TACT1, A Computer Program for
the Transient Thermal Analysis
of a Cooled Turbine Blade or Vane
Equipped With a Coolant Insert
II - Programmers Manual

Raymond E. Gaugler

JANUARY 1979
TACT1, A Computer Program for the Transient Thermal Analysis of a Cooled Turbine Blade or Vane Equipped With a Coolant Insert

II - Programmers Manual

Raymond E. Gaugler
Lewis Research Center
Cleveland, Ohio

NASA
National Aeronautics and Space Administration
Scientific and Technical Information Office
1979
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>OVERVIEW</td>
<td>2</td>
</tr>
<tr>
<td>Method of Analysis</td>
<td>2</td>
</tr>
<tr>
<td>Blade geometric model</td>
<td>2</td>
</tr>
<tr>
<td>Numerical model</td>
<td>3</td>
</tr>
<tr>
<td>General Program Description</td>
<td>3</td>
</tr>
<tr>
<td>DETAILED PROGRAM PROCEDURE</td>
<td>5</td>
</tr>
<tr>
<td>Block Data NGASDAT</td>
<td>6</td>
</tr>
<tr>
<td>Main Program NTTACT</td>
<td>6</td>
</tr>
<tr>
<td>Subroutine FLOWS</td>
<td>7</td>
</tr>
<tr>
<td>Subroutine FLSPLT</td>
<td>7</td>
</tr>
<tr>
<td>Subroutine GASTBL</td>
<td>8</td>
</tr>
<tr>
<td>Subroutine GAUSS</td>
<td>8</td>
</tr>
<tr>
<td>Subroutine GETIN</td>
<td>8</td>
</tr>
<tr>
<td>Function HCFRCD</td>
<td>9</td>
</tr>
<tr>
<td>Subroutine HCOOL</td>
<td>9</td>
</tr>
<tr>
<td>Subroutine HCPINS</td>
<td>9</td>
</tr>
<tr>
<td>Subroutine INPRT</td>
<td>9</td>
</tr>
<tr>
<td>Subroutine PARRAY</td>
<td>9</td>
</tr>
<tr>
<td>Subroutine PLNUM</td>
<td>10</td>
</tr>
<tr>
<td>Subroutine PLOTMF</td>
<td>10</td>
</tr>
<tr>
<td>Subroutine PREP</td>
<td>10</td>
</tr>
<tr>
<td>Subroutine TARRAY</td>
<td>10</td>
</tr>
<tr>
<td>Subroutine TCOEF</td>
<td>11</td>
</tr>
<tr>
<td>Subroutine THRCON</td>
<td>11</td>
</tr>
<tr>
<td>Subroutine WROUT</td>
<td>11</td>
</tr>
<tr>
<td>DICTIONARY</td>
<td>11</td>
</tr>
<tr>
<td>PROGRAM LISTING</td>
<td>53</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>150</td>
</tr>
</tbody>
</table>
SUMMARY

A FORTRAN IV computer program to calculate transient and steady-state temperatures, pressures, and flows in a cooled turbine blade or vane with an impingement insert has been developed and is described in this report. Coolant-side heat-transfer coefficients are calculated internally in the program, with the user specifying one of three modes of heat transfer at each station: (1) impingement, including the effect of crossflow; (2) forced-convection channel flow; or (3) forced convection over pin fins. Additionally, a limited capability to handle film cooling is available in the program. It is assumed that spent impingement air flows in a chordwise direction and is discharged through a split or drilled trailing edge and through film-cooling holes. The program does not allow for radial flow of the spent impingement air. The use of film cooling is restricted by a numerical model requirement for a continuous coolant-channel flow.

Input to the program includes a description of the blade geometry, coolant-supply conditions, outside thermal boundary conditions, and wheel speed. The user can divide the blade by chordwise cuts into as many as 15 slices and can divide each slice into as many as 79 stations around the blade. Each station in turn consists of four calculational nodes through the wall and one in the coolant channel. The blade wall can be two layers of different materials, such as a ceramic thermal-barrier coating over a metallic substrate. Program output includes the temperature at each node, the coolant pressures and flow rates, and the coolant-side heat-transfer coefficients.

INTRODUCTION

As core turbine-engine operating conditions become more severe, it becomes more difficult to effectively cool blades and vanes. Advanced transient thermal calculational techniques are needed to design reliable turbine blades. However, there appears to be no computer program generally available that uses these advanced techniques in combining the required heat-transfer and coolant-flow-distribution calculations. Thus, it was decided to create a computer program that would perform both transient and steady-state heat-transfer and coolant-flow analyses for a cooled blade, given the outside hot-gas boundary conditions, the coolant inlet pressure or flow rate, the geometry of the blade shell, and the cooling configuration.

The resulting program, TACT1, can handle a turbine blade or vane that is equipped with a central coolant-plenum insert from which coolant air flows through holes to impinge on the inner surface of the blade shell or directly into the trailing-edge region. It
is assumed that the spent impingement air then flows chordwise and is dumped through a split or drilled trailing edge and through film-cooling holes. The blade is modeled by dividing it by chordwise planes into as many as 15 slices, with each slice having as many as 79 calculational stations around the blade. Temperatures at each station are calculated for four points through the wall and one in the coolant channel. Included in this model is the capability to analyze a blade with a ceramic thermal-barrier coating. The ability of the program to model film cooling is limited by the numerical flow analysis requirement for a continuous coolant-channel flow.

The TACT1 program is used at the NASA Lewis Research Center on an IBM TSS/360–67 computer. The source program consists of approximately 6000 lines of code and the program requires about 60 000 words of storage. Typical running times for the program are 1.4 seconds of central processor unit (CPU) time per calculational station for a steady-state run and 0.4 second of CPU time per station per time step for a transient run.

The TACT1 program is reported in two parts. This report, part II, is a programmers manual and includes a complete program listing and a detailed description of the procedure. Part I (ref. 1) is a users manual and contains all the information necessary to run the program: a detailed description of the input, the method of solution, and the output as well as a sample problem.

OVERVIEW

Method of Analysis

The details of the analytical method are presented in part I (ref. 1). The blade model used in the analysis is described briefly in this section.

Blade geometric model. - The key to creating a usable computer program is to have as simple a geometric model as possible for the system being analyzed. In this program, the emphasis is on a blade or vane with a central coolant plenum and chordwise flow of the coolant after impingement. Therefore, it was decided that the primary calculational direction would also be chordwise. The blade is divided into layers that are bounded by chordwise cuts through the blade, as shown in figure 1. Each slice is treated separately in the program, with radial heat conduction in the wall the only communication between layers.

Figure 2 gives the details of the geometric model for a single blade slice and shows the breakdown of the blade or vane into calculational stations and nodes. Each calculational station consists of five nodes: one at the wall outer surface, one at the interface between the coating and blade metal, one at a point midway through the wall metal, one at the wall inner surface, and one in the middle of the coolant channel.
For input to the program, the following basic elements of the geometry are needed for each station: (1) the thicknesses of the wall coating and wall metal and the coolant-channel width, (2) the distance of each node from the adjacent lower-numbered node, and (3) the radial span for this slice. In addition, depending on the mode of heat transfer specified, the user must supply impingement-hole diameter and spacing or pin-fin diameter and spacing. Thermal properties of the blade materials must also be specified. The input is described in detail in reference 1.

**Numerical model.**—The numerical solution for the temperatures throughout the blade involves writing a transient energy-balance equation for each node and forming a set of equations to be solved for the temperature distribution. Similarly, the coolant pressure distribution is determined by writing the transient momentum equation for flow between adjacent fluid nodes and solving the resulting set of equations for static pressures.

The nodal energy balances are linearized, one-dimensional heat-conduction equations at the wall outer-surface node, at the coating-metal interface, and at the wall inner-surface node. At the midmetal node, a linearized, three-dimensional, heat-conduction equation is used. In the coolant channel, energy and momentum equations for one-dimensional compressible flow including friction and heat transfer are written for the elemental channel length between two coolant nodes. The equations used are presented in reference 1.

**General Program Description**

The TACT1 program is capable of performing a transient analysis as well as a steady-state analysis. In the case of a transient, the program first performs steady-state calculations to determine the initial conditions for the transient.

Figure 3 shows a schematic of the TACT1 calculational scheme. There are three basic, nested calculational loops that must converge for a steady-state solution to be reached. These loops are labeled A, B, and C in figure 3. The program begins a steady-state analysis with the coolant-supply pressure and total coolant flow fixed. The impingement flow is initially assumed to split uniformly at the leading-edge stagnation station, station 1. All coolant flows for the slice under consideration are calculated first, based on the current pressure distribution. The temperatures at each node are then calculated by solving simultaneously the energy equations presented in reference 1. The pressures at each coolant node are calculated by solving simultaneously the momentum equations presented in reference 1. This cycle, loop A in figure 3, is repeated until the pressure distribution no longer changes. The flow split between suction- and pressure-side coolant channels is then checked by comparing the pressures at the ends of the two channels. If they do not match, the impingement flow split at the leading edge
is adjusted and the inner loop calculations are repeated. This adjustment comprises loop B in figure 3. Once the proper flow split is achieved, the program moves up the blade to the next slice and repeats this sequence. After all N slices have converged, the total coolant mass flow used is compared with the inlet coolant mass flow. If there is an imbalance, either the inlet flow or the supply pressure is adjusted, depending on which was specified in the input; and the calculations start over. This is loop C in figure 3. Once the overall coolant mass flow balance is satisfied, the steady-state solution is complete and the transient calculations begin. During a transient calculation, loop B is bypassed because the coolant flow-split is primarily a function of blade geometry. Loop C is also bypassed because the inlet coolant mass flow rate at a given time is estimated based on the coolant mass flow used at the previous time step and the change in supply pressure.

The TACT1 subprograms and the calling relations are shown in figure 4. Block data subprogram NGASDAT contains air properties, for use in TACT1, tabulated as functions of temperature at a pressure of 20 atmospheres from reference 2. This subprogram must be loaded before execution of the program. The main program, NTTACT, calls other subroutines in their proper order.

The first call from NTTACT is to GETIN, a subroutine that controls the reading, storing, and printing of input data. Subroutine GETIN calls INPUT to print the input data if the user specifies INEDIT > 0. Subroutine INPRT has a call to PREP to put the input data in its proper form for use. All data are input by using a NAMELIST format.

After the input data have been read, the number of time steps, NTYM, to be used in the transient is determined in NTTACT. If only a steady-state solution is to be calculated, NTYM = 1. Time-dependent boundary conditions are then evaluated, with the initial entries assumed to be steady-state values. Then NTTACT loops through the blade, calling on subroutines PLNUM, PREP, and TCOEF for each slice. The first time through is a steady-state calculation.

Subroutine PLNUM calculates the pressure distribution in the impingement plenum for the current slice, given the inlet pressure and coolant flow-rate. PLNUM calls GASTBL for gas properties.

Subroutine PREP extracts the input data for the current slice from the input arrays.

Subroutine TCOEF controls loop A in figure 3, the iterative calculations of temperature and pressure for the nodes of the current slice. Each iteration in TCOEF requires calls to subroutines FLOWS, HCOOL, THRCON, TARRAY, PARRAY, and GAUSS.

Subroutine FLOWS computes the impingement jet flow rates, coolant-channel mass flow rates, and channel Mach numbers for each station around the blade, given the plenum pressure and temperature and the current pressure distribution in the coolant channel. FLOWS calls GASTBL for gas properties.
Subroutine HCOOL is called to calculate coolant heat-transfer coefficients for all the stations of this slice, based on the latest values of mass flow rate. HCOOL calls function HCFRCD to calculate forced-convection heat-transfer coefficients and GASTBL for gas properties.

Subroutine THRCON determines the wall thermal conductivity from the input table of conductivity as a function of temperature.

Subroutine TARRAY sets up the array of coefficients for the conduction and convection equations for each node. Calls are made to HCPINS for pin-fin heat-transfer coefficients, to HCFRCD for forced-convection heat-transfer coefficients, and to GASTBL for gas properties. TCOEF calls subroutine GAUSS to solve the set of equations for the temperature at each node.

Subroutine PARRAY sets up the array of coefficients for the momentum equations in the coolant channels and TCOEF calls subroutine GAUSS to solve the set of equations for the pressure at each coolant node.

After a new set of temperatures and pressures has been determined, convergence is checked by using the coolant-channel pressure at the blade leading edge. If this pressure stays within a tolerance band for four successive iterations, convergence is accepted. Once convergence is achieved, TCOEF calls subroutine FLSPLT to check the coolant flow-split between the pressure and suction sides. This is loop B in figure 3. Initially, the impingement jet flow at the forward stagnation station is assumed to split evenly between the suction- and pressure-side channels. If the coolant-channel pressures at the end of the impingement insert do not match, the flow split at the forward stagnation station is adjusted to increase the flow to the channel with the higher pressure at the end of the insert, and iteration loop A is repeated. Once a satisfactory flow split has been achieved, TCOEF calls subroutine WROUT to print the output for this slice and calls subroutine PLOTMF if there is to be graphical output. After NTTACT has calculated all blade slices, the total coolant mass flow is compared with the impingement-plenum inlet mass flow rate used to start the calculations. If the two flow rates are not close enough, the inlet mass flow or supply pressure is adjusted and the calculations are repeated. This is loop C in figure 3.

When the initial steady-state solution has been completed, the transient calculations are started. The transient is continued until the time reaches the specified maximum.

Subroutine PLOTMF makes use of a TSS/360 graphics package at the NASA Lewis Research Center to plot temperature and pressure distributions for the blade.

**DETAILED PROGRAM PROCEDURE**

Table I lists the names of each of the subprograms in TACT1, the corresponding
TSS/360 source module names, the COMMON blocks used in each, the names of the subroutines called by each, and the names of subroutines calling each. Table II is a cross-reference listing of named COMMON blocks and the subprograms using them. This section gives a detailed description of each subprogram used in TACT1. All variable names used are defined in the section DICTIONARY. The BLOCK DATA subprogram and the MAIN PROGRAM are discussed first and then each subprogram is described, in alphabetical order.

Block Data NGASDAT

A BLOCK DATA subprogram, NGASDAT, is used to provide a table of gas properties to the program. The properties are put in the array GS through a DATA statement with 5xNG entries, where NG is the number of table entries for each property. The first NG values are temperatures, the second are thermal conductivities, the third are specific heats, the fourth are Prandtl numbers, and the final NG values are viscosities. The property values included are taken from reference 2 at a pressure of 20 atmospheres.

Main Program NTTACT

The MAIN PROGRAM for TACT1, NTTACT, has overall control of the program. Figure 5 is a flow chart for NTTACT. During initialization, a call is made to a system subroutine, TIME, to get a unique label to be used to identify the plotted output for a given run. After the call to GETIN, where all the input data are read, NTTACT initiates the solution procedure by searching the transient boundary condition tables and using linear interpolation to extract the values for the current time. The next step is to begin the loop, labeled C in figure 3. The solution progresses from hub to tip. For each slice, NTTACT calls PLNUM to calculate coolant-supply conditions; PREP to extract the input data from the input tables; and TCOEF to calculate flows, temperatures, and pressures. After the return from TCOEF, NTTACT updates the total amount of coolant used, WUSED, by adding the amount used in the current slice, WIM. The amount of coolant-plenum flow available for the remaining slices, WPLEN, is updated by subtracting WIM. After all slices have been done, the overall amount of coolant used is printed and then checked against the assumed coolant flow-rate. If the absolute value of the difference, EXCESW, is more than 1 percent of the assumed flow, the assumed flow or the supply pressure is adjusted and the calculations are repeated. For transient runs, after the initial steady-state coolant-flow balance, there are no more iterations on coolant flow. Instead the flow for a given time step is based on the actual flow used in
the preceding time step and on the ratio of supply pressure for the two steps. Finally, once all loops have been completed, NTTACT calls PLOTMF to get a final summary plot of blade temperatures.

Subroutine FLOWS

Subroutine FLOWS is a routine to calculate the flow rates through all impingement and film-cooling holes, the friction factor in the coolant channels, and film-cooling effectiveness. FLOWS makes use of the current impingement-plenum mean pressure and temperature and coolant-channel pressure and temperature distributions. The impingement jet flow-rate, \( W_J \), is calculated for each station in the forward region and checked against the choked flow-rate, \( W_{CR} \). If \( W_J \) is greater than \( W_{CR} \), then \( W_J \) is set equal to \( W_{CR} \). If there is any film cooling on the blade, the film-cooling flow rates in the forward region, \( W_{FC} \), are also calculated. Then, the coolant-channel flow rates, \( W_{COR} \), are computed by considering a mass balance between stations, as illustrated in figure 6. Once the forward-region coolant flows have been determined, the Reynolds numbers - \( RE \) for the coolant channel, and \( RE_{FC} \) for the film-cooling flow - and the square of the coolant Mach number, \( AM^2 \), are computed for each forward station.

The next step is to calculate the amount of coolant, \( WDUMP \), dumped directly into the trailing-edge region from the coolant plenum. Then the total amount of coolant used for this slice, \( W_{IM} \), is determined by summing the impingement jet flows and \( WDUMP \). Following this, the flows in the trailing-edge region are computed, with the coolant flow being reduced by the amount of any film-cooling flow. Then, trailing-edge-region values of \( RE \), \( RE_{FC} \), and \( AM^2 \) are calculated.

After all the coolant flow-rates are determined, the friction factor, \( FF \), is calculated at each station. Finally, if there is any film cooling used, the film effectiveness is calculated by using the method of reference 3.

Subroutine FLSPLT

Subroutine FLSPLT is used to determine the location of the stagnation impingement jet, station \( JS \), and the fraction of that jet's flow that splits to each side of the blade, \( DELTAN \). Figure 7 is a detailed flow chart for subroutine FLSPLT. The primary variable carried into FLSPLT is the pressure-match parameter, \( EPSN \), which is defined as

\[
EPSN = \frac{P(2, ISLICE, NFWD-1) - P(2, ISLICE, NFWD)}{P(2, ISLICE, NFWD-1)}
\]

where the pressures are as illustrated in figure 8.
The magnitude and sign of EPSN are used to determine the adjustment of the stagnation impingement-jet row location and the fraction of that jet that splits to the suction-side channel. Initially, the stagnation jet row is located at station 1 and the split is $\text{DELTAN} = 0.50$. If EPSN is positive, DE\text{LTAN} is set to 0.75 to increase the flow down the suction-side channel; if EPSN is negative, DE\text{LTAN} is set to 0.25 to increase the flow down the pressure-side channel. For subsequent entries into FLSPLT, the value of DE\text{LTAN} is adjusted by passing a straight line through the last two points on a plot of EPSN versus DE\text{LTAN} and picking the value of DE\text{LTAN} where this line crosses the axis at EPSN $= 0$. If this intercept falls outside the DE\text{LTAN} range of 0 to 1, the stagnation station, JS, must be moved to an adjacent station and DE\text{LTAN} set to 0.50. Once a sign change is observed in EPSN, a fine-tuning process is triggered in FLSPLT. In this case, the values of DE\text{LTAN} and EPSN for the iteration preceding the sign change are saved and used as one of the points of the straight-line interpolation scheme for all subsequent iterations.

Subroutine GASTBL

Subroutine GASTBL is used to interpolate in the array GS for gas properties, given the absolute temperature. Linear interpolation is used.

Subroutine GAUSS

Subroutine GAUSS is a routine that uses Gaussian elimination to solve a set of simultaneous equations. The array of coefficients, TCOF, is in the form of a compressed, augmented band matrix. That is, only the matrix elements within the band and the constants from the right side are stored in TCOF. The matrix band width, $K$, is determined by the node-numbering system used. In TACT1, the temperature calculations require a band width of 23 elements, and the pressure calculations require 19.

Subroutine GETIN

Subroutine GETIN is a routine used to initialize input-data default values and to read and store input data. Input is in NAMELIST form as described in reference 1. The entire data set is read, and the input variables for each slice are stored in two arrays: INDCHN for integer data, and CHANL for real-number data. If the input is provided in SI units, subroutine GETIN converts it to U.S. customary units for internal use. If the user specifies INEDIT $> 0$, GETIN calls subroutine INPRT to print out the input data.
Function HCFRCD

Function subprogram HCFRCD is a routine to calculate a turbulent, forced-convection heat-transfer coefficient for channel flow as described in reference 1.

Subroutine HCOOL

Subroutine HCOOL is a routine containing the correlations for impingement heat transfer. The first part of HCOOL deals with leading-edge-region impingement cooling. In this part, the inner surface length from the stagnation impingement jet to the end of the leading-edge impingement region is determined and then used in a correlation to compute the average heat-transfer coefficient in this region. Beyond this region, for stations starting at ICOR, calculations are done by using an impingement-crossflow correlation.

Subroutine HCPINS

Subroutine HCPINS is a routine to calculate coolant-side heat-transfer coefficients in regions of the blade equipped with pin fins. In addition, the effective heat-transfer area, which accounts for the pin surface area and the pin-fin effectiveness, is calculated.

Subroutine INPRT

Subroutine INPRT is a routine to print a listing of the input data. Also, INPRT sets up the initial temperature distribution in the blade. Subroutine PREP is called for each slice to extract input data from the arrays INDCHN and CHANL.

Subroutine PARRAY

Subroutine PARRAY is a routine to set up the matrix to be solved for coolant-channel pressure distribution. The equations used are detailed in reference 1.

The array of coefficients generated in PARRAY, TCOF, is in the form of a compressed, augmented band matrix. Coefficients that would be on the main diagonal of the full matrix are stored in column 10 of the TCOF array. The terms from the right side of the equations are stored in column 20 of TCOF.
Subroutine PLNUM

Subroutine PLNUM is a routine to calculate the pressure and temperature distributions in the central coolant plenum. The mean plenum static pressure and temperature for each slice are used as the supply conditions for the impingement jets. The total temperature and pressure at the outlet of one plenum slice are used as input for the next slice.

There are five arguments used in the call statement for PLNUM: WXX is the mass flow rate into this plenum slice; PXX and TXX are the calculated, average static pressure and temperature; and PTEXIT and TTEXIT are total temperature and pressure, respectively. Going into the subroutine, PTEXIT and TTEXIT are the values at the entrance to this slice. On return, they are the values at the exit of this slice.

Subroutine PLOTMF

Subroutine PLOTMF is a routine that plots TACT1 output. PLOTMF makes use of a TSS/360 graphics package at the NASA Lewis Research Center. For an installation without this specific package, this subroutine would have to be revised or bypassed.

PLOTMF plots temperature and pressure versus surface distance from station 1 for each slice of the blade for a steady-state case. For transients, a set of two summary plots is made for each time step: the plots contain temperatures for all slices on one graph.

Subroutine PREP

Subroutine PREP is a routine to extract input data from storage and put it in the form used in the calculations. In PREP, the hot-gas-side boundary condition tables are searched and linear interpolation is done to extract the boundary condition values at each calculation station at the given time.

Subroutine TARRAY

Subroutine TARRAY is a routine to set up the matrix to be solved for the temperatures in each slice. The equations used are detailed in reference 1.

The array of coefficients generated in TARRAY, TCOF, is in the form of a compressed, augmented band matrix. The 12th column of TCOF contains the elements that would be on the main diagonal of a full matrix. The terms from the right side of the equations are stored in column 24.
Subroutine TCOEF

Subroutine TCOEF is a routine that controls the calculations for flow rates, temperatures, and pressures. The first time TCOEF is entered for each slice an initial estimate of the coolant-channel pressure distribution is set up. TCOEF controls the iterations in loops A and B in figure 3. Loop A consists of calls to subroutines FLOWS, TARRAY, and PARRAY. The variable IVERGE is used to count the number of iterations in loop A. Convergence is checked by comparing the four most-recent values of coolant-channel pressure at the flow-split point. When the ratio of the maximum difference among these four to the difference between coolant-supply pressure and trailing-edge exit pressure is less than PCNVRG, loop A is complete. Then the flow split at the stagnation impingement jet, JS, is checked by subroutine FLSPLT and adjusted if necessary. Loop B involves repeating loop A for a new flow split. The variable IDELT is used to count the number of flow-split iterations in loop B. Once flow-split convergence is achieved, WROUT is called to print the output for the current slice.

Subroutine THRCON

Subroutine THRCON is a routine that takes the wall temperatures and searches for the thermal conductivity values in the input tables.

Subroutine WROUT

Subroutine WROUT is a routine to control the printing of the output from TACT1. Output units are the same as the input data units.

