDIT(50)/RDA6/VARMIN(50)
     1/RDA7/VARMAX(50)/RDA8/FIINIT(50)/RDA9/PHINT(50)
     1/RDA10/CINT(50)/RDA11/EX(50)
     1/IPIP1/IP1(50)/HPIP2/IHP2(50)/RPIP1/RVAL(50)
     1/LPIP1/LVAL(50)
      CHARACTER*4 NAME,NSDA,NQ1,NQ2,NCOPY,IHP2,NDUM4,NSCRL,NDUM6*6
      CHARACTER*15 NHLP,NDUM15
C
C 6.5 Set dimension of common blocks for fluid-simulation library
C     files.
      CHARACTER*15 NFILES
      COMMON/NFLS/NFILES(9)
      COMMON/ILIST/LULIST,IDIRL(1001)
C
C 7   Set dimension indicators to correspond with above dimensions.
      CALL SUB4(MAXTCV,2500,MAXRUN,200,NBFC,NBFPAR,NUMPHI,50)
      CALL SUB4(NLG,20,NIG,50,NRG,100,NCG,10)
      CALL SUB4(NLSG,20,NISG,20,NRSG,100,NCSG,10)
      CALL SUB4(NIPIL,45,NRPIL,45,NPNAM,100,NSTACK,250)
      CALL SUB4(NXFR,100,NYFR,100,NZFR,100,NTFR,100)
C
C 8   Logical unit numbers & file names.
      DBGFIL=.FALSE.
      CALL DSCSAT(14,LUPR3,'  ',15,NDUM15,-11,16)
      CALL DSCSAT(9,LUPR2,'  ',15,NDUM15,11,16)
      CALL DSCSAT(4,LUPR1,'  ',15,NDUM15,11,16)
      CALL DSCSAT(-1,LUQ1,'  ',4,NQ1,0,0)
      CALL DSCSAT(-2,LUQ2,'  ',4,NQ2,0,0)
      CALL DSCSAT(-3,LUCOPY,'  ',4,NCOPY,0,0)
      CALL DSCSAT(-5,LUHELP,'  ',15,NHLP,0,0)
      CALL DSCSAT(-6,LULIST,'  ',15,NFILES(1),0,0)
      CALL DSCSAT(-7,LULIST,'  ',15,NFILES(2),0,0)
      CALL DSCSAT(-8,LULIST,'  ',15,NFILES(3),0,0)
      CALL DSCSAT(-10,LUSDA,'  ',4,NSDA,0,0)
      CALL DSCSAT(-15,LUSCRL,'  ',4,NSCRL,0,0)
      CALL DSCSAT(-17,LUGRID,'  ',4,NDUM4,0,0)
      CALL READQ1(LUQ1,NQ1,TALK,RUN,MAXRUN)
C
      CALL WRIT40('FILE FOR NOZZLE FLOWS USED                ')
      CALL PIPPA(TALK,MAXTCV,MAXRUN,NBFC,NUMPHI,NLG,NIG,NRG,NCG,
     1NLSG,NISG,NRSG,NCSG,NIPIL,NRPIL,NPNAM,NSTACK,NXFR,NYFR,NZFR,
     1NTFR,LUSDA,NSDA,LUQ1,NQ1,LUQ2,NQ2,LUCOPY,NCOPY,LUPR1,LUPR2,
     1LUPR3,LUHELP,NHLP,LUSCRL,NSCRL,LUGRID)
      CALL WAYOUT(0)
      END
C*************************************************************
      SUBROUTINE SAT
Cinclude "satear"
C FILE NAME SATEAR --- 170486
CNLIST
```

```
C
C------------------------------------------------------ ARRAYS
      COMMON/LDB1/DBGPHI(50)/IDA1/ITERMS(50)/IDA2/LITER(50)
     1/IDA3/IORCVF(50)/IDA4/IORCVL(50)/IDA5/ISLN(50)/IDA6/IPRN(50)
     1/HDA1/NAME(50)/RDA1/DTFALS(50)/RDA2/RESREF(50)
     1/RDA3/PRNDTL(50)/RDA4/PRT(50)/RDA5/ENDIT(50)/RDA6/VARMIN(50)
     1/RDA7/VARMAX(50)/RDA8/FIINIT(50)/RDA9/PHINT(50)
     1/RDA10/CINT(50)/RDA11/EX(50)
C------------------------------------------------------ LDATA
      COMMON/LDATA/CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32(17),SAVGEO,
     1RSTGEO,NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,
     1DONACC,PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,
     1USEGRD,USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP(11)
C
C------------------------------------------------------ LDEBUG
      COMMON/LDEBUG/DBGEOM,DBADJS,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37(9)
C
C------------------------------------------------------ IDATA
      COMMON/IDATA/NX,NY,NZ,LUPR1,LUPR2,LUPR3,LUPHUN,LUSDA,IPROF,
     1LUFI,LUDST,LUGRF,LUSAVE,LUOLD,LUDEP,LUPCO,LUDVL,
     1IRUNN,IOPTN,LITC,LITFLX,NRUN,LITHYD,FSTEP,LSTEP,
     1FSWEEP,LSWEEP,NPRINT,LIBREF,MEANDF,IXMON,IYMON,IZMON,IINIT,
     1NLSG1,NISG1,NRSG1,NCSG1,IPARAB,IDPHUN,NXFR1,NYFR1,NZFR1,
     1NTFR1,ENTH1,ENTH2,ISWR1,ISWR2,IXPRF,IXPRL,IYPRF,IYPRL,
     1NPRMNT,ISTPRL,ISTPRF,IZPRL,IZPRF,NUMCLS,TSTSWP,NYPRIN,NXPRIN,
     1NZPRIN,NPRMON,NTPRIN,NTZPRF,ISP66,IURINI,IURPRN,IURVAL,
     1IORTCV,NUMREG,NRTCV,ICHR,INTFRC,ITHC1,ISWC1,DEN1,DEN2,
     1VISL,INTMDT,ISWPRF,ISWPRL,IPSA,ISP84,IPLTF,IPLTL,NPLT,ITABL,
     1TEMP1,TEMP2,LEN1,LEN2,NLG1,NIG1,NRG1,NCG1,NPNAM1,
     1ISP98(3),LENREC,LUGEOM,IMB1,IMB2,PCOR,NCOLPF,NCOLCO,
     1NROWCO,EPOR,NPOR,HPOR,VPOR,KXFR,KYFR,KZFR,KTFR,IDATSP(2),
     1VIST,NPHI
C
C------------------------------------------------------ IDEBUG
      COMMON/IDEBUG/IZDB1,IZDB2,ITHDB1,ITHDB2,ISWDB1,ISWDB2,ISTDB1,
     1ISTDB2,INCHCK,IREGDB,NFMAX,IDBF0,IDBCMN,IDBGRD,IDEBSP(2)
C
C------------------------------------------------------ HDATA
      COMMON/HDATA/MESS(10),NBLANK,NAMGRD,NAMEJ,NAMEJ1,
     1NAMEM,NAMEM1,NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,
     1NPHUN,NHINIT,NDST,NAMSAT,NGEOM,NHDASP(2)
C
C------------------------------------------------------ HDEBUG
      COMMON/HDEBUG/NDBF0(2),NDBCMN(2),NHDBSP
C
C------------------------------------------------------ RDATA
      COMMON/RDATA/TINY,GREAT,RUPLIM,RLOLIM,AZDZ,AZXU,AZYV,
     1AZRI,AZAL,AZPH,XULAST,YVLAST,ZWLAST,TLAST,TFIRST,PBAR,SNALFA,
```

```
     1RINNER,ENUL,ENUT,RHO1,RHO2,CFIPS,CMDOT,CONMDT,GRND,HEATBL,
     1FIXFLU,READFI,ZMOVE1,ZDIFAC,DRH1DP,DRH2DP,U1AD,U2AD,V1AD,
     1V2AD,W1AD,W2AD,HUNIT,DIFCUT,ABSIZ,ORSIZ,OPPVAL,TMP1,TMP2,
     1EL1,EL2,GRND1,GRND2,GRND3,GRND4,GRND5,GRND6,GRND7,GRND8,GRND9
     1,GRND10,ZWADD,RINIT,SAME,FIXVAL,AXDZ,AYDZ,RDATSP(21)
C
C----------------------------------------------------- RDEBUG
      COMMON/RDEBUG/BGCHCK,SMCHCK,RDEBSP(5)
C
C---------------------------------LOGICAL DECLARATIONS
      LOGICAL LDAT,LDEB
      LOGICAL CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32,SAVGEO,RSTGEO,
     1NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,DONACC,
     1PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,USEGRD,
     1USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP
      LOGICAL DBGEOM,DBADJS,DBGPHI,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37
C
C---------------------------------INTEGER DECLARATIONS
      INTEGER FSTEP,FSWEEP,TSTSWP,ENTH1,ENTH2,DEN1,
     1DEN2,PCOR,VISL,EPOR,HPOR,VPOR,VIST,TEMP1,TEMP2
C---------------------------------CHARACTER DECLARATIONS
      CHARACTER*4 NHDAT,NHDEB
      CHARACTER*4 NAME
      CHARACTER*4 MESS,NBLANK,NAMGRD,NAMEJ,NAMEJ1,NAMEM,NAMEM1,
     1NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,NPHUN,NHINIT,
     1NDST,NAMSAT,NGEOM,NHDASP
      CHARACTER*4 NDBF0,NDBCMN,NHDBSP
C---------------------------------EQUIVALENT TRANSMISSION ARRAYS
      DIMENSION LDAT(84),LDEB(45),IDAT(120),IDEB(16),NHDAT(30),
     1NHDEB(5),RDAT(85),RDEB(7)
      EQUIVALENCE (LDAT(1),CARTES),(LDEB(1),DBGEOM),(IDAT(1),NX),
     1(IDEB(1),IZDB1),(NHDAT(1),MESS(1)),(NHDEB(1),NDBF0(1)),
     1(RDAT(1),TINY),(RDEB(1),BGCHCK)
CLIST
#include "satloc"
      CALL SATLIT
      RETURN
      END
C************************************************************
      SUBROUTINE SATLIT
Cinclude "satear"
C FILE NAME SATEAR --- 170486
CNLIST
C
C--------------------------------------------------- ARRAYS
      COMMON/LDB1/DBGPHI(50)/IDA1/ITERMS(50)/IDA2/LITER(50)
     1/IDA3/IORCVF(50)/IDA4/IORCVL(50)/IDA5/ISLN(50)/IDA6/IPRN(50)
     1/HDA1/NAME(50)/RDA1/DTFALS(50)/RDA2/RESREF(50)
```

```
     1/RDA3/PRNDTL(50)/RDA4/PRT(50)/RDA5/ENDIT(50)/RDA6/VARMIN(50)
     1/RDA7/VARMAX(50)/RDA8/FIINIT(50)/RDA9/PHINT(50)
     1/RDA10/CINT(50)/RDA11/EX(50)
C----------------------------------------------------- LDATA
      COMMON/LDATA/CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32(17),SAVGEO,
     1RSTGEO,NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,
     1DONACC,PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,
     1USEGRD,USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP(11)
C
C----------------------------------------------------- LDEBUG
      COMMON/LDEBUG/DBGEOM,DBADJS,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37(9)
C
C----------------------------------------------------- IDATA
      COMMON/IDATA/NX,NY,NZ,LUPR1,LUPR2,LUPR3,LUPHUN,LUSDA,IPROF,
     1LUFI,LUDST,LUGRF,LUSAVE,LUOLD,LUDEP,LUPCO,LUDVL,
     1IRUNN,IOPTN,LITC,LITFLX,NRUN,LITHYD,FSTEP,LSTEP,
     1FSWEEP,LSWEEP,NPRINT,LIBREF,MEANDF,IXMON,IYMON,IZMON,IINIT,
     1NLSG1,NISG1,NRSG1,NCSG1,IPARAB,IDPHUN,NXFR1,NYFR1,NZFR1,
     1NTFR1,ENTH1,ENTH2,ISWR1,ISWR2,IXPRF,IXPRL,IYPRF,IYPRL,
     1NPRMNT,ISTPRL,ISTPRF,IZPRL,IZPRF,NUMCLS,TSTSWP,NYPRIN,NXPRIN,
     1NZPRIN,NPRMON,NTPRIN,NTZPRF,ISP66,IURINI,IURPRN,IURVAL,
     1IORTCV,NUMREG,NRTCV,ICHR,INTFRC,ITHC1,ISWC1,DEN1,DEN2,
     1VISL,INTMDT,ISWPRF,ISWPRL,IPSA,ISP84,IPLTF,IPLTL,NPLT,ITABL,
     1TEMP1,TEMP2,LEN1,LEN2,NLG1,NIG1,NRG1,NCG1,NPNAM1,
     1ISP98(3),LENREC,LUGEOM,IMB1,IMB2,PCOR,NCOLPF,NCOLCO,
     1NROWCO,EPOR,NPOR,HPOR,VPOR,KXFR,KYFR,KZFR,KTFR,IDATSP(2),
     1VIST,NPHI
C
C----------------------------------------------------- IDEBUG
      COMMON/IDEBUG/IZDB1,IZDB2,ITHDB1,ITHDB2,ISWDB1,ISWDB2,ISTDB1,
     1ISTDB2,INCHCK,IREGDB,NFMAX,IDBF0,IDBCMN,IDBGRD,IDEBSP(2)
C
C----------------------------------------------------- HDATA
      COMMON/HDATA/MESS(10),NBLANK,NAMGRD,NAMEJ,NAMEJ1,
     1NAMEM,NAMEM1,NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,
     1NPHUN,NHINIT,NDST,NAMSAT,NGEOM,NHDASP(2)
C
C----------------------------------------------------- HDEBUG
      COMMON/HDEBUG/NDBF0(2),NDBCMN(2),NHDBSP
C
C----------------------------------------------------- RDATA
      COMMON/RDATA/TINY,GREAT,RUPLIM,RLOLIM,AZDZ,AZXU,AZYV,
     1AZRI,AZAL,AZPH,XULAST,YVLAST,ZWLAST,TLAST,TFIRST,PBAR,SNALFA,
     1RINNER,ENUL,ENUT,RHO1,RHO2,CFIPS,CMDOT,CONMDT,GRND,HEATBL,
     1FIXFLU,READFI,ZMOVE1,ZDIFAC,DRH1DP,DRH2DP,U1AD,U2AD,V1AD,
     1V2AD,W1AD,W2AD,HUNIT,DIFCUT,ABSIZ,ORSIZ,OPPVAL,TMP1,TMP2,
     1EL1,EL2,GRND1,GRND2,GRND3,GRND4,GRND5,GRND6,GRND7,GRND8,GRND9
     1,GRND10,ZWADD,RINIT,SAME,FIXVAL,AXDZ,AYDZ,RDATSP(21)
```

```
C
C---------------------------------------------------------- RDEBUG
      COMMON/RDEBUG/BGCHCK,SMCHCK,RDEBSP(5)
C
C-----------------------------------LOGICAL DECLARATIONS
      LOGICAL LDAT,LDEB
      LOGICAL CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32,SAVGEO,RSTGEO,
     1NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,DONACC,
     1PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,USEGRD,
     1USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP
      LOGICAL DBGEOM,DBADJS,DBGPHI,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37
C
C-----------------------------------INTEGER DECLARATIONS
      INTEGER FSTEP,FSWEEP,TSTSWP,ENTH1,ENTH2,DEN1,
     1DEN2,PCOR,VISL,EPOR,HPOR,VPOR,VIST,TEMP1,TEMP2
C-----------------------------------CHARACTER DECLARATIONS
      CHARACTER*4 NHDAT,NHDEB
      CHARACTER*4 NAME
      CHARACTER*4 MESS,NBLANK,NAMGRD,NAMEJ,NAMEJ1,NAMEM,NAMEM1,
     1NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,NPHUN,NHINIT,
     1NDST,NAMSAT,NGEOM,NHDASP
      CHARACTER*4 NDBF0,NDBCMN,NHDBSP
C-----------------------------------EQUIVALENT TRANSMISSION ARRAYS
      DIMENSION LDAT(84),LDEB(45),IDAT(120),IDEB(16),NHDAT(30),
     1NHDEB(5),RDAT(85),RDEB(7)
      EQUIVALENCE (LDAT(1),CARTES),(LDEB(1),DBGEOM),(IDAT(1),NX),
     1(IDEB(1),IZDB1),(NHDAT(1),MESS(1)),(NHDEB(1),NDBF0(1)),
     1(RDAT(1),TINY),(RDEB(1),BGCHCK)
CLIST
#include "satloc"
#include "bfcsat"
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION STARTS:
C
C 1   Set dimension of blank common arrays here as per MAIN.
      PARAMETER (NYPAR=100,NZPAR=300,NBFPAR=100000)
      COMMON F(1), TCVDA(2500),XFRAC(100),YFRAC(NYPAR),ZFRAC(NZPAR),
     1TFRAC(100),BFCS(NBFPAR)
C-pd---User dimensions are set here and are defined as follows: ------
C---       YN    --> array used to store radii
C---       SM0   --> array used to store mass fractions
C---       SMB   --> array used to store mass fractions
C---       SC    --> array used to store mass fractions/molecular wts
C---       PWRS  --> array for storing r/rt locations
C---       PWZS  --> array for storing z/rt locations
C---       SLOPE --> array used to store wall slopes
C---       ACELL --> array used to store inlet areas
C---
C---           NOTE:  Dimension PWRS PRZS & SLOPE to NRZS
```

```
C---
C-----------------------------------------------------------------------
      DIMENSION YN(NZPAR),SM0(3),SMB(3),SC(3),PWRS(18),PWZS(18),
     &          SLOPE(18),ACELL(NYPAR)
C
C 2   Modify data for GROUND as required.
      COMMON/LGRND/LG(20)/IGRND/IG(50)/RGRND/RG(100)/CGRND/CG(10)
      LOGICAL LG
      CHARACTER*4 CG
      CHARACTER*8 GNAME
C
C 3   Introduce SATLIT-only commons, arrays, equivalences.
      COMMON/LSG/DUDX,DVDX,DWDX,DUDY,DVDY,DWDY,DUDZ,DVDZ,DWDZ,GENK,
     1LSG1,LSG2,LSG3,LSG4,LSG5,LSG6,LSG7,LSG8,LSG9,LSG10
      LOGICAL DUDX,DVDX,DWDX,DUDY,DVDY,DWDY,DUDZ,DVDZ,DWDZ,GENK,
     1LSG1,LSG2,LSG3,LSG4,LSG5,LSG6,LSG7,LSG8,LSG9,LSG10
      COMMON/ISG/IZW1,ISG1,ISG2,ISG3,ISG4,ISG5,ISG6,ISG7,ISG8,ISG9,
     1ISG10,ISG11,ISG12,ISG13,ISG14,ISG15,ISG16,ISG17,ISG18,KELIN
      COMMON/RSG/TEMP0,PRESS0,ENULA,ENULB,ENULC,ENUTA,ENUTB,ENUTC,
     1CFIPA,CFIPB,CFIPC,CFIPD,CMDTA,CMDTB,CMDTC,CMDTD,WALLA,WALLB,
     1TMP1A,TMP1B,TMP1C,RHO1A,RHO1B,RHO1C,PRLH1A,PRLH1B,PRLH1C,
     1PRLC1A,PRLC1B,PRLC1C,PRLC3A,PRLC3B,PRLC3C,EL1A,EL1B,EL1C,
     1CINH1A,CINH1B,CINH1C,PHNH1A,PHNH1B,PHNH1C,
     1TMP2A,TMP2B,TMP2C,RHO2A,RHO2B,RHO2C,PRLH2A,PRLH2B,PRLH2C,
     1PRLC2A,PRLC2B,PRLC2C,PRLC4A,PRLC4B,PRLC4C,EL2A,EL2B,EL2C,
     1CINH2A,CINH2B,CINH2C,PHNH2A,PHNH2B,PHNH2C,
     1AZW1,BZW1,CZW1,DZW1,RSG1,RSG2,RSG3,RSG4,RSG5,RSG6,RSG7,RSG8,
     1RSG9,RSG10,RSG11,RSG12,RSG13,RSG14,RSG15,RSG16,RSG17,RSG18,
     1RSG19,RSG20,RSG21,RSG22,RSG23,RSG24,RSG25,RSG26,RSG27,RSG28,
     1RSG29,RSG30
      COMMON/CSG/CSG1,CSG2,CSG3,CSG4,CSG5,CSG6,CSG7,CSG8,CSG9,CSG10
      CHARACTER*4 CSG1,CSG2,CSG3,CSG4,CSG5,CSG6,CSG7,CSG8,CSG9,
     1CSG10
C
C 4   User places his data statements here.
      DATA GNAME/'INLET0  '/
C-pd---PWRS & PWZS defined above ------------------------------------
C---   NRZS is the number of data points in PWRS & PWZS and ---------
C---   the first value in each array is calculated in the program ---
      DATA PWRS / 0.000000, 1.134155, 1.403205, 1.713798, 2.141943,
     &            2.460367, 2.925982, 3.561216, 4.466252, 5.349017,
     &            5.795971, 6.276294, 6.780853, 7.297192, 7.806621,
     &            8.228043, 8.605041, 8.802905 /
      DATA PWZS / 0.000000, .3094342, .6749371, 1.112398, 1.744615,
     &            2.238402, 2.999905, 4.123805, 5.925968, 7.974671,
     &            9.152518, 10.55318, 12.21444, 14.18235, 16.50891,
     &            18.88728, 21.63390, 23.51770 /
      DATA NRZS / 18 /
C
C***********************************************************************
C-pd---This information is passed into satellite from Q1 ------------
C---         NZT    --> throat location
C---         IYBOT  --> iyf used in inlet patch
C---         IYTOP  --> iyl used in inlet patch
C---         NJETS  --> number of jets
```

```
C---          PU     --> power upto throat
C---          PD     --> power down from throat
C---          PR     --> power to wall
C---          FMIX   --> fuel mixture ratio
C---          ENTHH2 --> enthalpy for hydrogen   (cal/mole)
C---          ENTHO2 --> enthalpy for oxygen     (cal/mole)
C---          GA     --> gamma
C---          PRESIN --> inlet pressure          (psi)
C---          FRATE  --> flow rate               (lb/sec)
C--           GPI    --> geometric factor
C-----------------------------------------------------------------------
      NZT    = IG(21)
      IYBOT  = IG(22)
      IYTOP  = IG(23)
      NJETS  = IG(30)
      PU     = RG(1)
      PD     = RG(2)
      PR     = RG(3)
      FMIX   = RG(4)
      ENTHH2 = RG(5)
      ENTHO2 = RG(6)
      GA     = RG(7)
      PRESIN = RG(8)
      FRATE  = RG(9)
      GPI    = RG(12)
C***********************************************************************
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,
     122,23,24),IGR
C
C--- GROUP 1. Run title and other preliminaries
    1 CONTINUE
C-pd---NFO is the flag in satellite that controls printout -----------
C---   The higher the number the larger the amount of print ----------
      NFO=IG(19)
      IG(17)=1
C***********************************************************************
C-pd---this data is used to calculate the radius at any z-location ---
C---   and is defined as follows: ----------------------------------
C---        THETAO   --> inlet angle
C---        THETA1   --> angle associated with radius 1 and radius U
C---        THETAD   --> angle associated with radius D
C---        THETAE   --> exit angle
C---        RORT     --> radius at inlet divided by radius at throat
C---        R1RT     --> radius at 1 divided by radius at throat
C---        RURT     --> radius at U divided by radius at throat
C---                     (radius U is the upstream radius at throat)
C---        RDRT     --> radius at D divided by radius at throat
C---                     (radius D is the downstream radius at throat)
C---        RERT     --> radius at exit divided by radius at throat
C---        CONST1   --> converts inches to meters
C---        RT       --> radius at throat (inches)
C---        Z1       --> z distance at 1 (m)
C---        ZE       --> z distance at exit (m)
C---   **** NOTE: Location O is at the inlet and the begining of
C---                           the first straight segment
```

```
C---                 Location 1 is ending of the first straight segment
C---                            and the begining of the first arc
C---                 Location 2 is the ending of the first arc and the
C---                            begining of the second straight segment
C---                 Location 3 is ending of the second straight segment
C---                            and the begining of the second arc
C---                 Location T is the ending of the second arc, the
C---                            begining of the third arc and the
C---                            location of the throat
C---                 Location 4 is the ending of the third arc and
C---                            the begining of the bell shape
C---                 Location E is the exit and the ending of the
C---                            bell shape
C----------------------------------------------------------------------
      THETAO=0.0
      THETA1=25.4167
      THETAD=37.0
      THETAE=5.3738
      RORT=3.0**.5
      R1RT=1.73921
      RURT=1.0
      RDRT=0.392
      RERT=77.5**.5
      CONST1=0.0254
      RT=5.1527*CONST1
C-pd---Z1 & ZE ARE DISTANCES EXTRACTED FROM THE NOZZLE DATA FILE
      Z1=0.073137
      ZE=3.4336
C***********************************************************************
C-pd---Calculate Z2 Z3 ZT & Z4 along with the following radii --------
C---    YO Y1 Y2 Y3 & Y4 ----------------------------------------------
C---    **** NOTE:  The general nomenclature is as follows:
C---                RAD_     -->  radians of a given angle
C---                RADIS_   -->  radius at a given location
C---                CORD_    -->  is the half cord length
C---                DIST_    -->  distance to the largest cord
C---                H_MAX    -->  is the maximum rise
C---                CORD     -->  is the local half cord length
C---                DIST     -->  distance to the local cord
C---                HDIS     -->  is the local rise
C---                ZT       -->  z distance at throat
C---                DROP_    -->  drop between two locations
C---                RUN_     -->  length between two locations
C----------------------------------------------------------------------
      PI=3.14159254
      RADO=THETAO*PI/180.
      DROPO=Z1*TAN(RADO)
      YO=RORT*RT
      Y1=YO-DROPO
C
      RAD1=THETA1*PI/180.
      RADIS1=R1RT*RT
      CORD1=SIN(RAD1)*RADIS1
      Z2=Z1+CORD1
      DIST1=(RADIS1**2-CORD1**2)**.5
```

```
      H1MAX=RADIS1-DIST1
      Y2=Y1-H1MAX
C
      RADU=THETA1*PI/180.
      RADISU=RURT*RT
      CORDU=SIN(RADU)*RADISU
      DISTU=(RADISU**2-CORDU**2)**.5
      HUMAX=RADISU-DISTU
      Y3=RT+HUMAX
      DROP2=Y2-Y3
      RUN2=DROP2/TAN(RAD1)
      Z3=Z2+RUN2
      ZT=Z3+CORDU
C
      RADD=THETAD*PI/180.
      RADISD=RDRT*RT
      CORDD=SIN(RADD)*RADISD
      Z4=ZT+CORDD
      DISTD=(RADISD**2-CORDD**2)**.5
      HDMAX=RADISD-DISTD
      Y4=RT+HDMAX
C
      RADE=THETAE*PI/180.
C***********************************************************************
C-IWC--Properties and geometric data for Cooling Jacket Simulation ---
C---
C---   Physical properties for H are taken at P = 45 MPa and T = 160 K,
C---   mean values as estimated from supplied data.
C---
C---   Physical properties of metals taken as mean values from data
C---   tabulated in SUTTON, G P (1986) : 'Rocket Propulsion Elements'
C---
C---   The properties are defined as follows ------------------------
C---        CONCOP -->  Thermal conductivity of copper   (W/K-m)
C---        CONSTE -->  Thermal conductivity of steel    (W/K-m)
C---        FLXINL -->  Energy rate at inlet for lower   (J/s)
C---                    cooling jacket
C---        FLXINU -->  Energy rate at inlet for upper   (J/s)
C---                    cooling jacket
C---        PRHYD  -->  Prandtl No. for liquid hydrogen
C---        RATEL  -->  Mass flow at inlet for lower     (Kg/s)
C---                    cooling jacket
C---        RATEU  -->  Mass flow at inlet for upper     (Kg/s)
C---                    cooling jacket
C---        TLIQL  -->  Temperature of hydrogen at inlet (K)
C---        TLIQU  -->  Temperature of hydrogen at inlet (K)
C---        VISHYD -->  Dynamic viscosity of hydrogen    (Kg/m-s)
C---------------------------------------------------------------------
C---   The geometric data are defined as follows --------------------
C---        NCHA   -->  Number of channels in combustor jacket
C---        NTUB   -->  Number of tubes in nozzle jacket
C---        NCOM   -->  Number of data stations in combustor jacket
C---        NNOZ   -->  Number od data stations in nozzle jacket
C---------------------------------------------------------------------
       CONCOP = 364.8
```

```
      CONSTE = 50.5
      FLXINL = 6.9508472E6/(GPI)
      FLXINU = 2.004106E7/(GPI)
      PRHYD  = 0.8187
      RATEL  = 11.825/(GPI)
      RATEU  = 29.2839/(GPI)
      TLIQL  = 53.1283
      TLIQU  = 60.82167
      VISHYD = 92.4E-7
      NCHA   = 380
      NTUB   = 1080
      NCOM   = 100
      NNOZ   = 60
C***********************************************************************
      RETURN
C
C--- GROUP 2. Transience; time-step specification
    2 CONTINUE
      RETURN
C
C--- GROUP 3. X-direction grid specification
    3 CONTINUE
      RETURN
C
C--- GROUP 4. Y-direction grid specification
    4 CONTINUE
C-pd---Return if a grid generated grid is used -----------------------
      IF(IG(1).EQ.2) RETURN
C-pd---Setup y-fractions between 0 and 1 -----------------------------
C---    Calculate yfracs to wall ---------------------------------------
      YFRAC(NY)=1.
      DO 400 IY=1,NY-1
  400 YFRAC(NY-IY)=1.-(FLOAT(IY)/FLOAT(NY))**PR
C-pd---Print out yfracs ---------------------------------------------
      IF(NFO.GE.2) WRITE(6,425) (YFRAC(IY),IY=1,NY)
  425 FORMAT(' YFRAC'/(1P,5E11.3))
      RETURN
C
C--- GROUP 5. Z-direction grid specification
    5 CONTINUE
C-pd---Return if a grid generated grid is used -----------------------
      IF(IG(1).EQ.2) RETURN
C-pd---Setup z-fractions between 0 and 1 -----------------------------
      ZFRAC(NZT)=ZT/ZE
      ZFRAC(NZ)=1.
C-pd---Calculate zfracs up to throat ---------------------------------
      DO 500 IZ=1,NZT-1
  500 ZFRAC(NZT-IZ)=(1.-(FLOAT(IZ)/FLOAT(NZT))**PU)*ZFRAC(NZT)
C-pd---Calculate zfracs down from throat -----------------------------
      ZNOZ=ZFRAC(NZ)-ZFRAC(NZT)
      DO 505 IZ=NZT+1,NZ-1
  505 ZFRAC(IZ)=ZFRAC(NZT)+(FLOAT(IZ-NZT)/FLOAT(NZ-NZT))**PD*ZNOZ
C-pd---Print out zfracs ---------------------------------------------
      IF(NFO.GE.2) WRITE(6,525) (ZFRAC(IZ),IZ=1,NZ)
  525 FORMAT(' ZFRAC'/(1P,5E11.3))
```