DICTIONARY

All the important variable names used in the TACT1 code are defined in this section. The only names not defined are locally used indices. All dimensioned variables include the dimensions. The dictionary also indicates the COMMON block or subroutine in which each variable is used.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Common</th>
<th>Subroutine</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(400)</td>
<td>TCO</td>
<td></td>
<td>cross-sectional area normal to chord-wise direction, in^2, accessed by node number</td>
</tr>
<tr>
<td>AA</td>
<td>GETIN</td>
<td></td>
<td>outer-surface length between stations, in., used for calculating interpolated values of TDLX(2), TDLX(3), and TDLX(5)</td>
</tr>
<tr>
<td>AA</td>
<td>PLNUM</td>
<td></td>
<td>coolant-plenum cross-sectional area, in^2, used in plenum pressure-drop calculations</td>
</tr>
<tr>
<td>AB</td>
<td>PLNUM</td>
<td></td>
<td>maximum Mach number in coolant plenum</td>
</tr>
<tr>
<td>AC(5)</td>
<td>GASTBL</td>
<td></td>
<td>array of interpolated values of gas properties</td>
</tr>
<tr>
<td>ACH</td>
<td>PLNUM</td>
<td></td>
<td>coolant-plenum choked-flow indicator</td>
</tr>
<tr>
<td>ADUMP</td>
<td>TCO</td>
<td></td>
<td>area of slot or jets dumping coolant directly into trailing-edge region, in^2</td>
</tr>
<tr>
<td>ADUMPC</td>
<td>INPRT</td>
<td></td>
<td>same as ADUMP, but converted to cm^2 for input listing when input is in SI units</td>
</tr>
<tr>
<td>AHG</td>
<td>PREP</td>
<td></td>
<td>intermediate value of hot-gas-side heat-transfer coefficient, Btu/hr \cdot ft^2 \cdot ^\circ F, used for interpolating in input table during a transient</td>
</tr>
<tr>
<td>AHTRN1</td>
<td>TARRAY</td>
<td></td>
<td>inner-surface area for heat-transfer purposes, in^2</td>
</tr>
<tr>
<td>AHTTR</td>
<td>HCPINS</td>
<td></td>
<td>total surface area in pin-fin channel, in^2</td>
</tr>
<tr>
<td>AINTRV</td>
<td>PLOTMF</td>
<td></td>
<td>floating-point form of number of temperature intervals in summary plots</td>
</tr>
<tr>
<td>AJ</td>
<td>PLNUM</td>
<td></td>
<td>floating-point form of indicator J - 1</td>
</tr>
<tr>
<td>AJET(80)</td>
<td>TCO</td>
<td></td>
<td>total area of impingement jets at each station, in^2</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>------------</td>
<td>-----------</td>
</tr>
<tr>
<td>AJET(80)</td>
<td></td>
<td>FLSPLT</td>
<td>total area of impingement jets at each station, in$^2$, carried into subroutine as argument</td>
</tr>
<tr>
<td>AKC(15,80)</td>
<td></td>
<td>TCO</td>
<td>wall outer-coating thermal conductivity, Btu/hr $\cdot$ ft $\cdot$ °F</td>
</tr>
<tr>
<td>AKCTBL(20)</td>
<td></td>
<td>BOUND</td>
<td>input table of wall outer-coating thermal conductivity, Btu/hr $\cdot$ ft $\cdot$ °F, versus temperature, °F</td>
</tr>
<tr>
<td>AKW(15,80)</td>
<td></td>
<td>TCO</td>
<td>wall metal thermal conductivity, Btu/hr $\cdot$ ft $\cdot$ °F</td>
</tr>
<tr>
<td>AKWTBL(20)</td>
<td></td>
<td>BOUND</td>
<td>input table of wall metal thermal conductivity, Btu/hr $\cdot$ ft $\cdot$ °F, versus temperature, °F</td>
</tr>
<tr>
<td>ALABL(7)</td>
<td></td>
<td>PLOTMF</td>
<td>array containing time and date label for identification of output plots</td>
</tr>
<tr>
<td>ALPH(12)</td>
<td></td>
<td>NTTACT</td>
<td>alphameric array used to uniquely identify output of each job</td>
</tr>
<tr>
<td>ALPHA</td>
<td></td>
<td>FRIC</td>
<td>constant used in friction factor calculations</td>
</tr>
<tr>
<td>ALPH2(4)</td>
<td></td>
<td>PLOTMF</td>
<td>time and date information, generated in NTTACT and passed to plotting subroutine as argument</td>
</tr>
<tr>
<td>AM</td>
<td></td>
<td>HCOOL</td>
<td>exponent on Reynolds number in Kercher-Tabakoff impingement correlation</td>
</tr>
<tr>
<td>AMC(20)</td>
<td></td>
<td>PLNUM</td>
<td>Mach number distribution in coolant plenum for a given slice</td>
</tr>
<tr>
<td>AMCHOK</td>
<td></td>
<td>FLOWS</td>
<td>if any stations have a Mach number greater than 1.0, the value is saved in this variable and returned as an argument, to be printed by TCOEF</td>
</tr>
<tr>
<td>AMIN</td>
<td></td>
<td>FLOWS</td>
<td>area of coolant-flow channel at a given station, reduced by pin-fin blockage</td>
</tr>
<tr>
<td>HCPINS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>AM2(80)</td>
<td>TCO</td>
<td></td>
<td>array containing square of coolant-channel Mach number at each station, for a given slice</td>
</tr>
<tr>
<td>APG</td>
<td>PREP</td>
<td></td>
<td>intermediate value of hot-gas-side pressure, lbf/in², used for interpolating in input table during transient</td>
</tr>
<tr>
<td>APLEN</td>
<td>GETIN</td>
<td></td>
<td>input value of coolant-plenum area for given slice, cm² (in²)</td>
</tr>
<tr>
<td>APLN(15)</td>
<td>RADL</td>
<td></td>
<td>internal array to store plenum area for each slice, in²</td>
</tr>
<tr>
<td>AP1</td>
<td>GASTBL</td>
<td></td>
<td>interpolating factor in gas property table lookup</td>
</tr>
<tr>
<td>AP2</td>
<td>GASTBL</td>
<td></td>
<td>1.0 - AP1</td>
</tr>
<tr>
<td>AQG</td>
<td>PREP</td>
<td></td>
<td>intermediate value of hot-gas-side heat flux, Btu/hr · ft², used for interpolating in input table during transient</td>
</tr>
<tr>
<td>ASTG</td>
<td>TCOEF</td>
<td></td>
<td>inner-surface area under stagnation-point impingement jet, in²</td>
</tr>
<tr>
<td>ATG</td>
<td>PREP</td>
<td></td>
<td>intermediate value of hot-gas-side temperature, °R, used for interpolating in input table during transient</td>
</tr>
<tr>
<td>ATMAXP</td>
<td>PLOTMF</td>
<td></td>
<td>adjusted maximum temperature, °F, used as high endpoint on output plots</td>
</tr>
<tr>
<td>ATMINP</td>
<td>PLOTMF</td>
<td></td>
<td>adjusted minimum temperature, °F, used as low endpoint on output plots</td>
</tr>
<tr>
<td>ATYME</td>
<td>PLOTMF</td>
<td></td>
<td>value of time in transient, sec, used on output plots for identification</td>
</tr>
<tr>
<td>AVRGA</td>
<td>PARRAY</td>
<td></td>
<td>area ratio used in momentum equation at entrance to trailing edge</td>
</tr>
<tr>
<td>AZ</td>
<td>PLNUM</td>
<td></td>
<td>dummy variable, used as either diameter-area ratio or flow adjustment</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>A1</td>
<td>FLOWS</td>
<td></td>
<td>interpolating factor in friction factor calculation in transitional Reynolds number range</td>
</tr>
<tr>
<td>A1</td>
<td>TARRAY</td>
<td></td>
<td>upstream half of inside-wall heat-transfer area, in², associated with coolant-channel node</td>
</tr>
<tr>
<td>A2</td>
<td>FLOWS</td>
<td></td>
<td>interpolating factor in friction factor calculation in transitional Reynolds number range</td>
</tr>
<tr>
<td>A2</td>
<td>TARRAY</td>
<td></td>
<td>downstream half of inner surface heat-transfer area, in², associated with coolant-channel node</td>
</tr>
<tr>
<td>A3</td>
<td>TARRAY</td>
<td></td>
<td>same as A1, but on opposite wall, only used in trailing-edge region</td>
</tr>
<tr>
<td>A4</td>
<td>TARRAY</td>
<td></td>
<td>same as A2, but on opposite wall, only used in trailing-edge region</td>
</tr>
<tr>
<td>B</td>
<td>GETIN</td>
<td></td>
<td>ratio of length to thickness, used along with AA to calculate interpolated values of TDLX(2), TDLX(3), and TDLX(5)</td>
</tr>
<tr>
<td>B(20)</td>
<td>PLNUM</td>
<td></td>
<td>spanwise static-temperature distribution, °R, in coolant plenum for given slice</td>
</tr>
<tr>
<td>BC</td>
<td>GETIN</td>
<td></td>
<td>NAMELIST name</td>
</tr>
<tr>
<td>BCHGP(1000)</td>
<td>BOUND</td>
<td></td>
<td>input table of hot-gas, pressure-side heat-transfer coefficients, W/m²·K (Btu/hr·ft²·°F)</td>
</tr>
<tr>
<td>BCHGS(1000)</td>
<td>BOUND</td>
<td></td>
<td>input table of hot-gas, suction-side heat-transfer coefficients, W/m²·K (Btu/hr·ft²·°F)</td>
</tr>
<tr>
<td>BCPGP(1000)</td>
<td>BOUND</td>
<td></td>
<td>input table of hot-gas, pressure-side relative static pressure, kPa (lbf/in²)</td>
</tr>
<tr>
<td>BCPGS(1000)</td>
<td>BOUND</td>
<td></td>
<td>input table of hot-gas, suction-side relative static pressure, kPa (lbf/in²)</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>BCQGP(1000)</td>
<td>BOUND</td>
<td></td>
<td>input table of hot-gas, pressure-side heat flux, ( \text{W/m}^2 ) (( \text{Btu/hr} \cdot \text{ft}^2 ))</td>
</tr>
<tr>
<td>BCQGS(1000)</td>
<td>BOUND</td>
<td></td>
<td>input table of hot-gas, suction-side heat flux, ( \text{W/m}^2 ) (( \text{Btu/hr} \cdot \text{ft}^2 ))</td>
</tr>
<tr>
<td>BCTGP(1000)</td>
<td>BOUND</td>
<td></td>
<td>input table of hot-gas, pressure-side temperature, ( K \ (^o \text{F}) )</td>
</tr>
<tr>
<td>BCTGS(1000)</td>
<td>BOUND</td>
<td></td>
<td>input table of hot-gas, suction-side temperature, ( K \ (^o \text{F}) )</td>
</tr>
<tr>
<td>BCTIME(50)</td>
<td>BOUND</td>
<td></td>
<td>input table of time at which transient input tables are specified, sec</td>
</tr>
<tr>
<td>BCXP(400)</td>
<td>BOUND</td>
<td></td>
<td>input table of outer-surface, pressure-side locations at which hot-gas conditions are input, cm (in.)</td>
</tr>
<tr>
<td>BCXS(400)</td>
<td>BOUND</td>
<td></td>
<td>input table of outer-surface, suction-side locations at which hot-gas conditions are input, cm (in.)</td>
</tr>
<tr>
<td>BES</td>
<td></td>
<td>HCOOL</td>
<td>equivalent slot width, in., used in leading-edge impingement correlation</td>
</tr>
<tr>
<td>BETA</td>
<td></td>
<td>FRIC</td>
<td>constant used in friction factor calculations</td>
</tr>
<tr>
<td>BETA1</td>
<td></td>
<td>PLNUM</td>
<td>square of pressure at inlet to coolant plenum for given slice, ( (\text{lbf/in}^2)^2 )</td>
</tr>
<tr>
<td>BETTA(20)</td>
<td></td>
<td>PLNUM</td>
<td>spanwise static-pressure distribution in coolant plenum, lbf/in^2</td>
</tr>
<tr>
<td>BTA</td>
<td></td>
<td>TCO</td>
<td>indicates type of hot-gas boundary condition</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>FLOWS</td>
<td>gas thermal conductivity, ( \text{Btu/hr} \cdot \text{ft} \cdot \text{o} \text{F} )</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>CD</td>
<td>TCO</td>
<td></td>
<td>impingement-jet discharge coefficient</td>
</tr>
<tr>
<td>CDEN(2)</td>
<td>UNITS</td>
<td></td>
<td>conversion factor for density units</td>
</tr>
<tr>
<td>CD1(200)</td>
<td>NTTACT</td>
<td></td>
<td>dummy variable used to print selected intermediate temperature values</td>
</tr>
<tr>
<td>CEXCSW</td>
<td>NTTACT</td>
<td></td>
<td>amount of excess coolant flow, in SI units, kg/hr</td>
</tr>
<tr>
<td>CGASC(2)</td>
<td>UNITS</td>
<td></td>
<td>conversion factor for gas constant</td>
</tr>
<tr>
<td>CH(15)</td>
<td>PLNUM</td>
<td></td>
<td>coolant-channel choking indicator</td>
</tr>
<tr>
<td>CHANL(8000)</td>
<td>SPECL</td>
<td></td>
<td>array for storing input data</td>
</tr>
<tr>
<td>CHANLS</td>
<td>GETIN</td>
<td></td>
<td>NAMELIST name</td>
</tr>
<tr>
<td>CHFLX(2)</td>
<td>UNITS</td>
<td></td>
<td>conversion factor for heat-flux units</td>
</tr>
<tr>
<td>CHTC(2)</td>
<td>UNITS</td>
<td></td>
<td>conversion factor for heat-transfer-coefficient units</td>
</tr>
<tr>
<td>CIMP1</td>
<td>IMPCOR</td>
<td></td>
<td>user-supplied constants for general impingement correlation</td>
</tr>
<tr>
<td>CIMP2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIMP3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIMP4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIMP5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIMP6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIMP7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CINCH(2)</td>
<td>UNITS</td>
<td></td>
<td>conversion factor for length units</td>
</tr>
<tr>
<td>CMSFL(2)</td>
<td>UNITS</td>
<td></td>
<td>conversion factor for mass flow rate units</td>
</tr>
<tr>
<td>CNUM(80)</td>
<td>TCO</td>
<td></td>
<td>number of impingement jets at each station for given slice</td>
</tr>
<tr>
<td>CONDCT</td>
<td>HCOOL</td>
<td></td>
<td>coolant-air thermal conductivity, Btu/hr \cdot ft \cdot ^\circ F</td>
</tr>
<tr>
<td>CONTRL</td>
<td>GETIN</td>
<td></td>
<td>NAMELIST name</td>
</tr>
<tr>
<td>CP</td>
<td>TCO</td>
<td></td>
<td>gas specific heat at constant pressure, Btu/lbm \cdot ^\circ F</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>CPC(\text{80})</td>
<td>PRPS</td>
<td></td>
<td>coolant specific heat at constant pressure at each coolant node for given slice, Btu/lbm \cdot {^\circ}F evaluated at a mean temperature between bulk coolant temperature and wall temperature</td>
</tr>
<tr>
<td>CPIM</td>
<td>NTTACT</td>
<td>PLOTMF</td>
<td>mean impingement-plenum pressure for given slice, in SI units, kPa</td>
</tr>
<tr>
<td>CPM</td>
<td></td>
<td>FLOWS</td>
<td>hot-gas-stream specific heat at constant pressure, Btu/lbm \cdot {^\circ}F</td>
</tr>
<tr>
<td>CPO</td>
<td>PRPS</td>
<td></td>
<td>specific heat at constant pressure, Btu/lbm \cdot {^\circ}F, evaluated at impingement-jet supply temperature</td>
</tr>
<tr>
<td>CPRSR(\text{2})</td>
<td>UNITS</td>
<td></td>
<td>conversion factor for pressure units</td>
</tr>
<tr>
<td>CRHOVG(\text{2})</td>
<td>UNITS</td>
<td></td>
<td>conversion factor for density \times velocity units</td>
</tr>
<tr>
<td>CRITR</td>
<td></td>
<td>FLSPLT</td>
<td>coolant flow-split convergence criterion</td>
</tr>
<tr>
<td>CSPHT(\text{2})</td>
<td>UNITS</td>
<td></td>
<td>conversion factor for specific-heat units</td>
</tr>
<tr>
<td>CTCON(\text{2})</td>
<td>UNITS</td>
<td></td>
<td>conversion factor for thermal conductivity units</td>
</tr>
<tr>
<td>CTMPF(\text{2})</td>
<td>UNITS</td>
<td></td>
<td>conversion factor for temperature units</td>
</tr>
<tr>
<td>CT0G</td>
<td></td>
<td>NTTACT</td>
<td>mean impingement-plenum static temperature for given slice, in SI units, K</td>
</tr>
<tr>
<td>CURV</td>
<td></td>
<td>TARRAY</td>
<td>factor to account for wall curvature in heat-conduction equations</td>
</tr>
<tr>
<td>CVISC(\text{2})</td>
<td>UNITS</td>
<td></td>
<td>conversion factor for viscosity units</td>
</tr>
<tr>
<td>CWPLEN</td>
<td></td>
<td>NTTACT</td>
<td>coolant-plenum flow rate at entrance to given slice, in SI units, kg/hr</td>
</tr>
<tr>
<td>CWUSED</td>
<td></td>
<td>NTTACT</td>
<td>total amount of coolant air used, in SI units, kg/hr</td>
</tr>
<tr>
<td>CX</td>
<td></td>
<td>PLNUM</td>
<td>function of isentropic exponent (k), [-\frac{(k + 1)}{[2(k - 1)]}]</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>C1</td>
<td></td>
<td>PLNUM</td>
<td>function of isentropic exponent k, $2k/(k - 1)$</td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td>PLNUM</td>
<td>computed constant involving wheel speed and isentropic exponent</td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td>FLOWS</td>
<td>ratio of specific heats at constant pressure, coolant to hot gas</td>
</tr>
<tr>
<td>C5</td>
<td></td>
<td>PLNUM</td>
<td>computed constant involving isentropic exponent and gas constant</td>
</tr>
<tr>
<td>C6</td>
<td></td>
<td>PLNUM</td>
<td>function of isentropic exponent k, $(k - 1)/2$</td>
</tr>
<tr>
<td>C7</td>
<td></td>
<td>PLNUM</td>
<td>computed constant involving isentropic exponent and gas constant</td>
</tr>
<tr>
<td>C8</td>
<td></td>
<td>PLNUM</td>
<td>computed constant involving isentropic exponent and gas constant</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>PLNUM</td>
<td>convergence parameter in coolant-plenum pressure calculations</td>
</tr>
<tr>
<td>DD</td>
<td></td>
<td>PLNUM</td>
<td>coolant-plenum hydraulic diameter, in</td>
</tr>
<tr>
<td>DEH</td>
<td></td>
<td>HCOOL</td>
<td>hydraulic diameter of equivalent slot, in., used in leading-edge impingement correlation</td>
</tr>
<tr>
<td>DELAST</td>
<td></td>
<td>FLSPLT</td>
<td>variable used to save flow-split fraction at which pressure-match parameter, EPSN, changes sign</td>
</tr>
<tr>
<td>DELTA</td>
<td></td>
<td>FRIC</td>
<td>constant used in friction factor calculation</td>
</tr>
<tr>
<td>DELTAN(15)</td>
<td></td>
<td>FLOWS</td>
<td>fraction of stagnation-point impingement-jet flow that splits to suction-side</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FLSPLT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HCPINS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TARRAY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCOEF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WROUT</td>
<td></td>
</tr>
<tr>
<td>DELTAO</td>
<td></td>
<td>FLSPLT</td>
<td>value of DELTAN from previous flow-split iteration</td>
</tr>
</tbody>
</table>