```
      RETURN
C
C--- GROUP 6. Body-fitted coordinates or grid distortion
    6 CONTINUE
C-pd---Return if a grid generated grid is used -----------------------
      IF(IG(1).EQ.2) CALL READCO(CSG1)
      IF(IG(1).EQ.2) GOTO 650
C-pd---Begin loop to check for which section the given ZDIST falls ---
C---    into & calculate the associated radius (YN) ------------------
      DO 600 IZ=1,NZ-1
      ZDIST=ZFRAC(IZ)*ZE
C-pd---Calculate points up to first bend (up to Z1) -----------------
         IF(ZDIST.LE.Z1) THEN
           DROP=(Z1-ZDIST)*TAN(RADO)
           YN(IZ)=Y1+DROP
           GOTO 600
         ENDIF
C-pd---Calculate points in first bend (up to Z2) --------------------
         IF(ZDIST.LE.Z2) THEN
           CORD=ZDIST-Z1
           DIST=(RADIS1**2-CORD**2)**.5
           HDIS=RADIS1-DIST
           YN(IZ)=Y1-HDIS
           GOTO 600
         ENDIF
C-pd---Calculate points after first bend (up to Z3) -----------------
         IF(ZDIST.LE.Z3) THEN
           DIST=ZDIST-Z2
           YN(IZ)=(DIST/(Z3-Z2))*(Y3-Y2)+Y2
           GOTO 600
         ENDIF
C-pd---Calculate points in second bend (up to ZT) -------------------
         IF(ZDIST.LE.ZT) THEN
           CORD=ZT-ZDIST
           DIST=(RADISU**2-CORD**2)**.5
           HDIS=RADISU-DIST
           YN(IZ)=RT+HDIS
           GOTO 600
         ENDIF
C-pd---Calculate points in third bend (up to Z4) --------------------
         IF(ZDIST.LE.Z4) THEN
           CORD=ZDIST-ZT
           DIST=(RADISD**2-CORD**2)**.5
           HDIS=RADISD-DIST
           YN(IZ)=RT+HDIS
           GOTO 600
         ENDIF
C-pd---Calculate points after third bend (up to ZE) -----------------
         IF(ZDIST.LE.ZE) THEN
           CS=COS(RADD)
           SN=SIN(RADD)
           RWTD=RDRT
           ZWMAX=(ZE-ZT)/RT
           RWMAX=RERT
           PW101=1.0+RWTD*(1.0-CS)
```

```
          PW102=RWTD*SN
          IOPT=IG(2)
C-pd---IOPT = 1  gives a cone shape ---------------------------------
          IF(IOPT.EQ.1) THEN
            YE=RERT*RT
            FACT=(ZDIST-Z4)/(ZE-Z4)
            RISE=(YE-Y4)*FACT
            YN(IZ)=Y4+RISE
          ENDIF
C-pd---The following 3 options were taken from the TDK subroutine ----
C---   called WALL and modified so that for a known distance (ZDIST) -
C---   a radius (YN) could be calculated ----------------------------
C-pd---IOPT = 2 gives a parabolic shape ------------------------------
          IF(IOPT.EQ.2) THEN
            CC1=(ZWMAX-PW102)*SN/CS
            AA1=(RWMAX**2-PW101**2-2.0*PW101*CC1)/2.0/
     &          (RWMAX-PW101-CC1)
            BB1=2.0*(PW101-AA1)*SN/CS
            CC1=ZWMAX-(RWMAX-AA1)**2/BB1
            YN(IZ)=(AA1+SQRT(BB1*((ZDIST-ZT)/RT-CC1)))*RT
          ENDIF
C-pd---IOPT = 3 gives an arc shape -----------------------------------
          IF(IOPT.EQ.3) THEN
            AA1=((ZWMAX-PW102)**2+(RWMAX-PW101)**2)/(2.0*
     &          ((ZWMAX-PW102)*SN-(RWMAX-PW101)*CS))
            BB1=SN-(ZWMAX-PW102)/AA1
            THER=ATAN(BB1/SQRT(1.0-BB1**2))
            DLTHR=(THER-RADD)*(ZDIST-Z4)/(ZE-Z4)
            ANL=RADD+DLTHR
            YN(IZ)=(PW101+AA1*(COS(ANL)-CS))*RT
          ENDIF
C-pd---IOPT = 4 uses a spline fit to give the shape ------------------
          IF(IOPT.EQ.4) THEN
            IPASS=1
            PWRS(1) = PW101
            PWZS(1) = PW102
            SLOPE(1) = TAN(RADD)
            SLOPE(NRZS) = TAN(RADE)
            IF(IPASS.GT.1) GOTO 605
            CALL XSLP (PWZS,PWRS,NRZS,SLOPE,1)
  605       ZLOC =(ZDIST-ZT)/RT
C-pd---Find the two data points that ZLOC is between -----------------
C---   Warning will indicate inconsistency in data -------------------
            IF(ZLOC.LT.PWZS(1)) WRITE(6,*)' WARNING ZLOC BELOW
     &        FIRST DATA POINT'
            IF(ZLOC.GT.PWZS(NRZS)) WRITE(6,*)' WARNING ZLOC ABOVE
     &        LAST DATA POINT'
            DO 610 I=1,NRZS
  610       IF(ZLOC.LT.PWZS(I)) GOTO 620
  620       HH1=PWZS(I)-PWZS(I-1)
            DDX=ZLOC-PWZS(I-1)
            DDY=PWRS(I)-PWRS(I-1)
            YPS=SLOPE(I)+SLOPE(I-1)
            YN(IZ)=(PWRS(I-1)+DDX*(DDX*(DDX*(YPS*HH1 - 2.0*DDY)
     &             /HH1/HH1**2-(HH1*(SLOPE(I-1)+YPS)-3.0*DDY)
```

```
     &              /HH1**2)+SLOPE(I-1)))*RT
            IPASS=IPASS+1
          ENDIF
        ENDIF
  600 CONTINUE
C-pd---Calculate last radius ----------------------------------------
      YN(NZ)=RERT*RT
C-pd---Calculate interior points and set the values using SETPT ------
C-pd---X points are calculated as .01 of the y distance --------------
      DO 640 IZ=1,NZ+1
      DO 630 IY=1,NY+1
      YLOC=0.0
      IF(IY.GT.1.AND.IZ.EQ.1) YLOC=YFRAC(IY-1)*RORT*RT
      IF(IY.GT.1.AND.IZ.GT.1) YLOC=YFRAC(IY-1)*YN(IZ-1)
      ZLOC=0.0
      IF(IZ.GT.1) ZLOC=ZFRAC(IZ-1)*ZE
      XXW=-0.01*YLOC
      XXE=0.01*YLOC
      CALL SETPT(1,IY,IZ,XXW,YLOC,ZLOC)
  630 CALL SETPT(2,IY,IZ,XXE,YLOC,ZLOC)
  640 IF(NFO.GE.3)
     &      WRITE(6,'('' IZ YDIS ZDIST'',I4,1P,2E12.4)') IZ,YLOC,ZLOC
C-pd---Scale all points if necessary & calc. ropen fts rt and crossa
  650 SFAC=1.
      IF(IG(17).EQ.2) SFAC=100.
      CALL GSCALE(SFAC)
      LASTF=(NX+1)*(NY+1)*(NZ+1)
      NZTF=(NX+1)*(NY+1)*(NZT+1)
      WAVG=2000.*SFAC
      CALL GEOMTX(F(KXC+1),F(KYC+1),F(KZC+1),LASTF,NZTF,IYBOT,IYTOP,
     &            NY,NZ,WAVG,ROPEN,FTS,RT,CROSSA,GPI,ACELL)
      IF(IG(9).EQ.2) THEN
        ASUM=0.0
        DO 660 IY=1,NY
        DO 660 JJ=31,50
  660   IF(IG(JJ).EQ.IY) ASUM=ASUM+ACELL(IY)
      ENDIF
      RETURN
C
C--- GROUP 7. Variables stored, solved & named
    7 CONTINUE
      IF(IG(9).EQ.2) ONEPHS=.FALSE.
      CALL SOLUTN( 1,Y,Y,Y,N,N,N)
      CALL SOLUTN( 5,Y,Y,N,Y,N,N)
      IF(IG(9).EQ.2) CALL SOLUTN( 6,Y,Y,N,Y,N,N)
      CALL SOLUTN( 7,Y,Y,N,Y,N,N)
      IF(IG(9).EQ.2) CALL SOLUTN( 8,Y,Y,N,Y,N,N)
      IF(IG(9).EQ.2) CALL SOLUTN( 9,Y,Y,N,Y,N,N)
      IF(IG(9).EQ.2) CALL SOLUTN(10,Y,Y,N,Y,N,N)
      IF(IG(9).EQ.2) CALL SOLUTN(11,Y,Y,N,Y,N,N)
      IF(IG(3).EQ.1) CALL SOLUTN(12,Y,Y,N,N,N,N)
      IF(IG(3).EQ.1) CALL SOLUTN(13,Y,Y,N,N,N,N)
      CALL SOLUTN(14,Y,Y,N,N,N,N)
      CALL SOLUTN(16,Y,Y,N,N,N,N)
      CALL SOLUTN(17,Y,N,N,N,N,N)
```

```
CALL SOLUTN(18,Y,N,N,N,N,N)
CALL SOLUTN(19,Y,N,N,N,N,N)
CALL SOLUTN(20,Y,N,N,N,N,N)
CALL SOLUTN(21,Y,N,N,N,N,N)
CALL SOLUTN(22,Y,N,N,N,N,N)
CALL SOLUTN(23,Y,N,N,N,N,N)
CALL SOLUTN(24,Y,N,N,N,N,N)
CALL SOLUTN(25,Y,N,N,N,N,N)
CALL SOLUTN(26,Y,N,N,N,N,N)
CALL SOLUTN(27,Y,N,N,N,N,N)
CALL SOLUTN(28,Y,N,N,N,N,N)
CALL SOLUTN(29,Y,N,N,N,N,N)
CALL SOLUTN(30,Y,N,N,N,N,N)
IF(IG(9).EQ.2)CALL SOLUTN(31,Y,N,N,N,N,N)
IF(IG(8).EQ.3)CALL SOLUTN(32,Y,N,N,N,N,N)
CALL SOLUTN(45,Y,N,N,N,N,N)
CALL SOLUTN(46,Y,N,N,N,N,N)
IF(IG(9).EQ.2)CALL SOLUTN(47,Y,N,N,N,N,N)
IF(IG(9).EQ.2)CALL SOLUTN(48,Y,N,N,N,N,N)
CALL SOLUTN(49,Y,N,N,N,N,N)
CALL SOLUTN(50,Y,N,N,N,N,N)
NAME( 1) = 'P1  '
NAME( 5) = 'V1  '
IF(IG(9).EQ.2) NAME( 6) = 'V2  '
NAME( 7) = 'W1  '
IF(IG(9).EQ.2) NAME( 8) = 'W2  '
IF(IG(9).EQ.2) NAME( 9) = 'R1  '
IF(IG(9).EQ.2) NAME(10) = 'R2  '
IF(IG(9).EQ.2) NAME(11) = 'RS  '
IF(IG(3).EQ.1) NAME(12) = 'KE  '
IF(IG(3).EQ.1) NAME(13) = 'EP  '
NAME(14) = 'H1  '
NAME(16) = 'C1  '
NAME(17) = 'HH  '
NAME(18) = 'O2  '
NAME(19) = 'H2O '
NAME(20) = 'O   '
NAME(21) = 'H   '
NAME(22) = 'OH  '
NAME(23) = 'HO2 '
NAME(24) = 'ENUT'
NAME(25) = 'RHO1'
NAME(26) = 'TEMP'
NAME(27) = 'ETPY'
NAME(28) = 'GAMA'
NAME(29) = 'MACH'
NAME(30) = 'PSIA'
IF(IG(9).EQ.2) NAME(31) = 'AMDT'
IF(IG(8).EQ.3) NAME(32) = 'LTEM'
NAME(45) = 'YCOR'
NAME(46) = 'ZCOR'
IF(IG(9).EQ.2) NAME(47) = 'V2CR'
IF(IG(9).EQ.2) NAME(48) = 'W2CR'
NAME(49) = 'VCRT'
NAME(50) = 'WCRT'
```

```
      DEN1=25
      VIST=24
      INTMDT=31
      IF(IG(3).EQ.1) ENUT=GRND3
      IF(IG(3).EQ.1) EL1=GRND4
      RETURN
C
C--- GROUP 8. Terms (in differential equations) & devices
    8 CONTINUE
      NEWRH1=.FALSE.
      NEWENT=.FALSE.
      NEWENL=.FALSE.
      DIFCUT=0.0
      RETURN
C
C--- GROUP 9. Properties of the medium (or media)
    9 CONTINUE
C***********************************************************************
C-pd---Properties -----------------------------------------------------
C---    The properties and other quantities are defined as follows ----
C---          GMWH    -->  molecular weight of hydrogen
C---          GMWO    -->  molecular weight of oxygen
C---          CONST2 -->  converts psi to N/sq m
C---          CONST3 -->  converts cal/mole to J/kg mole
C---          CONST4 -->  converts kg to lb
C---          CONST5 -->  converts lb/(ft-sec) to kg/(m-sec)
C---          ENTHMX -->  enthalpy of mixture
C---          RGAS    -->  gas constant      (n-m/(deg K-kg mole))
C---          SM0(1) -->  mass fraction of H2 before combustion
C---          SM0(2) -->  mass fraction of O2 before combustion
C---          SM0(3) -->  mass fraction of H2O before combustion
C---          SC(1)   -->  molar concentration of H2
C---          SC(2)   -->  molar concentration of O2
C---          SC(3)   -->  molar concentration of H2O
C---          SMB(1) -->  mass fraction of H2 after combustion
C---          SMB(2) -->  mass fraction of O2 after combustion
C---          SMB(3) -->  mass fraction of H2O after combustion
C---          TGUESS -->  guess for temperature after combustion
C---          CTEMP   -->  temperature after combustion          (k)
C---          CMW     -->  molecular weight of combustion mixture
C---          RHOIN   -->  inlet density                   (kg/cu m)
C---          RHOEX   -->  rough guess for outlet density(kg/cu m)
C---          RCHAM   -->  radius of combustion chamber          (m)
C---          CROSSA -->  cross sectional area of chamber  (sq m)
C---          RHOVEL -->  density times velocity  (kg/(sec-sq m))
C---          WIN     -->  inlet velocity                     (m/sec)
C---          AVISC   -->  absolute viscosity         (lb/(ft-sec))
C----------------------------------------------------------------------
      GMWH    =  1.0079
      GMWO    =  15.9994
      IF(IG(17).EQ.1) THEN
        CONST2  =  6894.757
        CONST3  =  4186.0
        CONST4  =  2.2046
        CONST5  =  1.488
```

```
          RGAS     =  8314.32
        ELSE
          CONST2   =  6894.757*10.
          CONST3   =  4186.0*10000.
          CONST4   =  2.2046/1000.
          CONST5   =  1.488*10.
          RGAS     =  8314.32*10000.
        ENDIF
        PRESIN   =  PRESIN*CONST2
        SM0(1)   =  1./(1.+FMIX)
        SM0(2)   =  1.-SM0(1)
        SM0(3)   =  0.0
        ENTHH2   =  ENTHH2*CONST3/(2.*GMWH)
        ENTHO2   =  ENTHO2*CONST3/(2.*GMWO)
        ENTHMX   =  ENTHH2*SM0(1)+ENTHO2*SM0(2)
        IF(NFO.GE.1) WRITE(6,950) ENTHH2,ENTHO2,ENTHMX
C-pd---Calculate inlet molar concentrations (moles/kg)---------------
        SC(1)=SM0(1)/(2.*GMWH)
        SC(2)=SM0(2)/(2.*GMWO)
        SC(3)=SM0(3)/(2.*GMWH+GMWO)
C-pd---Calculate molar concentration assuming total combustion--------
        SC(1)=SC(1)-2.*SC(2)
        SC(3)=SC(3)+2.*SC(2)
        SC(2)=0.0
C-pd---Calculate mass fractions assuming total combustion-------------
        SMB(1)=SC(1)*(2.*GMWH)
        SMB(2)=SC(2)*(2.*GMWO)
        SMB(3)=SC(3)*(2.*GMWH+GMWO)
        TGUESS=4000.
C-pd---Call temper to calculate combustion temperature----------------
        CALL TEMPER(ENTHMX,TGUESS,CTEMP,CPDR,RGAS,SC,3,NFO)
        CMW=(SC(1)*2.*GMWH+SC(2)*2.*GMWO+SC(3)*(2.*GMWH+GMWO))/
       &(SC(1)+SC(2)+SC(3))
        RHOIN=PRESIN*CMW/(RGAS*CTEMP)
        RHOEX=6894.*CMW/(RGAS*1200.)
        IF(NFO.GE.1) WRITE(6,955) RHOIN,PRESIN,CMW,RGAS,CTEMP
        FRATE=FRATE/CONST4
        RHOVEL=FRATE/CROSSA
        WIN=RHOVEL/RHOIN
        IF(NFO.GE.1) WRITE(6,960) ROPEN,RT,CROSSA
        IF(NFO.GE.1) WRITE(6,965) FRATE,RHOVEL,WIN
  950 FORMAT(' H OF H2 O2 & MIX  = ',1P,3E11.3)
  955 FORMAT(' RHO P MW R T      = ',1P,5E11.3)
  960 FORMAT(' REQ RT CA         = ',1P,3E11.3)
  965 FORMAT(' MDOT RHO*W W      = ',1P,3E11.3)
C-pd---Other properties-------------------------------------------
        AVISC=2.9E-5
        AVISC=AVISC*CONST5
        IF(IG(3).LE.2) ENUL=GRND
        IF(IG(3).EQ.3) ENUL=0.0
        RHO1=GRND
        DRH1DP=GRND
        PRNDTL(H1)=.7
        PRT(H1)=.88
C-pd---Two phase calculations-------------------------------------
```

```
C---    The two phase quantities are defined as follows --------------
C---          RHO2    --> density of liquid oxygen         (kg/cu m)
C---          DIAJ1   --> inner jet diameter w/o wall            (m)
C---          DIAJ2   --> inner jet diameter w/ wall             (m)
C---          DIAJ3   --> outer jet diameter w/o wall            (m)
C---          TNOJET  --> total number of jets
C---          AREAJ1  --> flow area of oxygen                 (sq m)
C---          AREAXX  --> inner area unavail. for H2 flow     (sq m)
C---          AREAJ2  --> flow area of hydrogen               (sq m)
C---          XMDOTH  --> flow rate of hydrogen             (kg/sec)
C---          XMDOTO  --> flow rate of oxygen               (kg/sec)
C---          PEROX   --> % of reacted O2 at inlet
C---          XMDCOM  --> flow rate of reacted oxygen       (kg/sec)
C---          XTEMP   --> guess for inlet temperature            (k)
C---          XMDHOT  --> H2 flow rate in one jet           (kg/sec)
C---          CMW2    --> MW of inlet mixture
C---          XMDCLD  --> O2 flow rate in one jet           (kg/sec)
C---          VELH2   --> velocity of hydrogen               (m/sec)
C---          VELO2   --> velocity of oxygen                 (m/sec)
C---          STEN    --> surface tension of oxygen            (N/m)
C---          VISXY   --> viscosity of liquid oxygen (kg/(m-sec))
C---          CABS    --> factor used in stripping rate
C---          DSC     --> factor used in droplet diameter
C---          DROPDI  --> initial droplet diameter               (m)
C---------------------------------------------------------------------
      IF(IG(9).EQ.2) THEN
      RHO2=1275.19
      DIAJ1=4.7752E-3
      DIAJ2=5.8420E-3
      DIAJ3=8.8392E-3
      TNOJET=600.
      AREAJ1=PI*DIAJ1*DIAJ1/4.
      AREAXX=PI*DIAJ2*DIAJ2/4.
      AREAJ2=(PI*DIAJ3*DIAJ3/4.)-AREAXX
C-pd---Calculate flow rates-------------------------------------------
      XMDOTH=SM0(1)*FRATE
      XMDOTO=SM0(2)*FRATE
      PEROX=.1152
      XMDCOM=XMDOTO*PEROX
C-pd---Calculate inlet mass fractions---------------------------------
      SM0(1)=XMDOTH/(XMDOTH+XMDCOM)
      SM0(2)=1.-SM0(1)
      SM0(3)=0.0
C-pd---Calculate inlet molar concentrations (moles/kg)----------------
      SC(1)=SM0(1)/(2.*GMWH)
      SC(2)=SM0(2)/(2.*GMWO)
      SC(3)=SM0(3)/(2.*GMWH+GMWO)
C-pd---Calculate molar concentration assuming total combustion--------
      SC(1)=SC(1)-2.*SC(2)
      SC(3)=SC(3)+2.*SC(2)
      SC(2)=0.0
C-pd---Call temper to calculate combustion temperature----------------
      ENTMX2=ENTHH2*SM0(1)+ENTHO2*SM0(2)
      TGUESS=1500.
      IF(NFO.GE.1) WRITE(6,950) ENTHH2,ENTHO2,ENTMX2
```

```
      CALL TEMPER(ENTMX2,TGUESS,XTEMP,CPDR,RGAS,SC,3,NFO)
C-pd---Calculate density and flow rates-------------------------------
      CMW2=(SC(1)*2.*GMWH+SC(2)*2.*GMWO+SC(3)*(2.*GMWH+GMWO))/
     &(SC(1)+SC(2)+SC(3))
      RHOIN=PRESIN*CMW2/(RGAS*XTEMP)
      IF(NFO.GE.1) WRITE(6,955) RHOIN,PRESIN,CMW2,RGAS,XTEMP
      XMDHOT=(XMDOTH+XMDCOM)/TNOJET
      XMDCLD=(XMDOTO-XMDCOM)/TNOJET
      VELH2=XMDHOT/RHOIN/AREAJ2
      VELO2=XMDCLD/RHO2/AREAJ1
      XMDHOT=(XMDOTH+XMDCOM)/GPI/ASUM
      XMDCLD=(XMDOTO-XMDCOM)/GPI/ASUM
      DO 970 IY=1,NY
      DO 970 JJ=31,50
      IF(IG(JJ).EQ.IY) THEN
        RG(JJ+50)=XMDCLD*ACELL(IY)
      ENDIF
  970 CONTINUE
C-pd---Physical properties used to calculate a drop diameter----------
      STEN=.001
      VISXY=3.E-4
      CABS=.037854
      DSC=3.0553
      TERM1=VISXY*((STEN/RHO2)**.5)
      TERM2=RHOIN*((VELH2-VELO2)**2)
      DROPDI=DSC*((TERM1/TERM2)**.6666666)
      DROPDI=DIAJ1/20.
      ENDIF
C*********************************************************************
C-pd---This information is passed into ground from satellite ---------
C---         PRESIN --> total pressure                        (n/sq m)
C---         FRATE  --> flow rate                               (kg/s)
C---         ENTHMX --> enthalpy in                             (j/kg)
C---         CTEMP  --> combustion temperature                     (k)
C---         RGAS   --> gas constant             (n-m/(deg K-kg mole))
C---         RT     --> radius throat                              (m)
C---         CMW    --> combusted mixture molecular weight
C---         AVISC  --> viscosity                          (kg/(m-sec))
C---         CONST2 --> converts psi to N/sq m
C---         FTS    --> false time step                          (sec)
C---         DROPDI --> two phase droplet diameter                 (m)
C---         VELO2  --> velocity of oxygen                     (m/sec)
C---         XMDCLD --> O2 flow rate in one jet               (kg/sec)
C---         STEN   --> surface tension of oxygen                (N/m)
C---         VISXY  --> viscosity of liquid oxygen         (kg/(m-sec))
C---         CABS   --> factor used in stripping rate
C---         DIAJ1  --> inlet oxygen diameter                      (m)
C---         EHTHO2 --> enthalpy of oxygen                      (j/kg)
C---         CONCOP --> Thermal conductivity of copper         (W/K-m)
C---         CONSTE --> Thermal conductivity of steel          (W/K-m)
C---         FLXINL --> Energy rate at inlet for lower           (J/s)
C---                   cooling jacket
C---         FLXINU --> Energy rate at inlet for upper           (J/s)
C---                   cooling jacket
C---         PRHYD  --> Prandtl No. for liquid hydrogen
```

```
C---          RATEL  -->  Mass flow at inlet for lower          (Kg/s)
C---                     cooling jacket
C---          RATEU  -->  Mass flow at inlet for upper          (Kg/s)
C---                     cooling jacket
C---          TLIQL  -->  Temperature of hydrogen at inlet        (K)
C---          TLIQU  -->  Temperature of hydrogen at inlet        (K)
C---          VISHYD -->  Dynamic viscosity of hydrogen        (Kg/m-s)
C--------------------------------------------------------------------
      RG(8) =PRESIN
      RG(9) =FRATE
      RG(21)=ENTHMX
      RG(22)=CTEMP
      RG(23)=RGAS
      RG(24)=RT
      RG(28)=CMW
      RG(29)=AVISC
      RG(30)=CONST2
      RG(31)=FTS
      IF(IG(9).EQ.2) THEN
        RG(32)=DROPDI
        RG(33)=VELO2
        RG(34)=XMDCLD
        RG(35)=STEN
        RG(36)=VISXY
        RG(37)=CABS
        RG(38)=DIAJ1
        RG(6)=ENTHO2
      ENDIF
C-IWC--- Cooling jacket data passed to ground from satellite ----------
      IF(IG(8).EQ.3) THEN
        IG(25)=NCHA
        IG(26)=NTUB
        IG(27)=NCOM
        IG(28)=NNOZ
        RG(44)=FLXINL
        RG(45)=FLXINU
        RG(46)=CONCOP
        RG(47)=CONSTE
        RG(48)=PRHYD
        RG(49)=RATEL
        RG(50)=RATEU
        RG(51)=TLIQL
        RG(52)=TLIQU
        RG(53)=VISHYD
      ENDIF
C---------------------------------------------------------------------
C*********************************************************************
      RETURN
C
C--- GROUP 10. Inter-phase-transfer processes and properties
   10 CONTINUE
      IF(IG(9).EQ.2) CFIPS=GRND
      IF(IG(9).EQ.2) CMDOT=GRND
      RETURN
C
```

```
C--- GROUP 11. Initialization of variable or porosity fields
   11 CONTINUE
      TURBLV=.1
      SLENTH=.02*ROPEN
      TKIN=.5*(WIN*TURBLV)**2.
      EPIN=.1643*TKIN**1.5/SLENTH
      CALL PATCH('INITL   ',INIVAL,1,NX,1,NY,1,NZ,1,1)
      CALL INIT ('INITL   ',1   ,0.,GRND)
      CALL INIT ('INITL   ',7   ,0.,GRND)
      CALL INIT ('INITL   ',14  ,0.,GRND)
      CALL INIT ('INITL   ',25  ,0.,GRND)
      CALL INIT ('INITL   ',26  ,0.,GRND)
      IF(IG(9).EQ.2) THEN
      CALL PATCH('INITL1  ',INIVAL,1,NX,1,NY,1,NZT/2,1,1)
      CALL INIT ('INITL1  ',9   ,0.,0.98)
      CALL INIT ('INITL1  ',10  ,0.,0.02)
      CALL PATCH('INITL2  ',INIVAL,1,NX,1,NY,NZT/2+1,NZ,1,1)
      CALL INIT ('INITL2  ',9   ,0.,1.00)
      CALL INIT ('INITL2  ',10  ,0.,0.00)
      ENDIF
C     FIINIT (H1) = ENTHMX
      FIINIT (C1) = SM0(1)
      FIINIT (17) = SMB(1)
      FIINIT (19) = SMB(3)
      FIINIT (KE) = TKIN
      FIINIT (EP) = EPIN
      RETURN
C
C--- GROUP 12. Convection and diffusion adjustments
   12 CONTINUE
      RETURN
C
C--- GROUP 13. Boundary conditions and special sources
   13 CONTINUE
      IF(IG(3).GT.1) GOTO 1310
      KELIN=2
      CALL PATCH('KESOURCE',PHASEM,1,1,1,NY,1,NZ,1,1)
      CALL COVAL('KESOURCE',KE,GRND4,GRND4)
      CALL COVAL('KESOURCE',EP,GRND4,GRND4)
 1310 IF(IG(9).EQ.2) GOTO 1316
      IF(IG(6).NE.1) GOTO 1315
      CALL PATCH('INLET   ',LOW,1,1,IYBOT,IYTOP,1,1,1,1)
      CALL COVAL('INLET   ',P1,FIXFLU,RHOVEL)
      CALL COVAL('INLET   ',W1,ONLYMS,WIN)
      CALL COVAL('INLET   ',H1,ONLYMS,ENTHMX)
      CALL COVAL('INLET   ',C1,ONLYMS,SM0(1))
      CALL COVAL('INLET   ',KE,ONLYMS,TKIN)
      CALL COVAL('INLET   ',EP,ONLYMS,EPIN)
 1315 IF(IG(6).NE.2) GOTO 1320
      CALL PATCH('INLET   ',LOW,1,1,IYBOT,IYTOP,1,1,1,1)
      CALL COVAL('INLET   ',P1,0.1,PRESIN)
      CALL COVAL('INLET   ',W1,ONLYMS,WIN)
      CALL COVAL('INLET   ',H1,ONLYMS,ENTHMX)
      CALL COVAL('INLET   ',C1,ONLYMS,SM0(1))
      CALL COVAL('INLET   ',KE,ONLYMS,TKIN)
```