19
<table>
<thead>
<tr>
<th>Variable</th>
<th>Common</th>
<th>Subroutine</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DENOM</td>
<td></td>
<td>NTTACT</td>
<td>intermediate variable used in time interpolation of some boundary conditions</td>
</tr>
<tr>
<td>DH(80)</td>
<td>TCO</td>
<td></td>
<td>coolant-channel hydraulic diameter at each station, in.</td>
</tr>
<tr>
<td>DHF(80)</td>
<td>TCO</td>
<td></td>
<td>effective diameter of film-cooling hole at each station, in., defined as hydraulic diameter of one hole multiplied by square root of number of holes at station</td>
</tr>
<tr>
<td>DHJ(80)</td>
<td>TCO</td>
<td></td>
<td>actual hydraulic diameter of an impingement hole at each station, in.</td>
</tr>
<tr>
<td>DHYD</td>
<td></td>
<td>GETIN</td>
<td>input value of coolant-plenum hydraulic diameter for a slice, cm (in.)</td>
</tr>
<tr>
<td>DIFN</td>
<td></td>
<td>TCOEF</td>
<td>pressure difference parameter used in checking convergence</td>
</tr>
<tr>
<td>DIFO</td>
<td></td>
<td>TCOEF</td>
<td>maximum pressure difference parameter used in checking convergence</td>
</tr>
<tr>
<td>DIFTOL</td>
<td></td>
<td>PLNUM</td>
<td>tolerance on coolant-plenum pressure-drop calculations</td>
</tr>
<tr>
<td>DIMP1</td>
<td></td>
<td>IMPCOR</td>
<td>user-supplied constants for leading-edge impingement correlation</td>
</tr>
<tr>
<td>DIMP2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIMP3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIMP4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIMP5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIMP6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLTAOP</td>
<td></td>
<td>FLSPLT</td>
<td>best value of DELTAN in the event of an unstable flow split</td>
</tr>
<tr>
<td>DLYTME</td>
<td></td>
<td>TRNSNT</td>
<td>time step used in transient calculations, sec</td>
</tr>
<tr>
<td>DLX(400)</td>
<td>TCO</td>
<td></td>
<td>chordwise distance from each node to adjacent upstream node, in.</td>
</tr>
<tr>
<td>DP(80)</td>
<td>PRPS</td>
<td></td>
<td>diameter of pin fins at each station, in.</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DPLN(15)</td>
<td>RADL</td>
<td></td>
<td>coolant-plenum hydraulic diameter for each slice, in.</td>
</tr>
<tr>
<td>DR</td>
<td>PLNUM</td>
<td></td>
<td>radial length increment in coolant-plenum calculations, in.</td>
</tr>
<tr>
<td>DR2</td>
<td>PLNUM</td>
<td></td>
<td>radial increment squared, $\text{in}^2$</td>
</tr>
<tr>
<td>DUMR1(80)</td>
<td>PRPS</td>
<td></td>
<td>dummy variable, not currently used</td>
</tr>
<tr>
<td>DUMR2(80)</td>
<td>PRPS</td>
<td></td>
<td>dummy variable, used to carry impingement-jet Reynolds number to output subroutine</td>
</tr>
<tr>
<td>DUMTER</td>
<td>PARRAY</td>
<td></td>
<td>intermediate variable in momentum equation involving coolant dumped directly into trailing edge</td>
</tr>
<tr>
<td>DUM1(10)</td>
<td>INPRT</td>
<td>WROUT</td>
<td>dummy variables used to print input listings and program output</td>
</tr>
<tr>
<td>DUM2(10)</td>
<td>INPRT</td>
<td>WROUT</td>
<td></td>
</tr>
<tr>
<td>DUM3(10)</td>
<td>INPRT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM4(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM5(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM6(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM7(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM8(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM9(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM10(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM11(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM12(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM13(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM14(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM15(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM16(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>DUM17(10)</td>
<td></td>
<td>INPRT</td>
<td>dummy variables used to print input listings and program output</td>
</tr>
<tr>
<td>DUM18(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM19(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM20(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM25(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM52(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM53(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DUM55(10)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DX</td>
<td></td>
<td>PLNUM</td>
<td>spanwise step size used in calculating coolant-plenum pressure and temperature distributions</td>
</tr>
<tr>
<td>DXTEMP</td>
<td></td>
<td>PLNUM</td>
<td>variable to temporarily hold DX</td>
</tr>
<tr>
<td>DX1</td>
<td></td>
<td>TARRAY</td>
<td>path length between midwall node and adjacent upstream midwall node, in.</td>
</tr>
<tr>
<td>DX10</td>
<td></td>
<td>TARRAY</td>
<td>path length between outer-coating-wall junction node and adjacent downstream outer-coating-wall junction node, in.</td>
</tr>
<tr>
<td>DX2</td>
<td></td>
<td>TARRAY</td>
<td>path length between midwall node and adjacent downstream midwall node, in.</td>
</tr>
<tr>
<td>DX3</td>
<td></td>
<td>TARRAY</td>
<td>path length between outer-surface node and adjacent upstream outer-surface node, in.</td>
</tr>
<tr>
<td>DX4</td>
<td></td>
<td>TARRAY</td>
<td>path length between outer-surface node and adjacent downstream outer-surface node, in.</td>
</tr>
<tr>
<td>DX5</td>
<td></td>
<td>TARRAY</td>
<td>path length between inner-surface node and adjacent upstream inner-surface node, in.</td>
</tr>
<tr>
<td>DX6</td>
<td></td>
<td>TARRAY</td>
<td>path length between inner-surface node and adjacent downstream inner-surface node, in.</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>DX7</td>
<td>TARRAY</td>
<td></td>
<td>path length between coolant node and adjacent upstream coolant node, in.</td>
</tr>
<tr>
<td>DX9</td>
<td>TARRAY</td>
<td></td>
<td>path length between outer coating - wall junction node and adjacent upstream outer coating - wall junction node, in.</td>
</tr>
<tr>
<td>D1</td>
<td>PLNUM</td>
<td></td>
<td>computed constant used in coolant-plenum pressure equations to account for effect of pumping due to wheel rotation</td>
</tr>
<tr>
<td>D2</td>
<td>PLNUM</td>
<td></td>
<td>computed constant used in coolant-plenum temperature equations to account for effect of pumping due to wheel rotation</td>
</tr>
<tr>
<td>E</td>
<td>PLNUM</td>
<td></td>
<td>factor used in adjusting convergence rate in coolant-plenum calculations</td>
</tr>
<tr>
<td>EFAREA(80)</td>
<td>HCPINS</td>
<td>TARRAY</td>
<td>effective area, in², for heat transfer at stations with pin fins, including pin-fin effectiveness for heat transfer</td>
</tr>
<tr>
<td>EFTVNS</td>
<td>HCPINS</td>
<td></td>
<td>pin-fin effectiveness</td>
</tr>
<tr>
<td>EMES(80)</td>
<td>FLMCOL</td>
<td></td>
<td>for film cooling, ratio of coolant mass flux to free-stream mass flux, multiplied by equivalent slot width</td>
</tr>
<tr>
<td>EML</td>
<td>HCPINS</td>
<td></td>
<td>term used in pin-fin effectiveness calculation</td>
</tr>
<tr>
<td>ENDEFF</td>
<td>TARRAY</td>
<td></td>
<td>term in heat-transfer equations to account for convection to rear edge of blade when heat-transfer coefficients are input</td>
</tr>
<tr>
<td>ENDFLX</td>
<td>TARRAY</td>
<td></td>
<td>term in heat-transfer equations to account for convection to rear edge of blade when heat flux is input</td>
</tr>
<tr>
<td>EPLAST</td>
<td>FLSPLT</td>
<td></td>
<td>variable used to save latest value of pressure-match parameter, EPSN</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EPS</td>
<td>FRIC</td>
<td></td>
<td>constant used in friction factor calculations</td>
</tr>
<tr>
<td>EPSMIN</td>
<td>FLSPLT</td>
<td></td>
<td>minimum value attained by pressure-match parameter, EPSN, for unstable flow split</td>
</tr>
<tr>
<td>EPSN</td>
<td>FLSPLT</td>
<td>TCOEF</td>
<td>pressure-match parameter, defined as difference between suction- and pressure-side coolant-channel static pressures at end of insert, divided by suction-side coolant-channel static pressure</td>
</tr>
<tr>
<td>EPSO</td>
<td>FLSPLT</td>
<td></td>
<td>old value of pressure-match parameter, EPSN</td>
</tr>
<tr>
<td>ETAPRM</td>
<td>FLOWS</td>
<td></td>
<td>film-cooling effectiveness based on ratio of enthalpy differences</td>
</tr>
<tr>
<td>EXCESW</td>
<td>NTTACT</td>
<td></td>
<td>amount of excess coolant flow, difference between inlet flow and that actually used, lbm/hr</td>
</tr>
<tr>
<td>FACTOR</td>
<td>TARRAY</td>
<td></td>
<td>special variable to adjust amount of energy carried in by an impingement jet for case of a calculation station adjacent to flow-split station</td>
</tr>
<tr>
<td>FF(80)</td>
<td>TCO</td>
<td></td>
<td>value of friction factor at each flow station</td>
</tr>
<tr>
<td>FILM</td>
<td>PARRAY</td>
<td></td>
<td>term to account for momentum carried off by film-cooling air</td>
</tr>
<tr>
<td>FILMW</td>
<td>TARRAY</td>
<td></td>
<td>total film-cooling flow from given coolant node</td>
</tr>
<tr>
<td>FLMEFF(80)</td>
<td>FLMCOL</td>
<td></td>
<td>film-cooling effectiveness based on ratio of temperature differences</td>
</tr>
<tr>
<td>FM</td>
<td>GAUSS</td>
<td></td>
<td>multiplying factor used in Gauss elimination scheme</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>FUNP</td>
<td>PLNUM</td>
<td></td>
<td>statement function to calculate pressure difference in coolant plenum</td>
</tr>
<tr>
<td>FUNT</td>
<td>PLNUM</td>
<td></td>
<td>statement function to calculate temperature difference in coolant plenum</td>
</tr>
<tr>
<td>F1(20)</td>
<td>PLNUM</td>
<td></td>
<td>friction factor in coolant plenum</td>
</tr>
<tr>
<td>GAM</td>
<td>TCO</td>
<td></td>
<td>ratio of specific heats</td>
</tr>
<tr>
<td>GAMC(E80)</td>
<td>PRPS</td>
<td></td>
<td>ratio of specific heats at each coolant-channel node</td>
</tr>
<tr>
<td>GAMO</td>
<td>PRPS</td>
<td></td>
<td>ratio of specific heats at coolant-supply conditions</td>
</tr>
<tr>
<td>GEO</td>
<td>GETIN</td>
<td></td>
<td>NAMELIST name</td>
</tr>
<tr>
<td>GG</td>
<td>HCOOL</td>
<td></td>
<td>mass flux ratio, coolant crossflow to impingement-jet flow</td>
</tr>
<tr>
<td>GI</td>
<td>HCOOL</td>
<td></td>
<td>momentum flux ratio, coolant crossflow to impingement-jet flow</td>
</tr>
<tr>
<td>GMASS</td>
<td>HCOOL</td>
<td></td>
<td>mass flux from row of leading-edge impingement holes</td>
</tr>
<tr>
<td>GS(200)</td>
<td>GAAS</td>
<td></td>
<td>table of gas properties</td>
</tr>
<tr>
<td>G1</td>
<td>PLNUM</td>
<td></td>
<td>computed constant in coolant-plenum calculations, involving flow rate, gas constant, and specific heat at constant pressure</td>
</tr>
<tr>
<td>G2</td>
<td>PLNUM</td>
<td></td>
<td>computed constant in coolant-plenum calculations, involving flow rate and gas constant</td>
</tr>
<tr>
<td>HBAR</td>
<td>WROUT</td>
<td></td>
<td>average coolant-side heat-transfer coefficient for given slice, W/m²·K (Btu/hr·ft²·°F)</td>
</tr>
<tr>
<td>HC(E80)</td>
<td>TCO</td>
<td></td>
<td>coolant-side heat-transfer coefficients at each station for given slice, Btu/hr·ft²·°F</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>HCAL(4)</td>
<td>INPRT</td>
<td></td>
<td>alphameric array containing labels identifying type of coolant-side heat transfer</td>
</tr>
<tr>
<td>HG(80)</td>
<td>TCO</td>
<td></td>
<td>hot-gas-side heat-transfer coefficient at each station for given slice, Btu/hr \cdot ft^2 \cdot ^\circ F</td>
</tr>
<tr>
<td>HSTGMX</td>
<td>TCOEF</td>
<td></td>
<td>maximum physically possible value of coolant heat-transfer coefficient under stagnation jet, Btu/hr \cdot ft^2 \cdot ^\circ F</td>
</tr>
<tr>
<td>HUB1</td>
<td>TARRAY</td>
<td></td>
<td>term in conduction equation to account for specified hub temperature</td>
</tr>
<tr>
<td>HUB3</td>
<td>TARRAY</td>
<td></td>
<td>term in conduction equation to account for specified hub heat flux</td>
</tr>
<tr>
<td>HX</td>
<td>TARRAY</td>
<td></td>
<td>multiplying factor on coolant heat-transfer coefficient, initialized to 1.0 but may be changed dynamically</td>
</tr>
<tr>
<td>HYCOS</td>
<td>HCPINS</td>
<td></td>
<td>hyperbolic cosine term</td>
</tr>
<tr>
<td>HYSIN</td>
<td>HCPINS</td>
<td></td>
<td>hyperbolic sine term</td>
</tr>
<tr>
<td>IADJIN</td>
<td>SPECL</td>
<td></td>
<td>input variable that indicates which coolant-plenum supply property is to be held fixed</td>
</tr>
<tr>
<td>ICHK</td>
<td>PARRAY</td>
<td></td>
<td>indicates which side of blade a given station is on: 0 if suction side, 1 if pressure side</td>
</tr>
<tr>
<td>ICHNL</td>
<td>INPRT</td>
<td>PREP</td>
<td>slice number, carried through as argument</td>
</tr>
<tr>
<td>ICHOKE</td>
<td>FLOWS</td>
<td>PARRAY</td>
<td>number of station that shows choked coolant flow</td>
</tr>
<tr>
<td>ICOMP</td>
<td>TARRAY</td>
<td>TCOEF</td>
<td>number of station adjacent to impingement flow-split station in pressure-side direction</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>ICOMS</td>
<td>TARRAY</td>
<td>FLSPLT</td>
<td>number of station adjacent to impingement flow-split station in suction-side direction</td>
</tr>
<tr>
<td>ICONV</td>
<td></td>
<td>TCOEF</td>
<td>indicator for convergence of flow-split iterations</td>
</tr>
<tr>
<td>ICOR</td>
<td>TCO</td>
<td></td>
<td>station at which use of impingement-with-crossflow correlation is to begin</td>
</tr>
<tr>
<td>IDELT</td>
<td>FLSPLT</td>
<td>TCOEF</td>
<td>counter of number of flow-split iterations performed</td>
</tr>
<tr>
<td>IDN</td>
<td>PARRAY</td>
<td></td>
<td>downstream node number for coolant-channel pressure calculations</td>
</tr>
<tr>
<td>IDNS</td>
<td>PARRAY</td>
<td></td>
<td>downstream station number for coolant-channel pressure calculations</td>
</tr>
<tr>
<td>IDX</td>
<td>PARRAY</td>
<td></td>
<td>upstream node number for coolant-channel pressure calculations</td>
</tr>
<tr>
<td>IEND</td>
<td>GETIN</td>
<td></td>
<td>last point in CHANL array occupied by data for given slice</td>
</tr>
<tr>
<td>IFCP</td>
<td>FLOWS</td>
<td></td>
<td>indicator used in locating pressure-side film-cooling holes</td>
</tr>
<tr>
<td>IFCS</td>
<td>FLOWS</td>
<td></td>
<td>indicator used in locating suction-side film-cooling holes</td>
</tr>
<tr>
<td>IFILM</td>
<td>TCO</td>
<td></td>
<td>input indicator for film cooling</td>
</tr>
<tr>
<td>IFLU</td>
<td>PREP</td>
<td></td>
<td>coolant-channel node number</td>
</tr>
<tr>
<td>IFNL</td>
<td>PARRAY</td>
<td></td>
<td>number of coolant-channel nodes</td>
</tr>
<tr>
<td>IFNL</td>
<td>TCOEF</td>
<td></td>
<td>total number of stations, minus 1</td>
</tr>
<tr>
<td>IFSPLT</td>
<td>FLOWS</td>
<td></td>
<td>indicates in which direction film-cooling air flows from stagnation station</td>
</tr>
<tr>
<td>IGG(80)</td>
<td>HCOOL</td>
<td></td>
<td>array containing node numbers at which ratio of coolant crossflow to impingement-jet flow is out of Kercher-Tabakoff correlation range</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IGGC</td>
<td>HCOOL</td>
<td></td>
<td>counts number of entries in IGG array</td>
</tr>
<tr>
<td>IHC(80)</td>
<td>TCO</td>
<td>GETIN</td>
<td>indicates type of coolant-side heat transfer at each station for given slice</td>
</tr>
<tr>
<td>IHCT</td>
<td>GETIN</td>
<td></td>
<td>input value of IHC for given station</td>
</tr>
<tr>
<td>IHUB</td>
<td>TCO</td>
<td></td>
<td>indicates type of boundary to be used at hub end of blade</td>
</tr>
<tr>
<td>II</td>
<td>HCOOL</td>
<td></td>
<td>coolant-channel node number</td>
</tr>
<tr>
<td>IIHCTZ</td>
<td>GETIN</td>
<td></td>
<td>locates IHC array in overall array</td>
</tr>
<tr>
<td>ILEAD</td>
<td>HCOOL</td>
<td></td>
<td>last station in range of leading-edge impingement correlation</td>
</tr>
<tr>
<td>IMS</td>
<td>FLOWS</td>
<td></td>
<td>location of film–cooling hole preceding current station</td>
</tr>
<tr>
<td>INDCHN(2000)</td>
<td>SPECL</td>
<td></td>
<td>array for storing integer input data</td>
</tr>
<tr>
<td>INEDIT</td>
<td>GETIN</td>
<td></td>
<td>input variable to control listing of input data</td>
</tr>
<tr>
<td>INN</td>
<td>PREP</td>
<td></td>
<td>location of IHC data in INDCHN array</td>
</tr>
<tr>
<td>INSTRRT</td>
<td>GETIN</td>
<td></td>
<td>starting point for storage of integer data in INDCHN array for given slice</td>
</tr>
<tr>
<td>INUM</td>
<td>NTTACT</td>
<td></td>
<td>number of stations on each side of blade</td>
</tr>
<tr>
<td>IN1</td>
<td>PREP</td>
<td></td>
<td>location of end of group of single-valued variables in INDCHN array</td>
</tr>
<tr>
<td>IPLT</td>
<td>SPECL</td>
<td></td>
<td>input indicator to control plotting options</td>
</tr>
<tr>
<td>IPRES</td>
<td>WROUT</td>
<td></td>
<td>pressure-side, outer-surface node number</td>
</tr>
<tr>
<td>IPRSMN</td>
<td>WROUT</td>
<td></td>
<td>location of minimum outer-surface temperature on pressure side</td>
</tr>
<tr>
<td>IPRSMX</td>
<td>WROUT</td>
<td></td>
<td>location of maximum outer-surface temperature on pressure side</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IRE(80)</td>
<td></td>
<td>HCOOL</td>
<td>list of station numbers at which impingement-jet Reynolds number is out of Kercher-Tabakoff correlation range</td>
</tr>
<tr>
<td>IRL</td>
<td></td>
<td>PARRAY</td>
<td>coolant node number</td>
</tr>
<tr>
<td>ISBLOK</td>
<td>TCO</td>
<td></td>
<td>starting point in CHANL array of data for given slice</td>
</tr>
<tr>
<td>ISEN</td>
<td></td>
<td>TARRAY</td>
<td>indicates which side of blade station IS is on</td>
</tr>
<tr>
<td>ISENS</td>
<td></td>
<td>FLOWS</td>
<td>indicates which side of blade a given station is on</td>
</tr>
<tr>
<td>ISENS</td>
<td></td>
<td>TARRAY</td>
<td>indicates which side of blade a given trailing-edge region station is on</td>
</tr>
<tr>
<td>ISLICE</td>
<td>TCO</td>
<td></td>
<td>current slice number</td>
</tr>
<tr>
<td>ISTA</td>
<td></td>
<td>GETIN</td>
<td>input station number</td>
</tr>
<tr>
<td>ISTAT</td>
<td></td>
<td>WROUT</td>
<td>outer-surface node number for given station</td>
</tr>
<tr>
<td>ISTATD</td>
<td></td>
<td>WROUT</td>
<td>outer-surface node number immediately downstream of ISTAT</td>
</tr>
<tr>
<td>ISTB</td>
<td></td>
<td>GETIN</td>
<td>input station number</td>
</tr>
<tr>
<td>ISTRRT</td>
<td></td>
<td>HCOOL</td>
<td>station at which use of Kercher-Tabakoff correlation begins</td>
</tr>
<tr>
<td>ISTRRT</td>
<td></td>
<td>TCOEF</td>
<td>first station in trailing-edge region</td>
</tr>
<tr>
<td>ISUCMN</td>
<td></td>
<td>WROUT</td>
<td>location of minimum outer-surface temperature on suction side</td>
</tr>
<tr>
<td>ISUCMX</td>
<td></td>
<td>WROUT</td>
<td>location of maximum outer-surface temperature on suction side</td>
</tr>
<tr>
<td>ISUCT</td>
<td></td>
<td>WROUT</td>
<td>suction-side station number</td>
</tr>
<tr>
<td>ISUP</td>
<td></td>
<td>TARRAY</td>
<td>adjacent station, upstream of current station, IS</td>
</tr>
<tr>
<td>ISYM(5)</td>
<td></td>
<td>PLOTMF</td>
<td>data array containing plotting symbol codes</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>ITDHFZ</td>
<td>GETIN</td>
<td></td>
<td>locates film-cooling-hole data in CHANL array for given slice</td>
</tr>
<tr>
<td>ITDHJZ</td>
<td>GETIN</td>
<td></td>
<td>locates impingement-hole data in CHANL array for given slice</td>
</tr>
<tr>
<td>ITDLXZ</td>
<td>GETIN</td>
<td></td>
<td>locates node spacing data in CHANL array for given slice</td>
</tr>
<tr>
<td>ITDPZ</td>
<td>GETIN</td>
<td></td>
<td>locates pin-fin diameter data in CHANL array for given slice</td>
</tr>
<tr>
<td>ITHKZ</td>
<td>GETIN</td>
<td></td>
<td>locates wall and channel thickness data in CHANL array for given slice</td>
</tr>
<tr>
<td>ITIP</td>
<td>TCO</td>
<td></td>
<td>indicates type of boundary to be used at tip end of blade</td>
</tr>
<tr>
<td>ITRBG</td>
<td>INPRT</td>
<td></td>
<td>first station in trailing-edge region</td>
</tr>
<tr>
<td>ITRBG</td>
<td>WROUT</td>
<td></td>
<td>first station in trailing-edge region</td>
</tr>
<tr>
<td>ITREO</td>
<td>WROUT</td>
<td></td>
<td>last outside node at trailing edge</td>
</tr>
<tr>
<td>ITRRZ</td>
<td>GETIN</td>
<td></td>
<td>locates radial position data in CHANL array for given slice</td>
</tr>
<tr>
<td>ITSPZ</td>
<td>GETIN</td>
<td></td>
<td>locates pin-fin spacing data in CHANL array for given slice</td>
</tr>
<tr>
<td>ITXNZ</td>
<td>GETIN</td>
<td></td>
<td>locates impingement-hole spacing data in CHANL array for given slice</td>
</tr>
<tr>
<td>IUNITS</td>
<td>UNITS</td>
<td></td>
<td>indicates system of units used for input data</td>
</tr>
<tr>
<td>IUNSTB</td>
<td>FLSPLT</td>
<td></td>
<td>indicates whether flow split is stable or not</td>
</tr>
<tr>
<td>IUP</td>
<td>PARRAY</td>
<td></td>
<td>upstream node number for coolant-channel pressure calculations</td>
</tr>
<tr>
<td>IUPS</td>
<td>PARRAY</td>
<td></td>
<td>upstream station number for coolant-channel pressure calculations</td>
</tr>
<tr>
<td>IVARS(12)</td>
<td>PLOTMF</td>
<td></td>
<td>array of integer plotting controls</td>
</tr>
<tr>
<td>IVERGE</td>
<td>TCOEF</td>
<td>WROUT</td>
<td>pressure iteration loop counter</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>IWR</td>
<td></td>
<td>GAUSS</td>
<td>control on debugging output of coefficient matrix</td>
</tr>
<tr>
<td>IWRITE</td>
<td>SPECL</td>
<td></td>
<td>input control on amount of printed output</td>
</tr>
<tr>
<td>IXAX</td>
<td></td>
<td>PLOTMF</td>
<td>logical variable with value .TRUE.</td>
</tr>
<tr>
<td>IYAX</td>
<td></td>
<td>PLOTMF</td>
<td>logical variable with value .FALSE.</td>
</tr>
<tr>
<td>I1</td>
<td></td>
<td>PREP</td>
<td>starting point in INDCHN array for integer data for given slice, also used as starting point for nodal data in CHANL array for given slice</td>
</tr>
<tr>
<td>I3</td>
<td></td>
<td>PREP</td>
<td>starting point in CHANL array for station data</td>
</tr>
<tr>
<td>JDIS</td>
<td></td>
<td>FLOWS</td>
<td>number of stations that impingement flow-split station is displaced from station 1</td>
</tr>
<tr>
<td>JHCAL</td>
<td></td>
<td>INPRT</td>
<td>indicates type of heat transfer at given station</td>
</tr>
<tr>
<td>JLSTM</td>
<td></td>
<td>THRCON</td>
<td>size of wall thermal conductivity tables</td>
</tr>
<tr>
<td>JNUMS</td>
<td></td>
<td>FLSPLT</td>
<td>indicates whether previous call to FLSPLT resulted in unstable flow split</td>
</tr>
<tr>
<td>JOUTRG</td>
<td></td>
<td>FLSPLT</td>
<td>indicates attempt to split more than 100 percent of flow to one side</td>
</tr>
<tr>
<td>JPIV</td>
<td></td>
<td>GAUSS</td>
<td>pivotal column in matrix to be reduced</td>
</tr>
<tr>
<td>JS</td>
<td></td>
<td>FLOWS</td>
<td>station number at impingement flow-split point</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FLSPLT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HCOOL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PARRAY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TARRAY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCOEF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>WROUT</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>-------------</td>
<td>---------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>JSENS</td>
<td>FLOWS</td>
<td>GETIN</td>
<td>indicates which side of blade impingement flow-split is on</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FLSPLT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PARRAY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TARRAY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCOEF</td>
<td></td>
</tr>
<tr>
<td>JSGNCH</td>
<td></td>
<td>FLSPLT</td>
<td>indicates whether a sign change has occurred in EPSN</td>
</tr>
<tr>
<td>JSO(15)</td>
<td></td>
<td>TCOEF</td>
<td>array for saving converged-flow-split station number for each slice</td>
</tr>
<tr>
<td>JSOLDS(25)</td>
<td></td>
<td>FLSPLT</td>
<td>array to keep track of stations checked as flow-split stations</td>
</tr>
<tr>
<td>JTIMES</td>
<td></td>
<td>FLSPLT</td>
<td>indicates whether DELTAN is to be rough adjusted or fine tuned</td>
</tr>
<tr>
<td>K</td>
<td>GAUSS</td>
<td></td>
<td>total bandwidth of matrix to be solved</td>
</tr>
<tr>
<td>K</td>
<td>NTTACT</td>
<td></td>
<td>counter of number of overall coolant-flow iterations</td>
</tr>
<tr>
<td>KSIG</td>
<td>PLNUM</td>
<td></td>
<td>coolant-plenum iteration counter</td>
</tr>
<tr>
<td>L</td>
<td>TARRAY</td>
<td></td>
<td>midwall node number at given station</td>
</tr>
<tr>
<td></td>
<td>THRCON</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LCOOL</td>
<td>FLOWS</td>
<td></td>
<td>coolant-channel node number at given station</td>
</tr>
<tr>
<td></td>
<td>HCFRCD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HCOOL</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HCPINS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TARRAY</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TCOEF</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>WROUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LCOOLP</td>
<td>HCPINS</td>
<td></td>
<td>inner-surface node across coolant-channel from given station</td>
</tr>
<tr>
<td></td>
<td>TARRAY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LCUP</td>
<td>HCPINS</td>
<td></td>
<td>coolant-channel node number upstream of given station</td>
</tr>
<tr>
<td></td>
<td>TARRAY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LCUPP</td>
<td>TARRAY</td>
<td></td>
<td>inner-surface node across coolant channel and upstream of given station</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>LCUPS</td>
<td>TARRAY</td>
<td></td>
<td>inner-surface node upstream of given station</td>
</tr>
<tr>
<td>LDN</td>
<td>TARRAY</td>
<td></td>
<td>midwall node number downstream of given station</td>
</tr>
<tr>
<td>LIN</td>
<td>FLOWS</td>
<td>HCFRCD</td>
<td>inner-surface node number at given station</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HCOOL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HCPINS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TARRAY</td>
<td></td>
</tr>
<tr>
<td>LJ</td>
<td>INPRT</td>
<td>TARRAY</td>
<td>node number at junction of outer coating and wall metal at given station</td>
</tr>
<tr>
<td></td>
<td></td>
<td>THRCON</td>
<td></td>
</tr>
<tr>
<td>LOUT</td>
<td>TARRAY</td>
<td>THRCON</td>
<td>wall outer-surface node number at given station</td>
</tr>
<tr>
<td>LSP</td>
<td>PREP</td>
<td></td>
<td>starting point in BCXP array of data for given slice</td>
</tr>
<tr>
<td>LSS</td>
<td>PREP</td>
<td></td>
<td>starting point in BCXS array of data for given slice</td>
</tr>
<tr>
<td>LUP</td>
<td>TARRAY</td>
<td>THRCON</td>
<td>midwall node number upstream of given station</td>
</tr>
<tr>
<td>MACH1</td>
<td>PLNUM</td>
<td></td>
<td>indicator to keep track of step-size change</td>
</tr>
<tr>
<td>MD1</td>
<td>SPECL</td>
<td></td>
<td>indicator to control special condensed, on-line output</td>
</tr>
<tr>
<td>MD2</td>
<td>SPECL</td>
<td></td>
<td>indicator that job is complete and summary plots are to be produced</td>
</tr>
<tr>
<td>MD3</td>
<td>SPECL</td>
<td></td>
<td>plot counter</td>
</tr>
<tr>
<td>NBC</td>
<td>INPRT</td>
<td></td>
<td>maximum of NBCS and NBCP</td>
</tr>
<tr>
<td>N</td>
<td>GAUSS</td>
<td></td>
<td>number of rows in matrix to be solved</td>
</tr>
<tr>
<td>NAG</td>
<td>PLNUM</td>
<td></td>
<td>indicator whether calculations have progressed beyond initial station</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NBCP</td>
<td>BOUND</td>
<td>INPRT</td>
<td>input number of boundary condition points on pressure side</td>
</tr>
<tr>
<td>NBCS</td>
<td>BOUND</td>
<td>INPRT</td>
<td>input number of boundary condition points on suction side</td>
</tr>
<tr>
<td>NBFRP</td>
<td></td>
<td>INPRT</td>
<td>number of pressure-side boundary condition points preceding data for given slice at given time</td>
</tr>
<tr>
<td>NBFRS</td>
<td></td>
<td>INPRT</td>
<td>number of suction-side boundary condition points preceding data for given slice at given time</td>
</tr>
<tr>
<td>NBLKSZ</td>
<td>TCO</td>
<td></td>
<td>size of data block in CHANL array for given slice</td>
</tr>
<tr>
<td>NCC</td>
<td>PLNUM</td>
<td></td>
<td>loop counter</td>
</tr>
<tr>
<td>NCHAR</td>
<td>NTTACT</td>
<td></td>
<td>number of characters in ALPH2 array</td>
</tr>
<tr>
<td>NCOOL</td>
<td>PLOTMF</td>
<td></td>
<td>coolant node number</td>
</tr>
<tr>
<td>NEND</td>
<td>HCOOL</td>
<td></td>
<td>end of region that uses leading-edge impingement correlation</td>
</tr>
<tr>
<td>NFC</td>
<td>FLOWS</td>
<td></td>
<td>number of station containing film-cooling holes</td>
</tr>
<tr>
<td>NFCSUP(80)</td>
<td>FLMCOL</td>
<td></td>
<td>array identifying node supplying film cooling to each downstream node</td>
</tr>
<tr>
<td>NFLUID(200)</td>
<td></td>
<td>INPRT</td>
<td>array of coolant-channel node numbers for each station</td>
</tr>
<tr>
<td>NFWD</td>
<td>TCO</td>
<td></td>
<td>number of stations in forward region</td>
</tr>
<tr>
<td>NG</td>
<td>GAAS</td>
<td></td>
<td>number of temperature entries in gas property table, GS</td>
</tr>
<tr>
<td>NGEOS</td>
<td>GETIN</td>
<td></td>
<td>number of NAMELISTs /GEO/ to be read in</td>
</tr>
<tr>
<td>NINTRV</td>
<td>PLOTMF</td>
<td></td>
<td>number of temperature intervals in summary plots</td>
</tr>
<tr>
<td>NIT</td>
<td>TCOEF</td>
<td></td>
<td>counter of number of overall coolant-flow iterations</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>NL</td>
<td>INPRT</td>
<td>PLOTMF</td>
<td>output line counter</td>
</tr>
<tr>
<td>NLBLS</td>
<td>PLOTMF</td>
<td>NTTACT</td>
<td>number of points at which symbols are to be plotted</td>
</tr>
<tr>
<td>NMM</td>
<td>PLOTMF</td>
<td>WROUT</td>
<td>midwall node number</td>
</tr>
<tr>
<td>NMW</td>
<td>NTTACT</td>
<td>GETIN</td>
<td>outer-surface node number</td>
</tr>
<tr>
<td>NODM</td>
<td>GETIN</td>
<td>FLOWS</td>
<td>midwall node</td>
</tr>
<tr>
<td>NODOUT</td>
<td>FLOWS</td>
<td>INPRT</td>
<td>outer-surface node</td>
</tr>
<tr>
<td>NODSF</td>
<td>INPRT</td>
<td>PREP</td>
<td>number of nodes in forward region</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FLSPLT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PARRAY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TARRAY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCOEF</td>
<td></td>
</tr>
<tr>
<td>NODST</td>
<td>GETIN</td>
<td>INPRT</td>
<td>total number of nodes for given slice</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PREP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PARRAY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TARRAY</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCOEF</td>
<td></td>
</tr>
<tr>
<td>NODSTM</td>
<td>INPRT</td>
<td>NTTACT</td>
<td>total number of nodes minus 4</td>
</tr>
<tr>
<td>NOS</td>
<td>FLOWS</td>
<td>INPRT</td>
<td>outer-surface node number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WROUT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLOTMF</td>
<td></td>
</tr>
<tr>
<td>NPRCP</td>
<td>INPRT</td>
<td></td>
<td>number of points in each pressure-side boundary condition array for times preceding current time</td>
</tr>
<tr>
<td>NPRCS</td>
<td>INPRT</td>
<td></td>
<td>number of points in each suction-side boundary condition array for times preceding current time</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>NPRTP</td>
<td>INPRT</td>
<td></td>
<td>number of points in each pressure-side boundary condition array per time step</td>
</tr>
<tr>
<td>NPRTS</td>
<td>INPRT</td>
<td></td>
<td>number of points in each suction-side boundary condition array per time step</td>
</tr>
<tr>
<td>NPTS</td>
<td>PLOTMF</td>
<td></td>
<td>number of points on given plot</td>
</tr>
<tr>
<td>NROW</td>
<td>GAUSS</td>
<td></td>
<td>number of matrix row to be displayed by debug output</td>
</tr>
<tr>
<td>NSAWE</td>
<td>TCOEF</td>
<td></td>
<td>coolant node number just upstream of exit, location of TSAVE</td>
</tr>
<tr>
<td>NSLICE</td>
<td>TCO</td>
<td></td>
<td>current slice number</td>
</tr>
<tr>
<td>NSTA</td>
<td>TCO</td>
<td></td>
<td>number of stations per slice</td>
</tr>
<tr>
<td>NSTAPS</td>
<td>PLOTMF</td>
<td></td>
<td>number of stations on each side of blade</td>
</tr>
<tr>
<td>NSTNS</td>
<td>PLNUM</td>
<td></td>
<td>number of spanwise stations per slice in coolant plenum</td>
</tr>
<tr>
<td>NTBC</td>
<td>GETIN</td>
<td></td>
<td>number of entries in input BCTIME array</td>
</tr>
<tr>
<td>NTIMES</td>
<td>INPRT</td>
<td></td>
<td>number of entries in BCTIME array</td>
</tr>
<tr>
<td>NTTG</td>
<td>PREP</td>
<td></td>
<td>time step number</td>
</tr>
<tr>
<td>NTTACT</td>
<td>NTTACT</td>
<td></td>
<td>number of time steps in transient</td>
</tr>
<tr>
<td>NTYM</td>
<td>NTTACT</td>
<td></td>
<td>counter to force at least four attempts at a good flow split</td>
</tr>
<tr>
<td>NUMS</td>
<td>FLSPLT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P(2, 15, 80)</td>
<td>TCO</td>
<td></td>
<td>pressure at each node, for two consecutive time steps, lbf/in$^2$</td>