```
      CALL COVAL('INLET   ',EP,ONLYMS,EPIN)
      GOTO 1320
 1316 DO 1317 II=1,NJETS
      I10=II/10
      I1=II-I10*10
      WRITE(GNAME(7:7),'(I1)') I10
      WRITE(GNAME(8:8),'(I1)') I1
      CALL PATCH(GNAME,LOW,1,NX,IG(30+II),IG(30+II),1,1,1,1)
      CALL COVAL(GNAME,P1,1.E-15,XMDHOT*1.E15)
C
C     CALL COVAL(GNAME,P2,1.E-15,XMDCLD*1.E15)
C     CALL COVAL(GNAME,W2,ONLYMS,VELO2)
C
      CALL COVAL(GNAME,W1,ONLYMS,VELH2)
      CALL COVAL(GNAME,H1,ONLYMS,ENTMX2)
      CALL COVAL(GNAME,C1,ONLYMS,SM0(1))
      CALL COVAL(GNAME,KE,ONLYMS,TKIN)
 1317 CALL COVAL(GNAME,EP,ONLYMS,EPIN)
      CALL PATCH('CONSOR  ',CELL,1,NX,1,NY,1,NZ,1,1)
      CALL COVAL('CONSOR  ',H1,FIXFLU,GRND7)
      CALL COVAL('CONSOR  ',C1,GRND7,0.)
      CALL PATCH('JETSOR  ',CELL,1,NX,1,NY,1,NZT/2,1,1)
      CALL COVAL('JETSOR  ',P2,1.E-15,GRND8)
      CALL COVAL('JETSOR  ',W2,ONLYMS,VELO2)
 1320 IF(IG(4).NE.1) GOTO 1325
      IF(IG(3).EQ.3) GOTO 1330
      CALL PATCH('WALL    ',NWALL,1,NX,NY,NY,1,NZ,1,1)
      CALL COVAL('WALL    ',W1,GRND2,0.0)
      CALL COVAL('WALL    ',KE,GRND2,GRND2)
      CALL COVAL('WALL    ',EP,GRND2,GRND2)
      IF(IG(8).EQ.1) GOTO 1325
      CALL COVAL('WALL    ',H1,GRND2,GRND)
 1325 IF(IG(4).NE.2) GOTO 1330
      IF(IG(3).EQ.3) GOTO 1330
      CALL PATCH('MYWALL  ',NORTH,1,NX,NY,NY,1,NZ,1,1)
      CALL COVAL('MYWALL  ',W1,GRND,0.0)
      CALL COVAL('MYWALL  ',KE,GRND,GRND)
      CALL COVAL('MYWALL  ',EP,GRND,GRND)
      IF(IG(8).EQ.1) GOTO 1330
      CALL PATCH('WALL    ',NWALL,1,NX,NY,NY,1,NZ,1,1)
      CALL COVAL('WALL    ',H1,GRND2,GRND)
 1330 IF(IG(4).NE.3) GOTO 1335
      CALL PATCH('FAKEWALL',NORTH,1,NX,NY,NY,1,NZ,1,1)
 1335 CALL PATCH('FIXDEN  ',CELL,1,NX,1,NY,1,NZ,1,1)
      CALL COVAL('FIXDEN  ',V1,GRND,0.0)
      CALL COVAL('FIXDEN  ',W1,GRND,0.0)
      IF(IG(7).NE.1) GOTO 1340
      CALL PATCH('OUTLET  ',HIGH,1,1,1,NY,NZ,NZ,1,1)
      CALL COVAL('OUTLET  ',P1,FIXFLU,GRND)
      IF(IG(9).EQ.2) CALL COVAL('OUTLET  ',P2,FIXFLU,GRND)
      CALL COVAL('OUTLET  ',H1,ONLYMS,SAME)
      CALL COVAL('OUTLET  ',C1,ONLYMS,SAME)
 1340 IF(IG(7).NE.2) RETURN
      CALL PATCH('OUTLET  ',HIGH,1,1,1,NY,NZ,NZ,1,1)
      CALL COVAL('OUTLET  ',P1,0.1*RHOEX,GRND)
```

```
      IF(IG(9).EQ.2) CALL COVAL('OUTLET  ',P2,0.1*RHO2,GRND)
      CALL COVAL('OUTLET  ',H1,ONLYMS,SAME)
      CALL COVAL('OUTLET  ',C1,ONLYMS,SAME)
      RETURN
C
C--- GROUP 14. Downstream pressure for PARAB=.TRUE.
   14 CONTINUE
      RETURN
C
C--- GROUP 15. Termination of sweeps
   15 CONTINUE
      RETURN
C
C--- GROUP 16. Termination of iterations
   16 CONTINUE
      LITER(P1)=25
      ENDIT(P1)=0.01
      RETURN
C
C--- GROUP 17. Under-relaxation devices
   17 CONTINUE
      RETURN
C
C--- GROUP 18. Limits on variables or increments to them
   18 CONTINUE
      VARMAX(P1)=PRESIN*1.5
      VARMAX(EP)=1.0E+15*SFAC
      VARMAX(KE)=1.0E+10*SFAC
      RETURN
C
C--- GROUP 19. Data communicated by satellite to GROUND
   19 CONTINUE
      IF(IG(3).EQ.1) GENK=.TRUE.
      IF(NFO.GE.1) WRITE(6,'('' SATELLITE HAS JUST RUN '')')
      RETURN
C
C--- GROUP 20. Preliminary print-out
   20 CONTINUE
      RETURN
C
C--- GROUP 21. Print-out of variables
   21 CONTINUE
      RETURN
C
C--- GROUP 22. Spot-value print-out
   22 CONTINUE
      RETURN
C
C--- GROUP 23. Field print-out and plot control
   23 CONTINUE
      RETURN
C
C--- GROUP 24. Dumps for restarts
   24 CONTINUE
      RETURN
```

```
      END
C***********************************************************************
      SUBROUTINE ENTHAL(TEMP,HSUM,CPSUM,SC,NS,NFO)
C***********************************************************************
      DIMENSION SC(NS),ZS(7,2,3)
      DATA ZS/3.1,5.112E-4,5.264E-8,-3.491E-11,
     &        3.695E-15,-8.774E2,-1.963,3.057,2.667E-3,-5.81E-6,
     &        5.521E-9,-1.812E-12,-9.889E2,-2.3,3.622,7.362E-4,
     &       -1.965E-7,3.620E-11,-2.895E-15,-1.202E3,3.615,3.626,
     &       -1.878E-3,7.055E-6,-6.764E-9,2.156E-12,-1.048E3,4.305,
     &        2.717,2.945E-3,-8.022E-7,1.023E-10,-4.847E-15,-2.991E4,
     &        6.631,4.07,-1.108E-3,4.152E-6,-2.964E-9,8.07E-13,
     &       -3.028E4,-3.227E-1/
      K=1
      IF(TEMP.LT.1000.) K=2
      TEMP2=TEMP*TEMP
      HSUM=0.
      CPSUM=0.
      DO 100 IS=1,NS
      CP1=ZS(1,K,IS)
      CP2=ZS(2,K,IS)*TEMP
      CP3=ZS(3,K,IS)*TEMP2
      CP4=ZS(4,K,IS)*TEMP2*TEMP
      CP5=ZS(5,K,IS)*TEMP2*TEMP2
      CPSUM=CPSUM+SC(IS)*(CP1+CP2+CP3+CP4+CP5)
  100 HSUM =HSUM+
     1  SC(IS)*(CP1+.5*CP2+.33333*CP3+.25*CP4+.2*CP5+ZS(6,K,IS)/TEMP)
      RETURN
      END
C***********************************************************************
      SUBROUTINE TEMPER(HSTAT,T0,T,CPDR,RGAS,SC,NSC,NFO)
C***********************************************************************
C---------- SUBITERATIVE CALCULATION OF TEMPERATURE ---------------
      DIMENSION SC(NSC)
      DATA NITER,DT0,TMIN/12,50.,12.345/
      DT=DT0
      TEMP=T0
      CALL ENTHAL(TEMP,HHH,CPDR,SC,NSC,NFO)
      ENTH=HHH*RGAS*TEMP
      IF(HSTAT.LT.ENTH) DT=-DT
      TEMPL=TEMP
      IF(NFO.GE.4) WRITE(6,900) T0,ENTH,HSTAT,RGAS,SC(1),SC(2),SC(3)
      TEMP =TEMP+DT
      ITER=0
  100 ENTHL=ENTH
      ITER=ITER+1
      CALL ENTHAL(TEMP,HHH,CPDR,SC,NSC,NFO)
      ENTH =HHH*RGAS*TEMP
      RENTH=(HSTAT-ENTHL)/((ENTH-ENTHL) +1.E-9)
      IF(NFO.GE.4) WRITE(6,910) ITER,TEMP,ENTH,ENTHL,HSTAT,RENTH
      IF(ABS(ENTH-ENTHL).LT..001*ABS(ENTH)) RENTH=1.
      TEMP1=TEMPL+(TEMP-TEMPL)*RENTH
      TEMP1=AMAX1(TEMP1,.5*TEMP,TMIN)
      TEMP1=AMIN1(TEMP1,1.5*TEMP,5000.)
      TEMPL=TEMP
```

```
      TEMP=TEMP1
      AR=ABS(RENTH)
      IF( (AR.GT.1.005 .OR. AR.LT..995) .AND. ITER.LT.NITER) GO TO 100
      T=TEMP
      RETURN
  900 FORMAT(' TO E HS RG SC',1P,7E12.4)
  910 FORMAT(' IT T E EL HS RE',I3,1P,5E12.4)
      END
C***********************************************************************
      SUBROUTINE GSCALE(GFACT)
C***********************************************************************
Cinclude "satear"
C FILE NAME SATEAR --- 170486
CNLIST
C
C---------------------------------------------------- ARRAYS
      COMMON/LDB1/DBGPHI(50)/IDA1/ITERMS(50)/IDA2/LITER(50)
     1/IDA3/IORCVF(50)/IDA4/IORCVL(50)/IDA5/ISLN(50)/IDA6/IPRN(50)
     1/HDA1/NAME(50)/RDA1/DTFALS(50)/RDA2/RESREF(50)
     1/RDA3/PRNDTL(50)/RDA4/PRT(50)/RDA5/ENDIT(50)/RDA6/VARMIN(50)
     1/RDA7/VARMAX(50)/RDA8/FIINIT(50)/RDA9/PHINT(50)
     1/RDA10/CINT(50)/RDA11/EX(50)
C---------------------------------------------------- LDATA
      COMMON/LDATA/CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32(17),SAVGEO,
     1RSTGEO,NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,
     1DONACC,PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,
     1USEGRD,USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP(11)
C
C----------------------------------------------------- LDEBUG
      COMMON/LDEBUG/DBGEOM,DBADJS,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37(9)
C
C------------------------------------------------------ IDATA
      COMMON/IDATA/NX,NY,NZ,LUPR1,LUPR2,LUPR3,LUPHUN,LUSDA,IPROF,
     1LUFI,LUDST,LUGRF,LUSAVE,LUOLD,LUDEP,LUPCO,LUDVL,
     1IRUNN,IOPTN,LITC,LITFLX,NRUN,LITHYD,FSTEP,LSTEP,
     1FSWEEP,LSWEEP,NPRINT,LIBREF,MEANDF,IXMON,IYMON,IZMON,IINIT,
     1NLSG1,NISG1,NRSG1,NCSG1,IPARAB,IDPHUN,NXFR1,NYFR1,NZFR1,
     1NTFR1,ENTH1,ENTH2,ISWR1,ISWR2,IXPRF,IXPRL,IYPRF,IYPRL,
     1NPRMNT,ISTPRL,ISTPRF,IZPRL,IZPRF,NUMCLS,TSTSWP,NYPRIN,NXPRIN,
     1NZPRIN,NPRMON,NTPRIN,NTZPRF,ISP66,IURINI,IURPRN,IURVAL,
     1IORTCV,NUMREG,NRTCV,ICHR,INTFRC,ITHC1,ISWC1,DEN1,DEN2,
     1VISL,INTMDT,ISWPRF,ISWPRL,IPSA,ISP84,IPLTF,IPLTL,NPLT,ITABL,
     1TEMP1,TEMP2,LEN1,LEN2,NLG1,NIG1,NRG1,NCG1,NPNAM1,
     1ISP98(3),LENREC,LUGEOM,IMB1,IMB2,PCOR,NCOLPF,NCOLCO,
     1NROWCO,EPOR,NPOR,HPOR,VPOR,KXFR,KYFR,KZFR,KTFR,IDATSP(2),
     1VIST,NPHI
C
C------------------------------------------------------ IDEBUG
```

```
      COMMON/IDEBUG/IZDB1,IZDB2,ITHDB1,ITHDB2,ISWDB1,ISWDB2,ISTDB1,
     1ISTDB2,INCHCK,IREGDB,NFMAX,IDBF0,IDBCMN,IDBGRD,IDEBSP(2)
C
C------------------------------------------------------------ HDATA
      COMMON/HDATA/MESS(10),NBLANK,NAMGRD,NAMEJ,NAMEJ1,
     1NAMEM,NAMEM1,NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,
     1NPHUN,NHINIT,NDST,NAMSAT,NGEOM,NHDASP(2)
C
C------------------------------------------------------------ HDEBUG
      COMMON/HDEBUG/NDBF0(2),NDBCMN(2),NHDBSP
C
C------------------------------------------------------------ RDATA
      COMMON/RDATA/TINY,GREAT,RUPLIM,RLOLIM,AZDZ,AZXU,AZYV,
     1AZRI,AZAL,AZPH,XULAST,YVLAST,ZWLAST,TLAST,TFIRST,PBAR,SNALFA,
     1RINNER,ENUL,ENUT,RHO1,RHO2,CFIPS,CMDOT,CONMDT,GRND,HEATBL,
     1FIXFLU,READFI,ZMOVE1,ZDIFAC,DRH1DP,DRH2DP,U1AD,U2AD,V1AD,
     1V2AD,W1AD,W2AD,HUNIT,DIFCUT,ABSIZ,ORSIZ,OPPVAL,TMP1,TMP2,
     1EL1,EL2,GRND1,GRND2,GRND3,GRND4,GRND5,GRND6,GRND7,GRND8,GRND9
     1,GRND10,ZWADD,RINIT,SAME,FIXVAL,AXDZ,AYDZ,RDATSP(21)
C
C------------------------------------------------------------ RDEBUG
      COMMON/RDEBUG/BGCHCK,SMCHCK,RDEBSP(5)
C
C----------------------------------LOGICAL DECLARATIONS
      LOGICAL LDAT,LDEB
      LOGICAL CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32,SAVGEO,RSTGEO,
     1NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,DONACC,
     1PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,USEGRD,
     1USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP
      LOGICAL DBGEOM,DBADJS,DBGPHI,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37
C
C----------------------------------INTEGER DECLARATIONS
      INTEGER FSTEP,FSWEEP,TSTSWP,ENTH1,ENTH2,DEN1,
     1DEN2,PCOR,VISL,EPOR,HPOR,VPOR,VIST,TEMP1,TEMP2
C----------------------------------CHARACTER DECLARATIONS
      CHARACTER*4 NHDAT,NHDEB
      CHARACTER*4 NAME
      CHARACTER*4 MESS,NBLANK,NAMGRD,NAMEJ,NAMEJ1,NAMEM,NAMEM1,
     1NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,NPHUN,NHINIT,
     1NDST,NAMSAT,NGEOM,NHDASP
      CHARACTER*4 NDBF0,NDBCMN,NHDBSP
C----------------------------------EQUIVALENT TRANSMISSION ARRAYS
      DIMENSION LDAT(84),LDEB(45),IDAT(120),IDEB(16),NHDAT(30),
     1NHDEB(5),RDAT(85),RDEB(7)
      EQUIVALENCE (LDAT(1),CARTES),(LDEB(1),DBGEOM),(IDAT(1),NX),
     1(IDEB(1),IZDB1),(NHDAT(1),MESS(1)),(NHDEB(1),NDBF0(1)),
     1(RDAT(1),TINY),(RDEB(1),BGCHCK)
CLIST
```

```
#include "satloc"
#include "bfcsat"
      COMMON F(1)
      NI=NX+1
      NJ=NY+1
      NK=NZ+1
      JNNN=NI*NJ*NK
      CALL SCALEW(F(KXC+1),F(KYC+1),F(KZC+1),GFACT,JNNN)
      RETURN
      END
C***********************************************************************
      SUBROUTINE SCALEW(X,Y,Z,F,N)
C***********************************************************************
      DIMENSION X(*),Y(*),Z(*)
      DO 1 I=1,N
      X(I)=X(I)*F
      Y(I)=Y(I)*F
    1 Z(I)=Z(I)*F
      RETURN
      END
C***********************************************************************
      SUBROUTINE XSLP(X,Y,N,S,IC)
C***********************************************************************
C-pd---This subroutine is taken from the TDK program     For further -
C---    info contract KLAUS GROSS at MARSHALL SPACE FLIGHT CENTER -----
C
C COMPUTES SLOPES FOR CUBIC CHAIN FIT TO PLANAR SET OF DATA,
C INPUT DATA--
C (1) (X,Y)-ARRAY OF N POINTS.
C OUTPUT DATA--ARRAY OF SLOPES,S.
C
      DIMENSION DX(50),S(50),V(50),W(50),X(50),Y(50)
C
      NM1=N-1
      NM2=N-2
      NM3=N-3
      IF (IC. NE. 0)  GO TO 6
C     DEFINE FIRST AND LAST SLOPES
      V(1) = X(1)**2
      V(2) = X(2)**2
      V(3) = X(3)**2
      W(1) = X(2) - X(3)
      W(2) = X(3) - X(1)
      W(3) = X(1) - X(2)
      DX(1) = X(1)*(Y(1)*W(1) + Y(2)*W(2) + Y(3)*W(3))
      S(1) = (2.0*DX(1) + V(1)*(Y(2) - Y(3)) + V(2)*(Y(3) - Y(1)) + V(3)
     1       *(Y(1) - Y(2)))/(V(1)*W(1) + V(2)*W(2) + V(3)*W(3))
      V(1) = X(NM2)**2
      V(2) = X(NM1)**2
      V(3) = X(N)**2
      W(1) = X(NM1) - X(N)
      W(2) = X(N) - X(NM2)
      W(3) = X(NM2) - X(NM1)
      DX(1) = X(N)*(Y(NM2)*W(1) + Y(NM1)*W(2) + Y(N)*W(3))
      S(N) = (2.0*DX(1) + V(1)*(Y(NM1) - Y(N)) + V(2)*(Y(N) - Y(NM2)) +
```

```
     1       V(3)*(Y(NM2) - Y(NM1)))/(V(1)*W(1) + V(2)*W(2) + V(3)*W(3))
C VARIABLE SPACING, COMPUTE SUBINTERVAL ARRAY.
  6    DO 7 I = 1,NM1
 7     DX(I)=X(I+1)-X(I)
C INITIALIZE V(1).
       V(1) = DX(1)/(DX(1) + DX(2))*0.50
       IF (NM3 .LE. 0)  GO TO 309
C COMPUTE REST OF UPPER DIAGONAL TERMS,V(I).
       DO 509 I = 2,NM2
  509 V(I)=DX(I)/(2.*(DX(I)+DX(I+1))-DX(I+1)*V(I-1))
C INITIALIZE W(1).
  309 W(1)=V(1)/DX(1)*(3./DX(1)/DX(2)*(DX(1)*DX(1)*Y(3)
     1+(DX(2)*DX(2)-DX(1)*DX(1))*Y(2)-DX(2)*DX(2)*Y(1))
     2-DX(2)*S(1))
       IF(NM3 .LE. 0)  GO TO 209
       IF (N .EQ. 4)  GO TO 19
       DO 809 I = 2,NM3
C CALCULATE REMAINING TERMS,W(I), IN CONSTANT MATRIX.
  809 W(I)=V(I)/DX(I)*(3./DX(I)/DX(I+1)*(DX(I)*DX(I)*Y(I+2)
     1+(DX(I+1)*DX(I+1)-DX(I)*DX(I))*Y(I+1)-DX(I+1)*DX(I+1)*Y(I))
     2-DX(I+1)*W(I-1))
   19 W(NM2)=V(NM2)/DX(NM2)*(3./DX(NM2)/DX(NM1)*(DX(NM2)*DX(NM2)
     1*Y(N)+(DX(NM1)*DX(NM1)-DX(NM2)*DX(NM2))*Y(NM1)
     2-DX(NM1)*DX(NM1)*Y(NM2))-DX(NM2)*S(N)-DX(NM1)*W(NM3))
C COMPUTE SOLUTION SLOPES.
 209   S(NM1)=W(NM2)
       IF (NM3 .LE. 0)  RETURN
       DO 119 I = 1,NM3
 119   S(N-I-1) = W(N-I-2) - V(N-I-2)*S(N-I)
       RETURN
       END
C*********************************************************************
      SUBROUTINE GEOMTX(X,Y,Z,N,NZT,IYF,IYL,NY,NZ,WT,ROPEN,FTS,RT,CA,
     &                  GPI,ACELL)
C*********************************************************************
C-pd---This sub calculates an equivalent radius (ROPEN), the cross ---
C---   sectional open inlet area (CROSSA), a false time step (FTS) ---
C---   and the radius at the throat (RT) -----------------------------
      DIMENSION X(*),Y(*),Z(*),ACELL(*)
      TLENGT=Z(N)-Z(1)
      ADZ=TLENGT/NZ
      FTS=ADZ/WT
      DX1=X(IYF+NY+1)-X(IYF)
      DX2=X(IYL+1+NY+1)-X(IYL+1)
      DY=Y(IYL+1)-Y(IYF)
      DZ=Z(IYL+1)-Z(IYF)
      ROPEN=(DY**2+DZ**2)**.5
      HGT=(ROPEN**2-(DX2/2.)**2)**.5
      CA=(DX1+DX2)/2.*HGT*GPI
      RT=Y(NZT)
      DO 100 II=1,NY
      DX1=X(II+NY+1)-X(II)
      DX2=X(II+NY+2)-X(II+1)
      DY=Y(II+1)-Y(II)
      DZ=Z(II+1)-Z(II)
```

```
      SIDLN=(DY**2+DZ**2)**.5
      BASLN=(DX2-DX1)/2.
      HGT=(SIDLN**2-BASLN**2)**.5
      ACELL(II)=(DX1+DX2)/2.*HGT
  100 CONTINUE
      RETURN
      END
```

APPENDIX E

GROUND Program

LISTING E.1   GROUND FILE

```
      PROGRAM MAIN
C-pd---The following parameters have size limitations and may need ---
C---   increasing as your grid becomes larger.  The number in --------
C---   () is the number of times that parameter occurs in ground -----
C---          NFPAR  -->  now set at 600000    (1)
C---          JNX    -->  now set at 1         (1)
C---          JNY    -->  now set at 50        (1)
C---          JNZ    -->  now set at 300       (1)
C---          NXM    -->  now set at 1         (3)
C---          NYM    -->  now set at 50        (3)
C---          NZM    -->  now set at 300       (3)
C---------------------------------------------------------------------
C FILE NAME GROUND.FTN-------------------------------030386
C
C     PROGRAM MAIN
C
C 1   The following two COMMON's, which appear identically in the
C     satellite MAIN program, allow up to 25 dependent variables to
C     be solved for (or their storage spaces to be occupied by
C     other variables, such as density). If a larger number is
C     required, the 25's should be replaced, in the next 8 lines,
C     by the required larger number; and the 100 in COMMON/F01/
C     should be replaced by 4 times the required number. Numbers
C     less than 25 are not permitted.
C
      COMMON/LGE1/L1(50)/LGE2/L2(50)/LGE3/L3(50)/LGE4/L4(50)
     1/LDB1/L5(50)/IDA1/I1(50)/IDA2/I2(50)/IDA3/I3(50)/IDA4/I4(50)
     1/IDA5/I5(50)/IDA6/I6(50)/GI1/I7(50)/GI2/I8(50)/HDA1/IH1(50)
     1/GH1/IH2(50)/RDA1/R1(50)/RDA2/R2(50)/RDA3/R3(50)/RDA4/R4(50)
     1/RDA5/R5(50)/RDA6/R6(50)/RDA7/R7(50)/RDA8/R8(50)/RDA9/R9(50)
     1/RDA10/R10(50)/RDA11/R11(50)
     1/GR1/R12(50)/GR2/R13(50)/GR3/R14(50)/GR4/R15(50)
     1/IPIP1/IP1(50)/HPIP2/IHP2(50)/RPIP1/RVAL(50)/LPIP1/LVAL(50)
     1/IFPL/IPL0(50)/RFPL1/ORPRIN(50)/RFPL2/ORMAX(50)
     1/RFPL3/ORMIN(50)/RFPL4/CELAV(50)
      LOGICAL L1,L2,L3,L4,L5,DBGFIL,LVAL
      CHARACTER*4 IH1,IH2,IHP2,NSDA
C
      COMMON/F01/I9(200)
      COMMON/DISC/DBGFIL
      EXTERNAL WAYOUT
C
C 2   Set dimensions of data-for-GROUND arrays here.
      COMMON/LGRND/LG(20)/IGRND/IG(50)/RGRND/RG(100)/CGRND/CG(10)
      LOGICAL LG
      CHARACTER*4 CG
C
C 3   Set dimensions of data-for-GREX1 arrays here.
      COMMON/LSG/LSGD(20)/ISG/ISGD(20)/RSG/RSGD(100)/CSG/CSGD(10)
      LOGICAL LSGD
```

```
      CHARACTER*4 CSGD
C
C 4   Set dimension of patch-name array here.
      COMMON/NPAT/NAMPAT(100)
      CHARACTER*8 NAMPAT
C
C     Declare local CHARACTER varaibles.
      CHARACTER NDUM4*4,NDUM6*6,NDUM15*15
C
C 5   The numbers in the next two statements (which must be ident-
C     ical) indicate how much computer memory is to be set aside
C     for storing the main and auxiliary variables. The user may
C     alter them if he wishes, to accord with the number of
C     grid nodes and dependent variables he is concerned with.
C=====================================================================
      PARAMETER (NFPAR=600000)
      COMMON F(NFPAR)
      NFDIM=NFPAR
C=====================================================================
C
C 6   Logical-unit numbers and file names, not to be changed.
      DBGFIL=.FALSE.
      CALL DSCEAR(14,LUPR3,'  ',15,NDUM15,-11,16)
      CALL DSCEAR(6,LUDUM,'  ',4,NDUM4,9,33)
      CALL DSCEAR(-10,LUSDA,'  ',4,NSDA,0,0)
      CALL DSCEAR(-14,LUPR1,'  ',15,NDUM15,0,0)
      CALL DSCEAR(21,LUDST,'  ',4,NDUM4,9,33)
C
C     User may here change message transmitted to logical unit
C     LUPR3
      CALL WRIT40('GROUND STATION IS FOR NOZZLE FLOWS      ')
      CALL MAIN1(NFDIM,LUPR1,LUPR3,LUSDA,NSDA)
      CALL WAYOUT(0)
      STOP
      END
C*********************************************************************
      SUBROUTINE GROSTA
Cinclude "satear"
C FILE NAME SATEAR --- 170486
CNLIST
C
C------------------------------------------------------------- ARRAYS
      COMMON/LDB1/DBGPHI(50)/IDA1/ITERMS(50)/IDA2/LITER(50)
     1/IDA3/I0RCVF(50)/IDA4/I0RCVL(50)/IDA5/ISLN(50)/IDA6/IPRN(50)
     1/HDA1/NAME(50)/RDA1/DTFALS(50)/RDA2/RESREF(50)
     1/RDA3/PRNDTL(50)/RDA4/PRT(50)/RDA5/ENDIT(50)/RDA6/VARMIN(50)
     1/RDA7/VARMAX(50)/RDA8/FIINIT(50)/RDA9/PHINT(50)
     1/RDA10/CINT(50)/RDA11/EX(50)
C------------------------------------------------------------- LDATA
      COMMON/LDATA/CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32(17),SAVGEO,
     1RSTGEO,NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,
     1DONACC,PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,
```

```
     1USEGRD,USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP(11)
C
C---------------------------------------------------------- LDEBUG
      COMMON/LDEBUG/DBGEOM,DBADJS,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37(9)
C
C----------------------------------------------------------- IDATA
      COMMON/IDATA/NX,NY,NZ,LUPR1,LUPR2,LUPR3,LUPHUN,LUSDA,IPROF,
     1LUFI,LUDST,LUGRF,LUSAVE,LUOLD,LUDEP,LUPCO,LUDVL,
     1IRUNN,IOPTN,LITC,LITFLX,NRUN,LITHYD,FSTEP,LSTEP,
     1FSWEEP,LSWEEP,NPRINT,LIBREF,MEANDF,IXMON,IYMON,IZMON,IINIT,
     1NLSG1,NISG1,NRSG1,NCSG1,IPARAB,IDPHUN,NXFR1,NYFR1,NZFR1,
     1NTFR1,ENTH1,ENTH2,ISWR1,ISWR2,IXPRF,IXPRL,IYPRF,IYPRL,
     1NPRMNT,ISTPRL,ISTPRF,IZPRL,IZPRF,NUMCLS,TSTSWP,NYPRIN,NXPRIN,
     1NZPRIN,NPRMON,NTPRIN,NTZPRF,ISP66,IURINI,IURPRN,IURVAL,
     1IORTCV,NUMREG,NRTCV,ICHR,INTFRC,ITHC1,ISWC1,DEN1,DEN2,
     1VISL,INTMDT,ISWPRF,ISWPRL,IPSA,ISP84,IPLTF,IPLTL,NPLT,ITABL,
     1TEMP1,TEMP2,LEN1,LEN2,NLG1,NIG1,NRG1,NCG1,NPNAM1,
     1ISP98(3),LENREC,LUGEOM,IMB1,IMB2,PCOR,NCOLPF,NCOLCO,
     1NROWCO,EPOR,NPOR,HPOR,VPOR,KXFR,KYFR,KZFR,KTFR,IDATSP(2),
     1VIST,NPHI
C
C----------------------------------------------------------- IDEBUG
      COMMON/IDEBUG/IZDB1,IZDB2,ITHDB1,ITHDB2,ISWDB1,ISWDB2,ISTDB1,
     1ISTDB2,INCHCK,IREGDB,NFMAX,IDBF0,IDBCMN,IDBGRD,IDEBSP(2)
C
C----------------------------------------------------------- HDATA
      COMMON/HDATA/MESS(10),NBLANK,NAMGRD,NAMEJ,NAMEJ1,
     1NAMEM,NAMEM1,NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,
     1NPHUN,NHINIT,NDST,NAMSAT,NGEOM,NHDASP(2)
C
C----------------------------------------------------------- HDEBUG
      COMMON/HDEBUG/NDBF0(2),NDBCMN(2),NHDBSP
C
C----------------------------------------------------------- RDATA
      COMMON/RDATA/TINY,GREAT,RUPLIM,RLOLIM,AZDZ,AZXU,AZYV,
     1AZRI,AZAL,AZPH,XULAST,YVLAST,ZWLAST,TLAST,TFIRST,PBAR,SNALFA,
     1RINNER,ENUL,ENUT,RHO1,RHO2,CFIPS,CMDOT,CONMDT,GRND,HEATBL,
     1FIXFLU,READFI,ZMOVE1,ZDIFAC,DRH1DP,DRH2DP,U1AD,U2AD,V1AD,
     1V2AD,W1AD,W2AD,HUNIT,DIFCUT,ABSIZ,ORSIZ,OPPVAL,TMP1,TMP2,
     1EL1,EL2,GRND1,GRND2,GRND3,GRND4,GRND5,GRND6,GRND7,GRND8,GRND9
     1,GRND10,ZWADD,RINIT,SAME,FIXVAL,AXDZ,AYDZ,RDATSP(21)
C
C----------------------------------------------------------- RDEBUG
      COMMON/RDEBUG/BGCHCK,SMCHCK,RDEBSP(5)
C
C-------------------------------------LOGICAL DECLARATIONS
      LOGICAL LDAT,LDEB
      LOGICAL CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32,SAVGEO,RSTGEO,
```

```
     1NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,DONACC,
     1PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,USEGRD,
     1USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP
      LOGICAL DBGEOM,DBADJS,DBGPHI,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37
C
C----------------------------------INTEGER DECLARATIONS
      INTEGER FSTEP,FSWEEP,TSTSWP,ENTH1,ENTH2,DEN1,
     1DEN2,PCOR,VISL,EPOR,HPOR,VPOR,VIST,TEMP1,TEMP2
C----------------------------------CHARACTER DECLARATIONS
      CHARACTER*4 NHDAT,NHDEB
      CHARACTER*4 NAME
      CHARACTER*4 MESS,NBLANK,NAMGRD,NAMEJ,NAMEJ1,NAMEM,NAMEM1,
     1NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,NPHUN,NHINIT,
     1NDST,NAMSAT,NGEOM,NHDASP
      CHARACTER*4 NDBF0,NDBCMN,NHDBSP
C----------------------------------EQUIVALENT TRANSMISSION ARRAYS
      DIMENSION LDAT(84),LDEB(45),IDAT(120),IDEB(16),NHDAT(30),
     1NHDEB(5),RDAT(85),RDEB(7)
      EQUIVALENCE (LDAT(1),CARTES),(LDEB(1),DBGEOM),(IDAT(1),NX),
     1(IDEB(1),IZDB1),(NHDAT(1),MESS(1)),(NHDEB(1),NDBF0(1)),
     1(RDAT(1),TINY),(RDEB(1),BGCHCK)
CLIST
#include "grdloc"
#include "grdear"
C.... This subroutine directs control to the GROUNDs selected by
C     the satellite settings of USEGRX, NAMGRD & USEGRD.
C           Subroutine GREX1 contains much standard material, eg.
C     options for fluid properties, several turbulence models,
C     wall functions, etc.
C
          IF(USEGRX) CALL GREX1
          IF(USEGRD) CALL GROUND
C
C.... The data echo is now called at the preliminary print stage.
C
          IF(IGR.NE.20) RETURN
          IF(.NOT.ECHO) GO TO 20
      CALL DATPRN(Y,Y,Y,Y,  Y,Y,Y,Y,  Y,Y,Y,N,  Y,Y,Y,Y,
     1          Y,Y,Y,Y,  Y,Y,Y,Y)
      RETURN
   20 CALL DATPRN(Y,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N)
      RETURN
      END
C*************************************************************
      SUBROUTINE GROUND
Cinclude "satear"
C FILE NAME SATEAR --- 170486
CNLIST
C
C------------------------------------------------ ARRAYS
      COMMON/LDB1/DBGPHI(50)/IDA1/ITERMS(50)/IDA2/LITER(50)
```

```
     1/IDA3/IORCVF(50)/IDA4/IORCVL(50)/IDA5/ISLN(50)/IDA6/IPRN(50)
     1/HDA1/NAME(50)/RDA1/DTFALS(50)/RDA2/RESREF(50)
     1/RDA3/PRNDTL(50)/RDA4/PRT(50)/RDA5/ENDIT(50)/RDA6/VARMIN(50)
     1/RDA7/VARMAX(50)/RDA8/FIINIT(50)/RDA9/PHINT(50)
     1/RDA10/CINT(50)/RDA11/EX(50)
C------------------------------------------------------------ LDATA
      COMMON/LDATA/CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32(17),SAVGEO,
     1RSTGEO,NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,
     1DONACC,PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,
     1USEGRD,USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP(11)
C
C------------------------------------------------------------ LDEBUG
      COMMON/LDEBUG/DBGEOM,DBADJS,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37(9)
C
C------------------------------------------------------------ IDATA
      COMMON/IDATA/NX,NY,NZ,LUPR1,LUPR2,LUPR3,LUPHUN,LUSDA,IPROF,
     1LUFI,LUDST,LUGRF,LUSAVE,LUOLD,LUDEP,LUPCO,LUDVL,
     1IRUNN,IOPTN,LITC,LITFLX,NRUN,LITHYD,FSTEP,LSTEP,
     1FSWEEP,LSWEEP,NPRINT,LIBREF,MEANDF,IXMON,IYMON,IZMON,IINIT,
     1NLSG1,NISG1,NRSG1,NCSG1,IPARAB,IDPHUN,NXFR1,NYFR1,NZFR1,
     1NTFR1,ENTH1,ENTH2,ISWR1,ISWR2,IXPRF,IXPRL,IYPRF,IYPRL,
     1NPRMNT,ISTPRL,ISTPRF,IZPRL,IZPRF,NUMCLS,TSTSWP,NYPRIN,NXPRIN,
     1NZPRIN,NPRMON,NTPRIN,NTZPRF,ISP66,IURINI,IURPRN,IURVAL,
     1IORTCV,NUMREG,NRTCV,ICHR,INTFRC,ITHC1,ISWC1,DEN1,DEN2,
     1VISL,INTMDT,ISWPRF,ISWPRL,IPSA,ISP84,IPLTF,IPLTL,NPLT,ITABL,
     1TEMP1,TEMP2,LEN1,LEN2,NLG1,NIG1,NRG1,NCG1,NPNAM1,
     1ISP98(3),LENREC,LUGEOM,IMB1,IMB2,PCOR,NCOLPF,NCOLCO,
     1NROWCO,EPOR,NPOR,HPOR,VPOR,KXFR,KYFR,KZFR,KTFR,IDATSP(2),
     1VIST,NPHI
C
C------------------------------------------------------------ IDEBUG
      COMMON/IDEBUG/IZDB1,IZDB2,ITHDB1,ITHDB2,ISWDB1,ISWDB2,ISTDB1,
     1ISTDB2,INCHCK,IREGDB,NFMAX,IDBF0,IDBCMN,IDBGRD,IDEBSP(2)
C
C------------------------------------------------------------ HDATA
      COMMON/HDATA/MESS(10),NBLANK,NAMGRD,NAMEJ,NAMEJ1,
     1NAMEM,NAMEM1,NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,
     1NPHUN,NHINIT,NDST,NAMSAT,NGEOM,NHDASP(2)
C
C------------------------------------------------------------ HDEBUG
      COMMON/HDEBUG/NDBF0(2),NDBCMN(2),NHDBSP
C
C------------------------------------------------------------ RDATA
      COMMON/RDATA/TINY,GREAT,RUPLIM,RLOLIM,AZDZ,AZXU,AZYV,
     1AZRI,AZAL,AZPH,XULAST,YVLAST,ZWLAST,TLAST,TFIRST,PBAR,SNALFA,
     1RINNER,ENUL,ENUT,RHO1,RHO2,CFIPS,CMDOT,CONMDT,GRND,HEATBL,
     1FIXFLU,READFI,ZMOVE1,ZDIFAC,DRH1DP,DRH2DP,U1AD,U2AD,V1AD,
     1V2AD,W1AD,W2AD,HUNIT,DIFCUT,ABSIZ,ORSIZ,OPPVAL,TMP1,TMP2,
```

```
     1EL1,EL2,GRND1,GRND2,GRND3,GRND4,GRND5,GRND6,GRND7,GRND8,GRND9
     1,GRND10,ZWADD,RINIT,SAME,FIXVAL,AXDZ,AYDZ,RDATSP(21)
C
C---------------------------------------------------------- RDEBUG
      COMMON/RDEBUG/BGCHCK,SMCHCK,RDEBSP(5)
C
C---------------------------------LOGICAL DECLARATIONS
      LOGICAL LDAT,LDEB
      LOGICAL CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32,SAVGEO,RSTGEO,
     1NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,DONACC,
     1PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,USEGRD,
     1USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP
      LOGICAL DBGEOM,DBADJS,DBGPHI,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37
C
C---------------------------------INTEGER DECLARATIONS
      INTEGER FSTEP,FSWEEP,TSTSWP,ENTH1,ENTH2,DEN1,
     1DEN2,PCOR,VISL,EPOR,HPOR,VPOR,VIST,TEMP1,TEMP2
C---------------------------------CHARACTER DECLARATIONS
      CHARACTER*4 NHDAT,NHDEB
      CHARACTER*4 NAME
      CHARACTER*4 MESS,NBLANK,NAMGRD,NAMEJ,NAMEJ1,NAMEM,NAMEM1,
     1NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,NPHUN,NHINIT,
     1NDST,NAMSAT,NGEOM,NHDASP
      CHARACTER*4 NDBF0,NDBCMN,NHDBSP
C---------------------------------EQUIVALENT TRANSMISSION ARRAYS
      DIMENSION LDAT(84),LDEB(45),IDAT(120),IDEB(16),NHDAT(30),
     1NHDEB(5),RDAT(85),RDEB(7)
      EQUIVALENCE (LDAT(1),CARTES),(LDEB(1),DBGEOM),(IDAT(1),NX),
     1(IDEB(1),IZDB1),(NHDAT(1),MESS(1)),(NHDEB(1),NDBF0(1)),
     1(RDAT(1),TINY),(RDEB(1),BGCHCK)
CLIST
#include "grdloc"
#include "grdear"
      INTEGER HIGH,OLD,AUX
      LOGICAL STORE,SOLVE,PRINT
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION STARTS:
C
C 1   Set dimensions of satellite-to-GROUND data arrays to those
C     of the satellite.
      COMMON/LGRND/LG(20)/IGRND/IG(50)/RGRND/RG(100)/CGRND/CG(10)
      LOGICAL LG
      CHARACTER*4 CG
C
C 2   User dimensions own arrays here, for example:
C     DIMENSION UUH(10,10),UUC(10,10),UUX(10,10),UUZ(10)
C====================PARAMETER STATEMENTS===========================
      PARAMETER (JNX=1,JNY=50,JNZ=300,JNYP=JNY+1,JNZP=JNZ+1)
C====================DIMENSION (JNY,JNX)============================
```

```
      DIMENSION         CVAR(JNY,JNX),   GAEX(JNY,JNX),   GAH1(JNY,JNX),
     & GAMMA(JNY,JNX), GAN(JNY,JNX),    GC1(JNY,JNX),    GC4(JNY,JNX),
     & GC5(JNY,JNX),   GC6(JNY,JNX),    GC7(JNY,JNX),    GD1DP(JNY,JNX),
     & GENTRO(JNY,JNX),GENTH(JNY,JNX),  GFU(JNY,JNX),    GHPOR(JNY,JNX),
     & GMACH(JNY,JNX), GNPOR(JNY,JNX),  GOX(JNY,JNX),    GP(JNY,JNX),
     & GP1L(JNY,JNX),  GPIZ1(JNY,JNX),  GRH(JNY,JNX),    GTEMP(JNY,JNX),
     & GVPOR(JNY,JNX), GV1(JNY,JNX),    GW1(JNY,JNX),    GW1L(JNY,JNX),
     & GWA(JNY,JNX),   PHI(JNY,JNX),    VVAR(JNY,JNX),   XDIS(JNY,JNX),
     & YDIS(JNY,JNX),  ZDIS(JNY,JNX),   GLW1(JNY,JNX)
      DIMENSION         GW2(JNY,JNX),    GV2(JNY,JNX),    GR1(JNY,JNX),
     & GR2(JNY,JNX),   GRS(JNY,JNX),    ARRAY1(JNY,JNX),ARRAY2(JNY,JNX),
     & ARRAY3(JNY,JNX),GGAMA(JNY,JNX),  GTOT(JNY,JNX)
C====================DIMENSION (JNY)====================================
      DIMENSION         GWMOL(JNY)
C====================DIMENSION (JNYP)===================================
      DIMENSION         GYEXIT(JNYP)
C====================DIMENSION (JNZ)====================================
      DIMENSION         GANGWL(JNZ),     GARNWL(JNZ),     GENTHW(JNZ),
     & GHTCOE(JNZ),    GPAX(JNZ),       GPNW(JNZ),       GTAX(JNZ),
     & GTNW(JNZ),      GTWALL(JNZ),     GZNODE(JNZ)
C====================DIMENSION (JNZP)===================================
      DIMENSION         GZCELL(JNZP),    GYWALL(JNZP),    GDYNY(JNZP)
C====================DIMENSION OTHERS===================================
      DIMENSION         ATOMN(4),        ATOMW(4),        H0(7),
     & SM(7),           SMB(3),          SMI(3),          SMW(7),
     & SM1(7),          SN(7),           S0(7),           S1(7),
     & S2(7)
C=IWC================Cooling Jacket Simulation==========================
      DIMENSION         GTGAS(JNZ),      GTLIQ(0:JNZ),    GHGAS(JNZ),
     & GCPG(JNZ),       CXAREA(JNZ),     WTHK(JNZ),
     & DIAHYD(JNZ),     FT1(JNZ),        FT2(JNZ),        GDIST(JNZ)
C====================EQUIVALENCE========================================
      EQUIVALENCE       (NZTHRO,IG(21)),(INFO,IG(20)),   (GA,RG(7)),
     & (PTOT,RG(8)),    (POTOP,RG(10)), (POBOT,RG(11)), (ENTHIN,RG(21)),
     & (TTOT,RG(22)),   (RGAS,RG(23)),  (THROAT,RG(24)),(CMW,RG(28)),
     & (AVISC,RG(29)),  (CONST2,RG(30)),(GDDROP,RG(32)),(WJET,RG(33)),
     & (STEN,RG(35)),   (VISXY,RG(36)), (CABS,RG(37)),  (DIAOJ,RG(38)),
     & (ENTHO2,RG(6)),  (FRATE,RG(9)),  (GPI100,RG(12)),
C=IWC================Cooling Jacket Simulation==========================
     & (FLXINL,RG(44)),(FLXINU,RG(45)),(COPPK,RG(46)),
     & (STEEK,RG(47)),  (PRHYD,RG(48)), (RATEL,RG(49)), (RATEU,RG(50)),
     & (TLIQL,RG(51)),  (TLIQU,RG(52)), (VISHYD,RG(53)),
     & (LRSTRT,LG(1)),  (LHGEN,LG(2)),  (LHLEN,LG(3)),
     & (NCHA,IG(25)),   (NTUB,IG(26)),  (NCOM,IG(27)),  (NNOZ,IG(28))
C======================================================================
C 3   User places his data statements here, for example:
C     DATA NXDIM,NYDIM/10,10/
C-IWC---Cooling Jacket Simulation
      LOGICAL LEQUIL,LSWIT,LRSTRT,LHGEN,LHLEN
      DATA LSWIT      /.TRUE./
      DATA LEQUIL     /.TRUE./
      DATA ATOMN      /4HO   ,4HH   ,4HC   ,4HN   /
      DATA ATOMW      /15.9994,1.00797,12.01115,14.0067/
      DATA P0,Q0      /1.01325E5,0.0/
      DATA NS,NLM,NFR /7,2,10/
```

```
C
C 4   Index functions for GROUND-EARTH variable references.
      LOW(I)=NPHI+I
      HIGH(I)=2*NPHI+I
      OLD(I)=3*NPHI+I
      IN(I)=4*NPHI+I
      STORE(I)=MOD(ISLN(I),2).EQ.0
      SOLVE(I)=MOD(ISLN(I),3).EQ.0
      PRINT(I)=MOD(IPRN(I),2).EQ.0
C
C 5   Insert own coding below as desired, guided by GREX1 examples.
C     Note that the satellite-to-GREX1 special data in the labelled
C     COMMONs /RSG/, /ISG/, /LSG/ and /CSG/ can be included and
C     used below but the user must check GREX1 for any conflicting
C     uses. The same comment applies to the EARTH-spare working
C     arrays EASP1, EASP2,....EASP10. If the call to GREX1 has been
C     deactivated then they can all be used without reservation.
C
      IXL=IABS(IXL)
          IF(IGR.EQ.13) GO TO 13
          IF(IGR.EQ.19) GO TO 19
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,
     122,23,24),IGR
C*****************************************************************
C
C--- GROUP 1. Run title and other preliminaries
C
    1 GO TO (1001,1002),ISC
 1001 CONTINUE
C-pd---Provide local storage--------------------------------------
      CALL MAKE(EASP1)
      CALL MAKE(EASP10)
      RETURN
 1002 CONTINUE
C-pd---Calculate Pi-----------------------------------------------
      GPI=4.*ATAN(1.)
C-pd---Convert outlet pressure------------------------------------
      POTOP=POTOP*CONST2
      POBOT=POBOT*CONST2
      IF(IG(17).EQ.2) P0=P0*10.
C-pd---Chemic is called even if eq pack is not used to get mw-----
      CALL CHEMIC(0,INFO,LEQUIL,NS,NLM,TK,PA,P0,HSUB0,
     1 Q0,RGAS,S1,S2,ATOMN,ATOMW, SMW,RHOP,WMOL,H0,S0)
C-pd---Calculate mass fraction of stoichiometric reaction---------
      IF(IG(5).EQ.2) STOKFT=2.*ATOMW(2)/(ATOMW(1)+2.*ATOMW(2))
C-pd---Store flow rates of jets-----------------------------------
      IF(IG(9).EQ.2) THEN
        IX=1
        DO 1222 IY=1,NY
        DO 1222 JJ=31,50
        IF(IG(JJ).EQ.IY) GTOT(IY,IX)=RG(JJ+50)
 1222   CONTINUE
      ENDIF
      RETURN
C*****************************************************************
```

```
C
C--- GROUP 2. Transience; time-step specification
C
    2 CONTINUE
      RETURN
C*****************************************************************
C
C--- GROUP 3. X-direction grid specification
C
    3 CONTINUE
      RETURN
C*****************************************************************
C
C--- GROUP 4. Y-direction grid specification
C
    4 CONTINUE
      RETURN
C*****************************************************************
C
C--- GROUP 5. Z-direction grid specification
C
    5 CONTINUE
      RETURN
C*****************************************************************
C
C--- GROUP 6. Body-fitted coordinates or grid distortion
C
    6 CONTINUE
      RETURN
C*****************************************************************
C
C--- GROUP 7. Variables stored, solved & named
    7 CONTINUE
      RETURN
C*****************************************************************
C
C--- GROUP 8. Terms (in differential equations) & devices
C
    8 GO TO (81,82,83,84,85,86,87,88,89,810,811,812,813,814,815)
     1,ISC
   81 CONTINUE
C   * ------------------- SECTION  1 ---------------------------
C    For U1AD.LE.GRND--- phase 1 additional velocity (VELAD).
      RETURN
   82 CONTINUE
C   * ------------------- SECTION  2 ---------------------------
C    For U2AD.LE.GRND--- phase 2 additional velocity (VELAD).
      RETURN
   83 CONTINUE
C   * ------------------- SECTION  3 ---------------------------
C    For V1AD.LE.GRND--- phase 1 additional velocity (VELAD).
      RETURN
   84 CONTINUE
C   * ------------------- SECTION  4 ---------------------------
C    For V2AD.LE.GRND--- phase 2 additional velocity (VELAD).
```

```
      RETURN
   85 CONTINUE
C    * ------------------- SECTION  5 ---------------------------
C     For W1AD.LE.GRND--- phase 1 additional velocity (VELAD).
      RETURN
   86 CONTINUE
C    * ------------------- SECTION  6 ---------------------------
C     For W2AD.LE.GRND--- phase 2 additional velocity (VELAD).
      RETURN
   87 CONTINUE
C    * ------------------- SECTION 7 ---- VOLUMETRIC SOURCE FOR GALA
      RETURN
   88 CONTINUE
C    * ------------------- SECTION 8 --- CONVECTION FLUXES
      RETURN
   89 CONTINUE
C    * ------------------- SECTION 9 --- DIFFUSION COEFFICIENTS
      RETURN
  810 CONTINUE
C    * ------------------- SECTION 10 --- CONVECTION NEIGHBOURS
      RETURN
  811 CONTINUE
C    * ------------------- SECTION 11 --- DIFFUSION NEIGHBOURS
      RETURN
  812 CONTINUE
C    * ------------------- SECTION 12 --- LINEARISED SOURCES
      RETURN
  813 CONTINUE
C    * ------------------- SECTION 13 --- CORRECTION COEFFICIENTS
      RETURN
  814 CONTINUE
C    * ------------------- SECTION 14 --- USER'S SOLVER
      RETURN
  815 CONTINUE
C    * ------------------- SECTION 15 --- CHANGE SOLUTION
      RETURN
C    * Make all other group-8 changes in group 19.
C******************************************************************
C
C--- GROUP 9. Properties of the medium (or media)
C
C    The sections in this group are arranged sequentially in their
C    order of calling from EARTH. Thus, as can be seen from below,
C    the temperature sections (10 and 11) precede the density
C    sections (1 and 3); so, density formulae can refer to
C    temperature stores already set.
    9 GO TO (91,92,93,94,95,96,97,98,99,900,901,902,903),ISC
C******************************************************************
  900 CONTINUE
C    * ------------------- SECTION 10 ---------------------------
C     For TMP1.LE.GRND--------- phase-1 temperature Index AUX(TEMP1)
      RETURN
  901 CONTINUE
C    * ------------------- SECTION 11 ---------------------------
C     For TMP2.LE.GRND--------- phase-2 temperature Index AUX(TEMP2)
```

```
      RETURN
  902 CONTINUE
C   * ------------------- SECTION 12 ---------------------------
C    For EL1.LE.GRND---------- phase-1 length scale Index AUX(LEN1)
      RETURN
  903 CONTINUE
C   * ------------------- SECTION 13 ---------------------------
C    For EL2.LE.GRND---------- phase-2 length scale Index AUX(LEN2)
      RETURN
   91 CONTINUE
C   * ------------------- SECTION  1 ---------------------------
C    For RHO1.LE.GRND--- density for phase 1 Index AUX(DEN1).
C-pd---Get incell values-----------------------------------------------
      CALL GETYX(P1,GP,JNY,JNX)
      CALL GETYX(C1,GC1,JNY,JNX)
      CALL GETYX(C2,GFU,JNY,JNX)
      CALL GETYX(C3,GOX,JNY,JNX)
      CALL GETYX(C4,GWA,JNY,JNX)
      CALL GETYX(C5,GC4,JNY,JNX)
      CALL GETYX(C6,GC5,JNY,JNX)
      CALL GETYX(C7,GC6,JNY,JNX)
      CALL GETYX(C8,GC7,JNY,JNX)
      CALL GETYX(H1,GENTH,JNY,JNX)
      CALL GETYX(26,GTEMP,JNY,JNX)
      CALL GETYX(DEN1,GRH,JNY,JNX)
      IX=1
C-pd---Being iy loop---------------------------------------------------
      DO 9140 IY=1,NY
      SM(1)=GFU(IY,IX)
      SM(2)=GOX(IY,IX)
      SM(3)=GWA(IY,IX)
      SM(4)=GC4(IY,IX)
      SM(5)=GC5(IY,IX)
      SM(6)=GC6(IY,IX)
      SM(7)=GC7(IY,IX)
      TK=GTEMP(IY,IX)
      IF(GC1(IY,IX).LT.0.05) THEN
        IF(ISWEEP.EQ.LSWEEP-1)WRITE(6,*)
     &  ' IN C1 TRAP  Z Y C ',IZ,IY,GC1(IY,IX)
        GC1(IY,IX)=.05
      ENDIF
      SM1(1)=GC1(IY,IX)
      SM1(2)=1.-SM1(1)
      SM1(3)=0.0
      SM1(4)=0.0
      SM1(5)=0.0
      SM1(6)=0.0
      SM1(7)=0.0
C-pd---Calculate molar concentrations---------------------------------
C---   S1-incoming & S2-guess-----------------------------------------
      DO 9130 IS=1,NS
      S2(IS)=SM(IS)/SMW(IS)
 9130 S1(IS)=SM1(IS)/SMW(IS)
C-pd---Call equlibrium package to calc incell molar conc & temps------
C---   Call temper when eq not used to get temp at const molar conc---
```

```
C---   HSUB0=Incell H   PA=Incell P   GWMOL=Avg MW of eq mix----------
      HSUB0=GENTH(IY,IX)
      PA=GP(IY,IX)
      IF(IG(5).EQ.2) THEN
        SMI(1)=GC1(IY,IX)/(2.*ATOMW(2))
        SMI(2)=(1.-GC1(IY,IX))/(2.*ATOMW(1))
        SMI(3)=0.0
        IF(GC1(IY,IX).LT.STOKFT) THEN
          SMI(2)=SMI(2)-SMI(1)/2.
          SMI(3)=SMI(1)
          SMI(1)=0.0
        ELSE
          SMI(1)=SMI(1)-2.*SMI(2)
          SMI(3)=2.*SMI(2)
          SMI(2)=0.0
        ENDIF
        SMB(1)=SMI(1)*(2.*ATOMW(2))
        SMB(2)=SMI(2)*(2.*ATOMW(1))
        SMB(3)=SMI(3)*(2.*ATOMW(1)+ATOMW(2))
        WMOL=(SMI(1)*2.*ATOMW(2)+SMI(2)*2.*ATOMW(1)+SMI(3)*
     &       (2.*ATOMW(2)+ATOMW(1)))/(SMI(1)+SMI(2)+SMI(3))
        CALL TEMPER(GENTH(IY,IX),GTEMP(IY,IX),TK,CPDR,RGAS,SMI,3,INFO)
      ENDIF
      IF(IG(5).EQ.1) CALL CHEMIC(1,INFO,LEQUIL,NS,NLM,TK,PA,P0,HSUB0,
     1 Q0,RGAS,S1,S2,ATOMN,ATOMW,SMW,RHOP,WMOL,H0,S0)
      GWMOL(IY)=WMOL
C-pd---Calculate denstiy by PW/(RT)----------------------------------
      GRH(IY,IX)=GP(IY,IX)*WMOL/(RGAS*TK)
C-pd---Calculate compressibility term---------------------------------
      GD1DP(IY,IX)=1./GP(IY,IX)
C-pd---Assign incell temp---------------------------------------------
      GTEMP(IY,IX)=TK
C-pd---Calculate mass fractions---------------------------------------
      GFU(IY,IX)=S2(1)*SMW(1)
      GOX(IY,IX)=S2(2)*SMW(2)
      GWA(IY,IX)=S2(3)*SMW(3)
      GC4(IY,IX)=S2(4)*SMW(4)
      GC5(IY,IX)=S2(5)*SMW(5)
      GC6(IY,IX)=S2(6)*SMW(6)
      GC7(IY,IX)=S2(7)*SMW(7)
      IF(IG(5).EQ.2) THEN
        GFU(IY,IX)=SMB(1)
        GOX(IY,IX)=SMB(2)
        GWA(IY,IX)=SMB(3)
      ENDIF
C-pd---Calculate entropy and gamma on last sweep----------------------
      IF(ISWEEP.EQ.LSWEEP) THEN
        TLN=ALOG(TK)
        CALL HCPS(3,INFO,TK,TLN,NS,GHSUM,GCPSUM,S2,H0,S0,S0SUM)
        GENTRO(IY,1)=S0SUM*RGAS-RGAS*ALOG10(GP(IY,1)/P0)
        GCP=GCPSUM*RGAS
        GAMMA(IY,1)=GCP/(GCP-RGAS/WMOL)
      ENDIF
 9140 CONTINUE
C-pd---Set values into PHOENICS arrarys-------------------------------
```

```
      IF(ISWEEP.EQ.LSWEEP) THEN
        CALL SETYX(27,GENTRO,JNY,JNX)
        CALL SETYX(28,GAMMA,JNY,JNX)
      ENDIF
      CALL SETYX(AUX(DEN1),GRH,JNY,JNX)
      CALL SETYX(26,GTEMP,JNY,JNX)
      CALL SETYX(C2,GFU,JNY,JNX)
      CALL SETYX(C3,GOX,JNY,JNX)
      CALL SETYX(C4,GWA,JNY,JNX)
      CALL SETYX(C5,GC4,JNY,JNX)
      CALL SETYX(C6,GC5,JNY,JNX)
      CALL SETYX(C7,GC6,JNY,JNX)
      CALL SETYX(C8,GC7,JNY,JNX)
C----- SAVE TEMPERARURES AND PRESSURES (SAVE ALSO WALL ENTHALPY JULY86)
      IF(IG(8).GT.1) THEN
        CALL HCPS(1,INFO,GTWALL(IZ),0.,7,GHH,GCPSUM,S2,H0,S0,S0SUM)
        GENTHW(IZ)=GHH*RGAS*GTWALL(IZ)
        GCPG(IZ)=(GENTH(NY,1)-GENTHW(IZ))/(GTEMP(NY,1)-GTWALL(IZ))
      ENDIF
      GTNW(IZ)=GTEMP(NY,1)
      GTAX(IZ)=GTEMP(1,1)
      GPNW(IZ)=GP(NY,1)
      GPAX(IZ)=GP(1 ,1)
      RETURN
   92 CONTINUE
C   * ------------------- SECTION  2 --------------------------
C     For DRH1DP.LE.GRND--- D(LN(DEN))/DP for phase 1 (D1DP).
      CALL SETYX(D1DP,GD1DP,JNY,JNX)
      RETURN
   93 CONTINUE
C   * ------------------- SECTION  3 --------------------------
C     For RHO2.LE.GRND--- density for phase 2 Index AUX(DEN2).
      RETURN
   94 CONTINUE
C   * ------------------- SECTION  4 --------------------------
C     For DRH2DP.LE.GRND--- D(LN(DEN))/DP for phase 2 (D2DP).
      RETURN
   95 CONTINUE
C   * ------------------- SECTION  5 --------------------------
C     For ENUT.LE.GRND--- reference turbulent kinematic viscosity.
      RETURN
   96 CONTINUE
C   * ------------------- SECTION  6 --------------------------
C     For ENUL.LE.GRND--- reference laminar kinematic viscosity.
C-pd---Calculate kinematic viscosity--------------------------------
      CALL GETYX(DEN1,PHI,JNY,JNX)
      IX=1
      DO 9610 IY=1,NY
      GENUL=AVISC/(PHI(IY,IX)+1.0E-15)
 9610 PHI(IY,IX)=GENUL
      CALL SETYX(AUX(VISL),PHI,JNY,JNX)
      RETURN
   97 CONTINUE
C   * ------------------- SECTION  7 --------------------------
C     For PRNDTL( ).LE.GRND--- laminar PRANDTL nos., or diffusivity.
```