</tr>
<tr>
<td>PAVG</td>
<td>FLOWS</td>
<td></td>
<td>coolant-channel static pressure, lbf/in$^2$, used in calculating impingement hole flow rates</td>
</tr>
<tr>
<td>PBAR</td>
<td>FLOWS</td>
<td></td>
<td>pressure used in calculating square of coolant-channel Mach number</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>------------------</td>
<td>--------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PCNVRG</td>
<td></td>
<td>TCOEF</td>
<td>pressure-difference convergence criterion</td>
</tr>
<tr>
<td>PD</td>
<td></td>
<td>FLOWS</td>
<td>Prandtl number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GASTBL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HCFRCD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HCOOL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HCPINS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLNUM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TARRAY</td>
<td></td>
</tr>
<tr>
<td>PDTOG</td>
<td></td>
<td>HCOOL</td>
<td>Prandtl number based on coolant-supply temperature</td>
</tr>
<tr>
<td>PEX(400)</td>
<td></td>
<td>BOUND</td>
<td>input array containing tables of static pressure at trailing-edge coolant</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>exhaust, lbf/in²</td>
</tr>
<tr>
<td>PEXC</td>
<td></td>
<td>INPRT</td>
<td>static pressure at trailing-edge coolant exhaust in SI units, kPa, for given</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>slice</td>
</tr>
<tr>
<td>PEXIT(15)</td>
<td></td>
<td>TCO</td>
<td>static pressures at trailing-edge coolant exhaust for each slice at given</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>time, lbf/in²</td>
</tr>
<tr>
<td>PEXOLD(15)</td>
<td></td>
<td>TCOEF</td>
<td>saved value of exhaust static pressure, lbf/in², used in setting initial</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>guess of pressure distribution for subsequent time step</td>
</tr>
<tr>
<td>PEXTT</td>
<td></td>
<td>PLNUM</td>
<td>total pressure at exit of coolant plenum for given slice, lbf/in²</td>
</tr>
<tr>
<td>PG(80)</td>
<td></td>
<td>FLMCOL</td>
<td>array containing hot-gas-side static pressure, lbf/in², at each station</td>
</tr>
<tr>
<td>PI</td>
<td></td>
<td>HCOOL</td>
<td>constant, 3.14159</td>
</tr>
<tr>
<td>PIM</td>
<td></td>
<td>TCO</td>
<td>impingement-supply pressure, lbf/in²</td>
</tr>
<tr>
<td>PIMOLD(15)</td>
<td></td>
<td>TCOEF</td>
<td>saved value of impingement-supply pressure, lbf/in², for each slice</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>PIN(15)</td>
<td>RADL</td>
<td></td>
<td>coolant total pressure, lbf/in², at entrance to each slice</td>
</tr>
<tr>
<td>PINS</td>
<td>HCPINS</td>
<td>TARRAY</td>
<td>number of pin fins at given station</td>
</tr>
<tr>
<td>PIVOT</td>
<td>GAUSS</td>
<td></td>
<td>main diagonal term of row of matrix being solved</td>
</tr>
<tr>
<td>PLEGN(5)</td>
<td>PLOTMF</td>
<td></td>
<td>alphanemic array to label pressure-data plots</td>
</tr>
<tr>
<td>PLTYME(2)</td>
<td>PLOTMF</td>
<td></td>
<td>alphanemic variable to print transient time on each plot</td>
</tr>
<tr>
<td>POLD(15,80)</td>
<td>TCOEF</td>
<td></td>
<td>saved values of coolant-channel pressure, lbf/in², from previous iteration</td>
</tr>
<tr>
<td>PP</td>
<td>PLNUM</td>
<td></td>
<td>intermediate pressure term</td>
</tr>
<tr>
<td>PPLEN</td>
<td>NTTACT</td>
<td></td>
<td>impingement-supply pressure, lbf/in²</td>
</tr>
<tr>
<td>PROD</td>
<td>HCOOL</td>
<td></td>
<td>intermediate calculation result</td>
</tr>
<tr>
<td>PROPS</td>
<td>GETIN</td>
<td></td>
<td>NAMELIST name</td>
</tr>
<tr>
<td>PSAV(5)</td>
<td>TCOEF</td>
<td></td>
<td>array to save last four values of pressure at flow-split station, used to check convergence</td>
</tr>
<tr>
<td>PTEMP</td>
<td>PLNUM</td>
<td></td>
<td>intermediate pressure term</td>
</tr>
<tr>
<td>PTEXIT</td>
<td>PLNUM</td>
<td></td>
<td>coolant-plenum total pressure, lbf/in²: entrance value going into subroutine, exit value coming out</td>
</tr>
<tr>
<td>PTIN</td>
<td>NTTACT</td>
<td></td>
<td>coolant-supply pressure for a given time, lbf/in²</td>
</tr>
<tr>
<td>PTIO(50)</td>
<td>BOUND</td>
<td></td>
<td>input array of coolant-supply pressure, kPa (lbf/in²), as function of time, sec</td>
</tr>
<tr>
<td>PTIOC</td>
<td>INPRT</td>
<td></td>
<td>initial coolant-supply pressure in SI units, kPa</td>
</tr>
<tr>
<td>PTNOLD</td>
<td>NTTACT</td>
<td></td>
<td>previous value of PTIN, lbf/in²</td>
</tr>
<tr>
<td>PT1</td>
<td>PLNUM</td>
<td></td>
<td>calculated coolant-plenum inlet total pressure, lbf/in²</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------------</td>
<td>--------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PUMP(80)</td>
<td>TCO</td>
<td></td>
<td>term to account for coolant pumping due to wheel rotation</td>
</tr>
<tr>
<td>PUMTRM</td>
<td>PARRAY</td>
<td></td>
<td>term to account for coolant pumping due to wheel rotation</td>
</tr>
<tr>
<td>PXX</td>
<td>PLNUM</td>
<td></td>
<td>average static pressure, lbf/in(^2), in coolant plenum for given slice</td>
</tr>
<tr>
<td>QG(80)</td>
<td>TCO</td>
<td></td>
<td>hot-gas heat flux to blade at each station, Btu/hr \cdot ft(^2)</td>
</tr>
<tr>
<td>QHUB(80)</td>
<td>BOUND</td>
<td></td>
<td>heat flux conducted to blade wall from hub platform at each station,</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Btu/hr \cdot ft(^2)</td>
</tr>
<tr>
<td>QHUBIN(400)</td>
<td>BOUND</td>
<td></td>
<td>input table of hub heat flux at each station as function of time, W/m(^2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Btu/hr \cdot ft(^2))</td>
</tr>
<tr>
<td>QSNK(80)</td>
<td>TCO</td>
<td></td>
<td>term to account for heat removal from wall by film-cooling flow through wall</td>
</tr>
<tr>
<td>QTIP(80)</td>
<td>BOUND</td>
<td></td>
<td>heat flux from blade wall at tip for each station, Btu/hr \cdot ft(^2)</td>
</tr>
<tr>
<td>QTIPIN(400)</td>
<td>BOUND</td>
<td></td>
<td>input table of tip heat flux at each station as function of time, W/m(^2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Btu/hr \cdot ft(^2))</td>
</tr>
<tr>
<td>R</td>
<td>TCO</td>
<td></td>
<td>gas constant; value for air is build in, 53.35 ft-lbf/lbm \cdot (^{0})R</td>
</tr>
<tr>
<td>RATIO</td>
<td>THRCON</td>
<td></td>
<td>interpolating fraction</td>
</tr>
<tr>
<td>RCHRD</td>
<td>TARRAY</td>
<td></td>
<td>dimensionless ratio of time increment to chordwise length increment squared at each station</td>
</tr>
<tr>
<td>RCHRDM</td>
<td>TARRAY</td>
<td></td>
<td>maximum value of RCHRD for a given slice</td>
</tr>
<tr>
<td>RCVRY</td>
<td>TARRAY</td>
<td></td>
<td>recovery factor</td>
</tr>
<tr>
<td>RE(80)</td>
<td>PRPS</td>
<td></td>
<td>coolant-channel Reynolds number at each station</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------------</td>
<td>------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>REFC(80)</td>
<td>FLMCOL</td>
<td></td>
<td>film-cooling flow Reynolds number at each station</td>
</tr>
<tr>
<td>REJ(80)</td>
<td>HCOOL</td>
<td></td>
<td>impingement-jet Reynolds number at each station</td>
</tr>
<tr>
<td>REJ0VR(80)</td>
<td>HCOOL</td>
<td></td>
<td>array to save values of impingement-jet Reynolds number that are out of range of correlation</td>
</tr>
<tr>
<td>REY</td>
<td>PLNUM</td>
<td></td>
<td>coolant-plenum Reynolds number based on hydraulic diameter</td>
</tr>
<tr>
<td>RHOBAR</td>
<td>TARRAY</td>
<td></td>
<td>mean density in coolant channel, lbm/in^3</td>
</tr>
<tr>
<td>RHOC</td>
<td>TRNSNT</td>
<td></td>
<td>input density of outer coating, kg/m^3 (lbm/ft^3)</td>
</tr>
<tr>
<td>RHOM</td>
<td>TRNSNT</td>
<td></td>
<td>input density of wall metal, kg/m^3 (lbm/ft^3)</td>
</tr>
<tr>
<td>RHOVG(400)</td>
<td>BOUND</td>
<td></td>
<td>input table of hot-gas-side, free-stream mass velocity at each station as function of time, kg/m^2·sec (lbm/ft^2·sec)</td>
</tr>
<tr>
<td>RHOVGA(80)</td>
<td>FLMCOL</td>
<td></td>
<td>hot-gas-side, free-stream mass velocity at each station for given slice, lbm/ft^2·sec</td>
</tr>
<tr>
<td>RI</td>
<td>GETIN</td>
<td></td>
<td>input value of radial location of coolant-plenum inlet for given slice, cm (in.)</td>
</tr>
<tr>
<td>RIN(15)</td>
<td>RADL</td>
<td></td>
<td>table of RI values for each slice, in.</td>
</tr>
<tr>
<td>RO</td>
<td>GETIN</td>
<td></td>
<td>input value of radial location of coolant-plenum exit for given slice, cm (in.)</td>
</tr>
<tr>
<td>ROINVC</td>
<td>HCOOL</td>
<td></td>
<td>intermediate term in impingement correlation, ft^3/lbm</td>
</tr>
<tr>
<td>ROINVJ</td>
<td>HCOOL</td>
<td></td>
<td>intermediate term in impingement correlation, ft^3/lbm</td>
</tr>
<tr>
<td>ROOT</td>
<td>PARRAY</td>
<td></td>
<td>intermediate term in pressure calculations</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>ROUT(15)</td>
<td>RADL</td>
<td></td>
<td>table of RO values for each slice, in.</td>
</tr>
<tr>
<td>RR(80)</td>
<td>TCO</td>
<td></td>
<td>mean radial location of each station for a given slice</td>
</tr>
<tr>
<td>RRP</td>
<td></td>
<td>PLNUM</td>
<td>radial location, in.</td>
</tr>
<tr>
<td>RTEMP</td>
<td></td>
<td>PLNUM</td>
<td>radial location, in.</td>
</tr>
<tr>
<td>RTNARR(2)</td>
<td>PLOTMF</td>
<td></td>
<td>array containing maximum and minimum values of plot variables</td>
</tr>
<tr>
<td>RTRNV</td>
<td>TARRAY</td>
<td></td>
<td>dimensionless ratio of time increment to through-the-wall length increment squared</td>
</tr>
<tr>
<td>RTRNVM</td>
<td>TARRAY</td>
<td></td>
<td>maximum value of RTRNV for given slice</td>
</tr>
<tr>
<td>S(15)</td>
<td>TCO</td>
<td></td>
<td>span of each slice, in.</td>
</tr>
<tr>
<td>SEGMTS</td>
<td>PLNUM</td>
<td></td>
<td>number of segments in coolant plenum for given slice</td>
</tr>
<tr>
<td>SIGB</td>
<td>PLNUM</td>
<td></td>
<td>dummy variable used in coolant-plenum calculations</td>
</tr>
<tr>
<td>SIGC'</td>
<td>PLNUM</td>
<td></td>
<td>dummy variable used in coolant-plenum calculations</td>
</tr>
<tr>
<td>SIGMA (20)</td>
<td>PLNUM</td>
<td></td>
<td>coolant velocity distribution in coolant plenum</td>
</tr>
<tr>
<td>SLEGN(5)</td>
<td>PLOTMF</td>
<td></td>
<td>alphameric array to label suction-side plots</td>
</tr>
<tr>
<td>SLP</td>
<td></td>
<td>HCPINS</td>
<td>mean pin-fin length at given station, in.</td>
</tr>
<tr>
<td>SP(80)</td>
<td>PRPS</td>
<td></td>
<td>pin-fin spacing at each station, in.</td>
</tr>
<tr>
<td>SPAN</td>
<td>TCO</td>
<td></td>
<td>radial span of given slice, in.</td>
</tr>
<tr>
<td>SPANC</td>
<td></td>
<td>INPRT</td>
<td>radial span of given slice in SI units, cm</td>
</tr>
<tr>
<td>SPHTC</td>
<td>TRNSNT</td>
<td></td>
<td>input specific heat of outer coating, ( J/kg \cdot K ) (Btu/lbm \cdot ^{\circ}F)</td>
</tr>
<tr>
<td>SPHTM</td>
<td>TRNSNT</td>
<td></td>
<td>input specific heat of wall metal, ( J/kg \cdot K ) (Btu/lbm \cdot ^{\circ}F)</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>------------</td>
<td>--------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ST</td>
<td>HCOOL</td>
<td>ST</td>
<td>Stanton number calculated from user-supplied impingement correlation</td>
</tr>
<tr>
<td>STANMX</td>
<td>HCOOL</td>
<td>STANMX</td>
<td>Stanton number calculated from leading-edge impingement correlation</td>
</tr>
<tr>
<td>SV(3)</td>
<td>PLNUM</td>
<td></td>
<td>array to save values of SIGC</td>
</tr>
<tr>
<td>SYMBOL(10)</td>
<td>PLOTMF</td>
<td></td>
<td>array of integers to be used as plot symbols</td>
</tr>
<tr>
<td>SYMBOL</td>
<td>PLOTMF</td>
<td></td>
<td>particular entry from SYMBOL array</td>
</tr>
<tr>
<td>T[2,15,400]</td>
<td>TCO</td>
<td></td>
<td>calculated temperature at each node for each slice for two time steps, °F</td>
</tr>
<tr>
<td>TABOVE</td>
<td>TARRAY</td>
<td></td>
<td>midwall temperature at given station in slice above current slice, °F</td>
</tr>
<tr>
<td>TAU(400)</td>
<td>TCO</td>
<td></td>
<td>array of thickness values, in.</td>
</tr>
<tr>
<td>TBAR</td>
<td>FLOWS</td>
<td></td>
<td>coolant temperature, °R, used to calculate Mach number</td>
</tr>
<tr>
<td>TBAR</td>
<td>HCPINS</td>
<td></td>
<td>ratio of temperature drops in pin fins, pressure-side wall temperature minus mid-coolant-channel temperature to suction-side wall temperature minus mid-coolant-channel temperature</td>
</tr>
<tr>
<td>TBAR</td>
<td>WROUT</td>
<td></td>
<td>mean outer-surface temperature for given slice, °F</td>
</tr>
<tr>
<td>TBARMD</td>
<td>WROUT</td>
<td></td>
<td>mean midwall temperature, °F</td>
</tr>
<tr>
<td>TBELOW</td>
<td>TARRAY</td>
<td></td>
<td>midwall temperature at given station in slice below current slice, °F</td>
</tr>
<tr>
<td>TBULK</td>
<td>WROUT</td>
<td></td>
<td>overall blade bulk-metal temperature, °F</td>
</tr>
<tr>
<td>TC</td>
<td>THRCON</td>
<td></td>
<td>mean temperature of blade cladding material, °F</td>
</tr>
<tr>
<td>TCIN</td>
<td>NTTACT</td>
<td></td>
<td>coolant temperature, °F</td>
</tr>
<tr>
<td>TCOF(400,30)</td>
<td>MATRIX</td>
<td></td>
<td>array of coefficients to be solved for temperature or pressure</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TDHF</td>
<td></td>
<td>GETIN</td>
<td>function of film-cooling-hole size and spacing</td>
</tr>
<tr>
<td>TDHJ</td>
<td></td>
<td>GETIN</td>
<td>hydraulic diameter of impingement hole</td>
</tr>
<tr>
<td>TDLX(5)</td>
<td></td>
<td>GETIN</td>
<td>array containing lengths from nodes at upstream station to corresponding nodes at current station</td>
</tr>
<tr>
<td>TDP</td>
<td></td>
<td>GETIN</td>
<td>input pin-fin diameter, cm (in.)</td>
</tr>
<tr>
<td>TEM</td>
<td></td>
<td>INPRT</td>
<td>coolant-inlet absolute temperature, $K \left( ^\circ R \right)$</td>
</tr>
<tr>
<td>TEPS</td>
<td></td>
<td>TRNSNT</td>
<td>factor used to define time mean properties</td>
</tr>
<tr>
<td>TERM</td>
<td></td>
<td>FLSPLT</td>
<td>adjustment to flow-split parameter</td>
</tr>
<tr>
<td>TG(80)</td>
<td></td>
<td>TCO</td>
<td>hot-gas reference temperature at each station for given slice, $^\circ R$</td>
</tr>
<tr>
<td>THETA1</td>
<td></td>
<td>TARRAY</td>
<td>intermediate terms in nodal energy equations</td>
</tr>
<tr>
<td>THETA2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THETA3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THETA4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THETA5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THETA6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THETA8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THETA9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>THK(3)</td>
<td></td>
<td>GETIN</td>
<td>input array of thickness values, cm (in.)</td>
</tr>
<tr>
<td>THUB(80)</td>
<td></td>
<td>BOUND</td>
<td>specified temperature at each blade hub station for given time, $^\circ F$</td>
</tr>
<tr>
<td>THUBIN(400)</td>
<td></td>
<td>BOUND</td>
<td>input table of hub temperatures at each station as function of time, $K \left( ^\circ F \right)$</td>
</tr>
<tr>
<td>TIKLE(30)</td>
<td></td>
<td>GETIN</td>
<td>blank alphameric array used to initialize title array</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------------</td>
<td>--------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TIN(15)</td>
<td>RADL</td>
<td></td>
<td>inlet total temperature in coolant plenum for each slice, °F</td>
</tr>
<tr>
<td>TIP1</td>
<td>TARRAY</td>
<td></td>
<td>term in conduction equation to account for specified tip temperature</td>
</tr>
<tr>
<td>TIP3</td>
<td>TARRAY</td>
<td></td>
<td>term in conduction equation to account for specified tip heat flux</td>
</tr>
<tr>
<td>TITL</td>
<td>GETIN</td>
<td></td>
<td>NAMELIST name</td>
</tr>
<tr>
<td>TITLE(30)</td>
<td>SPECL</td>
<td></td>
<td>alphanumerical array of title information from input</td>
</tr>
<tr>
<td>TLABL1(21)</td>
<td>PLOTMF</td>
<td></td>
<td>alphanumerical arrays used to put input title on plots</td>
</tr>
<tr>
<td>TLABL2(9)</td>
<td>PLOTMF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMAXP</td>
<td>PLOTMF</td>
<td></td>
<td>maximum pressure-side temperature to be plotted</td>
</tr>
<tr>
<td>TMAXS</td>
<td>PLOTMF</td>
<td></td>
<td>maximum suction-side temperature to be plotted</td>
</tr>
<tr>
<td>TMFRAC</td>
<td>PREP</td>
<td></td>
<td>time-interpolating function</td>
</tr>
<tr>
<td>TMINP</td>
<td>PLOTMF</td>
<td></td>
<td>minimum pressure-side temperature to be plotted</td>
</tr>
<tr>
<td>TMINS</td>
<td>PLOTMF</td>
<td></td>
<td>minimum suction-side temperature to be plotted</td>
</tr>
<tr>
<td>TMP</td>
<td>GASTBL</td>
<td></td>
<td>temperature used to determine gas properties, °R</td>
</tr>
<tr>
<td>TMP1</td>
<td>GASTBL</td>
<td></td>
<td>temperature used to determine gas properties, °F</td>
</tr>
<tr>
<td>TOTSPN</td>
<td>WROUT</td>
<td></td>
<td>total span of blade, in.</td>
</tr>
<tr>
<td>TP</td>
<td>PLNUM</td>
<td></td>
<td>intermediate temperature variable</td>
</tr>
<tr>
<td>TPLEN</td>
<td>NTTACT</td>
<td></td>
<td>mean impingement-plenum static temperature, °F</td>
</tr>
<tr>
<td>TPM(500)</td>
<td>PLOTMF</td>
<td></td>
<td>table of pressure-side, midwall temperatures for summary plots, K (°F)</td>
</tr>
<tr>
<td>TPMAX</td>
<td>WROUT</td>
<td></td>
<td>maximum pressure-side, outer-surface temperature for given slice, K (°F)</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TMIN</td>
<td></td>
<td>WROUT</td>
<td>minimum pressure-side, outer-surface temperature for given slice, K°F</td>
</tr>
<tr>
<td>TPO(500)</td>
<td></td>
<td>PLOTMF</td>
<td>table of pressure-side, outer-surface temperatures for summary plots, K°F</td>
</tr>
<tr>
<td>TREEDGE</td>
<td></td>
<td>TARRAY</td>
<td>intermediate term in nodal energy equation in trailing-edge region</td>
</tr>
<tr>
<td>TREPS</td>
<td></td>
<td>PARRAY</td>
<td>same as TEPS</td>
</tr>
<tr>
<td>TRR</td>
<td></td>
<td>GETIN</td>
<td>input mean radial location of given station, cm (in.)</td>
</tr>
<tr>
<td>TRTRM</td>
<td></td>
<td>PARRAY</td>
<td>intermediate transient term in pressure equations</td>
</tr>
<tr>
<td>TRTRMC</td>
<td></td>
<td>TARRAY</td>
<td>intermediate transient term involving outer coating</td>
</tr>
<tr>
<td>TRTRMG</td>
<td></td>
<td>TARRAY</td>
<td>intermediate transient term involving coolant</td>
</tr>
<tr>
<td>TRTRM0</td>
<td></td>
<td>TARRAY</td>
<td>intermediate transient term for coolant channel at entrance to trailing edge</td>
</tr>
<tr>
<td>TSAVE</td>
<td></td>
<td>TCOEF</td>
<td>temperature in coolant channel, just upstream of exit</td>
</tr>
<tr>
<td>TSM(500)</td>
<td></td>
<td>PLOTMF</td>
<td>table of suction-side, midwall temperature for summary plots, K°F</td>
</tr>
<tr>
<td>TSMAX</td>
<td></td>
<td>WROUT</td>
<td>maximum suction-side, outer-surface temperature for given slice, K°F</td>
</tr>
<tr>
<td>TSMIN</td>
<td></td>
<td>WROUT</td>
<td>minimum suction-side, outer-surface temperature for given slice, K°F</td>
</tr>
<tr>
<td>TSO(500)</td>
<td></td>
<td>PLOTMF</td>
<td>table of suction-side, outer-surface temperatures for summary plots, K°F</td>
</tr>
<tr>
<td>TSP</td>
<td></td>
<td>GETIN</td>
<td>input pin-fin spacing, cm (in.)</td>
</tr>
<tr>
<td>TTEMP</td>
<td></td>
<td>PLNUM</td>
<td>variable to save temperature value, °R</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>TTEXIT</td>
<td>PLNUM</td>
<td></td>
<td>coolant total temperature in plenum, °F</td>
</tr>
<tr>
<td>TTIN</td>
<td>NTTACT</td>
<td></td>
<td>coolant-supply temperature, °F, at entrance to coolant plenum for given time</td>
</tr>
<tr>
<td>TTIO(50)</td>
<td>BOUND</td>
<td></td>
<td>input table of coolant-supply temperature, K (°F), as function of time</td>
</tr>
<tr>
<td>TTIP(80)</td>
<td>BOUND</td>
<td></td>
<td>table of blade tip temperature, °F, for given time</td>
</tr>
<tr>
<td>TTIPIN(400)</td>
<td>BOUND</td>
<td></td>
<td>input table of blade tip temperature, K (°F), as function of time</td>
</tr>
<tr>
<td>TTOTC(80)</td>
<td>TCOEF</td>
<td></td>
<td>coolant total temperature at each station for given slice, °F</td>
</tr>
<tr>
<td>TTX</td>
<td>PLNUM</td>
<td></td>
<td>coolant total temperature at inlet to coolant plenum for given slice, °R</td>
</tr>
<tr>
<td>TTYME</td>
<td>WROUT</td>
<td></td>
<td>current time in transient, sec</td>
</tr>
<tr>
<td>TT1(20)</td>
<td>PLNUM</td>
<td></td>
<td>total-temperature distribution, °F, in coolant plenum</td>
</tr>
<tr>
<td>TW</td>
<td>THRCON</td>
<td></td>
<td>midwall temperature for evaluating thermal conductivity</td>
</tr>
<tr>
<td>TXN</td>
<td>GETIN</td>
<td></td>
<td>input spanwise spacing of impingement jets at given station, cm (in.)</td>
</tr>
<tr>
<td>TXX</td>
<td>PLNUM</td>
<td></td>
<td>average static temperature in coolant plenum for given slice, °F</td>
</tr>
<tr>
<td>TYME</td>
<td>TRNSNT</td>
<td></td>
<td>time in transient calculations, sec</td>
</tr>
<tr>
<td>TYMMAX</td>
<td>TRNSNT</td>
<td></td>
<td>maximum time to which transient is carried, sec</td>
</tr>
<tr>
<td>T0G</td>
<td>TCO</td>
<td></td>
<td>impingement-jet temperature for given slice, °R</td>
</tr>
<tr>
<td>T1</td>
<td>PLNUM</td>
<td></td>
<td>inlet total temperature in coolant plenum for given slice, °R</td>
</tr>
<tr>
<td>UA(2)</td>
<td>INPRT</td>
<td></td>
<td>alphanumeric array containing area units</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UL (2)</td>
<td></td>
<td>INPRT</td>
<td>alphameric array containing length units</td>
</tr>
<tr>
<td>V</td>
<td></td>
<td>TCOEF</td>
<td>factor to accelerate convergence of pressure iterations</td>
</tr>
<tr>
<td>VARIB (15)</td>
<td></td>
<td>PLOTMF</td>
<td>alphameric array containing some plot labels</td>
</tr>
<tr>
<td>VARS (12)</td>
<td></td>
<td>PLOTMF</td>
<td>array containing plotting controls</td>
</tr>
<tr>
<td>VDP</td>
<td></td>
<td>HCPINS</td>
<td>pin-fin diameter at given station, in.</td>
</tr>
<tr>
<td>VOLBAR</td>
<td></td>
<td>TARRAY</td>
<td>coolant-channel volume element at entrance to trailing-edge region, in^3</td>
</tr>
<tr>
<td>VSP</td>
<td></td>
<td>HCPINS</td>
<td>pin-fin spacing at given station, in.</td>
</tr>
<tr>
<td>VI</td>
<td></td>
<td>PLNUM</td>
<td>intermediate term in coolant-plenum calculations</td>
</tr>
<tr>
<td>W(15)</td>
<td>RADL</td>
<td></td>
<td>coolant flow-rate at entrance to each slice, lbm/hr</td>
</tr>
<tr>
<td>WC</td>
<td></td>
<td>HCOOL</td>
<td>absolute value of coolant flow-rate, lbm/sec</td>
</tr>
<tr>
<td>WCHK (80)</td>
<td></td>
<td>CHKHOL</td>
<td>alphameric variable used to indicate choked flow in impingement jets</td>
</tr>
<tr>
<td>WCHKDM</td>
<td></td>
<td>CHKHOL</td>
<td>alphameric variable used to indicate choked flow in holes dumping coolant</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>to trailing-edge region</td>
</tr>
<tr>
<td>WCHOKE</td>
<td></td>
<td>PLNUM</td>
<td>coolant flow-rate at entrance to given slice, lbm/hr</td>
</tr>
<tr>
<td>WCR</td>
<td></td>
<td>FLOWS</td>
<td>critical flow-rate, lbm/sec</td>
</tr>
<tr>
<td>WCROS (2, 15, 80)</td>
<td>TCO</td>
<td></td>
<td>coolant-channel crossflow rate at each station for each slice for two time</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>steps, lbm/sec</td>
</tr>
<tr>
<td>WDUMP</td>
<td></td>
<td>TCO</td>
<td>rate of coolant flow being dumped directly from plenum to trailing-edge</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>region, lbm/sec</td>
</tr>
<tr>
<td>WFC (80)</td>
<td></td>
<td>TCO</td>
<td>film-cooling flow rate at each station for given slice, lbm/sec</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>--------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>WFCDUM</td>
<td>FLOWS</td>
<td></td>
<td>intermediate variable in film-cooling flow rate calculation</td>
</tr>
<tr>
<td>WFCDUM</td>
<td>PARRAY</td>
<td></td>
<td>total film-cooling flow rate at given station, lbm/sec</td>
</tr>
<tr>
<td>WIM</td>
<td>TCO</td>
<td></td>
<td>total impingement-jet flow rate for given slice, lbm/sec</td>
</tr>
<tr>
<td>WJ(15,80)</td>
<td>TCO</td>
<td></td>
<td>impingement-jet flow rate for each station for each slice, lbm/sec</td>
</tr>
<tr>
<td>WPLEN</td>
<td>BOUND</td>
<td></td>
<td>input initial guess at total coolant flow, kg/hr (lbm/hr)</td>
</tr>
<tr>
<td>WPLENC</td>
<td>INPRT</td>
<td></td>
<td>total coolant flow in SI units, kg/hr</td>
</tr>
<tr>
<td>WPLENO</td>
<td>NTTACT</td>
<td></td>
<td>variable to save previous estimate of total coolant flow, lbm/hr</td>
</tr>
<tr>
<td>WS</td>
<td>RADL</td>
<td></td>
<td>rotor speed at given time, rpm</td>
</tr>
<tr>
<td>WSVST(50)</td>
<td>BOUND</td>
<td></td>
<td>input table of rotor speed, rpm, versus time, sec</td>
</tr>
<tr>
<td>WUSED</td>
<td>NTTACT</td>
<td></td>
<td>cumulative amount of coolant used, up to current slice, lbm/hr</td>
</tr>
<tr>
<td>WXCP</td>
<td>TARRAY</td>
<td></td>
<td>coolant flow-rate times specific-heat term</td>
</tr>
<tr>
<td>WXX</td>
<td>PLNUM</td>
<td></td>
<td>coolant flow-rate at entrance to given slice, lbm/hr</td>
</tr>
</tbody>
</table>
| X(80)        | TCOEF  |            | coolant-channel node locations, in., used to set initial pressure distribu-
<p>|              |        |            | tion                                                                     |
| XBAR         | FLOWS  |            | term in film-cooling effectiveness correlation                             |
| XCC          | INPRT  |            | coolant-channel distance from station 1 to given station, cm (in.)        |
| XDUM         | FLOWS  |            | dummy variable used to save location of film-cooling hole                 |</p>
<table>
<thead>
<tr>
<th>Variable</th>
<th>Common</th>
<th>Subroutine</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>XFC(80)</td>
<td>FLMCOL</td>
<td></td>
<td>distance from station with film-cooling holes to downstream stations without film-cooling holes, in.</td>
</tr>
<tr>
<td>XIS</td>
<td>INPRT</td>
<td></td>
<td>inside-wall surface distance from station 1 to given station</td>
</tr>
<tr>
<td>XJN</td>
<td>INPRT</td>
<td></td>
<td>distance along junction of cladding and wall metal from station 1 to given station, cm (in.)</td>
</tr>
<tr>
<td>XK(4)</td>
<td>PLNUM</td>
<td></td>
<td>pressure change in coolant plenum</td>
</tr>
<tr>
<td>XL</td>
<td>HCOOL</td>
<td></td>
<td>length of inner surface used in leading-edge impingement correlation, in.</td>
</tr>
<tr>
<td>XL(4)</td>
<td>PLNUM</td>
<td></td>
<td>temperature change in coolant plenum</td>
</tr>
<tr>
<td>XLABL(29)</td>
<td>PLOTMF</td>
<td></td>
<td>alphameric array of plot labels</td>
</tr>
<tr>
<td>XLABL2(15)</td>
<td>PLOTMF</td>
<td></td>
<td>alphameric array of plot labels</td>
</tr>
<tr>
<td>XLBL(20)</td>
<td>PLOTMF</td>
<td></td>
<td>array of x-coordinates to be plotted as slice numbers</td>
</tr>
<tr>
<td>XMM</td>
<td>INPRT</td>
<td></td>
<td>midwall distance from station 1 to given station, cm (in.)</td>
</tr>
<tr>
<td>XMU</td>
<td>FLOWS</td>
<td>GASTBL</td>
<td>viscosity, lbm/ft. hr</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HCFRCD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HCOOL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>HCPINS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PLNUM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TARRAY</td>
<td></td>
</tr>
<tr>
<td>XMUC(80)</td>
<td>FLMCOL</td>
<td></td>
<td>coolant viscosity, lbm/ft. hr, at each station, evaluated at mean temperature between inner-wall surface and bulk coolant temperatures</td>
</tr>
<tr>
<td>XMUM</td>
<td>FLOWS</td>
<td></td>
<td>hot-gas viscosity, lbm/ft. hr</td>
</tr>
<tr>
<td>XMUTOG</td>
<td>HCOOL</td>
<td></td>
<td>coolant viscosity based on coolant-supply temperature, lbm/ft. hr</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>XN(80)</td>
<td>TCO</td>
<td></td>
<td>spanwise spacing of impingement holes at each station, in.</td>
</tr>
<tr>
<td>XNN</td>
<td>PLNUM</td>
<td></td>
<td>factor for increasing number of stations in coolant plenum</td>
</tr>
<tr>
<td>XOD</td>
<td>PREP</td>
<td></td>
<td>ratio of impingement-hole spacing to hydraulic diameter, at given station</td>
</tr>
<tr>
<td>XOS</td>
<td>INPRT</td>
<td></td>
<td>wall outer-surface distance from station 1 to given station, cm (in.)</td>
</tr>
<tr>
<td>XOVERD</td>
<td>HCOOL</td>
<td></td>
<td>ratio of impingement-hole spacing to hydraulic diameter, at given station</td>
</tr>
<tr>
<td>XOVRDL</td>
<td>HCPINS</td>
<td></td>
<td>location of zero temperature gradient in pin fins</td>
</tr>
<tr>
<td>XP</td>
<td>HCOOL</td>
<td></td>
<td>length of pressure-side inner-wall surface in leading-edge impingement region</td>
</tr>
<tr>
<td>XP(80)</td>
<td>PLOTMF</td>
<td></td>
<td>pressure-side, dimensionless distance along midwall plane from station 1 to each station</td>
</tr>
<tr>
<td>XP</td>
<td>PREP</td>
<td></td>
<td>distance of given pressure-side station from station 1, in.</td>
</tr>
<tr>
<td>XPF</td>
<td>PREP</td>
<td></td>
<td>interpolating fraction in BCXP table</td>
</tr>
<tr>
<td>XPL</td>
<td>PLOTMF</td>
<td></td>
<td>overall length along pressure-side, midwall plane, cm (in.)</td>
</tr>
<tr>
<td>XS</td>
<td>HCOOL</td>
<td></td>
<td>length of suction-side, inner-wall surface in leading-edge impingement region</td>
</tr>
<tr>
<td>XS(80)</td>
<td>PLOTMF</td>
<td></td>
<td>suction-side, dimensionless distance along midwall plane from station 1 to each station</td>
</tr>
<tr>
<td>XS</td>
<td>PREP</td>
<td></td>
<td>distance of given suction-side station from station 1, in.</td>
</tr>
<tr>
<td>XSF</td>
<td>PREP</td>
<td></td>
<td>interpolating function in BCXS table</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>XSL</td>
<td>PLOTMF</td>
<td></td>
<td>overall length along suction-side, mid-wall plane, cm (in.)</td>
</tr>
<tr>
<td>XTEST</td>
<td>PLNUM</td>
<td></td>
<td>convergence test variable</td>
</tr>
<tr>
<td>XTOT</td>
<td>WROUT</td>
<td></td>
<td>overall outer-surface length around blade, in.</td>
</tr>
<tr>
<td>XTOTMD</td>
<td>WROUT</td>
<td></td>
<td>overall midwall length around blade, in.</td>
</tr>
<tr>
<td>XXN</td>
<td>PLNUM</td>
<td></td>
<td>number of spanwise stations per slice in coolant plenum</td>
</tr>
<tr>
<td>Y(320)</td>
<td>PLOTMF</td>
<td></td>
<td>array containing temperature values to be plotted</td>
</tr>
<tr>
<td>YCNVUU</td>
<td>TARRAY</td>
<td></td>
<td>indicates forced-convection heat transfer at last forward-region station on pressure side</td>
</tr>
<tr>
<td>YCONV</td>
<td>TARRAY</td>
<td></td>
<td>indicates forced-convection heat transfer at given station</td>
</tr>
<tr>
<td>YCONVU</td>
<td>TARRAY</td>
<td></td>
<td>indicates forced-convection heat transfer at station immediately upstream of given station</td>
</tr>
<tr>
<td>YFINS</td>
<td>TARRAY</td>
<td></td>
<td>indicates pin-fin heat transfer at given station</td>
</tr>
<tr>
<td>YFINSU</td>
<td>TARRAY</td>
<td></td>
<td>indicates pin-fin heat transfer at station immediately upstream of given station</td>
</tr>
<tr>
<td>YFNSUU</td>
<td>TARRAY</td>
<td></td>
<td>indicates pin-fin heat transfer at last forward-region station on pressure side</td>
</tr>
<tr>
<td>YIMP</td>
<td>TARRAY</td>
<td></td>
<td>indicates impingement heat transfer at given station</td>
</tr>
<tr>
<td>YIMPU</td>
<td>TARRAY</td>
<td></td>
<td>indicates impingement heat transfer at station immediately upstream of given station</td>
</tr>
<tr>
<td>YIMPUU</td>
<td>TARRAY</td>
<td></td>
<td>indicates impingement heat transfer at last forward-region station on pressure side</td>
</tr>
<tr>
<td>Variable</td>
<td>Common</td>
<td>Subroutine</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>------------</td>
<td>-------------------------------------------------------------</td>
</tr>
<tr>
<td>YLABL(7)</td>
<td></td>
<td>PLOTMF</td>
<td>alphameric array for labeling plots</td>
</tr>
<tr>
<td>YLABL2(11)</td>
<td></td>
<td>PLOTMF</td>
<td>alphameric array for labeling plots</td>
</tr>
<tr>
<td>YLBL (20)</td>
<td></td>
<td>PLOTMF</td>
<td>array of coordinates of points to be plotted as slice numbers</td>
</tr>
<tr>
<td>YMAX</td>
<td></td>
<td>PLOTMF</td>
<td>maximum value of y-coordinates on plot</td>
</tr>
<tr>
<td>YMIN</td>
<td></td>
<td>PLOTMF</td>
<td>minimum value of y-coordinates on plot</td>
</tr>
<tr>
<td>YPLABL(10)</td>
<td></td>
<td>PLOTMF</td>
<td>alphameric array for labeling plots</td>
</tr>
<tr>
<td>YTEM(80)</td>
<td></td>
<td>PLOTMF</td>
<td>array to be plotted</td>
</tr>
<tr>
<td>ZED</td>
<td></td>
<td>PLNUM</td>
<td>coolant-plenum pressure-drop parameter for given slice</td>
</tr>
<tr>
<td>ZOVERD</td>
<td></td>
<td>HCOOL</td>
<td>ratio of coolant-channel width to impingement-hole hydraulic diameter</td>
</tr>
<tr>
<td>Z1(15)</td>
<td></td>
<td>PLNUM</td>
<td>coolant-plenum pressure-drop parameter for each slice</td>
</tr>
<tr>
<td>Z3</td>
<td></td>
<td>PLNUM</td>
<td>intermediate term involving coolant flow</td>
</tr>
<tr>
<td>Z4</td>
<td></td>
<td>PLNUM</td>
<td>intermediate term involving coolant flow</td>
</tr>
</tbody>
</table>
**PROGRAM LISTING**