```
      RETURN
   98 CONTINUE
C   * ------------------- SECTION  8 --------------------------
C    For PHINT( ).LE.GRND--- interface value of first phase(FII1).
      RETURN
   99 CONTINUE
C   * ------------------- SECTION  9 --------------------------
C    For PHINT( ).LE.GRND--- interface value of second phase(FII2)
      RETURN
C*****************************************************************
C
C--- GROUP 10. Inter-phase-transfer processes and properties
C
   10 GO TO (101,102,103,104),ISC
  101 CONTINUE
C   * ------------------- SECTION  1 --------------------------
C    For CFIPS.LE.GRND--- inter-phase friction coeff. AUX(INTFRC).
      CALL GETYX(W1  ,GW1  ,JNY,JNX)
      CALL GETYX(W2  ,GW2  ,JNY,JNX)
      CALL GETYX(V1  ,GV1  ,JNY,JNX)
      CALL GETYX(V2  ,GV2  ,JNY,JNX)
      CALL GETYX(R1  ,GR1  ,JNY,JNX)
      CALL GETYX(R2  ,GR2  ,JNY,JNX)
      CALL GETYX(VOL ,GVPOR,JNY,JNX)
      CALL GETYX(DEN1,GRH  ,JNY,JNX)
      CALL GETYX(RS  ,GRS  ,JNY,JNX)
C
      IX=1
      DO 1010 IY=1,NY
      GRATIO=GR2(IY,IX)/(GRS(IY,IX)+TINY)
      GDIA=GDDROP*GRATIO**0.333
      GDELV=GV1(IY,IX)-GV2(IY,IX)
      GDELW=GW1(IY,IX)-GW2(IY,IX)
      GVSLIP=SQRT(GDELV*GDELV+GDELW*GDELW)
      GRE=GRH(IY,IX)*GVSLIP*GDIA/AVISC
      GRE=AMAX1(GRE,TINY)
      GCD=24.*(1.+GRE**0.667/6.)/GRE+0.42/(1.+4.25E4*GRE**(-1.16))
      GCOEFF=GVPOR(IY,IX)*6.*GR2(IY,IX)*GCD*GRH(IY,IX)*GVSLIP/
     &             (4.*(GDIA+TINY))
 1010 ARRAY1(IY,IX)=0.5*GCOEFF
C
      CALL SETYX(AUX(INTFRC),ARRAY1,JNY,JNX)
      RETURN
  102 CONTINUE
C   * ------------------- SECTION  2 --------------------------
C    For CMDOT.EQ.GRND- inter-phase mass transfer Index AUX(INTMDT)
      CALL GETYX(V1    ,GV1  ,JNY,JNX)
      CALL GETYX(V2    ,GV2  ,JNY,JNX)
      CALL GETYX(W1    ,GW1  ,JNY,JNX)
      CALL GETYX(W2    ,GW2  ,JNY,JNX)
      CALL GETYX(R2    ,GR2  ,JNY,JNX)
      CALL GETYX(VOL   ,GVPOR,JNY,JNX)
      CALL GETYX(DEN1  ,GRH  ,JNY,JNX)
      CALL GETYX(26    ,GTEMP,JNY,JNX)
      CALL GETYX(RS    ,GRS  ,JNY,JNX)
```

```
C
      IX=1
      DO 1020 IY=1,NY
      GRATIO=GR2(IY,IX)/(GRS(IY,IX)+TINY)
      GDIA=GDDROP*GRATIO**0.333
      GDELV=GV1(IY,IX)-GV2(IY,IX)
      GDELW=GW1(IY,IX)-GW2(IY,IX)
      GVSLIP=SQRT(GDELV*GDELV+GDELW*GDELW)
      GRE=GRH(IY,IX)*GVSLIP*GDIA/AVISC
      GRE=AMAX1(GRE,TINY)
      GGAMA(IY,IX)=0.5*2.2011E-7*(ABS(GTEMP(IY,IX)/287.))**1.823+
     &             0.5*GGAMA(IY,IX)
      GGAMA(IY,IX)=AMAX1(TINY,GGAMA(IY,IX))
      GMDOT=GVPOR(IY,IX)*17.88*GR2(IY,IX)*GGAMA(IY,IX)*GDDROP*
     &      (GRH(IY,IX)**0.667)*(RHO2**0.333)/((GDIA+1.E-10)**3)
      GMDOT=GMDOT*(1.+0.244*SQRT(GRE))
      GMDOT=GMDOT*2.592*GRATIO**0.6142
 1020 ARRAY1(IY,IX)=AMAX1(TINY,GMDOT)
C
      CALL SETYX(AUX(INTMDT),ARRAY1,JNY,JNX)
      RETURN
  103 CONTINUE
C   * ------------------- SECTION  3 ---------------------------
C    For CINT( ).EQ.GRND--- phase1-to-interface transfer
C                                               coefficients (COI1)
      RETURN
  104 CONTINUE
C   * ------------------- SECTION  4 ---------------------------
C    For CINT( ).EQ.GRND--- phase2-to-interface transfer
C                                               coefficients (COI2)
      RETURN
C******************************************************************
C
C--- GROUP 11. Initialization of variable or porosity fields
C
   11 CONTINUE
C-pd---Calculate initial pressure field-------------------------------
      IF(INDVAR.NE.P1) GO TO 1110
      CALL GETPT(1,NY+1,IZ+1,XC,YN,ZC)
      AAT=(YN/THROAT)**2
      KS=1
      IF(IZ.LT.NZTHRO) KS=0
      CALL MSOLV(GA,KS,AAT,AM,S,TTT,PTP,RTR,VRT,QRT,RGAS)
      PSTAT=PTOT/PTP
      TSTAT=TTOT/TTT
      WFAC=SQRT(TTOT/CMW)
      WVEL=VRT*WFAC
      GEKIN=.5*WVEL**2
      GHSTAT=ENTHIN-GEKIN
      DO 1105 IX=1,NX
      DO 1105 IY=1,NY
 1105 PHI(IY,IX)=PSTAT
      CALL SETYX(VAL,PHI,JNY,JNX)
C-pd---Calculate initial enthalpy field-------------------------------
 1110 IF(INDVAR.NE.H1) GO TO 1120
```

```
      DO 1115 IX=1,NX
      DO 1115 IY=1,NY
 1115 PHI(IY,IX)= GHSTAT
      CALL SETYX(VAL,PHI,JNY,JNX)
C-pd---Calculate initial w-velocity field-----------------------------
 1120 IF(INDVAR.NE.W1) GO TO 1130
      DO 1125 IX=1,NX
      DO 1125 IY=1,NY
 1125 PHI(IY,IX)= WVEL
      CALL SETYX(VAL,PHI,JNY,JNX)
C-pd---Calculate initial temperature field----------------------------
 1130 IF(INDVAR.NE.26) GO TO 1140
      DO 1135 IX=1,NX
      DO 1135 IY=1,NY
 1135 PHI(IY,IX)= TSTAT
      CALL SETYX(VAL,PHI,JNY,JNX)
C-pd---Calculate initial density field--------------------------------
 1140 IF(INDVAR.NE.DEN1) GO TO 1150
      DO 1145 IX=1,NX
      DO 1145 IY=1,NY
      RHOIK=PSTAT*CMW/(RGAS*TSTAT)
 1145 PHI(IY,IX)= RHOIK
      CALL SETYX(VAL,PHI,JNY,JNX)
 1150 CONTINUE
      RETURN
C*****************************************************************
C
C--- GROUP 12. Convection and diffusion adjustments
C
   12 CONTINUE
      RETURN
C*****************************************************************
C
C--- GROUP 13. Boundary conditions and special sources
C
   13 CONTINUE
      GO TO (130,131,132,133,134,135,136,137,138,139,1310,
     11311,1312,1313,1314,1315,1316,1317,1318,1319,1320,1321),ISC
  130 CONTINUE
C------------------- SECTION  1 ------------- coefficient = GRND
C-pd---LONDON FIXIT---------------------------------------------------
      IF(NPATCH.NE.'FIXDEN') GOTO 13010
      CALL ONLYIF(U1,W2,'FIXDEN')
      CALL FN0(IN(105),AUX(DEN1))
      CALL FN1(CO,0.0)
C****************** W A L L   F U N C T I O N S **********************
13010 IF(NPATCH.NE.'MYWALL')  RETURN
C-pd---Get required data----------------------------------------------
      CALL GETYX(P1,GP,JNY,JNX)
      CALL GETYX(W1,GW1,JNY,JNX)
      CALL GETYX(DEN1,GRH,JNY,JNX)
      GWFP1=GP(NY,1)
      IF(IZ.LT.NZ) GWFW1=GW1(NY,1)
      GWFD1=GRH(NY,1)
C-pd---Get high values------------------------------------------------
```

```
      CALL GETYX(HIGH(P1),GP,JNY,JNX)
      CALL GETYX(HIGH(DEN1),GRH,JNY,JNX)
      GWFP1H=GP(NY,1)
      GWFD1H=GRH(NY,1)
      IF(IZ.EQ.NZ) THEN
      GWFP1H=GWFP1
      GWFD1H=GWFD1
      ENDIF
C-pd---Get low values---------------------------------------------
      CALL GETYX(LOW(W1),GW1L,JNY,JNX)
      CALL GETYX(LOW(P1),GP1L,JNY,JNX)
      GWFW1L=GW1L(NY,1)
      GWFP1L=GP1L(NY,1)
      IF(IZ.EQ.1) GWFW1L=GWFW1
      IF(IZ.EQ.1) GWFP1L=GWFP1
C-pd---Calculate gradient terms for h1 ke & ep---------------------
      GMULAM=AVISC
      GDZ=GZCELL(IZ+1)-GZCELL(IZ)
      GDY=.5*(GDYNY(IZ)+GDYNY(IZ+1))
      DPDZ=(GWFP1H-GWFP1L)/(2.*GDZ)
      GWAV=0.5*(GWFW1+GWFW1L)
      GARHO=GWFD1
C-pd---Calculate coefficient and value for w1----------------------
      IF(INDVAR.NE.W1) GO TO 13020
C-pd---Calculate gradient terms for w1-----------------------------
      GDZ=.5*(GZCELL(IZ+2)-GZCELL(IZ))
      DPDZ=(GWFP1H-GWFP1)/GDZ
      GARHO=0.5*(GWFD1+GWFD1H)
      GWAV=GWFW1
      CALL WALDP(IZ,ISWEEP,LSWEEP,TSTSWP,1,GDYNY(IZ),GMULAM,DPDZ,GWAV,
     1           GARHO,VALUE,COEF,INFO)
C-pd---Calculate coefficient and value for ke----------------------
13020 IF(INDVAR.NE.KE) GO TO 13030
      CALL WALDP(IZ,ISWEEP,LSWEEP,TSTSWP,2,GDY,GMULAM,DPDZ,GWAV,GARHO,
     1           VALUE,COEF,INFO)
C-pd---Calculate coefficient and value for ep----------------------
13030 IF(INDVAR.NE.EP) GO TO 13040
      CALL WALDP(IZ,ISWEEP,LSWEEP,TSTSWP,3,GDY,GMULAM,DPDZ,GWAV,GARHO,
     1           VALUE,COEF,INFO)
13040 CONTINUE
C-pd---Set coefficient and values----------------------------------
      CVAR(NY,1)=COEF
      VVAR(NY,1)=VALUE
      CALL SETYX(CO,CVAR,JNY,JNX)
      RETURN
  131 CONTINUE
C------------------ SECTION  2 ------------- coefficient = GRND1
      RETURN
  132 CONTINUE
C------------------ SECTION  3 ------------- coefficient = GRND2
C---
C-IWC---Cooling Jacket Simulation
C---
      IF(IG(8).EQ.3) THEN
        IF(NPATCH.EQ.'WALL'.AND.INDVAR.EQ.H1)THEN
```

```
          CALL GETYX(CO,CVAR,JNY,JNX)
          CALL GETYX(DEN1,GRH,JNY,JNX)
          CALL GETYX(W1,GW1,JNY,JNX)
          GHGAS(IZ)=CVAR(NY,1)*GRH(NY,1)*GW1(NY,1)*GCPG(IZ)
          IF(LHGEN)THEN
            GHGAS(IZ)=GHGAS(IZ)*FT1(IZ)
            CALL FN25(CO,FT1(IZ))
          END IF
        END IF
      ENDIF
C---
      RETURN
  133 CONTINUE
C-------------------- SECTION  4 ------------- coefficient = GRND3
      RETURN
  134 CONTINUE
C-------------------- SECTION  5 ------------- coefficient = GRND4
      RETURN
  135 CONTINUE
C-------------------- SECTION  6 ------------- coefficient = GRND5
      RETURN
  136 CONTINUE
C-------------------- SECTION  7 ------------- coefficient = GRND6
      RETURN
  137 CONTINUE
C-------------------- SECTION  8 ------------- coefficient = GRND7
      CALL GETYX(AUX(INTMDT),ARRAY1,JNY,JNX)
      CALL SETYX(CO,ARRAY1,JNY,JNX)
      RETURN
  138 CONTINUE
C-------------------- SECTION  9 ------------- coefficient = GRND8
      RETURN
  139 CONTINUE
C-------------------- SECTION 10 ------------- coefficient = GRND9
      RETURN
 1310 CONTINUE
C-------------------- SECTION 11 ------------- coefficient = GRND10
      RETURN
 1311 CONTINUE
C-------------------- SECTION 12 -------------------- value = GRND
C-pd---Use an extroplated exit boundary condition---------------------
      IF(NPATCH.NE.'OUTLET') GO TO 13116
      IF(INDVAR.NE.P1.AND.INDVAR.NE.P2) GOTO 13116
      IF(IZ.NE.NZ) GOTO 13116
      IF(IG(7).NE.1) GOTO 13114
      IF(IG(9).EQ.1) THEN
        CALL GETYX(AUX(DEN1),GRH,JNY,JNX)
        CALL GETYX(LOW(W1),GW1,JNY,JNX)
        DO 13112 IX=1,NX
        DO 13112 IY=1,NY
13112   PHI(IY,IX)=-GRH(IY,IX)*GW1(IY,IX)
        CALL SETYX(VAL,PHI,JNY,JNX)
      ELSE
        IF(INDVAR.EQ.P1) THEN
          CALL GETYX(AUX(DEN1),GRH,JNY,JNX)
```

```
          CALL GETYX(LOW(W1),GW1,JNY,JNX)
          CALL GETYX(R1      ,GR1,JNY,JNX)
          DO 13122 IX=1,NX
          DO 13122 IY=1,NY
13122     PHI(IY,IX)=-GRH(IY,IX)*GW1(IY,IX)*GR1(IY,IX)
          CALL SETYX(VAL,PHI,JNY,JNX)
        ELSE
          CALL GETYX(LOW(W2),GW2,JNY,JNX)
          CALL GETYX(R2      ,GR2,JNY,JNX)
          DO 13132 IX=1,NX
          DO 13132 IY=1,NY
13132     PHI(IY,IX)=-RHO2*GW2(IY,IX)*GR2(IY,IX)
          CALL SETYX(VAL,PHI,JNY,JNX)
        ENDIF
      ENDIF
C-pd---Use a fixed pressure boundary condition------------------------
13114 IF(IG(7).NE.2) GOTO 13116
      DISBOT=(GYEXIT(1)+GYEXIT(2))/2.
      DISTOP=(GYEXIT(NY+1)+GYEXIT(NY))/2.
      ELEGTH=DISTOP-DISBOT
      DO 13115 IX=1,NX
      DO 13115 IY=1,NY
      DISTP=(GYEXIT(IY+1)-GYEXIT(IY))/2.+GYEXIT(IY)
      GFACT=(DISTP-DISBOT)/ELEGTH
      GDELP=POTOP-POBOT
13115 PHI(IY,IX)=POBOT+GFACT*GDELP
      CALL SETYX(VAL,PHI,JNY,JNX)
C-pd---Set values when mywall is used---------------------------------
13116 IF(NPATCH.NE.'MYWALL') GO TO 13118
      IF(INDVAR.EQ.H1) THEN
        VVAR(NY,1)=GENTHW(IZ)
        CALL SETYX(VAL,VVAR,JNY,JNX)
      ENDIF
      IF(INDVAR.EQ.KE) CALL SETYX(VAL,VVAR,JNY,JNX)
      IF(INDVAR.EQ.EP) CALL SETYX(VAL,VVAR,JNY,JNX)
C-pd---Set value for h1 when standard phoenics wall is used-----------
13118 IF(NPATCH.NE.'WALL') RETURN
      IF(INDVAR.NE.H1) RETURN
      VVAR(NY,1)=GENTHW(IZ)
      CALL SETYX(VAL,VVAR,JNY,JNX)
      RETURN
 1312 CONTINUE
C-------------------- SECTION 13 ------------------- value = GRND1
      RETURN
 1313 CONTINUE
C-------------------- SECTION 14 ------------------- value = GRND2
      RETURN
 1314 CONTINUE
C-------------------- SECTION 15 ------------------- value = GRND3
      RETURN
 1315 CONTINUE
C-------------------- SECTION 16 ------------------- value = GRND4
      RETURN
 1316 CONTINUE
C-------------------- SECTION 17 ------------------- value = GRND5
```

```
      RETURN
 1317 CONTINUE
C------------------- SECTION 18 -------------------- value = GRND6
      RETURN
 1318 CONTINUE
C------------------- SECTION 19 -------------------- value = GRND7
C-pd---Two phase enthalpy source---------------------------------------
      CALL GETYX(AUX(INTMDT),ARRAY1,JNY,JNX)
      IX=1
      DO 13181 IY=1,NY
13181 ARRAY1(IY,IX)=ARRAY1(IY,IX)*ENTHO2
      CALL SETYX(VAL,ARRAY1,JNY,JNX)
      RETURN
 1319 CONTINUE
C------------------- SECTION 20 -------------------- value = GRND8
C-pd---Calculation of jet stripping-------------------------------------
      IX=1
      CAIN=GPI*DIAOJ*DIAOJ/4.
      CALL GETYX(DEN1,GRH,JNY,JNX)
      CALL GETYX(W1,GW1,JNY,JNX)
      CALL GETPT(1,1,IZ,XFST,YFST,ZFST)
      CALL GETPT(1,1,IZ+1,XLST,YLST,ZLST)
      DELZ=ZLST-ZFST
      DO 13191 IY=1,NY
      ARRAY3(IY,IX)=0.0
      DO 13192 JJ=31,50
13192 IF(IG(JJ).EQ.IY) GOTO 13195
      GOTO 13191
13195 UREL=GW1(IY,IX)-WJET
      TERM1=VISXY*RHO2/STEN*(GRH(IY,IX)*UREL*UREL)**2
      RFACT=(GTOT(IY,IX)-ARRAY2(IY,IX))/GTOT(IY,IX)
      DIAJ=(4.*RFACT*CAIN/GPI)**.5
      ARRAY3(IY,IX)=CABS*(TERM1**.333333)*GPI*DIAJ*DELZ
      IF(ARRAY3(IY,IX)+ARRAY2(IY,IX).GT.GTOT(IY,IX)) THEN
        ARRAY3(IY,IX)=GTOT(IY,IX)-ARRAY2(IY,IX)
      ENDIF
      ARRAY1(IY,IX)=ARRAY3(IY,IX)*1.E15
13191 CONTINUE
      CALL SETYX(VAL,ARRAY1,JNY,JNX)
      RETURN
 1320 CONTINUE
C------------------- SECTION 21 -------------------- value = GRND9
      RETURN
 1321 CONTINUE
C------------------- SECTION 22 -------------------- value = GRND10
      RETURN
C****************************************************************
C
C--- GROUP 14. Downstream pressure for PARAB=.TRUE.
C
   14 CONTINUE
      RETURN
C****************************************************************
C
C--- GROUP 15. Termination of sweeps
```

```
C
   15 CONTINUE
C    * Make changes for this group only in group 19.
      RETURN
C****************************************************************
C
C--- GROUP 16. Termination of iterations
C
   16 CONTINUE
C    * Make changes for this group only in group 19.
      RETURN
C****************************************************************
C
C--- GROUP 17. Under-relaxation devices
C
   17 CONTINUE
C    * Make changes for this group only in group 19.
      RETURN
C****************************************************************
C
C--- GROUP 18. Limits on variables or increments to them
C
   18 CONTINUE
C    * Make changes for this group only in group 19.
      RETURN
C****************************************************************
C
C--- GROUP 19. Special calls to GROUND from EARTH
C
   19 GO TO (191,192,193,194,195,196,197,198),ISC
  191 CONTINUE
C    * ------------------- SECTION 1 ---- START OF TIME STEP.
      RETURN
  192 CONTINUE
C    * ------------------- SECTION 2 ---- START OF SWEEP.
C-pd---Call flush is convex depentent-------------------------------
C     call flush(6)
C-pd---Reset arrays used in two phase calculations-------------------
      IF(IG(9).EQ.2) THEN
        IX=1
        DO 19205 JJ =1,NY
        ARRAY2(JJ,IX)=0.0
19205   ARRAY3(JJ,IX)=0.0
      ENDIF
      IF(ISWEEP.NE.FSWEEP) RETURN
C-pd---Get geometric data & calculate the distance form the near------
C---    wall cell center to the wall   zthro is z-dist at throat-------
      DO 19210 IZZ=1,NZ+1
      CALL GETPT(1,NY+1,IZZ ,XP,GYWALL(IZZ),GZCELL(IZZ))
      CALL GETPT(1,NY,IZZ,XP,GYNW,ZP)
C---
C-IWC---Cooling Jacket Simulation
C---
      IF(IG(8).EQ.3) THEN
        IF(LSWIT.AND.GYWALL(IZZ).GT.SQRT(5.*THROAT**2))THEN
```

```
          IBJ=IZZ-1
          LSWIT=.FALSE.
        END IF
      ENDIF
C---
      GDYNY(IZZ)=.5*(GYWALL(IZZ)-GYNW)
19210 IF(GDYNY(IZZ).LE.0) WRITE(6,*)' ****WARNING:  WALL TO CELL NODE DI
     1STANCE LESS THAN ZERO  (CHECK GRID CLOSELY)'
      ZTHRO=GZCELL(NZTHRO+1)
      DO 19220 IYY=1,NY+1
19220 CALL GETPT(1,IYY,NZ+1,XP,GYEXIT(IYY),ZP)
C-pd---Calclate the z-dist of the cell centers near the wall &--------
C---   the wall angles------------------------------------------------
      DO 19230 IZZ=1,NZ
      GZNODE(IZZ)=.5*(GZCELL(IZZ)+GZCELL(IZZ+1))
19230 GANGWL(IZZ)=ATAN((GYWALL(IZZ+1)-GYWALL(IZZ))/
     1                 (GZCELL(IZZ+1)-GZCELL(IZZ))+1.E-9)
C-pd---Calculate twall based on geometric locations & given data------
      IF(IG(8).EQ.3) THEN
        IF(.NOT.LRSTRT)CALL TWALBC(NZ,GTWALL,GZNODE,ZTHRO,THROAT)
        CALL DARTH(GZNODE,GYWALL,ZTHRO,NCHA,NTUB,NCOM,NNOZ,RATEL,RATEU,
     &             CXAREA,WTHK,DIAHYD,FT1,FT2,IBJ,GDIST,INFO,GPI100)
      ENDIF
C---
      IF(IG(8).EQ.2) CALL TWALBC(NZ,GTWALL,GZNODE,ZTHRO,THROAT)
      IF(IG(8).EQ.1) THEN
        DO 19240 IZZ=1,NZ
19240   GTWALL(IZZ)=0.0
      ENDIF
      RETURN
  193 CONTINUE
C   * ------------------- SECTION 3 ---- START OF IZ SLAB.
C---
C-IWC---Cooling Jacket Simulation
C---
      IF(IG(8).EQ.3) THEN
        IF(ISWEEP.EQ.FSWEEP.AND.LRSTRT)THEN
          CALL GETONE(32,XTEMP,NY,1)
          GTLIQ(IZ)=XTEMP
          CALL GETONE(32,XTEMP,NY-1,1)
          GTWALL(IZ)=XTEMP
        END IF
      ENDIF
C---
C-pd---Check NPOR HPOR VPOR for values < 1.E-10----------------------
      IF (ISWEEP.EQ.2)THEN
      CALL GTIZYX(4,IZ,GVPOR,JNY,JNX)
      CALL GTIZYX(7,IZ,GNPOR,JNY,JNX)
      CALL GTIZYX(9,IZ,GHPOR,JNY,JNX)
      DO 19310 II=1,NY
      IF (GVPOR(II,1).LT.1.E-10)
     &  WRITE(6,*)' ***WARNING*** VOLUME BELOW 1.E-10 AT IZ IY = ',IZ,II
      IF (GNPOR(II,1).LT.1.E-10)
     &  WRITE(6,*)' ***WARNING*** N-AREA BELOW 1.E-10 AT IZ IY = ',IZ,II
19310 IF (GHPOR(II,1).LT.1.E-10)
```

```
     &  WRITE(6,*)' ***WARNING*** H-AREA BELOW 1.E-10 AT IZ IY = ',IZ,II
        ENDIF
C-pd---Compute psia----------------------------------------------------
        IF (ISWEEP.EQ.LSWEEP) THEN
        GFACT=1./CONST2
        CALL GETYX(P1,GP,JNY,JNX)
        DO 19320 II=1,NX
        DO 19320 JJ=1,NY
19320   PHI(JJ,II)=GP(JJ,II)*GFACT
        CALL SETYX(30,PHI,JNY,JNX)
C-pd---Save the y and z cell centers-----------------------------------
C---    Change calls for V1.4---------------------------------------------
        CALL G3IZYX(38,IZ,XDIS,YDIS,ZDIS,JNY,JNX)
        CALL SETYX(45,YDIS,JNY,JNX)
        CALL SETYX(46,ZDIS,JNY,JNX)
C----------------------------------------------------------------------
        ENDIF
        RETURN
  194   CONTINUE
C    *  ------------------- SECTION 4 ---- START OF ITERATION.
        RETURN
  195   CONTINUE
C    *  ------------------- SECTION 5 ---- FINISH OF ITERATION.
        RETURN
  196   CONTINUE
C    *  ------------------- SECTION 6 ---- FINISH OF IZ SLAB.
C---
C-IWC---Cooling Jacket Simulation
C---
        IF(IG(8).EQ.3)THEN
        CALL GETYX(26,GTEMP,JNY,JNX)
        GTGAS(IZ)=GTEMP(NY,1)
          IF(IZ.EQ.NZ)THEN
           CALL TWCOOL(GTGAS,GTWALL,GTLIQ,GHGAS,IBJ,TLIQL,TLIQU,COPPK,
     1                 STEEK,VISHYD,PRHYD,FLXINL,FLXINU,RATEL,RATEU,GDIST,
     2                 CXAREA,DIAHYD,WTHK,FT2,LHLEN,INFO,GPI100)
          END IF
          IF(ISWEEP.EQ.LSWEEP)THEN
            IXF=1
            IXL=NX
            IYF=1
            IYL=NY-2
            CALL FN1(32,0.0)
            IYF=NY
            IYL=NY
            XTEMP=GTLIQ(IZ)
            CALL FN1(32,XTEMP)
            IYF=NY-1
            IYL=NY-1
            XTEMP=GTWALL(IZ)
            CALL FN1(32,XTEMP)
            IYF=1
            IYL=NY
          END IF
        END IF
```

```
C---
C-pd---Add up mass stripped from jet-----------------------------------
      IF(IG(9).EQ.2) THEN
        IX=1
        DO 19605 JJ =1,NY
        IF(ISWEEP.EQ.LSWEEP-1.AND.INFO.GT.1) THEN
          WRITE(6,*)' IZ IY A3',IZ,JJ,ARRAY3(JJ,IX)
        ENDIF
        ARRAY2(JJ,IX)=ARRAY2(JJ,IX)+ARRAY3(JJ,IX)
        IF(IZ.EQ.(NZTHRO/2)+1.AND.ISWEEP.EQ.LSWEEP-1) THEN
          IF(ARRAY2(JJ,IX).NE.GTOT(JJ,IX).AND.ARRAY3(JJ,IX).NE.0.) THEN
            WRITE(6,*)'IZ IY A2 GT ',IZ,JJ,ARRAY2(JJ,IX),GTOT(JJ,IX)
            WRITE(6,*)'STRIPPING RATE IS TOO LOW!!!'
          ENDIF
        ENDIF
19605 CONTINUE
      ENDIF
C-pd---Calculate mach number at last sweep----------------------------
      IF(ISWEEP.LT.LSWEEP) RETURN
      IX=1
      CALL GETCAR
      CALL GETYX(50,GW1,JNY,JNX)
      IF (IZ.GT.1 ) CALL GETYX(LOW(50),GLW1,JNY,JNX)
      IF (IZ.EQ.1 ) CALL GETYX(50,GLW1,JNY,JNX)
      IF (IZ.EQ.NZ) CALL GETYX(LOW(50),GW1,JNY,JNX)
      CALL GETYX(28,GAMMA,JNY,JNX)
      CALL GETYX(49,GV1,JNY,JNX)
      CALL GETYX(P1,GP,JNY,JNX)
      CALL GETYX(DEN1,GRH,JNY,JNX)
      DO 19610 IY=1,NY
      GWAV=.5*(GLW1(IY,1)+GW1(IY,1))
      GVS=0.0
      IF(IY.GT.1) GVS=GV1(IY-1,1)
      GVN=0.0
      IF(IY.LT.NY) GVN=GV1(IY,1)
      GVAV=.5*(GVN+GVS)
      GSOUND=SQRT(GAMMA(IY,1)*GP(IY,1)/GRH(IY,1))
19610 GMACH(IY,IX)=SQRT(GWAV**2+GVAV**2)/GSOUND
      CALL SETYX(29,GMACH,JNY,JNX)
C------CALCULATION OF AUXILIARY VARIABLES------------------------------
      IF(IZ.EQ.1) CALL GETYX(P1,GPIZ1,JNY,JNX)
      CALL GTIZYX(7,IZ,GAN,JNY,JNX)
      GARNWL(IZ)=GAN(NY,1)
      IF(IZ.EQ.1 ) CALL GTIZYX(9,IZ,GAH1,JNY,JNX)
      IF(IZ.EQ.NZ) CALL GTIZYX(9,IZ,GAEX,JNY,JNX)
      IF(IZ.EQ.NZ) CALL GETYX(DEN1,GRH,JNY,JNX)
C-pd---Calculate thrust at the exit-----------------------------------
C---   GSUMF1 is the pressure thrust & GSUMF2 is the momentum thrust--
      IF(IZ.EQ.NZ) THEN
      CALL GETYX(P1,GP,JNY,JNX)
      GSUMF1=0.0
      GSUMF2=0.0
      PATM=0.0
      GAEXT=0.0
      DO 19620 IY=1,NY
```

```
      GFORC1=(GP(IY,1)-PATM)*GAEX(IY,1)*GPI100
      GFORC2=GRH(IY,1)*GAEX(IY,1)*GW1(IY,1)*GPI100*GW1(IY,1)
      GAEXT=GAEXT+GAEX(IY,1)*GPI100
      GSUMF1=GSUMF1+GFORC1
      GSUMF2=GSUMF2+GFORC2
19620 CONTINUE
      ENDIF
C-pd---Calculate thrust and specific impulse (old way)----------------
      IF(IZ .LT.NZ) RETURN
      GPACON=0.
      GPAEXP=0.
      GPAINJ=0.
      DO 19630 IZZ=1,NZTHRO
19630 GPACON=GPACON+GPNW(IZZ)*GARNWL(IZZ)*SIN(GANGWL(IZZ))*GPI100
      DO 19640 IZZ=NZTHRO+1,NZ-1
19640 GPAEXP=GPAEXP+GPNW(IZZ)*GARNWL(IZZ)*SIN(GANGWL(IZZ))*GPI100
      DO 19650 IY=1,NY
19650 GPAINJ=GPAINJ+GPIZ1(IY,1)*GAH1(IY,1)*GPI100
      THRUST=GPACON+GPAEXP+GPAINJ
      GIMPLS=THRUST/(FLXOUT*9.80+1.E-15)
      TRST=GSUMF1+GSUMF2
      GISP=TRST/(FLXOUT*9.80+1.E-15)
C-pd---Write output summary-------------------------------------------
      WRITE(6,19690)
      WRITE(6,19691) GAEXT,FLXIN,FLXOUT,TRST,GISP
      WRITE(6,*)
      WRITE(6,19692)
      DO 19660 IZZ=1,NZ
19660 WRITE(6,19693) IZZ,.5*(GZCELL(IZZ)+GZCELL(IZZ+1)),GZNODE(IZZ),
     1 GYWALL(IZZ),GANGWL(IZZ)*180./GPI,GPNW(IZZ),GPAX(IZZ),
     2 GTWALL(IZZ),GTNW(IZZ),GTAX(IZZ)
      WRITE(6,19694)
      DO 19670 IZZ=1,NZ
19670 WRITE(6,19695) IZZ,GZNODE(IZZ),GYWALL(IZZ),GPNW(IZZ)/CONST2,
     1 GPAX(IZZ)/CONST2,GTWALL(IZZ)*1.8,GTNW(IZZ)*1.8,GTAX(IZZ)*1.8
C***********************************************************************
C-pd---Format statements----------------------------------------------
19690 FORMAT(' ************* OUTPUT SUMMARY *************')
19691 FORMAT(' EXIT   AREA                = ',1P,E12.4,/
     &        ' M FLUX THROUGH INLET     = ',1P,E12.4,/
     &        ' M FLUX THROUGH EXIT      = ',1P,E12.4,/
     &        ' THRUST                   = ',1P,E12.4,/
     &        ' SPECIFIC IMPULSE         = ',1P,E12.4)
19692 FORMAT('  IZ      ZG          ZND          YN          ANG',9X,
     1 'PW          PAX         TW           TNY          TAX')
19693 FORMAT(I5,1P,10E12.4)
19694 FORMAT(/' OUTPUT SUMMARY IN BRITISH UNITS '/' IZ',5X,' ZG:TH',8X,
     1 ' YN',7X,' PW(PSI)     PAX(PSI)     TW(R)     TNY(R)     TAX(R)')
19695 FORMAT(I5,1P,10E12.4)
C***********************************************************************
      RETURN
  197 CONTINUE
C   * ------------------ SECTION 7 ---- FINISH OF SWEEP.
C-pd---Get the inlet and outlet flux---------------------------------
      IF(ISWEEP.NE.LSWEEP-1) RETURN
```

```
      IF(IG(9).EQ.1) THEN
        CALL GETSOR('INLET',R1,FLXIN)
      ELSE
        FLXIN=FRATE/GPI100
      ENDIF
      CALL GETSOR('OUTLET',R1,FLXOUT)
      FLXIN=FLXIN*GPI100
      FLXOUT=-FLXOUT*GPI100
      RETURN
  198 CONTINUE
C   * -------------------- SECTION 8 ---- FINISH OF TIME STEP.
      RETURN
C***************************************************************
C
C--- GROUP 20. Preliminary print-out
C
   20 CONTINUE
      RETURN
C***************************************************************
C
C--- GROUP 21. Print-out of variables
C
   21 CONTINUE
C   * Make changes for this group only in group 19.
      RETURN
C***************************************************************
C
C--- GROUP 22. Spot-value print-out
   22 CONTINUE
C   * Make changes for this group only in group 19.
      RETURN
C***************************************************************
C
C--- GROUP 23. Field print-out and plot control
   23 CONTINUE
      RETURN
C***************************************************************
C
C--- GROUP 24. Dumps for restarts
C
   24 CONTINUE
      RETURN
      END
C**********************************************************************
      SUBROUTINE ENTHAL(TEMP,HSUM,CPSUM,SC,NS,NFO)
C**********************************************************************
C---------CALCULATION OF CP/R & H/RT-----------------------------------
      DIMENSION SC(NS),ZS(7,2,3)
      DATA ZS/3.1,5.112E-4,5.264E-8,-3.491E-11,
     &         3.695E-15,-8.774E2,-1.963,3.057,2.667E-3,-5.81E-6,
     &         5.521E-9,-1.812E-12,-9.889E2,-2.3,3.622,7.362E-4,
     &        -1.965E-7,3.620E-11,-2.895E-15,-1.202E3,3.615,3.626,
     &        -1.878E-3,7.055E-6,-6.764E-9,2.156E-12,-1.048E3,4.305,
     &         2.717,2.945E-3,-8.022E-7,1.023E-10,-4.847E-15,-2.991E4,
     &         6.631,4.07,-1.108E-3,4.152E-6,-2.964E-9,8.07E-13,
```

```
     &       -3.028E4,-3.227E-1/
      K=1
      IF(TEMP.LT.1000.) K=2
      TEMP2=TEMP*TEMP
      HSUM=0.
      CPSUM=0.
      DO 100 IS=1,NS
      CP1=ZS(1,K,IS)
      CP2=ZS(2,K,IS)*TEMP
      CP3=ZS(3,K,IS)*TEMP2
      CP4=ZS(4,K,IS)*TEMP2*TEMP
      CP5=ZS(5,K,IS)*TEMP2*TEMP2
      CPSUM=CPSUM+SC(IS)*(CP1+CP2+CP3+CP4+CP5)
  100 HSUM =HSUM+
     1  SC(IS)*(CP1+.5*CP2+.33333*CP3+.25*CP4+.2*CP5+ZS(6,K,IS)/TEMP)
      RETURN
      END
C***********************************************************************
      SUBROUTINE TEMPER(HSTAT,T0,T,CPDR,RGAS,SC,NSC,NFO)
C***********************************************************************
C----------- SUBITERATIVE CALCULATION OF TEMPERATURE ------------------
      DIMENSION SC(NSC)
      DATA NITER,DT0,TMIN/12,50.,12.345/
      DT=DT0
      TEMP=T0
      CALL ENTHAL(TEMP,HHH,CPDR,SC,NSC,NFO)
      ENTH=HHH*RGAS*TEMP
      IF(HSTAT.LT.ENTH) DT=-DT
      TEMPL=TEMP
      IF(NFO.GE.4) WRITE(6,900) T0,ENTH,HSTAT,RGAS,SC(1),SC(2),SC(3)
      TEMP=TEMP+DT
      ITER=0
  100 ENTHL=ENTH
      ITER=ITER+1
      CALL ENTHAL(TEMP,HHH,CPDR,SC,NSC,NFO)
      ENTH =HHH*RGAS*TEMP
      RENTH=(HSTAT-ENTHL)/((ENTH-ENTHL)+1.E-9)
      IF(NFO.GE.4) WRITE(6,910) ITER,TEMP,ENTH,ENTHL,HSTAT,RENTH
      IF(ABS(ENTH-ENTHL).LT..001*ABS(ENTH)) RENTH=1.
      TEMP1=TEMPL+(TEMP-TEMPL)*RENTH
      TEMP1=AMAX1(TEMP1,.5*TEMP,TMIN)
      TEMP1=AMIN1(TEMP1,1.5*TEMP,5000.)
      TEMPL=TEMP
      TEMP=TEMP1
      AR=ABS(RENTH)
      IF((AR.GT.1.005.OR.AR.LT..995).AND.ITER.LT.NITER) GO TO 100
      T=TEMP
      RETURN
  900 FORMAT(' T0 E HS RG SC',1P,7E12.4)
  910 FORMAT(' IT T E EL HS RE',I3,1P,5E12.4)
      END
C***********************************************************************
      SUBROUTINE MSOLV(GA,KS,AAT,AM,S,TTT,PTP,RTR,VRT,QRT,R)
C***********************************************************************
      G1=GA-1.
```

```
      G2=G1*.5
      G3=1./G1
      G4=GA/G1
      G5=GA+1.
      G6=G5/(2.*G1)
      NITER=0
      IF(KS.GT.0) GO TO 200
C---SUBSONIC-------------------------------------------------------
      IF(AM.GT..9999)AM=0.
  100 AML=AM
      AM=((2.+G1*AM*AM)/G5)**G6/AAT
      NITER=NITER+1
      IF(NITER.GT.100) GO TO 300
C-pd---Prevent mach number from exceding 1 in subsonic region---------
      IF(AM.GT.1.0) THEN
      AM=.98
      S=AM*AM
      GOTO 400
      ENDIF
      IF(ABS(AM-AML)/AM.LT..001) THEN
      S=AM*AM
      GO TO 400
      END IF
      GO TO 100
C----SUPERSONIC----------------------------------------------------
  200 G7=1/G6
      AAT=AMAX1(1.0000001,AAT)
      IF(AM.LT.1.0001) AM=1.0001
  250 S=((AAT*AM)**G7-2./G5)*G5/G1
      AML=AM
      AM=SQRT(S)
      NITER=NITER+1
      IF(NITER.GT.100) GO TO 300
      IF(ABS(AM-AML)/AM.LT..001) GO TO 400
      GO TO 250
  300 WRITE(6,900) AM,AML,AAT,KS
  400 TTT=1.+G2*S
      PTP=TTT**G4
      RTR=TTT**G3
      VRT=SQRT(GA*R/TTT)*AM
      QRT=SQRT(GA/R)*AM/TTT**G6
      RETURN
  900 FORMAT('   TOO MANY ITERATIONS   ',3E10.4,I4)
      END
C*********************************************************************
      SUBROUTINE TWALBC(NZ,TWALL,ZNOD,ZTHRO,THROAT)
C*********************************************************************
C----------CALCULATE WALL TEMPS BASED ON GIVEN DATA------------------
      DIMENSION TWALL(NZ),ZNOD(NZ)
      DIMENSION ZTQWD(27),ZTQW(27),TQW(27)
C----- ZTQWD=(z-zt)/rt AND TQW DATA--------------------------------
      DATA NTQW/27/
      DATA ZTQWD/-2.4842,-2.1348,-1.9407,-1.7467,-.9704,-.7763,-.3881,
     1 -.1941,.1941,.3881,.9704,1.5526,1.9407,1.9601,3.8815,5.8222,
     1 6.7926,7.7629,8.7333,9.7037,11.6444,13.5851,15.5258,17.4666,
```

```
     1 19.4073,21.3480,23.5231/
      DATA TQW/1360.,1500.,1510.,1500.,1470.,1470.,1490.,1490.,840.,
     1 830.,820.,790.,760.,1450.,1260.,1060.,960.,890.,850.,830.,
     1 795.,765.,745.,730.,720.,715.,710. /
C----- SORT TWALL FROM INPUT ZTQW AND TQW ARRAYS SET AS DATA ABOVE----
      WRITE(6,*)'   DATA            Z-DIST               TWALL'
      DO 100 IPP=1,NTQW
      ZTQW(IPP)=ZTQWD(IPP)*THROAT+ZTHRO
  100 WRITE(6,900)IPP,ZTQW(IPP),TQW(IPP)
C-pd---First cell center-------------------------------------------
      ZNODE=ZNOD(1)
C-pd---Find first data pt passed first cell center-------------------
      DO 200 IP1=1,NTQW
  200 IF(ZNODE.LT.ZTQW(IP1)) GO TO 220
  220 IPL=IP1
      IPLM=MAX0(IPL-1,1)
      TWALL(1)=TQW(IPL)+(ZTQW(IPL)-ZNODE)/(ZTQW(IPL)-ZTQW(IPLM)+
     1         1.E-10) *(TQW(IPLM)-TQW(IPL))
C-pd---All points up to first data point-----------------------------
      DO 240 IZ1=2,NZ
C-pd---Check if cell center exceeds first data poing------------------
      IF(ZNOD(IZ1).GT.ZTQW(IPL)) GO TO 300
  240 TWALL(IZ1)=TQW(IPL)+(ZTQW(IPL)-ZNOD(IZ1))/(ZTQW(IPL)-ZTQW(IPLM)+
     1          1.E-10)*(TQW(IPLM)-TQW(IPL))
C-pd---Calculations for most cells------------------------------------
  300 IZ=IZ1-1
      IPF=IPL
  320 IZ=IZ+1
      IF(IZ.GT.NZ) GO TO 400
      ZNODE=ZNOD(IZ)
C-pd---Cell center passed last data point-----------------------------
      IF(ZNODE.GE.ZTQW(NTQW)) GO TO 380
C-pd---Check for two data points in a cell----------------------------
      DO 340 IPP=IPF,NTQW
      IIIP1=MIN0(IPP+1,NTQW)
      IF(ZNODE.GE.ZTQW(IPP).AND.ZNODE.LT.ZTQW(IIIP1))GO TO 360
  340 CONTINUE
  360 IPF=MIN0(IPP,NTQW)
      IPFP=MIN0(IPF+1,NTQW)
C-pd---Twall for most interior points---------------------------------
      TWALL(IZ)=TQW(IPFP)+(ZTQW(IPFP)-ZNODE)/(ZTQW(IPFP)-ZTQW(IPF)+
     1          1.E-10) *(TQW(IPF)-TQW(IPFP))
      GO TO 320
  380 TWALL(IZ)=TQW(NTQW)
      GO TO 320
  400 CONTINUE
      DO 500 IZ=1,NZ
  500 WRITE(6,910) IZ,ZNOD(IZ),TWALL(IZ)
      RETURN
  900 FORMAT(3X,I4,8X,1P,E12.4,7X,0P,F10.2)
  910 FORMAT(' IZ Z ',I4,1P,E12.4,' TWALL=',1P,E12.4)
      END
C*********************************************************************
      SUBROUTINE WALDP(IZ,ISWEEP,LSWEEP,ISTSWP,MPHI,DY,AMU,DPDZP,WP,
     1                 RHOP,VALUE,COEF,INFO)
```

```
C*********************************************************************
C-------------------------------------------------------------------
C THIS SUBROUTINE CALCULATES THE WALL FUNCTIONS FOR FLOWS WITH SIGNI--
C FICANT AXIAL PRESSURE GRADIENTS. FOR REFERENCE SEE: T CEBECI AND----
C A.M.O. SMITH "A FINITE ...',ASME J. BAS. ENG.,1970,P 523, ALSO SEE--
C LAUNDER-SPALDING "MATHEMATICAL MODELS OF TURBULENCE', AP 1972.------
C-------------------------------------------------------------------
C-pd---The variables are defined as follows:------------------------
C---        DY     -->  wall to cell node distance
C---        NITER -->  number of iterations used to calculate s
C---        CAP    -->  von karman's constant
C---        CMUCD -->  turbulence constant
C---        TKMAX -->  maximum numerical value
C---        TKMIN -->  minimum numerical value
C---        Q116   -->  u+**3
C---        CC     -->  2(2-ln2) of Eq. B-30
C---        WP      ->  resultant velocity
C---        RHOP   -->  resultant density
C---        PPC    -->  p+/u+**3
C---        RE     -->  reynolds number
C---        E      -->  empirical constant
C---        CF     -->  skin friction coefficient
C---        S      -->  skin friction factor
C---        PPL    -->  p+
C---        DPDZP -->  resultant pressure gradient
C---        AMU    -->  absolute viscosity
C---        PPYP   -->  p+y+
C---        YPL    -->  y+
C---        UPL    -->  u+
C---        TAU    -->  wall shear stress
C---        UTAU   -->  friction velocity
C-------------------------------------------------------------------
C-pd---Constant initialization--------------------------------------
      DATA NITR,CAP,CMUCD,TKMAX,TKMIN /10,.4,.09,1.E5,1.E-5/
      Q116=11.6**3
      CC=4.-ALOG(4.)
      CAP2=CAP**2
C-pd---Calculation of skin friction factor for all variables---------
C-pd---Calculation of ppc and RE------------------------------------
      WP=ABS(WP)+1.E-5
      RHOP=AMAX1(.001,RHOP)
      PPC=DPDZP*AMU/((WP*RHOP)**2*WP)
      RE=RHOP*DY*WP/AMU
      E=9.
C-pd---Calculate coef for wl along with values for ke and ep---------
      VALUE=0.
      CF=AMU/DY
      S=.003
      IF(RE.LT.132.25) THEN
        IF(MOD(ISWEEP,ISTSWP).EQ.0.AND.MPHI.EQ.1) THEN
          WRITE(6,880)IZ,RE
          WRITE(6,890)RHOP,DY,WP,AMU
        ENDIF
        COEF=CF*(1.+PPC*RE/S)
        GOTO 200
```

```
      ENDIF
C-pd---Iterative calculation of the skin friction factor (s)----------
      DO 100 ITR=1,NITR
      PPYP=PPC*RE/S
      IF(PPYP.LE.-1.0) THEN
        IF(ITR.NE.NITR) GOTO 50
        IF(MOD(ISWEEP,ISTSWP).EQ.0.AND.MPHI.EQ.1)
     1     WRITE(6,900)PPYP,IZ,ISWEEP
   50   PPYP=-1.
      ENDIF
      SQ=1.+SQRT(1.+PPYP)
      SH=SQRT(S)
C-pd---S is calculated from eq B-30 with some rearrangement-----------
      S=CAP2/(ALOG(1.+E*RE*SH)-CC+2.*(SQ-ALOG(SQ)))**2
      S=AMAX1(1.E-6,S)
  100 CONTINUE
C-pd---Coef is set to rho*up*s----------------------------------------
      COEF=CF*RE*S
  200 IF(MPHI.NE.1) GOTO 300
      YPL=RE*SH
      UPL=1/SH
      PPL=PPC*UPL**3
      IF(INFO.GE.3) THEN
        WRITE(6,910) IZ,YPL,UPL,PPL,E,S,COEF
      ENDIF
      IF(INFO.LE.2.AND.ISWEEP.EQ.LSWEEP-1) THEN
        IF(IZ.EQ.1) WRITE(6,*)'  **************************  PRINTOUT OF
     1 WALL FUNCTION INFO AT LSWEEP  **************************'
        WRITE(6,910)IZ,YPL,UPL,PPL,E,S,COEF
      ENDIF
C-pd---Calculate value for ke-----------------------------------------
  300 IF(MPHI.NE.2) GO TO 400
      TAU=S*RHOP*WP**2
      VALUE=AMAX1(TKMIN,TAU/(RHOP*CMUCD**.5) )
      VALUE=AMIN1(TKMAX,VALUE)
      IF(INFO.GE.3) WRITE(6,*)' VALUE KE = ',VALUE
      COEF=1.E10
C-pd---Calculate value for ep-----------------------------------------
  400 IF(MPHI.NE.3) RETURN
      UTAU=WP*S**.5
      VALUE=UTAU**3/CAP/DY
      IF(INFO.GE.3) WRITE(6,*)' VALUE EP = ',VALUE
      COEF=1.E10
      RETURN
  880 FORMAT(' RE < 132.25 AT IZ =',I6,'    (RE) ',F8.2)
  890 FORMAT(' RHO D W U ',1P,4E12.4)
  900 FORMAT(' ****WARNING:  P+Y+ < -1.  (P+Y+,IZ,SWEEP)',1P,E12.4,2I7)
  910 FORMAT(' IZ Y+ U+ P+ E S COEF',I4,1P,6E12.4)
      END
C**********************************************************************
C     WRITE(6,*)' THE FOLLOWING VARIABLES ARE DEFINED AS:'
C     WRITE(6,*)'     INFO  -- A FLAG FOR PRINTOUT'
C     WRITE(6,*)'     TK    -- A GUESS FOR FINAL EQUILBRIUM TEMPERATURE'
C     WRITE(6,*)'     SM(1) -- GUESS FOR FINAL EQUIL. H2 MASS FRACTION'
C     WRITE(6,*)' THE FOLLOWING VARIABLES ARE DEFINED AS:'
```

```
C     WRITE(6,*)'    TENTH  -- THE INCOMING TEMPERATURE'
C     WRITE(6,*)'    SM1(1) -- THE IMCOMING H2 MASS FRACTION'
C*********************************************************************
      SUBROUTINE CHEMIC(ICALL,INFO,LEQUIL,NS,NLM,TK,PA,P0,HSUB0,
     1 Q0,GASCON,S1,S2,ATOMN,ATOMW,SMW,RHOP,WMOL,H0,S0)
C*********************************************************************
      PARAMETER (NSP=7,NLMP=4,NFRP=10)
      PARAMETER (NQP=NSP+2,NAP=NSP+3)
      DIMENSION S1(NS),S2(NS),ATOMN(NLM),ATOMW(NLM),SMW(NS)
      DIMENSION H0(NS),S0(NS)
      DIMENSION AL(NLMP,NSP),X(NQP),Y(NQP),B0(NLMP),PI(NLMP),A(NQP,NAP)
      DIMENSION ASUB(NSP,3),ID(NLMP,NFRP),BX(NFRP),TEN(NFRP),TACT(NFRP)
      DIMENSION BX2(NFRP),TEN2(NFRP),TACT2(NFRP)
      LOGICAL LEQUIL
      DATA NITRCH,TINYK,EPSS/50,1.E-20,.001/
      DATA TNY/-46.0517/
C
      IF(ICALL.GE.1) GO TO 200
      NSD=NSP
      NQD=NQP
      NAD=NAP
      NLMD=NLMP
      NFRD=NFRP
C----- INITIALIZATION AND INPUT DATA CHECKOUT.(NFREAC=NO OF FORWARD RE)
      RGSCN=1./GASCON
      LU1=4
      OPEN(LU1,FILE='H2.DAT')
      REWIND LU1
      N1 =NLM+1
      N2 =NLM+2
      N3 =NLM+3
      CALL CHEMIN(INFO,LU1,LEQUIL,NSD,NQD,NAD,NLMD,NFRD,NS,NSM,NA,NQ,
     1 NLM,N1,N2,N3,S2,ATOMN,ATOMW,AL,SMW,H0,S0,ASUB,BX,TEN,TACT,BX2,
     2 TEN2,TACT2,ID,X,Y)
      RETURN
C----- NORMALL CALL.
  200 TSAVE=TK
C
      CALL CHEMSO(INFO,NITRCH,LEQUIL,NS,NSM,NQ,NA,NLM,N1,N2,N3,SM,TK,
     1 PA,P0,GASCON,RGSCN,HSUB0,Q0,EPSS,TINYK,S1,S2,X,Y,H0,S0,B0,A,
     2 AL,PI,ASUB,TNY)
C
      WMOL=1./SM
      RETURN
      END
C*********************************************************************
      SUBROUTINE CHEMIN(INFO,LU1,LEQUIL,NSD,NQD,NAD,NLMD,NFRD,NS,NSM,
     1            NA,NQ,NLM,N1,N2,N3,S2,ATOMN,ATOMW,AL,SMW,H0,S0,ASUB,
     2            BX,TEN,TACT,BX2,TEN2,TACT2,ID,X,Y)
C*********************************************************************
C THIS IS THE INITIALIZING ROUTINE  READING INPUT DATA
C FOR: ELEMENT DATA DECK, THERMO DATA DECK, MECHANISM DATA DECK.
C BY : A.J. PRZEKWAS AND L.T. TAM ,  SEPT. 1986.
C---------------------------------------------------------------------
      COMMON/THERMD/Z(2,7,20)
```

```
      DIMENSION AL(NLM,NS),S2(NSD),ATOMN(NLMD),ATOMW(NLMD),
     1 H0(NSD),S0(NSD),SMW(NSD),ASUB(NSD,3),X(NQD),Y(NQD)
      DIMENSION DATA(12),AT(4),B(4),BX(NFRD),TEN(NFRD),TACT(NFRD),
     1          BX2(NFRD),TEN2(NFRD),TACT2(NFRD),ID(4,NFRD)
C
      DATA BLANK,REVRS,REMECH,THERM/4H    ,4HREVS,4HMECH,4HTHER/
      DATA THIRD,END ,GLOBAL/4HM   ,4HEND ,4HGLOB/
      DATA NPT,TFIT0,TFIT1,TENLN  /15,1000.,3000.,2.3025851/
C
    5 READ(LU1,'(12A4)') (DATA(I),I=1,12)
      IF(INFO.GE.8) WRITE(6,*) (DATA(I),I=1,12)
      IF(DATA(1).EQ.BLANK ) GO TO 5
      IF(DATA(1).EQ.THERM ) GO TO 20
      IF(DATA(1).EQ.REMECH) GO TO 30
      IF(DATA(1).EQ.END   ) GO TO 80
      GO TO 5
C=======READ THERMODYNAMIC JANNAF DATA TABLES. =======================
   20 IS=1
   21 READ(LU1,22) (DATA(I),I=1,3),(AT(J),B(J),J=1,4),T1,T2
   22 FORMAT(3A4,12X,4(A2,F3.0),1X,2F10.3)
      IF(DATA(1).EQ.BLANK) GO TO 29
      READ(LU1,'(5E15.8)') (Z(1,J,IS),J=1,5)
      READ(LU1,'(5E15.8)') (Z(1,J,IS),J=6,7),(Z(2,J,IS),J=1,3)
      READ(LU1,'(4E15.8)') (Z(2,J,IS),J=4,7)
      IF(INFO.LT.7) GO TO 23
      WRITE(6,22) (DATA(I),I=1,3),(AT(J),B(J),J=1,4),T1,T2
      WRITE(6,'(5E15.8)') (Z(1,J,IS),J=1,5)
      WRITE(6,'(5E15.8)') (Z(1,J,IS),J=6,7),(Z(2,J,IS),J=1,3)
      WRITE(6,'(4E15.8)') (Z(2,J,IS),J=4,7)
   23 CONTINUE
C----- ATOM STOICHIOMETRY .AL(L,N)=(KG-ATOMS OF ELEM. L PER
C       KG-MOLCULE OF SPEC J) AND ESTABLISH SPECIES MOL.WEIGTH SMW.
      DO 25 L=1,NLM
   25 AL(L,IS)=0.
      SUM=0.
      DO 27 K=1,4
      IF(B(K).EQ.0.) GO TO 27
        DO 26 L=1,NLM
        IF(ATOMN(L).NE.AT(K)) GO TO 26
        AL(L,IS)=AL(L,IS)+B(K)
        SUM=SUM+ATOMW(L)*B(K)
   26   CONTINUE
   27 CONTINUE
      SMW(IS)=SUM
C----- SAVE HOLLERITH NAME OF SPECIES.
      DO 28 I=1,3
   28 ASUB(IS,I)=DATA(I)
      IS=IS+1
      GO TO 21
   29 IS=IS-1
      NSM=IS+1
      NQ =IS+2
      NA =IS+3
      IF(INFO.GE.7)THEN
      WRITE(6,*)' AL PRINT , IS, NLM',IS,NLM
```

```
      DO 290 ISS=1,IS
  290 WRITE(6,*)(AL(L,ISS),L=1,NLM)
      ENDIF
      IF(IS.EQ.NS) GO TO 5
      WRITE(6,*)' WARNING YOU HAVE ABORTED EARLY  IS   NS',IS,NS
      STOP
C======READ MECHANISM/RATE DATA . DT1 (COL73-76) USED AS FLAG:=========
C  REVE --> REVERSE RATE DATA, IN SAME USITS AS FORWARD DATA.
C  GLOB --> GLOBAL RATE EXPRESSION.                    DT1 ,DT2 ARE
C  COMMENTS.  BX,TEN ARE RATE CONSTANT DEFINED AS  10**BX * TEMP**TEN,
C  TACT IS ACTIVATION TEMPERATURE (E/R) OR(EACT/GASCON) ,DEG K.
C
   30 JJ=1
   31 READ(LU1,32) (DATA(I),I=1,12),BX(JJ),TEN(JJ),TACT(JJ),DT1,DT2
CPD CHANG OF FORMAT
C  32 FORMAT(12A4,3F8.3,2A4)
   32 FORMAT(12A4,F7.4,F4.1,F11.2,2X,2A4)
      IF(DATA(1).EQ.BLANK) GO TO 49
CPD   THE FOLLOWING TWO LINES WERE MOVED FROM ABOVE THE IF
      IF(INFO.GE.7)
     1 WRITE(6,32) (DATA(I),I=1,12),BX(JJ),TEN(JJ),TACT(JJ),DT1,DT2
      IF(DT1    .NE.REVRS) GO TO 33
      J=JJ-1
      TEN2 (J)=TEN (JJ)
      TACT2(J)=TACT(JJ)
      BX2  (J)=10.**BX(JJ)
      GO TO 31
   33 BX(JJ)=10.**BX(JJ)
C---- SET ID(I,J) AS NUMBER OF THE I-TH SPECIE IN J-TH REACTION
C                   I=1,4 AS NO DISTINCT 3RD BODIES ARE CONSIDERED
      DO 34 I=1,4
   34 ID(I,JJ)=0
      ND=1
      DO 40 N=1,6
      K=N*2-1
      IF(DATA(K).EQ.BLANK) GO TO 40
      IF(DATA(K).NE.THIRD) GO TO 35
      DATA(K)=BLANK
      GO TO 40
   35 DO 36 I=1,NS
      IF(DATA(K  ).NE.ASUB(I,1)) GO TO 36
      IF(DATA(K+1).NE.ASUB(I,2)) GO TO 36
      II=I
      GO TO 37
   36 CONTINUE
   37 IF(K.GT.3) GO TO 38
      ID(ND,JJ)=II
      ND=ND+1
      GO TO 40
   38 IF(ND.EQ.2) ND=3
      ID(ND,JJ)=II
      ND=ND+1
  40  CONTINUE
C
C---- THE FOLLOWING SECTION UP TO STATEMENT 47 IS NOT USED IF REVERSE
```

```
C AS WELL AS FORWARD RATE DATA IS SUPPLIED FOR **ALL** REACTIONS.
C ELSE WE NEED TO USE LEAST-SQUARE LINEAR REGRESSION ANALYSIS FOR
C REVERSE RATE BASED ON FORWARD RATE DATA AND EQUILIBRIUM CONSTANTS.
C NPT= 15 POINTS ARE USED FOR FIT BETWEEN 1000 AND 3000 K.
C----- NOTE X=1/TEMPERATUE.
      IF(DT1.EQ.GLOBAL) GO TO 48
      XMAX =1./TFIT0
      XMIN =1./TFIT1
      DX   =(XMAX-XMIN)/FLOAT(NPT-1)
      SUMX =0.
      SUMY =0.
      DO 45 I=1,NPT
      X(I)=XMIN+DX*FLOAT(I-1)
      SUMX=SUMX+X(I)
      TK  =1./X(I)
      TLN=ALOG(TK)
      TKINV=1./TK
C
      CALL HCPS(4,INFO,TK,TLN,NS,HSUM,CPSUM,S2,H0,S0,S0SUM)
C
      SUM1=0.
      DO 42 ND=1,4
      K=ID(ND,JJ)
      IF(K.EQ.0) GO TO 42
      GF=H0(K)-S0(K)
      IF(ND.LT.3) SUM1=SUM1+GF
      IF(ND.GE.3) SUM1=SUM1-GF
   42 CONTINUE
      SUM1=EXP(SUM1)
      TM1=1.
      IF(ID(2,JJ).EQ.0) TM1=.082057*TK
      IF(ID(4,JJ).EQ.0) TM1=1./(.082057*TK)
      AK=BX(JJ)*EXP(-TACT(JJ)*TKINV) *TK**TEN(JJ)
      AK=AK*TM1/SUM1
      Y(I)=ALOG(AK)
   45 SUMY=SUMY+Y(I)
      XBAR=SUMX*6.66666667E-2
      YBAR=SUMY*6.66666667E-2
      SUMX=0.
      SUMY=0.
      SUM1=0.
      DO 47 I=1,NPT
      SUMX=SUMX + (X(I)-XBAR)*Y(I)
      SUMY=SUMY + (Y(I)-YBAR)**2
   47 SUM1=SUM1 + (X(I)-XBAR)**2
      TEN2 (JJ)=0.
      TACT2(JJ)=-SUMX/SUM1
      BX2  (JJ)=10.**((YBAR+TACT2(JJ)*XBAR)/TENLN)
   48 JJ=JJ+1
      GO TO 31
   49 JJ=JJ-1