C----SOURCE.NTACT---THIS IS THE MAIN PROGRAM. BLOCK DATA SUBPROGRAM
GASDAT MUST BE LOADED FIRST.

**TRANIENT THERMAL ANALYSIS OF A COOLED TURBINE BLADE**

TRANSIENT THERMAL ANALYSIS OF A COOLED TURBINE BLADE

COMMON /BOUND/ BCXS(400), BCXP(400), BCHGS(1000), BCHGP(1000)
Z
BCTGS(1000), BCPG(1000), BCGGS(1000), BCQGP(1000)
Z
QUBIN(400), QHUB(80), TIPIN(400), TIP(80)
Z
QUBIN(400), QHUB(80), TIPIN(400), TIP(80)
Z
WSVST(50), TTI0(50), PTO(50), WPLEN
Z
WSVST(50), AKCTBL(20), AKWTBL(20), NBCS, HBCP

COMMON /FLMCOL/ RHOVGA(80), FG(80), XFC(80), FLMEFF(80)
Z
QHUBIN(400), QHUB(80), TTI0(50), WPLEN
Z
QHUBIN(400), QHUB(80)
Z
TTIPIN(400), TTIP(80), WPLEN
Z
TTIPIN(400), TTIP(80)
Z
RHOVG(400), PEX(400)
Z
RHOVG(400)
Z
ACTIME(50), TTO(50), PTO(50)
Z
ACTIME(50)
Z

COMMON /FLMCOL/ RHOGA(80), PG(80), XPC(80), PLMEFF(80)
Z
XMUC(80), EMES(80), REPC(80), NFCSUP(80)
Z

COMMON /GAAS/ GS(200), NG
Z

COMMON /RADL/ APLN(15), DPLN(15), RIN(15), ROUT(15)
Z
PIN(15), TIN(15), W(15), WS
Z

COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000)
Z
IPLOT, 1D1, XD2, MD3, IADJIN, IWRITE
Z

COMMON /TCO/ ADUMP, BTA, CD, CP
Z
GAM, PIM, R, SFAN, TOG
Z
WDUMP, WIM, AKC(15,80), AKW(15,80)
Z
A(400), AJET(80), AM2(80), CMUM(80)
Z
DH(80), DHF(80), DHJ(80)
Z
DLX(400), FF(80), HC(80), HG(80)
Z
P(2,15,80), PEXIT(15), PUMP(80), QC(80)
Z
QSNK(80), BR(80), S(15), T(2,15,400)
Z
TG(80), TAU(400), WFC(80)
Z
WJ(15,80), WCROS(2,15,80), XSS(80)
Z
ICOB, IFILM, IHUB, ITIP
Z
ISBLOK, ISLICE, NLKSB, NSLICE
Z
NFWD, NSTA, IHC(80)
Z

COMMON /TRANSNT/ RHOC, RHOM, SPHTC, SPHTM
Z
DLTYME, TYME, TEPS, TMAX
Z

COMMON /UNITS/ CINCH(2), CHTC(2), CHPLX(2), CPTSR(2), CMSFL(2)
Z
CTMPF(2), CTCON(2), CDEN(2), CSPHT(2), CGASC(2)
Z
CVISC(2), CRHOG(2), UNITS
Z

DIMENSION DP(80), SP(80), ALPH(12), ALPH2(4), CD1(200)
Z

ITTO = TOTAL TEMPERATURE OF BLADE COOLING AIR AT INLET
WPLEN = ESTIMATE OF COOLANT FLOW RATE - USED AS FIRST GUESS
PTIO = TOTAL PRESSURE OF BLADE COOLING AIR AT INLET
PEX = EXTERNAL GAS STREAM STATIC PRESSURE AT TRAILING EDGE

DATA ALPH/ 'THI', 'S JO', 'B WA', 'S ST', 'ARTE', 'D AT', 'Z'
'M', 'ON', '!', '/

DATA NCCHAR/16/

MD1 = 0
MD2 = 0
Tyme = -1.0
Dltyme = 0.0

C TO GET AN ABBREVIATED OUTPUT OF MID-WALL TEMPERATURES AT THE TERMINAL
C FOR EACH SLICE, ENTER:
C AT TACT.50; SET TACT.MD1=1
C
MD3 = 0
K = 1
C
C RECORD STARTING TIME, TO BE USED TO IDENTIFY MICROFILM PLOTS
C
CALL TIME(NCHAR,ALPH2)
ALPH(7) = ALPH2(3)
ALPH(8) = ALPH2(4)
ALPH(10) = ALPH2(1)
ALPH(11) = ALPH2(2)
WRITE(6,425) (ALPH(I),I=1,12)
WRITE(8,425) (ALPH(I),I=1,12)

C
C READ IN DATA
C
CALL GETIN(IWRITE,TYMMAX,WSVST,IADJIN)

C
C WRITE TITLE PAGE
C
WRITE(6,400)
WRITE(6,425) (ALPH(I),I=1,12)
WRITE(8,430) (TITLZ(1),I=1,30)

C
C
Ttin = TTIO(1)
Ptin = PTIO(1)
Weleno = WPLEN
Ptinold = PTIN

C
C DO 1100 ITYM = 1,NTYM
C ITYM = ITYM+1
C Ntyn = ITYM*DLTYME
C IF (DLTYME.GT.0.) NTYM = TYMMAX/DLTYME + 1
C Nodst = 5*Nsta

C
C START MARCHING
C
DO 1100 ITYM = 1,NTYM
C ITYM = ITYM+1
C Ntyn = ITYM*DLTYME
C IF (ITYM.EQ.1) Tyme = -1.

C
C-- EVALUATE TIME DEPENDENT BOUNDARY CONDITIONS ----------------------------

C
Ptin = PTIO(1)
C IF (Tyme.LT.0.0) GO TO 490
C-- LOCATE COOLANT SUPPLY PRESSURE FOR TYME
C
DO 450 I = 4,50,2
PTIN = PTIO(I-3)
IF (PTIO(I).LE.0.0) GO TO 460
IPTIO = I-1
IF (TYME.LE.PTIO(I).AND.TYME.GT.PTIO(I-2)) GO TO 455
CONTINUE

450 CONTINUE

455 DENOM = PTIO(IPTIO+1)-PTIO(IPTIO-1)
IF (DENOM.GT.0.) PTIN = PTIO(IPTIO-2) +
      (PTIO(IPTIO)-PTIO(IPTIO-2))*(TYME-PTIO(IPTIO-1))/DENOM
CONTINUE

C-- LOCATE COOLANT SUPPLY TEMPERATURE FOR TYME
C
DO 470 I = 4,50,2
TTIN = TTIO(I-3)
IF (TTIO(I).LE.0.0) GO TO 490
ITTIO = I-1
IF (TYME.LE.TTIO(1).AND.TYME.GT.TTIO(I-2)) GO TO 475
CONTINUE

475 DENOM = TTIO(ITTIO+1)-TTIO(ITTIO-1)
IF (DENOM.GT.0.) TTIN = TTIO(ITTIO-2) +
      (TTIO(ITTIO)-TTIO(ITTIO-2))*(TYME-TTIO(ITTIO-1))/DENOM
CONTINUE

C LOCATE THE VALUE OF THE WHZEL SPEED FOR THE CURRENT TIME.
C
WS = WSVST(1)
IF (TYME.LE.0.0) GO TO 530
DO 510 I = 4,50,2
WS = WSVST(I-3)
IF (WSVST(I).LE.0.0) GO TO 530
WS = WSVST(I-1)
IF (TYME.LE.WSVST(I).AND.TYME.GT.WSVST(I-2)) GO TO 520
CONTINUE

510 CONTINUE

520 DENOM = WSVST(IWS+1)-WSVST(IWS-1)
IF (DENOM.GT.0.) WS = WSVST(IWS-2) +
      (WSVST(IWS)-WSVST(IWS-2))*(TYME-WSVST(IWS-1))/DENOM
CONTINUE

C LOCATE THE VALUE FOR PEXIT, GAS STATIC PRESSURE AT EXIT OF BLADE,
C FOR THE CURRENT TIME AND ALL SLICES.
C
IF (TYME.GT.0.0) GO TO 533
DO 532 I = 1,NSLICE
PEXIT(I) = PEX(I)
IF (PEX(I).LE.0.0) PEXIT(I) = PEX(1)
CONTINUE

532 CONTINUE

533 CONTINUE

IF (BCTIME(2).LE.0.0) GO TO 545
DO 535 I = 2,50
IPEX = I-1
IF (TYME.LE.BCTIME(I).AND.TYME.GT.BCTIME(I-1)) GO TO 540
CONTINUE

535 CONTINUE

540 DENOM = BCTIME(IPEX+1)-BCTIME(IPEX)
IF (DENOM.GT.0.0) GO TO 545
TYMFR = (TYME - BCTIME(IPEX))/DENOM
DO 542 I = 1,NSLICE
      PEXIT(I) = PEXIT(I) +
      (BCTIME(IPEX)-BCTIME(IPEX-1))*(TYME-BCTIME(IPEX-1))/DENOM
CONTINUE

542 CONTINUE

C
IC = (IPEX-1)*NSLICE + I
PEXIT(I) = PEX(IC) + (PEX(IC+NSLICE) - PEX(IC))*TYMFR

CONTINUE
IF (IFILM.EQ.2) GO TO 545

C-- SET INTERPOLATED VALUES OF FREE STREAM RHO*V FOR THIS TIME
C
DO 543 I = 1,NSTA
IRO = (IPEX-1)*NSTA + I
IRN = IPEX*NSTA + I

RHOVGA(I) = RHOV(IRO) + TYMFR*(RHOV(IRN) - RHOV(IRO))

CONTINUE

C-- SET TIME INTERPOLATED VALUES OF QHUB OR THUB AND QTIP OR TTIP.
IF (BCTIME(2).LE.0.0) GO TO 555
DO 550 I = 1,NSTA
IQO = (IPEX-1)*NSTA + I
IQN = IPEX*NSTA + I

Z = TYMFR*(THUB(I) - THUB(IQO))
Z = TYMFR*(QHUB(IQO) - QHUB(I))
Z = TYMFR*(TTIP(IQO) - TTIP(I))
Z = TYMFR*(QTIP(IQO) - QTIP(I))

CONTINUE

GO TO 565

C

555 CONTINUE

DO 560 I = 1,NSTA
IF (IHUB.EQ.1) THUB(I) = THUBN(IQO) +
Z = TYMFR*(THUBN(IQN) - THUBN(IQO))
Z = TYMFR*(QHUBN(IQO) - QHUBN(I))
Z = TYMFR*(TTIPN(IQO) - TTIPN(I))
Z = TYMFR*(QTIPN(IQO) - QTIPN(I))

CONTINUE

GO TO 565

C

560 CONTINUE

C-- CALCULATE TEMPERATURE AND PRESSURES FOR EACH SLICE OF THE BLADE
C
DO 1000 I = 1,NSLICE
ISLICE = I

C FIRST DETERMINE IMPINGEMENT PLENUM CONDITIONS
CALL PLNUM(WPLEN,PPLEN,PTLEN,TLEN,TCIN)
TEN = TPLEN + 460.
PIM = PPLEN
IF (UNITX.EQ.1) GO TO 860
WRITE(6,800) I,PIM,TEN

FORMAT(1HZ/*10X,100(*')/30X,THE IMPINGEMENT PLENUM CONDITIONS)
Z ' FOR SLICE NO.',I2,' ARE: */60X,'PIM = ',F7.2.

800 CONTINUE

56
Z 'PSIA,'//60X,'TOG = ',F7.2,' DEG. B'//10X,100('**))

C IF (I.EQ.1) WRITE(6,850) WPLEN
GO TO 890
C WPLEN=CPlN/1.8
CPIM=PIM/CPRSR(I)
WFITE(6,E7G) I,CPIM,CTOG
D70 FORMAT(1H2//1@X,100
(*I)
//30X,'THE IMPELLER PLNUM CONDITIONS'
Z FOR SLICE NO.'62,' ARE: '/60X,'PIM = ',F7.2,' K
Z KPA, '/60X,'TOG = ',F7.2,'K '/10X,100('**))
CWPLEN=WPLEN/CPSFL(I)
IF (I.EQ.1) WRITE (6,880) CWPLEN
C CONTINUE
C C THEN COMPUTE TEMPERATURES AND PRESSURES
C CALL PREP(I,NTTG)
C CALL TCOZP(IWRITE,WS,K,IPLOT,ALPH2)
IF (IPLOT.GT.0) CALL PLOTMF(ALPH2)
IF (MD1.EQ.O) GO TO 975
L C THIS BLOCK PRINTS SPECIAL CONDENSED OUTPUT TO THE TERMINAL IF MD1 > 0
C IC1=0
DO 955 II=1,NSTA,2
IC1=IC1+1
NMW=5*II-4
555 CD1(IC1)=T(2,I,NMW)-460.
DO 360 II=2,NSTA,2
IC1=IC1+1
NMW=5*II-4
360 CD1(IC1)=T(2,I,NMW)-460.
WRITE(8,962) I,K
S2 FORMAT('/1 BLADE SLICE ',I2,' OVERALL FLOW LOOP ',I2,'/
Z SURFACE TEMPERATURE, (F), STARTING AT LEADING EDGE'/)
INUM=NSTA/2+1
WRITE(8,964) (CD1(II),II=1,INUM)
5f0 FORMAT('/ PRESSURE SIDE'/10(2X,F7.1))
INUM=INUM+1
WRITE(8,966) CD1(1),(CD1(II),II=INUM,NSTA)
9tf FORMAT('/ SUCTION SIDE'/10(2X,F7.1))
C C CONTINUE
C C CHECK HOW MUCH PLNUM FLOW IS LEFT
WUSED=WUSED+3600.*WIM
EXCESW=WPLEN-3600.*WIM
IF (EXCESW.GT.0.AND.I.LT.NSLICE) WPLEN=EXCESW
IF (EXCESW.LT.0.AND.I.LT.NSLICE) WRITE (6,985) I
985 FORMAT('/ RAN OUT OF MASS FLOW IN BRANCH NO.',I2)
100 CONTINUE
C C
IF (IUNITS.EQ.2) WRITE(*,1010) WUSED
1010 FORMAT(*,30X,'AMOUNT OF COOLANT ACTUALLY USED AT THIS TIME'
     Z 'STEP IS ',F6.1,' LBM/HR')
CWUSED = WUSED/CMSFL(1)
IF (IUNITS.EQ.1) WRITE(*,1011) CWUSED
1011 FORMAT(*,30X,'AMOUNT OF COOLANT ACTUALLY USED AT THIS TIME'
     Z 'STEP IS ',F6.1,' KG/HR')
C DO 1040 I = 1,NSLICE
  DO 1020 J = 1,NODST
    T(1,I,J) = T(2,I,J)
  DO 1040 J = 1,NSTA
    P(1,ISLICE,J) = P(2,ISLICE,J)
  IF (TNYME.GT.0.0) GO TO 1100
C C----> ADJUST COOLANT FLOW AND RECALCULATE TEMPERATURES, ETC. FOR STEADY
C STATE CASE OR TIME =0.0
C EXCESW = WPLENO - WUSED
  IF (ABS(EXCESW).LT.0.01*WPLENO) GO TO 1100
  IF (IADJIN.GT.0.0) GO TO 1050
C C----> NORMAL ADJUSTMENT IS ON WPLEN
C WPLEN = WPLENO - .995*EXCESW
  K = K + 1
  PTIN = PTNOLD
  TCIN = TTIN
  GO TO 570
C C-----> SPECIAL CASE, FOR IADJPT > 0, ADJUSTMENT IS ON PTIN
C 1050 PTIN = PEXIT(1) + (PTNOLD-PEXIT(1)) *(WPLENO/WUSED)**1.6
  WPLEN = WPLENO
  TCIN = TTIN
  K = K + 1
  GO TO 570
C 1100 CONTINUE
C C IF (IUNITS.EQ.1) GO TO 3860
3000 WRITE(*,3500) K,WPLENO
3500 FORMAT(*,20X,80('-'),/,'LOOP(S) ON INITIAL COOLANT FLOW'
     Z 'WERE USED. FINAL VALUE IS ',F7.2,' LB/HR')
  WRITE(*,3600) EXCESW
3600 FORMAT(5X,'RESIDUAL COOLING AIR FLOW IS ',F9.4,' LBM/HR',/ Z 20X,80('-'))
  WRITE(*,3870) (ALPH(1),I=1,12)
C C GO TO 3900
C 3860 CWPLEN = WPLENO/CMSFL(1)
  CEXCSW = EXCESW/CMSFL(1)
  WRITE(*,3880) K,CWPLEN
3870 FORMAT(*,20X,80('-'),/,'LOOP(S) ON INITIAL COOLANT FLOW'
     Z 'WERE USED. FINAL VALUE IS ',F7.2,' KG/HR')
  WRITE(*,3880) CEXCSW
C 3880 FORMAT(5X,'RESIDUAL COOLING AIR FLOW IS ',F9.4,' KG/HR',/ Z 20X,80('-'))
WRITE (6,425) (ALPH(I),I=1,12)  
C  3900 CONTINUE  
MD2 = 1  
IF (NSLICE.GT.1.AND.IPLT.GT.0) CALL PLOTPF(ALPH2)  
WRITE (6,425) (ALPH(I),I=1,12)  
STOP  
END  
C-----SOURCE.NFLOEST  
SUBROUTINE FLOWS(JS,DELTAN,ICHOKE,AMCHOK)  
C  COMMON /CHKHOL/ WCHK(80), WCHKDM  
C  COMMON /PLMCOL/ RHOVGA(80), PG(80), XFC(80), PLMEFF(80),  
Z XMUC(80), EMES(80), REFC(80), NFCSUP(80)  
C  COMMON /FRIC/ ALPHA, BETA, DELTA, EPS  
C  COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80),  
Z CPC(80), GAMC(80), DUMR1(80), DUMR2(80)  
C  COMMON /TCO/ ADUMP, BTA, CD, CF,  
Z GAM, PIN, R, SPAN, TOG,  
Z WDUMP, WIM, AKC(15,80), AKW(15,80),  
Z A(400), AJET(80), AM2(80), CNUM(80),  
Z DH(80), DHF(80), DHJ(80),  
Z DLX(400), FP(80), HC(80), HG(80),  
Z P(2,15,80), PEXIT(15), PUMP(80), QG(80),  
Z QSNK(80), RR(80), S(15), T(2,15,400),  
Z TG(80), TAU(400), WFC(80),  
Z WJ(15,80), WCROS(2,15,90),  
Z ICO, IFILM, IHUB, ITIP,  
Z TSBLK, ISLICE, NLKSSZ, NSLICE,  
Z NFWD, NSTA, ICU(80)  
C  COMMON /TRNSNT/ RHOC, RROH, SPHTC, SPHTM,  
Z DLTYME, TYME, TEP, TTYMMA  
C  DATA CHKD/'A'/, UNCHKD/' '  
C  DIMENSION DELTAN(15)
C FOLLOWING VARIABLES NEEDED TO CALCULATE FILM COOLING EFFECTIVENESS.
C THEY ARE TRANSMITTED THROUGH COMMON FILMCOL.
C WHERE-
C    PG IS GAS SIDE STATIC PRESSURE DISTRIBUTION; XFC IS THE
C    DISTANCE A STATION IS DOWNSTREAM FROM THE NEAREST
C    ROW OF FILM COOLING HOLES, (IN); FILM EFFECTIVENESS IS THE CALCULATED
C    FILM EFFECTIVENESS AT EACH STATION;
C    XMUC IS COOLANT VISCOSITY BASED ON LOCAL COOLANT TEMPERATURE;
C    EMES IS THE PRODUCT M*S, WHERE M IS THE BLOWING
C    RATE, AND S IS AN EQUIVALENT SLOT WIDTH; REFC IS THE FILM
C    REYNOLDS NUMBER, BASED ON S; AND NFCSUP IDENTIFIES
C    THE STATION NUMBER SUPPLYING FILM COOLING TO DOWNSTREAM STATIONS.
C
100 CONTINUE
C INITIALIZE HOLE CHOKING INDICATOR TO UNCHOKED
DO 101 I = 1,NSTA
101  WCHK(I) = UNCHK
ICHOKE = 0
N = NSTA-1
C
C N = NODE NUMBER OF LAST FLOW CHANNEL NODE, AT EXIT OF TRAILING EDGE
C
C CALCULATE IMPINGEMENT FLOWS, AND FORWARD REGION FILM COOLING FLOWS
C
TMP = T0G
CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)
GAM0 = GAM
CP0 = CP
NODSF = 5*NFWD
C
PAVG = P(2,ISLICE,1)
WJ(ISLICE,1) = 0.0
IF (PAVG.GT.PIM) GO TO 120
WCR=CD*PAVG*AJET(1)/(R*T0G)*SQRT(64.4*GAM0*R*T0G/(GAM0+1.)) * 
   (PIM/PAVG)**((GAM0-1.0)/GAM0)
WJ(ISLICE,1) = PAVG/(R*T0G)*AJET(1)*CD* 
   SQRT(64.4*GAM0*R*T0G/(GAM0+1.)* (1.- (PAVG/PIM)**((GAM0-1.0)/GAM0)) 

WPC(1) = 0.0
IF (P(2,ISLICE,1).GT.PG(1)) WFC(1) = CD*.25*3.1415926* (DHF(1)**2) 
   Z *SQRT(32.2*P (2,ISLICE,1) * (P (2,ISLICE,1) -PG(1))/ (R*T (2, ISLICE, 5)) ) 
   IF(WCR.LT.WJ(ISLICE,1)) WCHK(1) = CHKD 
   IF(WCR.LT.WJ(ISLICE,1)) WJ(ISLICE,1) = WCR
C
120 PAVG = P(2,ISLICE,2)
WJ(ISLICE,2) = 0.0
IF (PAVG.GT.PIM) GO TO 130
WCR=CD*PAVG*AJET(2)/(R*T0G)*SQRT(64.4*GAM0*R*T0G/(GAM0+1.)) * 
   (PIM/PAVG)**((GAM0-1.0)/GAM0)
WJ(ISLICE,2) = PAVG/(R*T0G)*AJET(2)*CD* 
   Z *SQRT(64.4*GAM0*R*T0G/(GAM0+1.)*(1.- (PAVG/PIM)**((GAM0-1.0)/GAM0)) 
   Z *PIM/PAVG)**((GAM0-1.0)/GAM0)
WPC(2) = 0.0
IF (P(2,ISLICE,2).GT.PG(2)) WFC(2) = CD*.25*3.1415926* (DHF(2)**2) 
   Z *SQRT(32.2*P (2,ISLICE,2) * (P (2,ISLICE,2) -PG(2))/ (R*T (2, ISLICE, 10)) ) 
   IF(WCR.LT.WJ(ISLICE,2)) WCHK(2) = CHKD 
   IF(WCR.LT.WJ(ISLICE,2)) WJ(ISLICE,2) = WCR
C
130  PAVG = P(2, ISLICE, 3)  
WJ(ISLICE, 3) = 0.0  
IF (PAVG.GT.PIH) GO TO 140  
WCR=CD*PAVG*A*JET(3)/(R*TOG)*SQRT((64.4*GAMO*R*TOG)/(GAMO+1.))  
Z (PIH/PAVG)**((GAMO-1.0)/GAMO)  
WJ(ISLICE, 3)=PAVG/(R*TOG)*A*JET(3)*CD*  
Z *SQRT((64.4*GAMO*R*TOG)/(GAMO+1.))  
Z *SQR((64.4*GAMO*R*TOG)/(GAMO+1.))*(1.-PAVG)**((GAMO-1.0)/GAMO)) 
WFC(3) = 0.0  
IF (P(2, ISLICE, 3).GT.PG(3)) YFC(3)  
= CD*.25*3.1415926*(DHP(3)**2)  
Z*SQRT(32.2*P(2, ISLICE, 3)*(P(2, ISLICE, 3)-PG(3))/Z (R*T(I5,JS)))  
IF(WCR.LT.WJ(ISLICE, 3)) WCHK(2) = CHKD  
IF(WCR.GT.WJ(ISLICE, 3)) YJ(ISLICE, 3) = WCR  
CONTINUE  
140 CONTINUE  
C  
DO 150 I = 4, NFWD  
PAVG = P(2, ISLICE, I)  
WJ(ISLICE, I) = 0.0  
IF (PAVG.GT.PIH) GO TO 150  
WCR=CD*PAVG*A*JET(I)/(R*TOG)*SQRT((64.4*GAMO*R*TOG)/(GAMO+1.))  
Z (PIH/PAVG)**((GAMO-1.0)/GAMO)  
WJ(ISLICE, I)=PAVG/(R*TOG)*A*JET(I)*CD*  
Z *SQRT((64.4*GAMO*R*TOG)/(GAMO+1.))*(1.-PAVG)**((GAMO-1.0)/GAMO))  
WFC(I) = 0.0  
IF (P(2, ISLICE, I).GT.PG(I)) WFC(I) = CD*.25*3.1415926*(DHP(I)**2)  
Z*SQRT(32.2*P(2, ISLICE, I)*(P(2, ISLICE, I)-PG(I))/Z (R*T(I5,JS)))  
IF(WCR.LT.WJ(ISLICE, I)) WCHK(I) = CHKD  
IF(WCR.GT.WJ(ISLICE, I)) YJ(ISLICE, I) = WCR  
CONTINUE  
C  
C CALCULATE CROSSFLOW RATE AT EACH STATION  
C  
200 WCROS(2, ISLICE, JS) = 0.0  
JDIS = JS/2  
C**** JDIS IS THE DISPLACEMENT OF THE FLOW SPLIT STATION FROM STATION 1  
JSENS = JS - 2*JDIS  
C********* JSENS = 0, STATION IS ON SUCTION SIDE (EVEN);  
C = 1, STATION IS ON PRESSURE SIDE (ODD)  
C  
JP = JS + 2  
JM = JS - 2  
IF (JM.LT.1) JM = 1  
JPIN = JS + 1  
JSTART = JS + 3  
IF (JS.EQ.1) GO TO 220  
IF (JSENS.EQ.1) GO TO 230  
IF (JSENS.EQ.0) GO TO 270  
C  
220 CONTINUE  
C***** WCROS AT A GIVEN STATION IS THE CROSSFLOW AT UPSTREAM STATION  
C PLUS IMPINGEMENT JET  
C FLOW FROM UPSTREAM STATION MINUS ANY FILM COOLING  
C FLOW AT THIS STATION.  
C  
C** THIS BLOCK IS EXECUTED IF FLOW SPLIT OCCURS AT STATION 1. ISTART=5  
WCROS(2, ISLICE, 2) = DELTAN(ISLICE)*(WJ(ISLICE, 1)-WFC(1)) - WFC(2)  
WCROS(2, ISLICE, 3) = (1.-DELTAN(ISLICE))*(WJ(ISLICE, 1)-WFC(1)) - WFC(3)  
GO TO 320
CONTINUE

C********** THIS BLOCK IS EXECUTED IF FLOW SPLIT STATION IS ON
C THE PRESSURE SIDE. JSENS=1

WCROS(2,ISLICE,JP) = (1. - DELTAN(ISLICE)) * (WJ(ISLICE,JS) - WFC(JS))
Z = WCROS(2,ISLICE,JP)

Z = WCROS(2,ISLICE,JS) - WCROS(2,ISLICE,JP) - WFC(JP)

IF (JM.EQ.1) GO TO 250
IRNG = JDIS - 1
DO 240 I = 1,IRNG
IBK = (JM - Z*I)
IBKP = IBK + 2

240 WCROS(2,ISLICE,IBK) = WCROS(2,ISLICE,IBKP) + WJ(ISLICE,IBKP) - WFC(IBK)

250 JNOD = JFIN
DO 260 I = 2,JNOD,2
IM8 = I - 2
IF (IM8.EQ.0) IM8 = 1

260 WCROS(2,ISLICE,I) = WCROS(2,ISLICE,IM8) + WJ(ISLICE,IM8) - WFC(I)
GO TO 320

C

270 CONTINUE

C********** THIS BLOCK IS EXECUTED IF FLOW SPLIT STATION IS ON
C SUCTION SIDE. JSENS=0

WCROS(2,ISLICE,JP) = DELTAN(ISLICE) * (WJ(ISLICE,JS) - WFC(JS)) - WCROS(2,ISLICE,JP)
DELTA(ISLICE) = WJ(ISLICE,JS) - WFC(JS)

DIFS = WJ(ISLICE,JS) - WFC(JS)

Z = WCROS(2,ISLICE,JP)

Z = WCROS(2,ISLICE,JP) - WCROS(2,ISLICE,IBK) + WJ(ISLICE,IBK) - WFC(IBK)

Z = WCROS(2,ISLICE,IBK) - WFC(IBK)

Z = WCROS(2,ISLICE,IBK) - WCROS(2,ISLICE,1) + WJ(ISLICE,1) - WFC(1)

290 WCROS(2,ISLICE,1) = WCROS(2,ISLICE,2) + WJ(ISLICE,2) - WFC(1)

300 CONTINUE

C-------NOW UP THE PRESSURE SIDE

JNOD = JFIN
DO 310 I = 3,JNOD,2

310 WCROS(2,ISLICE,I) = WCROS(2,ISLICE,I-2) + WJ(ISLICE,I-2) - WFC(I)

320 CONTINUE

ISTART = JSTART
DO 330 I = ISTART,NFWD
WCROS(2,ISLICE,I) = WCROS(2,ISLICE,I-2) + WJ(ISLICE,I-2) - WFC(I)

330 CONTINUE

C

C CALCULATE CROSSFLOW RE & MACH NUMBER SQUARED, AND FILM COOLING RE,
C FOR THE FORWARD REGION.

C

DO 360 I = 1,NFWD
LCOOL = 5*I
LIN = LCOOL - 1
AMIN = A(LCOOL)

C

C EVALUATE COOLANT PROPERTIES AT MEAN TEMPERATURE BETWEEN WALL
C AND COOLANT BULK
C

C ----------------------------------

62
\[
\text{TMP} = \frac{(T(2, \text{ISLICE}, \text{LCOOL}) + T(2, \text{ISLICE}, \text{LIN}))}{2}.
\]

\[
\text{CALL GASTBL(TMP, C, CP, GAH, PD, R, IMU)}
\]

\[
\text{XMUC(I) = XMU}
\]

\[
\text{CPC(I) = CP}
\]

\[
\text{GAMC(I) = GAM}
\]

c

\[
\text{RE(I) = 12. * 3600. * \text{ABS(WCROS}(2, \text{ISLICE}, I))} \cdot \text{DH}(I) / (\text{AMIN} \cdot \text{XMU})
\]

\[
\text{REFC(I) = 12. * 3600. * \text{WC}(I) / (S(\text{ISLICE}) \cdot \text{XMU})}
\]

\[
\text{IF (ICH(I) .EQ. 3) AMIN = A(\text{LCOOL}) \cdot (SP(I) - DP(I)) / SP(I)}
\]

\[
W = \text{WCROS}(2, \text{ISLICE}, I)
\]

\[
P\text{BAR} = P(2, \text{ISLICE}, I)
\]

\[
T\text{BAR} = T(2, \text{ISLICE}, \text{LCOOL})
\]

\[
335 \quad \text{AM2}(I) = \left(\frac{W}{(P\text{BAR} \cdot \text{AMIN})} \right)^{2} \cdot \text{TBAR} / \text{GAMC}(I) / 32.2
\]

\[
\text{IF (AM2(I) .LT. 1.0) GO TO 340}
\]

\[
\text{ANCOK = AM2}(I)
\]

\[
\text{ICHOKE = I}
\]

\[
340 \quad \text{IF (IAC(I) .NE. 3) GO TO 350}
\]

\[
\text{AH2}(I) = \left(\frac{W}{(P\text{BAR} \cdot \text{ABIN})} \right)^{2} \cdot \text{R} \cdot \text{TBAR} / \text{GAMC}(I) / 32.2
\]

\[
\text{IF (AH2}(I).LT.1.0) \text{GO TO 340}
\]

\[
\text{AHCHOK} = \text{n2}(I)
\]

\[
\text{ICHOKE} = \text{I}
\]

\[
350 \quad \text{CONTINUE}
\]

\[
360 \quad \text{CONTINUE}
\]

\[
\text{C CALCULATE FLOW DUMPED DIRECTLY INTO TRAILING EDGE REGION.}
\]

\[
\text{PAVG} = 0.5 \cdot (P(2, \text{ISLICE}, \text{NPWD}) + P(2, \text{ISLICE}, \text{NPWD}-1))
\]

\[
\text{IF (PAVG .GT. PIH) GO TO 370}
\]

\[
W\text{CR} = \text{C}\text{D} \cdot P\text{AVG} \cdot \text{A}\text{DUMP} / (R \cdot \text{TOG}) \cdot \text{SQRT}(64.4 \cdot \text{GAMO} \cdot R \cdot \text{TOG} / (\text{GAMO} + 1.))
\]

\[
Z = (\text{PIN} / \text{PAVG}) \cdot ((\text{GAMO} - 1.0) / \text{GAMO})
\]

\[
W\text{DUMP} = \text{PAVG} / (R \cdot \text{TOG}) \cdot \text{A}\text{DUMP} \cdot \text{CD} \cdot Z \cdot \text{SQRT}(64.4 \cdot \text{GAMO} \cdot R \cdot \text{TOG} / (\text{GAMO} - 1.)) \cdot (1. - (\text{PAVG} / \text{PIN}) \cdot ((\text{GAMO} - 1.0) / \text{GAMO}))
\]

\[
Z = (\text{PIN} / \text{PAVG}) \cdot ((\text{GAMO} - 1.0) / \text{GAMO})
\]

\[
\text{IF (WCR .LT. WDUHP) WCHKDH = CHKD}
\]

\[
\text{IF (WCR .LT. WDUHP) WDUMP = WCR}
\]

\[
\text{r}
\]

\[
\text{I}
\]

\[
\text{C ADD UP TOTAL FLOW FROM IMPINGEMENT PLENUM, WIM.}
\]

\[
370 \quad \text{WIM} = \text{WDUMP}
\]

\[
\text{DO 380 I = 1, NSTA}
\]

\[
380 \quad \text{WIM} = \text{WIM} + WJ(\text{ISLICE}, I)
\]

\[
\text{C TRAILING EDGE REGION, CALCULATE FILM COOLING FLOWS.}
\]

\[
\text{ISTRT} = \text{NPWD}+1
\]

\[
\text{DO 400 I = ISTRT, N, 2}
\]

\[
\text{LCOOL} = 5 \cdot I
\]

\[
\text{WFCDUM} = 0.0
\]

\[
390 \quad \text{IF (P(2, \text{ISLICE}, I).GT. PG(I)) WFCDUM = CD} \cdot 25 \cdot 3.14 \cdot 15926 \cdot Z \cdot \text{SQRT}(32.2 \cdot P(2, \text{ISLICE}, I) \cdot (P(2, \text{ISLICE}, I) - PG(I)) / (R \cdot T(2, \text{ISLICE}, \text{LCOOL})))
\]

\[
Z = (R \cdot T(2, \text{ISLICE}, \text{LCOOL}))
\]

\[
\text{WF}(I) = \text{WFCDUM} \cdot (\text{DHP} (I) )^{*} \cdot 2)
\]

\[
\text{WF}(I+1) = \text{WFCDUM} \cdot (\text{DHP} (I+1) )^{*} \cdot 2)
\]

\[
400 \quad \text{CONTINUE}
\]

\[
\text{C TRAILING EDGE REGION, CALCULATE CROSSFLOW, RE, MACH NUMBER}
\]

\[
\text{C SQUARED, AND FILM COOLING RE.}
\]

\[
\text{WCROS}(2, \text{ISLICE}, \text{ISTRT}) = \text{WCROS}(2, \text{ISLICE}, \text{NPWD}-1) + Z
\]

\[
\text{WCROS}(2, \text{ISLICE}, \text{NPWD}) + \text{WDUMP} + WJ(\text{ISLICE}, \text{NPWD}-1)
\]

\[
Z + WJ(\text{ISLICE}, \text{NPWD}) - \text{WF}(\text{ISTRT}) - \text{WF}(\text{ISTRT+1})
\]

\[
\text{WCROS}(2, \text{ISLICE}, \text{ISTRT+1}) = \text{WCROS}(2, \text{ISLICE}, \text{ISTRT})
\]

\[
\text{AMIN} = A(\text{NODSP+5})
\]

63
EVALUATE COOLANT PROPERTIES AT MEAN TEMPERATURE BETWEEN WALL AND COOLANT BULK

\[ \text{THP} = \frac{\text{T}(2, \text{ISLICE}, \text{ISTRT}) + \text{T}(2, \text{ISLICE}, \text{ISTRT}-1)}{2}. \]

CALL GASTBL(TMP, C, CP, GAM, PD, R, XMU)

XMUC(ISTRT) = XMU
XMUC(ISTRT+1) = XMU
CPC(ISTRT) = CP
CPC(ISTRT+1) = CP
GAME(ISTRT) = GAM
GAME(ISTRT+1) = GAM

\[ \text{RE}(\text{ISTRT}) = 12. * 3600 * \text{ABS} \left( \frac{\text{WCROS}(2, \text{ISLICE}, \text{ISTRT})}{\text{A}(\text{LCOOL})} \right) * \text{DH} / \text{XMU} \]

405 \ \text{AM2(ISTR+1)} = ((\text{AMIN} + \text{A}(\text{NODSF}+5)) * \text{SP(ISTR)}) / \text{SP(IS)}

410 IF (IHC(IS) .EQ. 3) AMIN = \text{A}(\text{NODSF}+5) * \text{SP(IS)} / \text{SP(IS)}

420 IF (IHC(IS) .EQ. 3) AMIN = \text{A}(\text{NODSF}+5) * \text{SP(IS)} / \text{SP(IS)}

EVALUATE COOLANT PROPERTIES AT MEAN TEMPERATURE BETWEEN WALL AND COOLANT BULK

\[ \text{THP} = \frac{\text{T}(2, \text{ISLICE}, \text{LCOOL}) + \text{T}(2, \text{ISLICE}, \text{LCOOL}-1)}{2}. \]

CALL GASTBL(TMP, C, CP, GAM, PD, R, XMU)

XMUC(I) = XMU
XMUC(I+1) = XMU
CPC(I) = CP
CPC(I+1) = CP
GAME(I) = GAM
GAME(I+1) = GAM

\[ \text{RE}(\text{I}) = 12. * 3600 * \text{ABS} \left( \frac{\text{WCROS}(2, \text{ISLICE}, \text{I})}{\text{A}(\text{LCOOL})} \right) / \text{XMU} \]

425 \ \text{AM2(I)} = \left( \frac{\text{W}(\text{PBAR}, \text{AMIN})}{\text{PBAR}} \right) * \text{TBAR} / \text{GAME}(\text{I}) / 32.2
IF (AM2(I) .LT. 1.6) GO TO 430
AMCHOK = AM2(I)
ICHOKE = I

430 IF (IHC(IS) .NE. 3) GO TO 440
AM2(I) = AM2(I) * ((AMIN/A(LCOOL)) ** 2)
480 AM2(I+1) = AM2(I)
450 CONTINUE
C
C CALCULATE COOLANT CHANNEL FRICTION FACTOR AT EACH STATION.
C
DO 560 I = 1, NSTA
LCOOL = 5 * I
C
C DETERMINE IF RE IS LAMINAR, TRANSITIONAL, OR TURBULENT
C AND CALCULATE THE FRICTION FACTOR
C
IF (IHC(I) .EQ. 3) GO TO 540
IF (RE(I) .GT. 2300.) GO TO 510
500 FF(I) = DELTA * RE(I) ** EPS
GO TO 560
510 IF (RE(I) .LT. 4000.) GO TO 530
520 FF(I) = ALPHA * RE(I) ** BETAS
530 A1 = DELTA * 2300. ** EPS
A2 = ALPHA * 4000. ** BETAS
GO TO 560
540 CONTINUE
C
C FOR A PIN FIN ARRAY:
FF(I) = 1.060 * (RE(I) ** (-0.3301))
GO TO 560
550 CONTINUE
FF(I) = 0.0
560 CONTINUE
C
C THE FOLLOWING BLOCK IS USED TO COMPUTE THE FILM COOLING EFFECTIVENESS
C IF IFILM IS SET = 2
I
IF (IFILM .LT. 2) GO TO 690
IFSPLT = 0
DO 610 I = 1, NSTA
610 XFC(I) = 0.0
N = NSTA - 1
C LOCATE FILM COOLING HOLES AND SET UP THE XFC ARRAY
C
C--- IFSPLT IS AN INDICATOR THAT TELLS WHICH SIDE OF THE BLADE STATION
C--- IS TO BE CONSIDERED A PART OF FOR FILM COOLING PURPOSES.
C--- = 0 IS THE DEFAULT, AND INDICATES SUCTION SIDE
C--- = 1 WILL INDICATE PRESSURE SIDE.
C FIRST, MARCH DOWN THE PRESSURE SIDE SEARCHING FOR FILM COOLING HOLES
IF (DHF(1) .GT. 0.0) NFC = IFSPLT
IF (NFC .EQ. 0) NFC = NSTA + 1
XDUM = 0.0
DO 615 I = 3, NSTA, 2
NOS = 5 * I - 4
IF (I .GT. NFC) XFC(I) = XDUM + DLX(NOS)
NFC = NFC + 1
615 IF (DHF(I) .GT. 0.0) NFC = IFSPLT
610 CONTINUE
65
IF (DHF(I).GT.0.0) GO TO 612
XDUM = XFC(I)
GO TO 615

612  NFC = I  XDUM = 0.0
615  CONTINUE

C  C--- SUCTION SIDE  C
C
IF (DHF(I).GT.0.0) NFC = 1 - IPSPLT
IF (NFC.EQ.0) NFC = NSTA + 1
XDUM = 0.0
DO 625 I = 2,N,2
NOS = 5*I - 4
IF (I.GT.NFC) XPC(I) = XDUM + DLX(NOS)
IF (DHF(I).GT.0.0) GO TO 622
XDUM = XFC(I)
GO TO 625

622  NFC = I  XDUM = 0.0
625  CONTINUE

C  C INT. J. HT. & MASS TRANS., VA, 1965, PP 55-65  C
C  FLHEFP = 3.09*(((X/(M*S))*((BEMUC/MUG)**(-1/4) + 4.1)**(-.8)
C
IFCS = 0
IFCP = 0
IF (WFC(I).LE.0.0) GO TO 630
IFCS = 1 - IPSPLT
IFCP = IPSPLT
IF (RHOVGA(I).GT.0.0) EMES(I) = 144.*WFC(I)/(RHOVGA(I)*S(ISLICE))
CONTINUE
DO 650 I=2,NSTA
ISENS = I - 2*(I/2)
IF (RHOVGA(I).GT.0.0) EMES(I) = 144.*WFC(I)/(RHOVGA(I)*S(ISLICE))
FLMEFF(I) = 0.0
IF (ISENS.EQ.0) GO TO 660

C  C PRESSURE SIDE SUPPLY HOLE LOCATIONS  C
C  IF (WFC(I).GT.0.0) IFCP = I
NFCSUP(I) = IFCS
GO TO 650

640  CONTINUE

C  C SUCTION SIDE SUPPLY HOLE LOCATIONS  C
C  IF (WFC(I).GT.0.0) IFCS = I
NFCSUP(I) = IFCP
GO TO 650

650  CONTINUE

C  TMP = TG(1)
CALL GASTBL(TMP,C,CPM,GAM,PD,R,XMUM)

C  C FINALLY, CALCULATE THE EFFECTIVENESS  C
C
DO 680 I = 1,NSTA
IMS = NFCSUP(I)
IF (XFC(I).EQ.0.0.OR.EMES(IMS).EQ.0.0.OR.REPC(IMS).EQ.0.0)
Z GO TO 680
C3 = CPC(IMS)/CPM
XBAR = (XFC(I)/EMES(IMS))*((REPC(IMS)*XMUC(I)/XMUM)**(-.25))
ETAPRM = 3.09*(XBAR+4.1)**(-.8)
FLMEFF(I) = C3*ETAPRM/(1.0 + (C3-1.0)*ETAPRM)
IF (FLMEFF(I).GT.1.0) FLMEFF(I) = 1.0

Z GO TO 680
C---- SOURCE.NFLSPLP

SUBROUTINE FLSPLT(AJET, EPSN, ISLICE, NODSF, IDELT, JS, DELTAN, ICONV)
  DIMENSION DELTAN(15), AJET(80), JSOLDS(25)

C- SOURCE.NFLSPLP---A SUBROUTINE TO SET THE STATION AT WHICH COOLING
  AIR FLOW SPLITS BETWEEN THE SUCTION
  AND THE PRESSURE SIDE FLOW CHANNELS.