   80 IF(INFO.LT.6) RETURN
C----- PRINTOUT ------------------------------------------
      DO 85 J=1,JJ
```

```
      DO 83 N=1,6
      K=N*2-1
      L=N
      IF(N.GT.3) L=N-1
      DATA(K  )=BLANK
      DATA(K+1)=BLANK
      IF(N.EQ.3) GO TO 83
      IF(N.EQ.6) GO TO 83
      IF(ID(L,J).EQ.0) GO TO 83
      DATA(K  )=ASUB(ID(L,J),1)
      DATA(K+1)=ASUB(ID(L,J),2)
   83 CONTINUE
      IF(ID(2,J).EQ.0) DATA(5)=THIRD
      IF(ID(4,J).EQ.0) DATA(5)=THIRD
      DATA(11)=DATA(5)
      WRITE(6,84) J,(DATA(K),K=1,12)
   84 FORMAT(5X,I5,1H.,5X,6A4,5H---->,4X,6A4/)
   85 CONTINUE
C
      RETURN
      END
C*********************************************************************
      SUBROUTINE CHEMSO(INFO,NITRCH,LEQUIL,NS,NSM,NQ,NA,NLM,N1,N2,N3,
     1 SM,TK,PA,P0,GASCON,RGSCN,HSUB0,Q0,EPSS,TINYK,S1,S2,X,Y,H0,S0,
     2 B0,A,AL,PI,ASUB,TNY)
C*********************************************************************
C THIS ROUTINE CALLS CMPCHE TO COMPUTE THE CORRECTIONS TO THE CHEMICAL
C SPECIES AND TEMPERATURE AND DETERMINES THE UNDERRELAXATION PRIOR TO
C THE APPLICATION OF THESE CORRECTIONS TO THE ESTIMATES FOR BOTH EQUI-
C LIBRIUM AND KINETIC STATIONARY STATES FOR EACH ITERATION.
C CHEMSO ALSO CONTROLLS THE CONVERGENCE TESTS.
C----------------------------------------------AJP & LTT SEPT,1986.
      COMMON/THERMD/Z(2,7,20)
      DIMENSION S1(NS),S2(NS),X(NQ),Y(NQ),H0(NS),S0(NS),B0(NLM),
     1 PI(NLM),A(NQ,NA),AL(NLM,NS),ASUB(NS,3)
      LOGICAL LCONVG,LEQUIL
      DATA ALN1E4   /9.2103404/
C
      LCONVG=.FALSE.
      SM=0.
      DO 10 IS=1,NS
      S2(IS)=AMAX1(S2(IS),TINYK)
      SM=SM+S2(IS)
      Y(IS)=ALOG(S2(IS))
   10 X(IS)=0.
      IF(INFO.GE.7) WRITE(6,*)' S1',(S1(IS),IS=1,NS)
      IF(INFO.GE.7) WRITE(6,*)' S2',(S2(IS),IS=1,NS)
      Y(NSM)=ALOG(SM)
      X(NSM)=0.
      TLN=ALOG(TK)
      TKINV =1./TK
      PRAT=PA/P0
      PPLN=ALOG(PRAT)
      SMINV =1./SM
      Y(NQ) =TLN
```

```
      IMAT  =NQ
      IF(LEQUIL) IMAT=N2
      KMAT=IMAT+1
      IF(INFO.GE.7) WRITE(6,*)' SM,TK,PA,P0',SM,TK,PA,P0
C----- ITERATION LOOP.  SOLVE FOR CORRECTIONS.
      DO 500 ITRCH=1,NITRCH
      IHCPS=3
      IF(.NOT.LEQUIL) IHCPS=2
      CALL HCPS(IHCPS,INFO,TK,TLN,NS,HSUM,CPSUM,S2,H0,S0,S0SUM)
C
      CALL COMPCH(INFO,NLM,NQ,NA,NS,NSM,N1,N2,N3,LEQUIL,GASCON,RGSCN,
     1  TK,TKINV,PPLN,SM,HSUB0,Q0,HSUM,CPSUM,H0,S0,B0,S2,A,AL,Y,S1)
C
      IF(INFO.LE.5) GO TO 105
      WRITE(6,'('' ELEMENTS A(I,K) OF CORRECTION  MATRIX''/)')
      DO 103 K=1,IMAT
  103 WRITE(6,'(1P,11E10.2)') (A(K,I),I=1,KMAT)
C
  105 CONTINUE
C----- SOLVE FOR CORRECTIONS BY PIVOTAL GAUSIIAN ELIMINATION.
C       WE CAN TEST FOR SINGULAR MATRIX (A(NN,NN)=0.) INSIDE 120 LOOP.
      DO 120 NN=1,IMAT
      K=NN+1
      DTM1=1./A(NN,NN)
      DO 112 J=K,KMAT
  112 A(NN,J)=A(NN,J)*DTM1
      IF(K.EQ.KMAT) GO TO 120
      DO 115 I=K,IMAT
      IF(A(I,NN).EQ.0.) GO TO 115
      DO 114 J=K,KMAT
  114 A(I,J)=A(I,J)-A(I,NN)*A(NN,J)
  115 CONTINUE
  120 CONTINUE
C----- BACKSUBSTITUTION.
      K=IMAT
  131 J=K+1
      SUM=0.
      X(K)=0.
      IF(IMAT.LT.J) GO TO 134
      DO 133 I=J,IMAT
  133 SUM=SUM+A(K,I)*X(I)
  134 X(K)=A(K,KMAT)-SUM
      K=K-1
      IF(K.NE.0) GO TO 131
C
      IF(.NOT.LEQUIL) GO TO 150
C----- EQUILIBRIUM -CONSTRUCTION OF CORRECTIONS FOR SPECIES.
      DO 141 L=1,NLM
  141 PI(L)=X(L)
      X(NSM)=X(N1)
      X(NQ )=X(N2)
      DO 142 IS=1,NS
      X(IS)=H0(IS)*X(NQ)-(H0(IS)-S0(IS)+Y(IS)+PPLN-Y(NSM))+X(NSM)
      DO 142 L=1,NLM
  142 X(IS)=X(IS)+AL(L,IS)*PI(L)
```

```
C----CALCULATE UNDERRELAXATION PARAMETER ETA. UNDERRELAXATIONS ARE
C    DIFFERENT FOR MAJOR AND MINOR SPECIES WITH: ETA=MIN(ETA1,ETA2,1.)
C       FOR MAJOR SPECIES  ---> S2(I)/SM >1.E-8  USE ETA1,
C       FOR MINOR SPECIES  ---> S2(I)/SM <1.E-8  USE ETA2   ,  AND
C    ANY POSITIVE CORRECTION CHANGES FOR MOLE NUMBERS ARE MONITORED.
C
  150 ETA1=1.
      SUM=AMAX1(ABS(X(NSM)),ABS(X(NQ)))
      DO 155 IS=1,NS
      IF(X(IS).LE.0.) GO TO 155
      SUM=AMAX1(X(IS),SUM)
      IF(S2(IS)/SM.LE.1.E-8) ETA1=
     1   AMIN1((ABS(Y(NSM)-Y(IS)-ALN1E4)/ABS(X(IS)-X(NSM))),ETA1)
  155 CONTINUE
      ETA=AMIN1(ETA1,2./(SUM+1.E-10))
C----- APPLY CORRECTIONS TO ESTIMATES.
      DO 160 IS=1,NS
      Y(IS)=AMAX1(Y(IS)+ETA*X(IS),TNY)
  160 S2(IS)=EXP(Y(IS))
      Y(NSM)=Y(NSM)+ETA*X(NSM)
      SM    =EXP(Y(NSM))
      SMINV =1./SM
      Y(NQ) =Y(NQ)+ETA*X(NQ)
      TLN=Y(NQ)
      TK    =EXP(TLN)
      TKINV =1./TK
C
      IF(INFO.GE.7) WRITE(6,166) ITRCH,ETA,SM,TK,X(NSM),X(NQ),
     1                          (ASUB(IS,1),S2(IS),Y(IS),X(IS) ,IS=1,NS)
  166 FORMAT(' ITER=',I4,' ETA=',1P,E11.3,' SM TK X(NM),X(MQ)',
     1 1P,4E11.3/15X,'S2',12X,'LOGS2',8X,'D(LODS2)'/(2X,A4,1P,3E14.6))
C----- CONVERGENCE CHECK. ALL MOLE NUMBER CORRECTIONS MUST BE < 1.
      IF(ETA.LT.1.) GO TO 500
      DO 170 IS=1,NS
      IF(S2(IS).LE.TINYK*1.001) GO TO 170
      IF(ABS(X(IS)).GT.EPSS) GO TO 500
  170 CONTINUE
      LCONVG=.TRUE.
      HSUB0=HSUM*GASCON*TK
C
      IF(.NOT.LEQUIL) GO TO 200
      DO 180 IS=1,NS
CPD   1.E-8 WAS CHANGED TO 1.E-10
  180 S2(IS)=AMAX1(S2(IS),1.E-20)
      GO TO 900
C
C----- KINETICS CONTINUES FURTHER ...........................
  200 CONTINUE
C
  500 CONTINUE
C
  900 IF(INFO.GE.7)WRITE(6,'('' END OF ITER-CHEMSO ITER='',I4)')ITRCH
      RETURN
      END
C**********************************************************************
```

```
      COMMON/LDATA/CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32(17),SAVGEO,
     1RSTGEO,NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,
     1DONACC,PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,
     1USEGRD,USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP(11)
C
C---------------------------------------------------------- LDEBUG
      COMMON/LDEBUG/DBGEOM,DBADJS,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37(9)
C
C---------------------------------------------------------- IDATA
      COMMON/IDATA/NX,NY,NZ,LUPR1,LUPR2,LUPR3,LUPHUN,LUSDA,IPROF,
     1LUFI,LUDST,LUGRF,LUSAVE,LUOLD,LUDEP,LUPCO,LUDVL,
     1IRUNN,IOPTN,LITC,LITFLX,NRUN,LITHYD,FSTEP,LSTEP,
     1FSWEEP,LSWEEP,NPRINT,LIBREF,MEANDF,IXMON,IYMON,IZMON,IINIT,
     1NLSG1,NISG1,NRSG1,NCSG1,IPARAB,IDPHUN,NXFR1,NYFR1,NZFR1,
     1NTFR1,ENTH1,ENTH2,ISWR1,ISWR2,IXPRF,IXPRL,IYPRF,IYPRL,
     1NPRMNT,ISTPRL,ISTPRF,IZPRL,IZPRF,NUMCLS,TSTSWP,NYPRIN,NXPRIN,
     1NZPRIN,NPRMON,NTPRIN,NTZPRF,ISP66,IURINI,IURPRN,IURVAL,
     1I0RTCV,NUMREG,NRTCV,ICHR,INTFRC,ITHC1,ISWC1,DEN1,DEN2,
     1VISL,INTMDT,ISWPRF,ISWPRL,IPSA,ISP84,IPLTF,IPLTL,NPLT,ITABL,
     1TEMP1,TEMP2,LEN1,LEN2,NLG1,NIG1,NRG1,NCG1,NPNAM1,
     1ISP98(3),LENREC,LUGEOM,IMB1,IMB2,PCOR,NCOLPF,NCOLCO,
     1NROWCO,EPOR,NPOR,HPOR,VPOR,KXFR,KYFR,KZFR,KTFR,IDATSP(2),
     1VIST,NPHI
C
C---------------------------------------------------------- IDEBUG
      COMMON/IDEBUG/IZDB1,IZDB2,ITHDB1,ITHDB2,ISWDB1,ISWDB2,ISTDB1,
     1ISTDB2,INCHCK,IREGDB,NFMAX,IDBF0,IDBCMN,IDBGRD,IDEBSP(2)
C
C---------------------------------------------------------- HDATA
      COMMON/HDATA/MESS(10),NBLANK,NAMGRD,NAMEJ,NAMEJ1,
     1NAMEM,NAMEM1,NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,
     1NPHUN,NHINIT,NDST,NAMSAT,NGEOM,NHDASP(2)
C
C---------------------------------------------------------- HDEBUG
      COMMON/HDEBUG/NDBF0(2),NDBCMN(2),NHDBSP
C
C---------------------------------------------------------- RDATA
      COMMON/RDATA/TINY,GREAT,RUPLIM,RLOLIM,AZDZ,AZXU,AZYV,
     1AZRI,AZAL,AZPH,XULAST,YVLAST,ZWLAST,TLAST,TFIRST,PBAR,SNALFA,
     1RINNER,ENUL,ENUT,RHO1,RHO2,CFIPS,CMDOT,CONMDT,GRND,HEATBL,
     1FIXFLU,READFI,ZMOVE1,ZDIFAC,DRH1DP,DRH2DP,U1AD,U2AD,V1AD,
     1V2AD,W1AD,W2AD,HUNIT,DIFCUT,ABSIZ,ORSIZ,OPPVAL,TMP1,TMP2,
     1EL1,EL2,GRND1,GRND2,GRND3,GRND4,GRND5,GRND6,GRND7,GRND8,GRND9
     1,GRND10,ZWADD,RINIT,SAME,FIXVAL,AXDZ,AYDZ,RDATSP(21)
C
C---------------------------------------------------------- RDEBUG
      COMMON/RDEBUG/BGCHCK,SMCHCK,RDEBSP(5)
C
```

```
      SUBROUTINE COMPCH(INFO,NLM,NQ,NA,NS,NSM,N1,N2,N3,LEQUIL,GASCON,
     1  RGSCN,TK,TKINV,PPLN,SM,HSUB0,Q0,HSUM,CPSUM,H0,S0,B0,S2,A,AL,
     2  Y,S1)
C*********************************************************************
C THIS SUBROUTINE CONSTRUCTS THE NEWTON-RAPHSON DERIVATIVE MATRIX FOR
C BOTH KINETIC AND EQUILIBRIUM SOLUTIONS AND SOLVES IT BY PIVOTAL
C GAUSSIAN REDUCTION. CALLED FROM IREATION LOOP OF SUBROUTINE  CHEMSO.
C---------------------------------------------------------------------
      DIMENSION A(NQ,NA),B0(NLM),H0(NS),S0(NS),S2(NS),Y(NQ),
     1          AL(NLM,NS),S1(NS)
      LOGICAL LEQUIL
C
      DO 10 I=1,NQ
      DO 10 K=1,NA
   10 A(I,K)=0.
C----- NONDIMENSIONAL ENTHALPY (H/RT) AND HEAT SOURCE (Q/RT).
      HIN   =HSUB0*RGSCN*TKINV
      Q     =Q0   *RGSCN*TKINV
C
      IF(LEQUIL) GO TO 400
C----- KINETIC SOLUTION SETUP ----------
C
      RETURN
C----- EQUILIBRIUM SOLUTION SETUP ------
  400 HSUM=0.
      SUM =0.
      IF(INFO.GE.7)THEN
      WRITE(6,*)' AL PRINT , NS, NLM',NS,NLM
      DO 290 ISS=1,NS
  290 WRITE(6,292)ISS,S2(ISS),(AL(L,ISS),L=1,NLM)
  292 FORMAT(' ISS S2',I3,1P,E11.3,' AL ',1P,6E11.3)
      ENDIF
      DO 410 L=1,NLM
  410 B0(L)=0.
      DO 450 IS=1,NS
      SUM =SUM +S2(IS)
      TM1 =H0(IS)*S2(IS)
      HSUM=HSUM+TM1
      TM2 =(H0(IS)-S0(IS)+Y(IS)-Y(NSM)+PPLN)*S2(IS)
      A(N1,N3)=A(N1,N3)+TM2
      A(N2,N2)=A(N2,N2)+H0(IS)*TM1
      A(N2,N3)=A(N2,N3)+H0(IS)*TM2
      DO 440 L=1,NLM
      IF(AL(L,IS).EQ.0.) GO TO 440
      TM3=AL(L,IS)*S2(IS)
C----- CROSS-DERIVATIVES OF ELEMENT EQUATIONS D( F(L) )/D( PI(K) ).
      DO 430 K=L,NLM
  430 IF(AL(K,IS).NE.0.) A(L,K)=A(L,K)+AL(K,IS)*TM3
C----- DERIVATIVES OF L-ELEMENT EQN W.R.T.  LN(SM) AND LN(T)
      A(L,N1)=A(L,N1)+TM3
      A(L,N2)=A(L,N2)+AL(L,IS)*TM1
      A(L,N3)=A(L,N3)+AL(L,IS)*TM2
      B0(L)  =B0(L)  +AL(L,IS)*S1(IS)
  440 CONTINUE
  450 CONTINUE
```

```
C----- NEGATIVE OF L-ELEMENTEQTS  F(L).
      DO 460 L=1,NLM
  460 A(L,N3)=A(L,N3)+B0(L)-A(L,N1)
      A(N1,N1)=SUM-SM
      A(N1,N2)=HSUM
      A(N1,N3)=A(N1,N3)-(SUM-SM)
      A(N2,N2)=A(N2,N2)+CPSUM
      A(N2,N3)=A(N2,N3)+HIN-HSUM-Q
C-----STORE SYMMETRIC ELEMENTS OF MATRIX
      DO 470 I=1,N2
      DO 470 J=1,N2
  470 A(J,I)=A(I,J)
      IF(INFO.LE.6) GO TO 105
      WRITE(6,'('' ELEMENTS A(I,K) OF CORRECTION  MATRIX''/)')
      DO 103 K=1,N2
  103 WRITE(6,'(1P,11E10.2)') (A(K,I),I=1,N3)
C
  105 CONTINUE
C----- INTERCHANGE SM-EQN WITH ELEMENT ROW L WITH LARGEST A(L,N1)
C      AVOID POTENTIAL ZERO IN DIAGONAL ALEMENT(N1,N1)
      TM1=0.
      DO 482 L=1,NLM
      IF(A(L,N1).LT.TM1) GO TO 482
      TM1=A(L,N1)
      LL=L
  482 CONTINUE
      DO 484 J=1,N3
      TM1=A(N1,J)
      A(N1,J)=A(LL,J)
      A(LL,J)=TM1
  484 CONTINUE
C
      RETURN
      END
C*******************************************************************
      SUBROUTINE HCPS(IHCPS,INFO,TK,TLN,NS,HSUM,CPSUM,SC,H0,S0,S0SUM)
C*******************************************************************
C THIS SUBSOUTINE CALCULATES THE NONDIMENSIONAL, 1-ATM VALUES OF EN-
C THALPY, CPECIFIC HEAT AND ENTROPY FOR A GIVEN TEMPERATURE TK(DEG K).
C-------------------------------------------------------------------
      COMMON/THERMD/Z(2,7,20)
      DIMENSION SC(NS),H0(NS),S0(NS)
C
      K=1
      IF(TK.LT.1000.) K=2
      TKINV=1./TK
      TK2  =TK*TK
      TK3  =TK*TK2
      TK4  =TK*TK3
      TM   =.5      *TK
      TM2  =.3333333*TK2
      TM3  =.25     *TK3
      TM4  =.2      *TK4
      GO TO (10,20,20,30),IHCPS
C----- IHCPS=1 -----> H0(NS) AND MIXTURE ENTHALPY HSUM.
```

```
   10 HSUM=0.
      DO 11 IS=1,NS
      H0(IS)=TM4*Z(K,5,IS)+TM3*Z(K,4,IS)+TM2*Z(K,3,IS)+TM *Z(K,2,IS)+
     1      TKINV*Z(K,6,IS)+    Z(K,1,IS)
   11 HSUM=HSUM+H0(IS)*SC(IS)
      RETURN
C----- IHCPS=2 -----> H0(NS) ENTHALPY AND CP FOR MIXTURE.
   20 HSUM =0.
      CPSUM=0.
      DO 21 IS=1,NS
      H0(IS)=TM4*Z(K,5,IS)+TM3*Z(K,4,IS)+TM2*Z(K,3,IS)+TM*Z(K,2,IS)+
     1      TKINV*Z(K,6,IS)+    Z(K,1,IS)
      CPSUM=CPSUM+ SC(IS)*(TK4*Z(K,5,IS)+TK3*Z(K,4,IS)+TK2*Z(K,3,IS)+
     1                  TK *Z(K,2,IS)+    Z(K,1,IS))
   21 HSUM=HSUM+H0(IS)*SC(IS)
      IF(IHCPS.EQ.3) GO TO 30
      RETURN
C----- IHCPS=3 -----> S0(NS)
   30 TK2 =TK2*.5
      TK3 =TK3*.333333333
      TK4 =TK4*.25
CPD--SUM S0
      S0SUM=0.0
      DO 31 IS=1,NS
C  31 S0(IS)=TK4*Z(K,5,IS)+TK3*Z(K,4,IS)+TK2*Z(K,3,IS)+TK*Z(K,2,IS)+
C    1        TLN*Z(K,1,IS)+    Z(K,7,IS)
      S0(IS)=TK4*Z(K,5,IS)+TK3*Z(K,4,IS)+TK2*Z(K,3,IS)+TK*Z(K,2,IS)+
     1        TLN*Z(K,1,IS)+    Z(K,7,IS)
   31 S0SUM=S0SUM+S0(IS)*SC(IS)
      IF(IHCPS.EQ.4) GO TO 40
      RETURN
C----- IHCPS=4 -----> H0(NS)  AND S0(NS).
   40 DO 41 I=1,NS
   41 H0(IS)=TM4*Z(K,5,IS)+TM3*Z(K,4,IS)+TM2*Z(K,3,IS)+TM*Z(K,2,IS)+
     1      TKINV*Z(K,6,IS)+    Z(K,1,IS)
      RETURN
      END
C*********************************************************************
      SUBROUTINE DARTH(GZNODE,GYWALL,ZTHRO,NCHA,NTUB,NCOM,NNOZ,RATEL,
     &                 RATEU,CXAREA,WTHK,DIAHYD,FT1,FT2,IBJ,GDIST,
     &                 INFO,GPI100)
C*********************************************************************
C---
Cinclude "satear"
C FILE NAME SATEAR --- 170486
CNLIST
C
C--------------------------------------------------- ARRAYS
      COMMON/LDB1/DBGPHI(50)/IDA1/ITERMS(50)/IDA2/LITER(50)
     1/IDA3/IORCVF(50)/IDA4/IORCVL(50)/IDA5/ISLN(50)/IDA6/IPRN(50)
     1/HDA1/NAME(50)/RDA1/DTFALS(50)/RDA2/RESREF(50)
     1/RDA3/PRNDTL(50)/RDA4/PRT(50)/RDA5/ENDIT(50)/RDA6/VARMIN(50)
     1/RDA7/VARMAX(50)/RDA8/FIINIT(50)/RDA9/PHINT(50)
     1/RDA10/CINT(50)/RDA11/EX(50)
C--------------------------------------------------- LDATA
```

```
C--------------------------------LOGICAL DECLARATIONS
      LOGICAL LDAT,LDEB
      LOGICAL CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32,SAVGEO,RSTGEO,
     1NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,DONACC,
     1PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,USEGRD,
     1USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP
      LOGICAL DBGEOM,DBADJS,DBGPHI,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37
C
C--------------------------------INTEGER DECLARATIONS
      INTEGER FSTEP,FSWEEP,TSTSWP,ENTH1,ENTH2,DEN1,
     1DEN2,PCOR,VISL,EPOR,HPOR,VPOR,VIST,TEMP1,TEMP2
C--------------------------------CHARACTER DECLARATIONS
      CHARACTER*4 NHDAT,NHDEB
      CHARACTER*4 NAME
      CHARACTER*4 MESS,NBLANK,NAMGRD,NAMEJ,NAMEJ1,NAMEM,NAMEM1,
     1NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,NPHUN,NHINIT,
     1NDST,NAMSAT,NGEOM,NHDASP
      CHARACTER*4 NDBF0,NDBCMN,NHDBSP
C--------------------------------EQUIVALENT TRANSMISSION ARRAYS
      DIMENSION LDAT(84),LDEB(45),IDAT(120),IDEB(16),NHDAT(30),
     1NHDEB(5),RDAT(85),RDEB(7)
      EQUIVALENCE (LDAT(1),CARTES),(LDEB(1),DBGEOM),(IDAT(1),NX),
     1(IDEB(1),IZDB1),(NHDAT(1),MESS(1)),(NHDEB(1),NDBF0(1)),
     1(RDAT(1),TINY),(RDEB(1),BGCHCK)
CLIST
#include "grdear"
#include "grdloc"
C---
      PARAMETER(NXM=1,NYM=50,NZM=300,NZP=NZM+1,NDM=160)
C---
      DIMENSION GZNODE(NZM),GYWALL(NZP),WTHK(NZM),XAREA(NZM),
     &          CXAREA(NZM),DIAHYD(NZM),FT1(NZM),FT2(NZM),GDIST(NZM)
      DIMENSION XJ(NDM),ACJ(NDM),DELJ(NDM),FTA(NDM),FTB(NDM)
C---
      DATA CONV,CONV2,GPI/39.37008,1550.,3.1415926/
      DATA XJ/-13.9999,-13.5001,-12.7648,-12.0001,-11.4998,
     &        -11.0000,-10.8001,-10.6001,-10.4002,-10.1998,
     &         -9.9998, -9.7999, -9.6000, -9.4001, -9.2001,
     &         -9.0002, -8.7998, -8.5999, -8.3999, -8.2000,
     &         -8.0001, -7.8002, -7.6002, -7.3998, -7.1999,
     &         -6.9999, -6.8000, -6.6001, -6.4002, -6.2002,
     &         -5.9998, -5.7999, -5.6000, -5.4000, -5.2001,
     &         -5.0000, -4.8000, -4.6000, -4.4000, -4.2000,
     &         -4.0000, -3.9000, -3.8000, -3.7000, -3.6000,
     &         -3.5000, -3.4000, -3.3000, -3.2000, -3.1000,
     &         -3.0000, -2.9000, -2.8000, -2.7000, -2.6000,
     &         -2.5000, -2.4000, -2.3000, -2.2000, -2.1000,
     &         -2.0000, -1.9000, -1.8000, -1.7000, -1.6000,
```

```
&         -1.5000, -1.4000, -1.3000, -1.2000, -1.1000,
&         -1.0000, -0.9000, -0.8000, -0.7000, -0.6000,
&         -0.5000, -0.4000, -0.3000, -0.2000, -0.1000,
&          0.0000,  0.1000,  0.2000,  0.3000,  0.4000,
&          0.6000,  0.8000,  1.0000,  1.5000,  2.0000,
&          2.5000,  3.0000,  3.5000,  4.0000,  4.5000,
&          5.0000,  5.9998,  6.9999,  8.0001,  9.7298,
&          9.7500, 10.0000, 10.5000, 11.0000, 11.5000,
&         12.0000, 12.5000, 13.0000, 13.5000, 14.0000,
&         14.5000, 15.0000, 15.5000, 16.0000, 16.5000,
&         17.0000, 17.5000, 18.0000, 18.5000, 19.0000,
&         19.5000, 20.0000, 20.5000, 21.0000, 21.5000,
&         22.0000, 23.0000, 24.0000, 26.0000, 28.0000,
&         30.0000, 32.0000, 34.0000, 36.0000, 38.0000,
&         40.0000, 42.0000, 44.0000, 46.0000, 48.0000,
&         52.0000, 56.0001, 60.0001, 64.0001, 68.0002,
&         72.0002, 75.9997, 79.9998, 83.9998, 87.9999,
&         91.9999, 96.0000,100.0000,104.0000,108.0000,
&        112.0001,116.0002,117.9999,119.0001,119.5782/
 DATA ACJ/0.01550,0.01426,0.00608,0.00608,0.00608,
&         0.00608,0.00608,0.00608,0.00608,0.00608,
&         0.00608,0.00608,0.00608,0.00608,0.00608,
&         0.00608,0.00608,0.00608,0.00608,0.00608,
&         0.00608,0.00608,0.00608,0.00608,0.00608,
&         0.00587,0.00565,0.00542,0.00521,0.00501,
&         0.00493,0.00492,0.00490,0.00480,0.00464,
&         0.00449,0.00433,0.00436,0.00424,0.00412,
&         0.00399,0.00393,0.00387,0.00380,0.00374,
&         0.00368,0.00362,0.00356,0.00349,0.00343,
&         0.00337,0.00333,0.00337,0.00341,0.00345,
&         0.00349,0.00354,0.00358,0.00362,0.00366,
&         0.00370,0.00374,0.00378,0.00382,0.00386,
&         0.00390,0.00394,0.00399,0.00403,0.00407,
&         0.00411,0.00415,0.00419,0.00423,0.00427,
&         0.00432,0.00436,0.00440,0.00444,0.00448,
&         0.00452,0.00462,0.00473,0.00486,0.00505,
&         0.00548,0.00605,0.00680,0.00768,0.00780,
&         0.00784,0.00952,0.01138,0.01244,0.01277,
&         0.01337,0.01420,0.01488,0.01488,0.01488,
&         0.00474,0.00495,0.00531,0.00571,0.00613,
&         0.00656,0.00696,0.00737,0.00783,0.00827,
&         0.00866,0.00906,0.00947,0.00988,0.01025,
&         0.01068,0.01113,0.01158,0.01200,0.01247,
&         0.01294,0.01343,0.01393,0.01429,0.01453,
&         0.01468,0.01498,0.01520,0.01559,0.01594,
&         0.01620,0.01642,0.01660,0.01673,0.01678,
&         0.01682,0.01713,0.01774,0.01872,0.01973,
&         0.02167,0.02364,0.02548,0.02717,0.02904,
&         0.03101,0.03230,0.03385,0.03537,0.03675,
&         0.03798,0.03930,0.04040,0.04158,0.04267,
&         0.04374,0.04464,0.04501,0.04445,0.04453/
 DATA DELJ/32*0.035,57*0.028,11*0.035,34*0.0077,
&          0.00780,
&          0.00805,0.00815,0.00835,0.00855,0.00875,
&          0.00895,0.00925,0.00945,0.00975,0.00985,
```