C INPUT TO FLSPLT IS THE PRESSURE MATCH PARAMETER, EPSN; THE NO. OF
  NODES (NODSF) IN THE IMPINGEMENT REGION;
  JS COMES IN AS THE CURRENT FLOW SPLIT STATION NO., AND IS
  RETURNED AS THE NEW STATION IF A CHANGE IS NEEDED.
  DELTAN COMES IN AS THE CURRENT FRACTION OF FLOW SPLIT TO
  SUCTION SIDE FROM AN IMPINGEMENT
  JET AT JS. IF A CHANGE IN JS IS NOT NEEDED, DELTAN IS
  USED TO FINE TUNE THE SPLIT.
  ICONV INDICATES IF CONVERGENCE IS COMPLETE.
  = 0--NOT DONE; = 1--OK.

NFWD = NODSF/5
IF (IUNSTB.EQ.1) GO TO 280
IF (IDELT.NE.1) GO TO 220
JNUMS = 0
IUNSTB = 0
NUMS = 0
JSGNCH=0
JOUTRG=0
DO 210 I = 1,25
  JSOLDS(1) = 0
  CONTINUE
CRITF = .002
ICONV = 0
JSENS = JS - 2*(JS/2)

C******* (SUCTION - PRESSURE SIDE Pressures)/ SUCTION SIDE = EPSN
  IF (ABS(EPSN).LT.CRITR) GO TO 280

C**********IF EPSN < 0.0; NEED TO INCREASE FLOW TO PRESSURE SIDE
C********** EPSN > 0.0; NEED TO INCREASE FLOW TO SUCTION SIDE

C IF (JTIMES.EQ.0) GO TO 246
C
C********** JTIMES = 0, THIS IS FIRST CHECK AT THIS STATION,
C ********** SO ROUGH ADJUST DELTAN;
C********** 1, HAVE BEEN HERE BEFORE, SO FINE TUNE DELTAN.

C IF ( JSGNCH.GT.0 ) GO TO 247
C
C********** JSGNCH = 0, THERE HAS NOT BEEN A PRIOR SIGN CHANGE IN EPSN;
C********** 1, THERE HAS BEEN A SIGN CHANGE BEFORE,
C ********** SO STAY AT THIS STATION

242 IF (EPSO/EPSN.LT.0.) JSGNCH = 1
C
247  CONTINUE
    IF (EPLAST/EPSN.GE.0) GO TO 243
    DELTAO = DELLAST
    EPSO = EPLAST
243  CONTINUE
    IF (JSGNCH.EQ.0) GO TO 252
    IF (NUMS.GT.0) GO TO 248
    EPSMIN = ABS(EPSN)
    DELTAO = DELTAN(ISLICE)
248  CONTINUE
    NUMS = NUMS + 1
    IF (ABS(EPSN).GT.EPSMIN) GO TO 249
    EPSMIN = ABS(EPSN)
    DELTAO = DELTAN(ISLICE)
249  CONTINUE
    IF (NUMS.LT.4) GO TO 252
    IF (JNUMS.EQ.1) GO TO 250
    NUMS = 0
    JNUMS = 1
    DELTAN(ISLICE) = DELTAO
    JTIMES = 0
    JSGNCH = 0
    JOUTRG = 0
    GO TO 290
250  CONTINUE
    DELAST = DELTAN(ISLICE)
    DELTAN(ISLICE) = DELTAO
    JINSTB = 1
    GO TO 290
C
C
246  JTIMES = 1
    EPSO = EPSN
    DELTAO = DELTAN(ISLICE)
    IF (EPSO.GT.0.0) DELTAN(ISLICE) = (1.0+DELTAN(ISLICE))/2.0
    IF (EPSO.LT.0.0) DELTAN(ISLICE) = DELTAN(ISLICE)/2.
    IF (DELTAN(ISLICE).EQ.DELTAO) DELTAN(ISLICE) = DELTAN(ISLICE)+
                    Z (.5-DELTAN(ISLICE))/5.
    GO TO 290
C
C
252  CONTINUE
    TERM = EPSN*(DELTAO-DELTAN(ISLICE))/(EPSO-EPSN)
    IF (TERM.EQ.0.0) TERM = .05
    IF (JSGNCH.GT.0) GO TO 255
    DELTAO = DELTAN(ISLICE)
    EPSO = EPSN
255  CONTINUE
    DELAST = DELTAN(ISLICE)
    DELTAN(ISLICE) = DELTAN(ISLICE)-TERM
    IF ( DELTAN(ISLICE).LT.1.0.AND.DELTAN(ISLICE).GT.0.0) GO TO 290
    IF (JOUTRG.GT.0) GO TO 258
    IF (DELTAN(ISLICE).LT.0.0) DELTAN(ISLICE) = .01
    IF (DELTAN(ISLICE).GT.1.0) DELTAN(ISLICE) = .99
    JOUTRG = 1
    GO TO 290
C
C
258  CONTINUE
    JOUTRG = 0
C

JSGNCH = 0
JTIMES = 0
JSOLDS(JS) = 1
IF (DELTAN(ISLICE).LT.1.) GO TO 265

C*** MOVE JS IN PRESSURE DIRECTION
IF (JSENS.EQ.0) GO TO 262
JS = JS + 2
IF (AJET(JS).LE.0.) GO TO 261
GO TO 285

C

262 CONTINUE
IF (JS.EQ.2) JS = 1
IF (JS.GT.2) JS = JS - 2
IF (AJET(JS).LE.0.) GO TO 262
GO TO 285

C

C*** MOVE JS IN SUCTION DIRECTION
IF (JSENS.EQ.0) GO TO 267
JS = JS + 2
IF (AJET(JS).LE.0.) GO TO 267
GO TO 285

C

C********** GET READY TO LEAVE SUBROUTINE

C THIS BLOCK IS EXECUTED IF CONVERGENCE WAS DETECTED

ICONV = 1
JTIMES = 0

IF (IUNSTB.EQ.1) WRITE(6,284) ISLICE,IDELT,JS,DELTAN(ISLICE)
IF (IUNSTB.EQ.1) WRITE(8,284) ISLICE,IDELT,JS,DELTA(ISLICE)
IUNSTB = 0
EPSN = EPSN
RETURN

284 FORMAT(1H2,40(''),40('**')/ISLICE,'I2,'POOR FLOW SPLIT','Z
I3,'ITERATIONS, SPLIT AT STATION ','Z I2,'BEST SPLIT IS AT DELTA = ',F6.4)

RETURN

CONTINUE

WRITE(6,792) DELTAN(ISLICE)
WRITE(8,792) DELTAN(ISLICE)
STOP

792 FORMAT(/5X,****** FLOW SPLIT CANNOT BE MADE AS SPECIFIED,'Z
' DELTA = ',F9.5)
C
C
285  IF ( JSOLSJS(JS) .EQ. 1 ) GO TO 789
DELTAN(ISLICE) = .50
C
C THIS BLOCK IS THE USUAL EXIT AFTER ADJUSTING THE FLOW SPLIT
C
290  CONTINUE
EPLAST = EPSN
IF ( JSGNEE .EQ. 0 ) DELAST = DELTAO
IDELT = IDELT + 1
RETURN
END
C---- source: NGASDAT
BLOCK DATA
C-- source: NGASDAT---
C
COMMON /GAAS/ GS(200), NG
DATA GS/620., 1160., 1700., 2240., 2780., 3320.,
Z .02564, .03580, .04548, .05467, .06435, .07475,
Z .2511, .2681, .2814, .2939, .3070, .3214,
Z .706, .705, .703, .702, .699,
Z .07233, .09458, .11369, .13063, .14683, .16256,
Z 170*0.0/, NG /6/
C
C---- GS IS TABLE OF AIR PROPERTIES VS TEMPERATURE AT CONSTANT PRESSURE
C-- PROPERTY VALUES ARE FROM PEPERL & SVEHLA, TN D-7488, AT 20 ATM.
C---- NG IS THE NUMBER OF TEMPERATURE ENTRIES IN THE TABLE
C-- ENTRIES IN GS ARE:
C-- 1ST NG ARE TEMPERATURE,(F)
C-- 2ND NG ARE THERMAL CONDUCTIVITY,(BTU/ (HR*FT*R))
C-- 3RD NG ARE SPECIFIC HEAT, (BTU/(LB*M))
C-- 4TH NG ARE PRANDTL NUMBER
C-- 5TH NG ARE VISCOSITY, (LBM/(FT*HR))
C
END
C---- source: NGASTB
SUBROUTINE GASTBL(TMP,C,CP,GAM,PD,R,XMU)
C-- source: NGASTB
C
A SUBROUTINE TO LOOK UP GAS PROPERTIES IN AN INPUT TABLE (GS(200))
C WHERE TMP = TEMPERATURE AT WHICH PROPERTIES ARE TO BE EVALUATED (R)
C G = GAS TEMPERATURE (BTU/(HR*FT*R))
C CP = GAS SPECIFIC HEAT (BTU/(LB*M))
C GAM = RATIO OF SPECIFIC HEATS
C PD = PRANDTL NUMBER
C R = SPECIFIC GAS CONSTANT (FT*LB)/(LB*M)
C XMU = VISCOSITY (LBM/(FT*HR))
C
COMMON /GAAS/ GS(200), NG
DIMENSION AC(5)
C
TMP1=TMP - 460,
IF(TMP1.GT.GS(1)) GO TO 200
100 AP1=0.0
AP2=1.0
I1=2
I2=1
GO TO 500
200 DO 300 I=1,NG
IF(GS(I).GT_TMP1) GO TO 400
CONTINUE
1=I
300 CONTINUE
500 DO 600 J=1,4
AC(J) =AP1*GS(I1) +AP2*GS(I2)
AC(5) =1.0/(1.0-R/(778.2*AC(2)))
RETURN
END
C---- SOURCE.NGAUS
SUBROUTINE GAUSS(N,K)
C COMMON /HATRX/ TCOP(400,30)
IWR = 0
NEW = 0
IF (IWR.EQ.0) GO TO 63
C DEBUGGING OUTPUT:
C IWR CAN BE SET DYNAMICALLY IN ORDER TO GET DEBUG OUTPUT OF
C SELECTED ROWS OF THE MATRIX, BEFORE OR AFTER REDUCTION.
C WRITE(6,57)
57 FORMAT(' ENTER NUMBER OF ROW TO BE DISPLAYED. USE I3 FORMAT')
READ(7,59) NROW
59 FORMAT(I3)
IF (NROW.EQ.0) GO TO 63
KP = K+1
WRITE(8,60) NROW
WRITE(8,61) (I,TCOP(NROW,I),I=1,KP)
60 FORMAT(' ENTER ANOTHER ROW NO. OR 000 TO CONTINUE PROCESSING')
61 FORMAT(' ENTER NUMBER OF ROW TO BE DISPLAYED. USE I3 FORMAT')
READ(7,62) NROW
WRITE(8,63) (I,TCOP(NROW,I),I=1,KP)
62 FORMAT(' ENTER NUMBER OF ROW TO BE DISPLAYED. USE I3 FORMAT')
GO TO 58

CONTINUE

JPIV = K/2 + 1
N1 = N-1
DO 100 I = 1,N1
JS = I+1
JF = I + K/2
IF (JF.GT.N) JF = N
PIVOT = TCOF(I,JPIV)
IF (PIVOT.EQ.0.0) GO TO 130
DO 90 J = JS,JF
JR = JPIV+1-LS
IF (LR.LT.LS) GO TO 85
80 TCOF(J,L) = TCOF(J,L) - FM*TCOF(I,LR)
85 TCOF(J,K+l) = TCOF(J,K+l) - PM*TCOF(I,K+l)
90 CONTINUE
100 CONTINUE

IF (IWR.EQ.0) GO TO 163

DEBUGGING OUTPUT:

WRITE(8,57)
READ(7,59) NROW
IF (NROW.EQ.0) GO TO 163
KP = K+1
WRITE(8,60) NROW
WRITE(8,61) (I,TCOF(NROW,I),I=1,KP)
WRITE(8,62)
GO TO 158

CONTINUE

TCOF(N,K+l) = TCOF(N,K+l)/TCOF(N,JPIV)
DO 120 I = 1,N1
IIN = N-I
JF = J+JPIV
IF (IJ.GT.N) GO TO 117
115 SUM = SUM - TCOF(IJ,K+l)*TCOF(IIN,JP)
117 CONTINUE
120 TCOF(IIN,K+l) = SUM/TCOF(IIN,JPIV)
125 CONTINUE
RETURN
130 WRITE(7,135) I
135 FORMAT('DIAGONAL ELEMENT FOR ROW ',I2,' IS ZERO. NO',NGAUS 1141,NGAUS 1142,NGAUS 1143,NGAUS 1144,NGAUS 1145,NGAUS 1146,NGAUS 1147,NGAUS 1148,NGAUS 1149,NGAUS 1150,NGAUS 1151,NGAUS 1152,NGAUS 1153,NGAUS 1154,NGAUS 1155,NGAUS 1156,NGAUS 1157,NGAUS 1158,NGAUS 1159,NGAUS 1160,NGAUS 1161,NGAUS 1162,NGAUS 1163,NGAUS 1164,NGAUS 1165,NGAUS 1166,NGAUS 1167,NGAUS 1168,NGAUS 1169,NGAUS 1170,NGAUS 1171,NGAUS 1172,NGAUS 1173,NGAUS 1174,NGAUS 1175,NGAUS 1176,NGAUS 1177,NGAUS 1178,NGAUS 1179,NGAUS 1180,NGAUS 1181,NGAUS 1182,NGAUS 1183,NGAUS 1184,NGAUS 1185,NGAUS 1186,NGAUS 1187,NGAUS 1188,NGAUS 1189,NGAUS 1190,NGAUS 1191,NGAUS 1192,NGAUS 1193,NGAUS 1194,NGAUS 1195,NGAUS 1196,NGAUS 1197,NGAUS 1198,NGAUS 1199,NGAUS 1200
C 
'THROUGH ATTEMPT TO SOLVE WILL BE MADE.') 
GO TO 125 
END 

C----SOURCE.NGETINT 

SUBROUTINE GETIN 

C 

SOURCE.NGETINT---- 

COMMON /HEADER/ BCXS(400), BCXP(400), BCHGS(1000), BCHGP(1000), NGETINT 1209 
Z 
BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000), NGETINT 1210 
Z 
BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80), NGETINT 211 
Z 
QHUBIN(400), QHUB(80), TTIPIN(400), TTIP(80), NGETINT 1212 
Z 
QTIME(50), TTIO(50), PTIO(50), WPLEN, NGETINT 1214 
Z 
WSVT(50), AKCTBL(20), AKWTBL(20), NBC, NBCP NGETINT 1215 
Z 

COMMON /PLMCOL/ RHOVG(80), PG(80), XPC(80), FLMEFF(80), NGETINT 1217 
Z 
XMUC(80), EMES(80), REFPC(80), NFCSUP(80), NGETINT 1218 
Z 

COMMON /IMPINC/ CIMP1, CIMP2, CIMP3, CIMP4, CIMP5, CIMP6, CIMP7, NGETINT 1220 
Z 
DIMP1, DIMP2, DIMP3, DIMP4, DIMP5, DIMP6, NGETINT 1221 
Z 

COMMON /RADL/ APLN(15), DPLN(15), RIN(15), ROUT(15), NGETINT 1223 
Z 
PIN(15), TIN(15), W(15), WS, NGETINT 1224 
Z 

COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000), NGETINT 1226 
Z 
IPLLOT, MD1, MD2, MD3, IADJIN, IWRITE, NGETINT 1227 
Z 

COMMON /TCO/ ADUMP, BTA, CD, CP, NGETINT 1230 
Z 
GAM, PIN, F, SPAN, TOG, NGETINT 1231 
Z 
WDMCP, WIM, AKC(15,80), AKW(15,80), NGETINT 1232 
Z 
A(400), AJET(80), AM2(80), CNUM(80), NGETINT 1233 
Z 
DH(80), DHF(80), DHJ(80), NGETINT 1234 
Z 
DLX(400), FF(80), HC(80), HG(80), NGETINT 1235 
Z 
P(2,15,80), PEXIT(15), PUMP(80), QG(80), NGETINT 1236 
Z 
QSNK(80), RR(80), S (15), T(2,15,400), NGETINT 1237 
Z 
TGG(80), TAU(400), WFC(80), NGETINT 1238 
Z 
WJ(15,80), WCROS(2,15,80), NGETINT 1239 
Z 
XN(80), NGETINT 1240 
Z 
ICOR, IFILM, ITZ, ITZ, NGETINT 1241 
Z 
ISBLOK, ISLICE, NLBKSZ, NSLICE, NGETINT 1242 
Z 
NFWD, NTA, IHC(80), NGETINT 1243 
Z 

COMMON /TRANS/ RHOC, RMOM, SPHTC, SFHTM, NGETINT 1244 
Z 
DLTME, TIME, TEP, TMMAX, NGETINT 1245 
Z 

COMMON /UNITS/ CINCH(2), CHTC(2), CRFLX(2), CPRSR(2), CMSFL(2), NGETINT 1246 
Z 
CTMPF(2), CTCON(2), CDEN(2), CSGHT(2), CGASC(2), NGETINT 1247 
Z 
CIVISC(2), CRHOVG(2), IUNITS, NGETINT 1248 
Z 

DIMENSION THK(3), TDLX(5), TPLMHL(10), NGETINT 1249 
Z 

NAMELIST /TITL/ TITLE, NGETINT 1250 
Z 

NAMELIST /CHNL/ NSLICE, NTA, INDCHN, IWRITE, NGETINT 1251 
Z 
MD1, MD2, MD3, IADJIN, IADJIN, NGETINT 1252 
Z 

NAMELIST /BC/ NBCS, NBCP, BCX, BCXP, BCHGS, BCHGP, NGETINT 1253 
Z 
BCTGS, BCTGP, BCQGS, BCQGP, BCP, BCPGP, NGETINT 1254 
Z 
THUBIN, QHUBIN, TTIPIN, QTIPIN, RHOVG, NGETINT 1255 
Z 
PXX, BCTIME, TTIO, PTIO, WPLEN, NGETINT 1256 
Z 

Z WSVST, AKCTBL, AKWTBL, RHOC, RHOM, NGETINT 1261
Z SPHTC, SPHTM, DITYME, TEPs, TMMAX NGETINT 1262
C NAMELIST /CONTRI/ NFWD, ICOR, NGE0 NGETINT 1263
C NAMELIST /PROPS/ CD, SPAN, ADUMP, DHYD, APLEN, RO, NGETINT 1264
Z RL, CIMP1, CIMP2, CIMP3, CIMP4, CIMP5, NGETINT 1265
Z CIMP6, CIMP7, DIMP1, DIMP2, DIMP3, DIMP4, NGETINT 1266
Z DIMP5, DIMP6 NGETINT 1267
C NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN, NGETINT 1268
Z TDHF, TRR, IHCT, TDP, TSP, TFLMHL NGETINT 1269
C DATA TIKLE/* */ NGETINT 1270
C NSLICE = THE NO. OF SLICES OF THE BLADE THAT ARE BEING CONSIDERED NGETINT 1271
C IHUB = 1 INDICATES A SPECIFIED TEMPERATURE DISTRIBUTION IS GIVEN AT THE HUB END NGETINT 1272
C = 2 INDICATES AN ADIABATIC SURFACE AT THE HUB END NGETINT 1273
C = 3 INDICATES HEAT FLUX IS SPECIFIED AT HUB END (BTU/HR FT**2 R) NGETINT 1274
C ITIP = 1 INDICATES A SPECIFIED TEMPERATURE DISTRIBUTION IS GIVEN AT THE TIP END NGETINT 1275
C = 2 INDICATES AN ADIABATIC SURFACE AT THE TIP END NGETINT 1276
C = 3 INDICATES HEAT FLUX IS SPECIFIED AT TIP END (BTU/HR FT**2 R) NGETINT 1277
C IADJIN = 0, MEANS TO HOLD PTIO CONSTANT AND ADJUST WPLEN; NGETINT 1278
C > 0, MEANS TO PIX WPLEN AND ADJUST PTIO. NGETINT 1279
C ISTA = FIRST STATION NUMBER FOR THIS DATA SET NGETINT 1280
C IF ISTB IS SPECIFIED, IT IS THE LAST STATION NUMBER FOR THIS DATA SET NGETINT 1281
C IF ISTB IS SPECIFIED, IT MUST BE EQUL TO ISTA + A MULTIPLE OF 2. NGETINT 1282
C THK = (1) - COATING THICKNESS, (2) - METAL THICKNESS, AND (3) - CHANNEL WIDTH. ALL IN INCHES. NGETINT 1283
C TDLX = DISTANCE FROM UPSTREAM NODE (INCHES) NGETINT 1284
C TDHJ = HYDRAULIC DIAMETER OF IMPINGEMENT JET HOLE (INCHES) - STORED UNDER STATION NUMBER NGETINT 1285
C TDHF = EFFECTIVE DIAMETER OF FILM COOLING HOLE IF PRESENT (INCHES) - STORED UNDER STATION NUMBER NGETINT 1286
C = DIAMETER OF ONE HOLE*SQRT(NO. OF HOLES AT THIS STATION IN THIS SLICE) NGETINT 1287
C TXN = SPANWISE SPACING OF IMPINGEMENT JETS (INCHES) NGETINT 1288
C TRR = RADIAL LOCATION OF THIS STATION (INCHES) NGETINT 1289
C IHCT INDICATES THE TYPE OF INSIDE HEAT TRANSFER AT THIS STATION, NGETINT 1290
C = 1 FOR IMPINGEMENT WITH CROSSFLOW NGETINT 1291
C = 2 FOR FORCED CONVECTION CHANNEL FLOW NGETINT 1292
C = 3 FOR PIN FIN ARRAY OF PINS. NGETINT 1293
C TDP = THE PIN PIN DIAMETER (IN) IF PINS ARE USED; NGETINT 1294
C TSP = THE PIN PIN SPACING (IN), ASSUMING AN EQUILATERAL TRIANGULAR ARRAY OF PINS. NGETINT 1295
C AKCTBL= TABLE OF CLADDING THERMAL CONDUCTIVITY (BTU/HR FT R) VS TEMPERATURE (F) NGETINT 1296
C AKWTBL= TABLE OF WALL METAL THERMAL CONDUCTIVITY (BTU/HR FT R) VS TEMPERATURE (F) NGETINT 1297
C RHOVG = HOT GAS FREE STREAM MASS VELOCITY, DENSITY*VELOCITY, FOR FILM COOLING USE, AT EACH FILM COOLING STATION. NGETINT 1298
C RHOC = DENSITY OF OUTER COATING (LBM/FT**3) NGETINT 1299
C RHOM = DENSITY OF WALL METAL (LBM/FT**3) NGETINT 1300
C SPHTC = SPECIFIC HEAT OF COATING (BTU/LBM FT R) NGETINT 1301
C SPHTM = SPECIFIC HEAT OF WALL METAL NGETINT 1302
DLTYME = TIME STEP SIZE FOR TRANSIENT CALCULATIONS (SEC)

TMYMAX = MAX. TIME (SEC) TO WHICH TRANSIENT IS CARRIED.

TEPS = FRACTION OF TIME STEP AT WHICH TEMP. IS EVALUATED. (NEW = OLDS)

WSVST = TABLE OF WHEEL SPEED VS TIME, (RPM VS SEC), ODD SUBSCRIPTS ARE SPEED, EVEN ARE TIME, WSVST(2)=0.0

CIMP1 TO CIMP5 ARE EXPONENTS TO BE USED IN A GENERAL IMPINGEMENT WITH CROSSFLOW CORRELATION. IF NOT SPECIFIED, THEN THE BUILT IN KIRCHER-TABAKOFF CORRELATION IS USED.

SEE SUBROUTINE HCOLT FOR DESCRIPTION OF GENERAL CORRELATION.