```
          GO TO 30
        END IF
 25     CONTINUE
 30   IF(JZ.LE.IBJ)THEN
        TOTARE=FLOAT(NCHA)*XAREA(JZ)
      ELSE
        TOTARE=FLOAT(NTUB)*XAREA(JZ)
      END IF
      RIN=GYWALL(JZ)+WTHK(JZ)
      ROUT=SQRT(TOTARE/GPI+RIN**2)
      CXAREA(JZ)=TOTARE/GPI100
      DIAHYD(JZ)=2.0*(ROUT-RIN)
      IF(JZ.EQ.1.AND.INFO.EQ.2)THEN
        WRITE(6,*)'*****************************************************'
        WRITE(6,*)'***....... COOLING JACKET SIMULATION DATA ......***'
        WRITE(6,*)'*****************************************************'
        WRITE(6,300)
        WRITE(6,350)
      END IF
      GDIST(JZ)=DELT/CONV
      IF(INFO.EQ.2)
     1 WRITE(6,400)JZ,GDIST(JZ),CXAREA(JZ),TOTARE,DIAHYD(JZ),GYWALL(JZ),
     1              WTHK(JZ),FT1(JZ),FT2(JZ)
 20   CONTINUE
C---
      IF(JLST.NE.NZ)THEN
        JFST=IBJ+1
        JLST=NZ
        NFST=NLST+1
        NLST=NDEL
        GO TO 15
      END IF
C**********************************************************************
 300  FORMAT(/2X,
     1     '-------------------- DEFINITIONS --------------------',//2X,
     1     ' IZ       : AXIAL GRID NUMBER',/2X,
     2     ' ZDIST    : AXIAL DISTANCE FROM THROAT (m)',/2X,
     3     ' FLWA     : CROSS SECTIONAL FLOW AREA PER TUBE (m2)',/2X,
     4     ' TOTAR    : TOTAL CROSS SECTIONAL FLOW AREA (m2)',/2X,
     5     ' DHYD     : HYDRAULIC DIAMETER (m)',/2X,
     6     ' RAD1     : HOT GAS WALL RADII (m)',/2X,
     7     ' DELJ     : WALL THICKNESS (m)',/2X,
     8     ' FT1,FT2  : PARAMETRIC FACTORS',//2X,
     9     '-----------------------------------------------------')
 350  FORMAT(/2X,'  IZ   ZDIST       FLWA       TOTAR       DHYD        ',
     1          'RAD1        DELJ        FT1         FT2     ',
     2     /2X,'         (m)        (m2)       (m2)        (m)         ',
     3          '(m)         (m)',
     4     /2X,'================================================',
     5          '=========================================')
 400  FORMAT(2X,I4,8(1P,E11.3))
C**********************************************************************
      RETURN
      END
C**********************************************************************
```

```
      SUBROUTINE TWCOOL(TGAS,TGASW,TLIQ,GHGAS,IBJ,TLIQL,TLIQU,COPPK,
     1                  STEEK,VISHYD,PRHYD,FLXINL,FLXINU,RATEL,RATEU,
     2                  GDIST,CXAREA,DIAHYD,WTHK,FT2,LHLEN,INFO,GPI100)
C***********************************************************************
C----------------------------------------------------------------------
C---  This subroutine calculates the heat flux across the thrust
C---  chamber wall into a simulated cooling jacket.  With application
C---  of appropriate heat balance equations 1-D profiles of the
C---  following variables are determined along the thrust chamber wall.
C---
C---      GHGAS  -->  Heat transfer coefficient on the gas side
C---      GHWAL  -->  Heat transfer coefficient through the chamber wall
C---      GHLIQ  -->  Heat transfer coefficient on the liquid side
C---      TGAS   -->  Gas temperature
C---      TGASW  -->  Wall temperature on the gas side
C---      TLIQW  -->  Wall temperature on the coolant side
C---      TLIQ   -->  Coolant temperature
C---      FLXG   -->  Heat flux from the gas to coolant
C---      FLX1   -->  Heat flux within the cooling jacket
C----------------------------------------------------------------------
Cinclude "satear"
C FILE NAME SATEAR --- 170486
CNLIST
C
C------------------------------------------------------------ ARRAYS
      COMMON/LDB1/DBGPHI(50)/IDA1/ITERMS(50)/IDA2/LITER(50)
     1/IDA3/IORCVF(50)/IDA4/IORCVL(50)/IDA5/ISLN(50)/IDA6/IPRN(50)
     1/HDA1/NAME(50)/RDA1/DTFALS(50)/RDA2/RESREF(50)
     1/RDA3/PRNDTL(50)/RDA4/PRT(50)/RDA5/ENDIT(50)/RDA6/VARMIN(50)
     1/RDA7/VARMAX(50)/RDA8/FIINIT(50)/RDA9/PHINT(50)
     1/RDA10/CINT(50)/RDA11/EX(50)
C------------------------------------------------------------ LDATA
      COMMON/LDATA/CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32(17),SAVGEO,
     1RSTGEO,NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,
     1DONACC,PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,
     1USEGRD,USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP(11)
C
C------------------------------------------------------------ LDEBUG
      COMMON/LDEBUG/DBGEOM,DBADJS,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37(9)
C
C------------------------------------------------------------ IDATA
      COMMON/IDATA/NX,NY,NZ,LUPR1,LUPR2,LUPR3,LUPHUN,LUSDA,IPROF,
     1LUFI,LUDST,LUGRF,LUSAVE,LUOLD,LUDEP,LUPCO,LUDVL,
     1IRUNN,IOPTN,LITC,LITFLX,NRUN,LITHYD,FSTEP,LSTEP,
     1FSWEEP,LSWEEP,NPRINT,LIBREF,MEANDF,IXMON,IYMON,IZMON,IINIT,
     1NLSG1,NISG1,NRSG1,NCSG1,IPARAB,IDPHUN,NXFR1,NYFR1,NZFR1,
     1NTFR1,ENTH1,ENTH2,ISWR1,ISWR2,IXPRF,IXPRL,IYPRF,IYPRL,
     1NPRMNT,ISTPRL,ISTPRF,IZPRL,IZPRF,NUMCLS,TSTSWP,NYPRIN,NXPRIN,
```

```
     &           0.00965,0.01035,0.01045,0.01065,0.01085,
     &           0.01105,0.01115,0.01135,0.01145,0.01145,
     &           0.01150,0.01155,0.01180,0.01280,0.01280/
      DATA FTA/160*1.0/
      DATA FTB/160*1.0/
C---
      NDEL=NCOM+NNOZ
      DO 10 II=1,NDEL
      IF(II.LE.NCOM)THEN
        TOTAR=ACJ(II)*FLOAT(NCHA)/CONV2
      ELSE
        TOTAR=ACJ(II)*FLOAT(NTUB)/CONV2
      END IF
 10   CONTINUE
C
      JFST=1
      JLST=IBJ
      NFST=1
      NLST=NCOM
 15   DO 20 JZ=JFST,JLST
      DELT=(GZNODE(JZ)-ZTHRO)*CONV
        DO 25 JJ=NFST,NLST
        IF(XJ(JJ).GT.DELT.AND.JJ.EQ.1)THEN
          TDIF=XJ(JJ+1)-XJ(JJ)
          D1=ACJ(JJ+1)-ACJ(JJ)
          D2=DELJ(JJ+1)-DELJ(JJ)
          D4=FTA(JJ+1)-FTA(JJ)
          D5=FTB(JJ+1)-FTB(JJ)
        ELSE IF(XJ(JJ).GT.DELT.OR.JJ.EQ.NLST)THEN
          TDIF=XJ(JJ)-XJ(JJ-1)
          D1=ACJ(JJ)-ACJ(JJ-1)
          D2=DELJ(JJ)-DELJ(JJ-1)
          D4=FTA(JJ)-FTA(JJ-1)
          D5=FTB(JJ)-FTB(JJ-1)
        END IF
        IF(XJ(JJ).GT.DELT.AND.JJ.EQ.1)THEN
          ADIF=XJ(JJ)-DELT
          XAREA(JZ)=(ACJ(JJ)-ADIF*D1/TDIF)/CONV2
          WTHK(JZ)=(DELJ(JJ)-ADIF*D2/TDIF)/CONV
          FT1(JZ)=FTA(JJ)-ADIF*D4/TDIF
          FT2(JZ)=FTB(JJ)-ADIF*D5/TDIF
          GO TO 30
        ELSE IF(XJ(JJ).GT.DELT.AND.JJ.LT.NLST)THEN
          ADIF=XJ(JJ)-DELT
          XAREA(JZ)=(ACJ(JJ-1)+ADIF*D1/TDIF)/CONV2
          WTHK(JZ)=(DELJ(JJ-1)+ADIF*D2/TDIF)/CONV
          FT1(JZ)=FTA(JJ-1)+ADIF*D4/TDIF
          FT2(JZ)=FTB(JJ-1)+ADIF*D5/TDIF
          GO TO 30
        ELSE IF(JJ.EQ.NLST)THEN
          ADIF=DELT-XJ(JJ)
          XAREA(JZ)=(ACJ(JJ)+ADIF*D1/TDIF)/CONV2
          WTHK(JZ)=(DELJ(JJ)+ADIF*D2/TDIF)/CONV
          FT1(JZ)=FTA(JJ)+ADIF*D4/TDIF
          FT2(JZ)=FTB(JJ)+ADIF*D5/TDIF
```

```
     1NZPRIN,NPRMON,NTPRIN,NTZPRF,ISP66,IURINI,IURPRN,IURVAL,
     1IORTCV,NUMREG,NRTCV,ICHR,INTFRC,ITHC1,ISWC1,DEN1,DEN2,
     1VISL,INTMDT,ISWPRF,ISWPRL,IPSA,ISP84,IPLTF,IPLTL,NPLT,ITABL,
     1TEMP1,TEMP2,LEN1,LEN2,NLG1,NIG1,NRG1,NCG1,NPNAM1,
     1ISP98(3),LENREC,LUGEOM,IMB1,IMB2,PCOR,NCOLPF,NCOLCO,
     1NROWCO,EPOR,NPOR,HPOR,VPOR,KXFR,KYFR,KZFR,KTFR,IDATSP(2),
     1VIST,NPHI
C
C------------------------------------------------------- IDEBUG
      COMMON/IDEBUG/IZDB1,IZDB2,ITHDB1,ITHDB2,ISWDB1,ISWDB2,ISTDB1,
     1ISTDB2,INCHCK,IREGDB,NFMAX,IDBF0,IDBCMN,IDBGRD,IDEBSP(2)
C
C------------------------------------------------------- HDATA
      COMMON/HDATA/MESS(10),NBLANK,NAMGRD,NAMEJ,NAMEJ1,
     1NAMEM,NAMEM1,NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,
     1NPHUN,NHINIT,NDST,NAMSAT,NGEOM,NHDASP(2)
C
C------------------------------------------------------- HDEBUG
      COMMON/HDEBUG/NDBF0(2),NDBCMN(2),NHDBSP
C
C------------------------------------------------------- RDATA
      COMMON/RDATA/TINY,GREAT,RUPLIM,RLOLIM,AZDZ,AZXU,AZYV,
     1AZRI,AZAL,AZPH,XULAST,YVLAST,ZWLAST,TLAST,TFIRST,PBAR,SNALFA,
     1RINNER,ENUL,ENUT,RHO1,RHO2,CFIPS,CMDOT,CONMDT,GRND,HEATBL,
     1FIXFLU,READFI,ZMOVE1,ZDIFAC,DRH1DP,DRH2DP,U1AD,U2AD,V1AD,
     1V2AD,W1AD,W2AD,HUNIT,DIFCUT,ABSIZ,ORSIZ,OPPVAL,TMP1,TMP2,
     1EL1,EL2,GRND1,GRND2,GRND3,GRND4,GRND5,GRND6,GRND7,GRND8,GRND9
     1,GRND10,ZWADD,RINIT,SAME,FIXVAL,AXDZ,AYDZ,RDATSP(21)
C
C------------------------------------------------------- RDEBUG
      COMMON/RDEBUG/BGCHCK,SMCHCK,RDEBSP(5)
C
C-----------------------------------LOGICAL DECLARATIONS
      LOGICAL LDAT,LDEB
      LOGICAL CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32,SAVGEO,RSTGEO,
     1NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,DONACC,
     1PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,USEGRD,
     1USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP
      LOGICAL DBGEOM,DBADJS,DBGPHI,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37
C
C-----------------------------------INTEGER DECLARATIONS
      INTEGER FSTEP,FSWEEP,TSTSWP,ENTH1,ENTH2,DEN1,
     1DEN2,PCOR,VISL,EPOR,HPOR,VPOR,VIST,TEMP1,TEMP2
C-----------------------------------CHARACTER DECLARATIONS
      CHARACTER*4 NHDAT,NHDEB
      CHARACTER*4 NAME
      CHARACTER*4 MESS,NBLANK,NAMGRD,NAMEJ,NAMEJ1,NAMEM,NAMEM1,
     1NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,NPHUN,NHINIT,
```

```
     1NDST,NAMSAT,NGEOM,NHDASP
      CHARACTER*4 NDBF0,NDBCMN,NHDBSP
C----------------------------------EQUIVALENT TRANSMISSION ARRAYS
      DIMENSION LDAT(84),LDEB(45),IDAT(120),IDEB(16),NHDAT(30),
     1NHDEB(5),RDAT(85),RDEB(7)
      EQUIVALENCE (LDAT(1),CARTES),(LDEB(1),DBGEOM),(IDAT(1),NX),
     1(IDEB(1),IZDB1),(NHDAT(1),MESS(1)),(NHDEB(1),NDBF0(1)),
     1(RDAT(1),TINY),(RDEB(1),BGCHCK)
CLIST
#include "grdear"
#include "grdloc"
C---
      PARAMETER (NXM=1,NYM=50,NZM=300)
C---
      DIMENSION TGAS(NZM),TGASW(NZM),TLIQ(0:NZM),TLIQW(NZM),
     1          SLAREA(NYM,NXM),CXAREA(NZM),DIAHYD(NZM),WTHK(NZM),
     2          GHGAS(NZM),GHLIQ(NZM),GHWAL(NZM),GHCOMP(NZM),
     3          PIPARE(NZM),FLX1(0:NZM),FT2(NZM),GDIST(NZM),
     4          FLXG(NZM),SUMGAS(2),GTOUT(2),GQOUT(2)
C---
      LOGICAL LHLEN
C---
      DATA GPI/3.1415926/
      DATA AA,BB,CC,DD/-2767.481,298.3292,-1.347217,1.838612E-3/
C---
      DO 10 JZ=NZ,1,-1
C-----------------------------------------------------------------
C---  COOLANT INITIALISATION
C-----------------------------------------------------------------
        IF(JZ.EQ.NZ)THEN
          FLX1(JZ)=FLXINU
          RATE=RATEU
          TLIQ(JZ)=TLIQU
        ELSE IF(JZ.EQ.IBJ)THEN
          FLX1(JZ)=FLXINL
          RATE=RATEL
          TLIQ(JZ)=TLIQL
        END IF
C-----------------------------------------------------------------
C---  WALL HEAT TRANSFER COEFFICIENT
C-----------------------------------------------------------------
      IF(JZ.LE.IBJ)THEN
        GHWAL(JZ)=COPPK/WTHK(JZ)
      ELSE
        GHWAL(JZ)=STEEK/WTHK(JZ)
      END IF
C-----------------------------------------------------------------
C---  LIQUID FILM COEFFICIENT
C-----------------------------------------------------------------
        CPHYD=AA+BB*TLIQ(JZ)+CC*TLIQ(JZ)**2+DD*TLIQ(JZ)**3
        REYHYD=DIAHYD(JZ)*RATE/(CXAREA(JZ)*VISHYD)
        GHLIQ(JZ)=0.023*CPHYD*RATE/CXAREA(JZ)
        GHLIQ(JZ)=GHLIQ(JZ)*REYHYD**(-0.2)*PRHYD**(-2/3)
        IF(LHLEN)GHLIQ(JZ)=GHLIQ(JZ)*FT2(JZ)
C-----------------------------------------------------------------
```

```
C---  COMPOSITE HEAT TRANSFER COEFFICIENT
C-----------------------------------------------------------------------
      GC1=GHGAS(JZ)
      GC2=GHWAL(JZ)
      GC3=GHLIQ(JZ)
      GHCOMP(JZ)=GC1*GC2*GC3/(GC1*(GC2+GC3)+GC2*GC3)
C-----------------------------------------------------------------------
C---  HEAT TRANSFER AREA
C-----------------------------------------------------------------------
        IF(ISWEEP.EQ.FSWEEP)THEN
          CALL GTIZYX(7,JZ,SLAREA,NYM,NXM)
          PIPARE(JZ)=SLAREA(NY,1)
        END IF
C-----------------------------------------------------------------------
C---  CALCULATE HEAT FLUX TO COOLANT AND UPDATE TEMPERATURES
C-----------------------------------------------------------------------
        FLXG(JZ)=GHCOMP(JZ)*(TGAS(JZ)-TLIQ(JZ))
        TLIQW(JZ)=TLIQ(JZ)+FLXG(JZ)/GHLIQ(JZ)
        TGASW(JZ)=TLIQW(JZ)+FLXG(JZ)/GHWAL(JZ)
        FLX1(JZ-1)=FLX1(JZ)+FLXG(JZ)*PIPARE(JZ)
        TLIQ(JZ-1)=TLIQ(JZ)+FLXG(JZ)*PIPARE(JZ)/(RATE*CPHYD)
        TOTGAS = TOTGAS + FLXG(JZ)*PIPARE(JZ)
C---
      IF(MOD(ISWEEP,NPRMON).EQ.0.0.OR.ISWEEP.EQ.LSWEEP)THEN
        IF(JZ.EQ.IBJ+1)THEN
          WRITE(6,*)'COOLANT TEMPERATURE AT EXIT FROM NOZZLE JACKET : ',
     1                TLIQ(IBJ),' K'
          GTOUT(1)=TLIQ(IBJ)
          GQOUT(1)=FLX1(IBJ)
          SUMGAS(1)=TOTGAS
          TOTGAS=0.0
        ELSE IF(JZ.EQ.1)THEN
          WRITE(6,*)'COOLANT TEMP AT EXIT FROM COMBUSTOR JACKET : ',
     1                TLIQ(0),' K'
          GTOUT(2)=TLIQ(0)
          GQOUT(2)=FLX1(0)
          SUMGAS(2)=TOTGAS
          TOTGAS=0.0
        END IF
      END IF
C---
 10   CONTINUE
C-----------------------------------------------------------------------
C---  PRINTOUT HEAT BALANCE DATA
C-----------------------------------------------------------------------
      IF(ISWEEP.EQ.LSWEEP)THEN
        CALL CJPRNT(JZ,GDIST,PIPARE,GHGAS,GHWAL,GHLIQ,FLXG,FLX1,
     &              TGAS,TGASW,TLIQW,TLIQ,IBJ,GTOUT,GQOUT,SUMGAS,
     &              INFO,GPI100)
      END IF
C---
      RETURN
      END
C***********************************************************************
      SUBROUTINE CJPRNT(JZ,GDIST,PIPARE,GHGAS,GHWAL,GHLIQ,FLXG,FLX1,
```

```
     &                      TGAS,TGASW,TLIQW,TLIQ,IBJ,GTOUT,GQOUT,SUMGAS,
     &                      INFO,GPI100)
C***********************************************************************
Cinclude "satear"
C FILE NAME SATEAR --- 170486
CNLIST
C
C----------------------------------------------------- ARRAYS
      COMMON/LDB1/DBGPHI(50)/IDA1/ITERMS(50)/IDA2/LITER(50)
     1/IDA3/I0RCVF(50)/IDA4/I0RCVL(50)/IDA5/ISLN(50)/IDA6/IPRN(50)
     1/HDA1/NAME(50)/RDA1/DTFALS(50)/RDA2/RESREF(50)
     1/RDA3/PRNDTL(50)/RDA4/PRT(50)/RDA5/ENDIT(50)/RDA6/VARMIN(50)
     1/RDA7/VARMAX(50)/RDA8/FIINIT(50)/RDA9/PHINT(50)
     1/RDA10/CINT(50)/RDA11/EX(50)
C----------------------------------------------------- LDATA
      COMMON/LDATA/CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32(17),SAVGEO,
     1RSTGEO,NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,
     1DONACC,PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,
     1USEGRD,USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP(11)
C
C----------------------------------------------------- LDEBUG
      COMMON/LDEBUG/DBGEOM,DBADJS,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37(9)
C
C----------------------------------------------------- IDATA
      COMMON/IDATA/NX,NY,NZ,LUPR1,LUPR2,LUPR3,LUPHUN,LUSDA,IPROF,
     1LUFI,LUDST,LUGRF,LUSAVE,LUOLD,LUDEP,LUPCO,LUDVL,
     1IRUNN,IOPTN,LITC,LITFLX,NRUN,LITHYD,FSTEP,LSTEP,
     1FSWEEP,LSWEEP,NPRINT,LIBREF,MEANDF,IXMON,IYMON,IZMON,IINIT,
     1NLSG1,NISG1,NRSG1,NCSG1,IPARAB,IDPHUN,NXFR1,NYFR1,NZFR1,
     1NTFR1,ENTH1,ENTH2,ISWR1,ISWR2,IXPRF,IXPRL,IYPRF,IYPRL,
     1NPRMNT,ISTPRL,ISTPRF,IZPRL,IZPRF,NUMCLS,TSTSWP,NYPRIN,NXPRIN,
     1NZPRIN,NPRMON,NTPRIN,NTZPRF,ISP66,IURINI,IURPRN,IURVAL,
     1I0RTCV,NUMREG,NRTCV,ICHR,INTFRC,ITHC1,ISWC1,DEN1,DEN2,
     1VISL,INTMDT,ISWPRF,ISWPRL,IPSA,ISP84,IPLTF,IPLTL,NPLT,ITABL,
     1TEMP1,TEMP2,LEN1,LEN2,NLG1,NIG1,NRG1,NCG1,NPNAM1,
     1ISP98(3),LENREC,LUGEOM,IMB1,IMB2,PCOR,NCOLPF,NCOLCO,
     1NROWCO,EPOR,NPOR,HPOR,VPOR,KXFR,KYFR,KZFR,KTFR,IDATSP(2),
     1VIST,NPHI
C
C----------------------------------------------------- IDEBUG
      COMMON/IDEBUG/IZDB1,IZDB2,ITHDB1,ITHDB2,ISWDB1,ISWDB2,ISTDB1,
     1ISTDB2,INCHCK,IREGDB,NFMAX,IDBF0,IDBCMN,IDBGRD,IDEBSP(2)
C
C----------------------------------------------------- HDATA
      COMMON/HDATA/MESS(10),NBLANK,NAMGRD,NAMEJ,NAMEJ1,
     1NAMEM,NAMEM1,NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,
     1NPHUN,NHINIT,NDST,NAMSAT,NGEOM,NHDASP(2)
C
```

```
C------------------------------------------------------ HDEBUG
      COMMON/HDEBUG/NDBF0(2),NDBCMN(2),NHDBSP
C
C------------------------------------------------------ RDATA
      COMMON/RDATA/TINY,GREAT,RUPLIM,RLOLIM,AZDZ,AZXU,AZYV,
     1AZRI,AZAL,AZPH,XULAST,YVLAST,ZWLAST,TLAST,TFIRST,PBAR,SNALFA,
     1RINNER,ENUL,ENUT,RHO1,RHO2,CFIPS,CMDOT,CONMDT,GRND,HEATBL,
     1FIXFLU,READFI,ZMOVE1,ZDIFAC,DRH1DP,DRH2DP,U1AD,U2AD,V1AD,
     1V2AD,W1AD,W2AD,HUNIT,DIFCUT,ABSIZ,ORSIZ,OPPVAL,TMP1,TMP2,
     1EL1,EL2,GRND1,GRND2,GRND3,GRND4,GRND5,GRND6,GRND7,GRND8,GRND9
     1,GRND10,ZWADD,RINIT,SAME,FIXVAL,AXDZ,AYDZ,RDATSP(21)
C
C------------------------------------------------------ RDEBUG
      COMMON/RDEBUG/BGCHCK,SMCHCK,RDEBSP(5)
C
C----------------------------------LOGICAL DECLARATIONS
      LOGICAL LDAT,LDEB
      LOGICAL CARTES,XANGLE,YZPR,ONEPHS,YANGLE,SAVE,ZANGLE,
     1XCYCLE,XZPR,EQDVDP,UCONV,UDIFF,UCONNE,UDIFNE,USOURC,UCORCO,
     1USOLVE,UCORR,STEADY,BFC,AUTOPS,EQUVEL,ADDDIF,NOWIPE,ECHO,
     1WARN,NOSORT,NOADAP,UGEOM,NEWENT,NEWENL,LSP32,SAVGEO,RSTGEO,
     1NEWRH1,NEWRH2,LINIT,SUBWGR,INIADD,INIFLD,SWTCH,GALA,DONACC,
     1PARAB,CONICL,DEBUG,DISTIL,PICKUP,NONORT,HIGHLO,EARTH,USEGRD,
     1USEGRX,PILBUG,SMPLR,VOID,DARCY,LDATSP
      LOGICAL DBGEOM,DBADJS,DBGPHI,DBCOMP,DBINDX,
     1DBFLUX,DBMAIN,DBSOL1,DBSOL2,DBSOL3,DBEMU,DBRHO,DBEXP,DBSODA,
     1DBONLY,DBT,DBL,DBCMPE,DBCMPN,DBCMPH,DBCONV,DBGAM,DBCMP2
     1,DBSHFT,DBOUT,DBCMPR,DBMDOT,DBCFIP,DBPRBL,DBEDGE,DBGRND,
     1FLAG,MONITR,SEARCH,DBCONT,TEST,TSTGNK,LDBS37
C
C----------------------------------INTEGER DECLARATIONS
      INTEGER FSTEP,FSWEEP,TSTSWP,ENTH1,ENTH2,DEN1,
     1DEN2,PCOR,VISL,EPOR,HPOR,VPOR,VIST,TEMP1,TEMP2
C----------------------------------CHARACTER DECLARATIONS
      CHARACTER*4 NHDAT,NHDEB
      CHARACTER*4 NAME
      CHARACTER*4 MESS,NBLANK,NAMGRD,NAMEJ,NAMEJ1,NAMEM,NAMEM1,
     1NAMEP,NAMEQ,NAMEQ1,NAMFI,NSDA,NSAVE,NGRF,NPHUN,NHINIT,
     1NDST,NAMSAT,NGEOM,NHDASP
      CHARACTER*4 NDBF0,NDBCMN,NHDBSP
C----------------------------------EQUIVALENT TRANSMISSION ARRAYS
      DIMENSION LDAT(84),LDEB(45),IDAT(120),IDEB(16),NHDAT(30),
     1NHDEB(5),RDAT(85),RDEB(7)
      EQUIVALENCE (LDAT(1),CARTES),(LDEB(1),DBGEOM),(IDAT(1),NX),
     1(IDEB(1),IZDB1),(NHDAT(1),MESS(1)),(NHDEB(1),NDBF0(1)),
     1(RDAT(1),TINY),(RDEB(1),BGCHCK)
CLIST
#include "grdear"
#include "grdloc"
C---
      PARAMETER (NXM=1,NYM=50,NZM=300)
C---
      DIMENSION TGAS(NZM),TGASW(NZM),TLIQ(0:NZM),TLIQW(NZM),
     1          GHGAS(NZM),GHLIQ(NZM),GHWAL(NZM),
     2          PIPARE(NZM),FLX1(0:NZM),GDIST(NZM),
```

```
     3            FLXG(NZM),SUMGAS(2),GTOUT(2),GQOUT(2)
C---
      DATA GPI/3.141592654/
      DATA CN1,CN2,CN3/100.,1.8,459.67/
      DATA CN4,CN5,CN6/3.397345E-7,6.114919E-7,9.484067E-4/
C---
      WRITE(6,*)'****************************************************'
      WRITE(6,*)'***......... COOLING JACKET SIMULATION .........***'
      WRITE(6,*)'***......... (SI UNITS)                .........***'
      WRITE(6,*)'****************************************************'
      WRITE(6,1961)
      DO 10 JZ=NZ,1,-1
        WRITE(6,1963)JZ,GDIST(JZ),PIPARE(JZ),GHGAS(JZ),GHWAL(JZ),
     1              GHLIQ(JZ),FLXG(JZ),FLX1(JZ),TGAS(JZ),TGASW(JZ),
     2              TLIQW(JZ),TLIQ(JZ)
        IF(JZ.EQ.IBJ+1)THEN
          WRITE(6,*)'COOLANT TEMPERATURE AT EXIT FROM NOZZLE JACKET : ',
     1              GTOUT(1),' K'
          WRITE(6,*)'HEAT FLOW AT EXIT FROM NOZZLE JACKET : ',
     1              GQOUT(1),' W'
          WRITE(6,*)'TOTAL HEAT INPUT TO THE NOZZLE JACKET : ',
     1              SUMGAS(1),' W'
          WRITE(6,1961)
        ELSE IF(JZ.EQ.1)THEN
          WRITE(6,*)'COOLANT TEMP AT EXIT FROM COMBUSTOR JACKET : ',
     1              GTOUT(2),' K'
          WRITE(6,*)'HEAT FLOW AT EXIT FROM COMBUSTOR JACKET : ',
     1              GQOUT(2),' W'
          WRITE(6,*)'TOTAL HEAT INPUT TO THE COMBUSTOR JACKET : ',
     1              SUMGAS(2),' W'
        END IF
 10   CONTINUE
C---
      DO 11 JZ=NZ,1,-1
      IF(JZ.EQ.NZ.OR.JZ.EQ.IBJ)SUMQ=0.0
      SUMQ=SUMQ+FLXG(JZ)*PIPARE(JZ)*GPI100
        IF(JZ.EQ.NZ)THEN
         WRITE(6,*)'****************************************************'
         WRITE(6,*)'***......... COOLING JACKET SIMULATION .........***'
         WRITE(6,*)'***......... (TABLE DATA)              .........***'
         WRITE(6,*)'****************************************************'
         WRITE(6,1964)
        END IF
        WRITE(6,1965)JZ,GDIST(JZ)*CN1,TGASW(JZ)*CN2-CN3,
     1              TLIQW(JZ)*CN2-CN3,TLIQ(JZ)*CN2,GHLIQ(JZ)*CN4,
     2              GHGAS(JZ)*CN4,FLXG(JZ)*CN5,SUMQ*CN6
        IF(JZ.EQ.IBJ+1)WRITE(6,1964)
 11   CONTINUE
C---
 1961   FORMAT(/2X,' IZ    XDIST     XAREA     GHGAS     GHWAL     ',
     1           ' GHLIQ     FLXG      FLXOUT    TGAS     ',
     2           '   TGASW     TLIQW     TLIQ    ',
     3       /2X,'          (m)        (m)      (W/K-m2)  (W/K-m2)  ',
     4           '(W/K-m2)  (W/m2)    (W)       (K)      ',
     5           '    (K)       (K)       (K)     ',
```

```
     6         /2X,'===============================================',
     7              '=======================================',
     8              '================================')
 1963   FORMAT(2X,I3,(1P,E11.3),10(1P,E10.3))
 1964   FORMAT(/2X,' IZ     XDIST      TGASW      TLIQW      TLIQ    ',
     1              '    GHLIQ      GHGAS      FLXG       SUMQ    ',
     2         /2X,'         (cm)        (F)        (F)        (R)    ',
     3              '    (BTU/in2-s-F)  (BTU/in2-s)  (BTU/s) ',
     4         /2X,'============================================',
     5              '==========================================')
 1965   FORMAT(2X,I3,(1P,E11.3),7(1P,E10.3))
C---
      RETURN
      END
```