INITIALIZE:

100 CONTINUE

C

GAS CONSTANT FOR AIR, FT LBF/LBM R

R = 53.35

C-- SET VALUES FOR UNITS CORRECTION FACTORS---

(1) CONVERTS FROM SI TO ENGLISH, (2) MAKES NO CONVERSION--

ALREADY IN ENGLISH

C--- CINCH(1) IS CONVERSION FACTOR FROM (CM) TO (IN)

CINCH(1) = .39370

CINCH(2) = 1.0

C--- CHTC(1) IS CONVERSION FACTOR FROM (WATTS/M**2 K) TO (BTU/HR FT**2 R)

CHTC(1) = .17623

CHTC(2) = 1.0

C--- CHFLX(1) IS CONVERSION FACTOR FROM (WATTS/M**2) TO (BTU/HR FT**2)

CHFLX(1) = .31721

CHFLX(2) = 1.0

C--- CPRSR(1) IS CONVERSION FACTOR FROM (KILOPASCALS) TO (PSIA)

CPRSR(1) = .14503

CPRSR(2) = 1.0

C--- CMSPL(1) IS CONVERSION FACTOR FROM (KG/HR) TO (LBM/HR)

CMSPL(1) = 2.67924

CMSPL(2) = 1.0

C--- CTMPF(1) IS CONVERSION FACTOR FROM (K) TO (R)

CTMPF(1) = 1.8

CTMPF(2) = 1.0

C--- CTCON(1) IS CONVERSION FACTOR FROM (WATTS/M K) TO (BTU/HR FT R)

CTCON(1) = .57817

CTCON(2) = 1.0

C--- CDEN(1) IS CONVERSION FACTOR FROM (KG/M**3) TO (LBM/FT**3)

CDEN(1) = .06243
C--- CSPHT(1) IS CONVERSION FACTOR FROM (J/KG K) TO (BTU/LB-FT)
CSPHT(1) = .000239
CSPHT(2) = 1.0

C--- CVISC(1) IS CONVERSION FACTOR FROM (PA SEC) TO (LB-FT/LBM)
CVISC(1) = 2419.096
CVISC(2) = 1.0

C--- CGASC(1) IS CONVERSION FROM (J/KG K) TO (FT LBF/LBM)
CGASC(1) = .16602
CGASC(2) = 1.0

CDEN(2) = 1.0

C--- CRHOVG IS CONVERSION FROM (KG/SEC M**2) TO (LBM/SEC FT**2)
CRHOVG(1) = .0204823
CRHOVG(2) = 1.0

DO 105 I = 1,30
  TITLE(I) = TIKLE
  BCHGS(I) = 0.0
  BCHGP(I) = 0.0
  BCTGS(I) = 0.0
  BCTGP(I) = 0.0
  BCQGS(I) = 0.0
  BCQGP(I) = 0.0

  BCFGP(I) = 0.0
  RHOC = 0.0
  RHOM = 0.0
  SPHTC = 0.0
  SPHTM = 0.0

  DO 107 I = 1,400
    THUBIN(I) = 0.0
    QHUBIN(I) = 0.0
    TTI PIN(I) = 0.0
    QTIPIN(I) = 0.0
    RHOVG(I) = 0.0

  PEX(I) = 0.0

  DO 108 I = 1,50
    BCTIME(I) = 0.0
    TTIO(I) = 0.0
    PTIO(I) = 0.0
    WSVST(I) = 0.0

  RR(I) = 0.0

  DO 110 I = 1,6000
    CHANL(I) = 0.0

  DO 112 I = 1,15
    PEXIT(I) = 0.0

  DO 112 J = 1,80
  AKC(I,J) = 0.0

  AKW(I,J) = 0.0

  DO 115 I = 1,2000
    INDCHN(I) = 0

  DO 116 I = 1,20
  AKCTBL(I) = 0.0

  AKWBL(I) = 0.0
  IPILOT = 0
  IWRITE = 0

105
106
107
108
110
112
115
116
INEDIT = 0
TEPS = 1.0
DLTIME = 0.0
READ (5, TITL)
READ (5, CHANLS)
READ (5, BC)
IF (BCHGS (1).EQ.0.0) BTA=1.0
IF (BCQGS (1) .EQ.0.0) BTA = 0.0
IF (TTPIN (1).GT.0.0) ITIP = 1
IF (THUBIN (1) .GT.0.0) IHUB = 1
IF (ABS(QTIPIN (1)).GT.0.0) ITIP = 3
IF (ABS(QHUBIN(1)).GT.0.0) IHUB = 3
WS = WSVST (1)

C
PEXIT (1) = PEX (1)
DO 175 ICHLN0 = 1, NSLICE

C ICHLN0 IS THE CHANNEL NUMBER; = 1 AT THE HUB, = NSLICE AT THE TIP

C
READ (5, CONTRL)
NODSF = 5*NFWD
C--NODSF IS THE NUMBER OF NODES IN THE FORWARD REGION
C
NODST = 5*NSTA
C--NODST IS THE TOTAL NUMBER OF NODES IN THE BLADE SLICE ICHLNO
C
NBLKSZ = (15 + 2*NODST) + 8*NSTA
C--NBLKSZ IS THE SIZE OF THE DATA BLOCK RESERVED IN CHANL ARRAY FOR THIS SLICE ICHLNO
C
SLICE ICHLNO
C
ISBLOK = IEND + 1
C--ISBLOK IS THE STARTING POINT IN CHANL ARRAY FOR THIS BLOCK OF DATA
C
INSTRT = 15 + (ICHLNO-1)*(15 + NSTA)
C--INSTRT IS THE STARTING POINT IN INDCHN ARRAY FOR THIS BLOCK OF INTEGER DATA
C
INDCHN (ICHLNO) = INSTRT
INDCHN (INSTRT) = ICHLNO
INDCHN (INSTRT+1) = IFILM
INDCHN (INSTRT+2) = ICOR
INDCHN (INSTRT+3) = NFWD
INDCHN (INSTRT+4) = NSTA
INDCHN (INSTRT+5) = ISBLOK
INDCHN (INSTRT+6) = NBLKSZ
INDCHN (INSTRT+7) = IPOINT
INDCHN (INSTRT+8) = MD1
INDCHN (INSTRT+9) = MD2
INDCHN (INSTRT+10) = MD3
INDCHN (INSTRT+12) = IHUB
INDCHN (INSTRT+15) = ITIP
IIIHCTZ = INSTRT + 14
C--IIIHCTZ IS THE RELATIVE ZERO POINT IN INDCHN FOR STORAGE OF THE C INDICATOR IHC

READ (5, PROPS)
S (ICHLNO) = SPAN * CINCH (IUNITS)
APLN (ICHLNO) = APLN * CINCH (IUNITS) * CINCH (IUNITS)
DPLN (ICHLNO) = DPLN * CINCH (IUNITS)
ROUT (ICHLNO) = ROUT * CINCH (IUNITS)
RIN (ICHLNO) = RIN * CINCH (IUNITS)
NOW, GEO IS READ, NGE0 TIMES, AND THE DATA STORED IN CHANL ARRAY. 
ISBLOK = THE STARTING POINT IN CHANL ARRAY FOR CHANNEL ICHLN0 DATA
FIRST, STORE THE SINGLE VALUED DATA
CHANL(ISBLOK) = CD 
CHANL(ISBLOK+1) = ALPHA 
CHANL(ISBLOK+2) = BETA 
CHANL(ISBLOK+3) = DELTA 
CHANL(ISBLOK+4) = EPS 
CHANL(ISBLOK+6) = ADUMP*CINCH(IUNITS)**2 
CHANL(ISBLOK+7) = SPAN*CINCH(IUNITS) 
CHANL(ISBLOK+8) = ETA 
CHANL(ISBLOK+9) = DLYME 
CHANL(ISBLOK+10)= TEPS 

THE FOLLOWING ARE STORED BY NODE NUMBER:
THK (TAU), TDLX (DLX)

THE REST ARE STORED BY STATION NUMBER:
TDHJ (DHJ), TDHF (DHF), TXN (XN), TRR (RR)

THK(1) = 0.0
DO 170 I = 1,NGEO
   ISTB = 0
   READ(5,GEO)
   IF (THK(1).LE.0.0) THK(1) = .0001*THK(2)
   IF (TDLX(1).GT.0.*TDLX(4).OR.TDLX(4).GT. 12*TDLX(1)) WRITE(8,136) ICHLN0,ISTA
      136 FORMAT(' CHANNEL ',12,*, STATION ',13,
      Z 'DO NOT LOOK RIGHT')
   IF (TDHJ.GT.0. .AND.TXN.LT. 1.1*TDHJ) WRITE (8,137) ICHLN0,ISTA
      137 FORMAT(' CHANNEL ',12,*, STATION ',13,
      Z 'HOLE SPACING AND DIAMETER DO NOT LOOK RIGHT')
   IF (ISTB.EQ.0) ISTB = ISTA
   DO 165 J = ISTA,ISTB,2
      Z

J REPRESENTS THE STATION NUMBER IN THIS CASE
JSENS = J - 2*(J/2)
JSENS = 0 INDICATES THAT STATION NO. IS EVEN AND STATION IS ON SUCTION SIDE
JSENS = 1 INDICATES THAT STATION NO. IS ODD AND STATION IS ON PRESSURE SIDE
C

**IARG = ITDHJZ + J**
**CHANL(IARG) = TDHJ*CINCH(IUNITS)**

**IARG = ITDHFZ + J**
**CHANL(IARG) = TDHF*CINCH(IUNITS)**

**IARG = ITXNZ + J**
**CHANL(IARG) = TXN*CINCH(IUNITS)**

**IARG = ITRRZ + J**
**CHANL(IARG) = TRR*CINCH(IUNITS)**

**IARG = ITDPZ + J**
**CHANL(IARG) = TDP*CINCH(IUNITS)**

**IARG = ITSPZ + J**
**CHANL(IARG) = TSP*CINCH(IUNITS)**

**IARG = IIHCTZ + J**
**INDCHN(IARG) = IHCT**

**NODOUT = 5*J - 4**

C

**NODOUT IS THE NODE NO. ON THE OUTSIDE SURFACE AT STATION J**

**5 IS THE NUMBER OF NODES AT STATION J**

C

**145 CONTINUE**

**LOCA = ITDLXZ + NODOUT**

**IF (TDLX(3) .LE.0.) GO TO 155**

**DO 150 L = 1,5**

**LOCAL = LOCA + L - 1**

**CHANL(Local) = TDLX(L) *CINCH(IUNITS)**

**GO TO 160**

**155 CHANL(LOCA) = TDLX(1) *CINCH(IUNITS)**

**CHANL(LOCA+3) = TDLX(4) *CINCH(IUNITS)**

**AA = TDLX(1)**

**B = (TDLX(4) - TDLX(1)) / (THK(1) + THK(2))**

**CHANL(LOCA+1) = (AA + B*THK(1)) *CINCH(IUNITS)**

**CHANL(LOCA+2) = (AA + B*(THK(1) + THK(2)/2.)) *CINCH(IUNITS)**

**CHANL(LOCA+4) = (AA + B*(THK(1) + THK(2) + THK(3)/2.)) *CINCH(IUNITS)**

**160 CONTINUE**

**LOCA = ITHKZ + NODOUT**

**CHANL(LOCA) = THK(1) *CINCH(IUNITS)**

**CHANL(LOCA+2) = THK(2) *CINCH(IUNITS)**

**CHANL(LOCA+4) = THK(3) *CINCH(IUNITS)**

C

**165 CONTINUE**

**170 CONTINUE**

**175 CONTINUE**

C

**--- CONVEPT UNITS ON BC DATA**

**IF (IUNITS .EQ.2) GO TO 300**

**NTBC = 1**

**DO 205 I = 2,50**

**IF (BCTIME(I) .LE.0.0) GO TO 210**

**205 NTBC = NTBC + 1**

**210 NSETS = NBCS*NSLICE*NTBC**

**NSETP = NBCP*NSLICE*NTBC**

C

**DO 215 I = 1,NSETS**

**BCXS(I) = BCXS(I) *CINCH(1)**

**BCCHGS(I) = BCHGS(I) *CHTC(1)**

**BCTGS(I) = BCTGS(I) *CTMPF(1) - 460.**

**BCQGS(I) = BCQGS(I) *CHFLX(1)**

**215 BCPS(I) = BCPS(I) *CPRSR(1)**
DO 220 I = 1, NSET
BCXP(I) = BCXP(I) * CINCH(1)
BCHGP(I) = BCHGP(I) * CHTC(1)
BCTGP(I) = BCTGP(I) * CTMPF(1) - 460.
BCQGP(I) = BCQGP(I) * CHFLX(1)
BCGP(I) = BCGP(I) * CPRSR(1)
220
C
NSET = NSTA * NTBC
DO 225 I = 1, NSET
RHOVGA(I) = RHOVGA(I) * CHOC(1)
THUBIN(I) = THUBIN(I) * CTMPF(1) - 460.
QHUBIN(I) = QHUBIN(I) * CHFLX(1)
TTHIPI(I) = TTHIPI(I) * CTMPF(1) - 460.
225
QTHIPI(I) = QTHIPI(I) * CHFLX(1)
C
NSET = NSLICE * NTBC
DO 230 I = 1, NSET
PEX(I) = PEX(I) * CPRSR(IUNITS)
230
DO 235 I = 1, 49, 2
TTIO(I) = TTIO(I) * CTMPF(IUNITS) - 460.
235
PTIO(I) = PTIO(I) * CPRSR(IUNITS)
WPLEN = WPLEN * CMSFL(IUNITS)
RHOC = RHOC * CDEN(IUNITS)
RHO3 = RHOC * CDEN(IUNITS)
SPHTC = SPHTC * CSPHT(IUNITS)
SPHTN = SPHTN * CSPHT(IUNITS)
DO 280 I = 1, 19, 2
AKCTBL(I) = AKCTBL(I) * CTMPF(IUNITS) - 460.
AKTBL(I+1) = AKTBL(I+1) * CTCON(IUNITS)
AKWTBL(I) = AKWTBL(I) * CTMPF(IUNITS) - 460.
AKWTBL(I+1) = AKWTBL(I+1) * CTCON(IUNITS)
280
CONTINUE
C
IF (IFILM.NE.2) GO TO 320
DO 310 I = 1, NSTA
RHOVGA(I) = RHOVGA(I)
310
CONTINUE
C
IF INEDIT .GT. 0, PRINT AN INPUT EDIT
C
DO 180 I = 1, NSLICE
CALL INPRT(I, INEDIT)
180
CONTINUE
RETURN
END

C----SOURCE.NHCFRCT
FUNCTION HCFCD(IS, LC00L, LIN)
C
C----SOURCE.NHCFRCT----
C
COMMON /TCO/ ADUMP, BTA, CD, CP, SPAN, T0G,
Z GAN, PIM, R, WDDMP, WIM, AKC(15, 80), AKW(15, 80),
Z A(400), AJET(80), AN2(80), CNUM(90),
Z DH(90), DNF(80), DHJ(80),

HCFRCT 1671
HCFRCT 1672
HCFRCT 1673
HCFRCT 1674
HCFRCT 1675
HCFRCT 1676
HCFRCT 1677
HCFRCT 1678
HCFRCT 1679
HCFRCT 1680
UNIT 139 DLX(400), FF(80), HC(80), HG(80),
UNIT 140 P(2,15,80), PEXIT(15), PUMP(80), QG(80),
UNIT 141 QSNK(80), RR(80), S(15), T(2,15,400),
UNIT 142 DG(80), TA(400), WFC(80),
UNIT 143 WJ(15,80), WCROS(2,15,80), XN(80),
UNIT 144 ICOR, IFILM, IHUB, ITIP,
UNIT 145 ISBLOK, ISLICE, NBLNSZ, NSLICE,
UNIT 146 NFWD, NSTA, IHC(80)

C COMPUTE TURBULENT Heat TRANSFER COEFFICIENT IN CHANNEL FLOW:
C
NU = .023*(RE**.8)*(PD**.333)

C

C CONTINUE
C
TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.
CALL GASTBL(TMPC,CP,GAM,P,D,XMU)
EE = 12.*3600.*ABS(WCROS(2,ISLICE,IS))!*DH(IS)/((A(LCOOL)*XMU)
HCFCRCD = .023*12.*((C/DH(IS))*(RE**.8)*(PD**.333)

C

C---- SOURCE.NHCOOLT
C
SUBROUTINE NHCOOLT(JS)
C
C
COMMON /ICMPOR/ CIMP1, CIMP2, CIMP3, CIMP4, CIMP5, CIMP6, CIMP7,
C
DIM1, DIM2, DIM3, DIM4, DIM5, DIM6
C
COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80),
C
CPC(80), GAMC(80), DUMR1(80), DUMR2(80)
C
COMMON /TCO/ ADUMP, BTA, CD, CP,
C
GAM, EIM, R, SPAN, TOG,
C
WDUMP, WIM, AKC(15,80), AKW(15,80),
C
A(400), AJET(80), AM2(80), CNUM(80),
C
DH(80), DHF(80), DHJ(80),
C
DLX(400), FF(80), HC(80), HG(80),
C
QSNK(80), RR(80), S(15), T(2,15,400),
C
DG(80), TAU(400), WFC(80),
C
WJ(15,80), WCROS(2,15,80), XN(80),
C
ICOR, IFILM, IHUB, ITIP,
C
ISBLOK, ISLICE, NBLNSZ, NSLICE,
C
NFWD, NSTA, IHC(80)
C

C DIMENSION IGG(80), IRE(80), REJ(80), REJOVR(80)
C
CONTINUE
C
TMP=TOG
C
CALL GASTBL(TMPC,C,CP,GAM,P,D,XMU)
CONDC = C
XMUTOG = XMU
PDTOG = PD
PI=3.14159
IF (JS.GT.1) GO TO 101
IF (ICOR.EQ.1) GO TO 101
IF (WJ(ISLICE,JS).LE.0.0) GO TO 101
C
C---- LEADING EDGE HEAT TRANSFER CORRELATION FOR STATIONS FORWARD OF ICORNHCOOLT
C---- CORRELATION OF METZGER ET AL, J. ENG. POWER, JULY 1969, PP 149-158 NHCOOLT
**C**

5 NEND=ICOR+1
XS = 0.0
XP = 0.0
DO 50 J = 3,NEND,2
LINS = 5*(J-1) - 1
LINP = 5*J - 1
XS = XS + DLX(LINS)
XP = XP + DLX(LINP)
50 CONTINUE

55 XL = (XS + XP)/2.
IF (AJET(JS).GT.0.) GMASS = WJ(ISLICE,JS)/AJET(JS)
BES = PI*DHJ(JS)**2/(4.*Xi(JS))
DEN = 2.*BES
REJ(JS) = 12.*3600.*GMASS*DBH/XMU
PROD = REJ(JS)**.27*(XL/BES)**.52
STANMX = .355/PROD
HC(JS) = STANMX*CP*GMASS*144.*3600.

75 FORMAT(1H /****WARNING*** LEADING EDGE IMPINGEMENT JET REYNOLDS****,NHCOOLT 1761
Z 'NUMBER IS ',F8.1/' RANGE OF THE CORRELATION IS 1150,- 1762
Z ' < REJ < 6300')
ILOAD = ICOR - 1
IF (ILOAD.LT.2) GO TO 85
DO 80 I = 2,ILOAD
IF(WJ(ISLICE,I).GT.0.0) GO TO 90
80 HC(I) = HC(JS)
85 CONTINUE
GO TO 101

90 WRITE(8,95) ICOR
95 FORMAT(1H /' SOLUTION TERMINATED*** TOO MANY ROWS OF IMPINGEMENT',NHCOOLT 1772
Z ' HOLES FORWARD OF STATION',I3,'. HOLES ARE ',NHCOOLT 1773
Z 'ALLOWED ONLY AT STATION 1.')
STOP

C

--KIRCHER-TABAKOFF CORRELATION, IMPINGEMENT WITH CROSS FLOW
C--ICOR = STATION NUMBER APPLICATION OF THIS CORRELATION BEGINS

101 IGGC = 0
IREC = 0
ISTRT=ICOR
IF (JS.GT.1) ISTRT= 1

C

IF (CIMP1.NE.0.0) GO TO 400
DO 130 I = ISTRT,NFWD
WC = ABS(WCRS(2,ISLICE,I))
II = 5*I
REI(I) = 0.0
IF (HC(I).NE.1.0) GO TO 103
LCOOL = 5*I
LIN = LCOOL - 1
HC(I) = HCFRCD(I,LCOOL,LIN)
GO TO 130
103 CONTINUE

IF (AJET(I).EQ.0.0) GO TO 128
IF (WJ(ISLICE,I).LE.0.) GO TO 128
TMP = (T(2,ISLICE,LIN)+T0) /2.
CALL GASTBL (TMP,C,CP,GAM,PD,R,XMU)
CONDCT = C
REJ(I) = WJ(1SLICE, I) * AJET(I) / (XMUTOG/3600.) * 12.0
GG = WC/A1) / WJ(1SLICE, I) * AJET(I)
IF (GG <= 2.0) GO TO 110
IGGC = IGGC + 1
IGG(I) = I
CONTINUE
IF (REJ(I) .LT. 3000.) GO TO 115
IREC = IREC + 1
IRE(I) = I
CONTINUE
AM = -.002517 * (XN(I) / DHJ(I)) ** 2 + .068485 * XN(I) / DHJ(I) + .506994
HC(I) = REJ(I) ** AM
HC(I) = HC(I) * EXP(.02596 * (XN(I) / DHJ(I)) ** 2 - .8259 * XN(I) / DHJ(I) + .3985)
HC(I) = HC(I) / (1. + .4696 * Z)
GO TO 125
AM = -.023152 * (XN(I) / DHJ(I)) ** 2 + .042838 * (XN(I) / DHJ(I)) + .516548
HC(I) = REJ(I) ** AM
HC(I) = HC(I) * EXP(.0126 * (XN(I) / DHJ(I)) ** 2 - .5106 * XN(I) / DHJ(I) - .2057)
HC(I) = HC(I) / (1. + .4215 * Z)
GO TO 125
HC(I) = HC(I) * CONDCT / DHJ(I) * 12.0 * PDTOG ** .33 * (TAU(I) / DHJ(I)) ** .965
GO TO 130
CONTINUE
IF (I.GT.2) HC(I) = HC(I-2)
IF (I.EQ.2) HC(I) = HC(I)
IF (I.EQ.1) HC(I) = HC(3)
CONTINUE
IST = NFWD + 1
DO 150 I = IST, NSTA, 2
IF (IH(I).NE.1) GO TO 155
HC(I) = HC(I-2)
150             CONTINUE
155 IST = NFWD + 2
DO 160 I = IST, NSTA, 2
IF (IH(I).NE.1) GO TO 165
HC(I) = HC(I-2)
160             CONTINUE
165 IF (IGGC.GT.0) WRITE(6, 140) (IGG(I), I = 1, IGGC)
DO 132 I = 1, IREC
ISTRN = IRE(I)
REJOVR(I) = REJ(I)
CONTINUE
132 IF (IREC.GT.0) WRITE(6, 145) (IRE(I), REJOVR(I), I = 1, IREC)
CONTINUE
140 FORMAT(1H' ***** WARNING ***** RATIO OF CROSSFLOW TO ',
          *JET-FLOW IS OUT OF THE RANGE OF ',
          'THE CORRELATION AT THE FOLLOWING STATIONS: 23X,20(I4,'),'))
CONTINUE
145 FORMAT(1H' ***** WARNING ***** JET REYNOLDS NUMBER IS ',
          'OUT OF THE RANGE OF THE CORRELATION ',
          'AT THE FOLLOWING STATIONS: 1X,8(''***',I2,'---',',F8.1,''))
CONTINUE
DO 301 I = 1, NFWD
DUMH2(I) = REJ(I)
301 CONTINUE
RETURN
C---- GENERAL CORRELATION FOR IMPINGEMENT WITH CROSSFLOW IS EVALUATED HERE.
C E
C---- FORM OF CORRELATION IS:
C = CIMP1*(GG**CIMP2)*GI**CIMP3)*((Z/D)**CIMP4)
C *((X/D)**CIMP5)*REJ**CIMP6)*PDTOG**CIMP7
C---- WHERE GG IS THE MASS FLUX RATIO, FREE STREAM TO JET, AND
C---- GI IS THE MOMENTUM FLUX RATIO.
C
400 CONTINUE
DO 450 I = ISTRT,NFWD
WC = ABS(WCHOS(2,ISLICE,I))
II = 5*I
BOINVC = P*TG/(144.*P(2,ISLICE,I))
SKINVC = R*T(2,ISLICE,II)/(144.*R(2,ISLICE,I))
REJ(I) = 0.0
IF (IHC(I).EQ.1) GO TO 403
LCOL = 5*I
LIN = LCOOL - 1
HC(I) = HCFRCD(I,LCOOL,LIN)
GO TO 450
403 CONTINUE
IF (AJET(I).EQ.0.0) GO TO 445
IF (WJ(ISLICE,I).LE.0.) GO TO 445
TMP= (T(2,ISLICE,LIN) +TOG)/2
CALL GASTBL (TMP,C,CP,GAM,PD,R,XXU)
CONDCT = C
XMUTOG = XMU
PDTOG = PD
405 REJ(I)=WJ(ISLICE,I)/AJET(I) *DHJ(I)/(XMUTOG/3600.)*12.?
GG= (WC/A(II))/(WJ(ISLICE,I)/AJET(I))
GI = (WC/A(II))**2*BOINVC/(WJ(ISLICE,I)/AJET(I))**2*KOINVJ
ZOVERD = TAU(I)/DHJ(I)
XOVERD = XN(I)/DHJ(I)
ST = CIMP1*GG**CIMP2)*GI**CIMP3)*ZOVERD**CIMP4)
Z = (*XOVERD**CIMP5)*REJ(I)**CIMP6)*PDTOG**CIMP7
HC(I) = 144.*3600.*ST*CP*WJ(ISLICE,I)/AJET(I)
GO TO 450
445 CONTINUE
IF (I.GT.2) HC(I) = HC(I-2)
IF (I.EQ.2) HC(I) = HC(I-2)
IF (I.EQ.1) HC(I) = HC(I-3)
450 CONTINUE
IST = NFWD+1
DO 460 I = IST,NSTA,2
IF (IHC(I).NE.1) GO TO 465
460 HC(I) = HC(I-2)
465 IST = NFWD+2
DO 470 I = IST,NSTA,2
IF (IHC(I).NE.1) GO TO 475
470 HC(I) = HC(I-2)
475 CONTINUE
C
485 CONTINUE
RETURN
END

C---- SOURCE.NHCPINT
SUBROUTINE HCPINS(IS,DELTAN,LCOOL,LCUP,LIN,LCOOLP,PINS,ZAREA)

84
SOURCE.NHCPINT----

COMMON /PRPS/, CPC(80), GAMS, DP(80), SP(80), RE(80),
     CP(CPC(80), GAMS(80), DMR1(80), DMR2(80))

COMMON /TCO/, ADUMP, BTA, CD, CP,
     GAM, PIM, R, SPAN, TOG,
     WDTMP, WIM, AKE(15,80), AKW(15,80),
     DH(80), DHF(80), DHJ(80),
     DLX(400), FCP(80), HCC(80), HGR(80),
     QSNK(80), BR(80), S(15),
     TG(80), TAU(400), WFC(80),
     WJ(15,80), WCROS(2,15,80), XN(80),
     ICOR, IPM, IHUB, ITIP,
     NFWD, NSTA, IHC(80)

DIMENSION EFAREA(80), DELTAN(15)

COMPUTE THE HEAT TRANSFER COEFFICIENT AND EFFECTIVENESS FOR A
TRIANGULAR ARRAY OF PIN PINS

WHERE DP IS PIN DIAMETER IN INCHES AND SP IS PIN SPACING IN INCHES

VDP = DP(IS)
VSP = SP(IS)

CONTINUE

TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.

CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)

NO. OF PINS AT THIS STATION IS:

PINS = SPAN*DLX(LCOOL)/(.86603*VSP**2)

AVERAGE LENGTH OF PINS:

SLF = (TAU(LCOOL) + TAU(LCUP))/2.

MINIMUM FLOW AREA:

AMIN = SLF*SPAN*(VSP-VDP)/VSP

TOTAL SURFACE AREA:

AHTR = 2.*DLX(LCOOL)*SPAN + 3.14159*PINS*(VDP*SLP-VDP**2/4.)

CHANNEL HYDRAULIC DIAMETER:

DH(IS) = 4.*AMIN*DLX(LCOOL)/AHTR

REDH = 12.*3600.*ABS(WCROS(2,ISLICE,IS))*DH(IS)/(AMIN*XMU)

TERM1 = -.89*(VSP/SLP)**2.5075

TERM2 = -3.094*VDP/VSP

TERM3 = 4.143*EXP(TERM1 + TERM2)/(REDH**.9246)

CONTINUE

HC(IS) = (12.*C/DH(IS))*(.023 + TERM3)*(REDH**.8)*(PD**.333)

EML = SQRT(4.*HC(IS)*SLP*2/(AWK(ISLICE,IS)*VDP))

EFTVNS = TANH(EML)/EML

CHECK LOCATION OF HEAT FLOW SPLIT POINT IF THIS IS A TRAILING
EDGE REGION STATION

IF (IS.LE.NFWD) GO TO 160

TBAR = (T(2,ISLICE,LCOOLP) - T(2,ISLICE,LCOOL))/
     (T(2,ISLICE,LIN) - T(2,ISLICE,LCOOL))

HCOS = COSH(EML)

HYSIN = SINH(EML)

IF (HCOS-TBAR.LT.HYSIN) GO TO 120
WRITE(6, 110) LCOOLP, LIN, DELTAN(ISLICE)

110 FORMAT(1H //" **** WARNING **** NODE", I3, 
Z(1H * IS RECEIVING HEAT FROM NODE", I3,' THROUGH THE PINS.'
Z(1H * RESULTS ARE INVALID. DELTAN =", F7.4)

GO TO 140

120 CONTINUE

IF (HYCOS - TBAR .GT. 0.) GO TO 130

WRITE(6, 110) LIN, LCOOLP, DELTAN(ISLICE)

GO TO 140

130 CONTINUE

XOVRL = (HYCOS - TEAR) / BHSIN

XOVRL = ALOG((1.+XOVRL)/(1.-XOVRL))/(2.*EML)

140 CONTINUE

150 CONTINUE

EFAREA(IS) = DLX(LIN) * SPAN

Z = 3.14159 * PINS * (VDP**2/4. - EFTVNS*VDP*SLP*XOVRL)

IF (IS.GT.NPWD) EFAREA(IS+1) = DLX(LCOOLP) * SPAN

Z = 3.14159 * PINS * (VDP**2/4. - EFTVNS*VDP*SLP* (1.-XOVRL)

EFAREA(IS) = DLX(LIN) * SPAN

Z = 3.14159 * PINS * (VDP**2/4. - EFTVNS*VDP*SLP)

160 CONTINUE

RETURN

END

C---- SOURCE.NINPRTT

SUBROUTINE INPRT(ICHNL, INEDIT)

C SOURCE.NINPRTT----

COHRON /BOUND/ BCXS(400), BCXP(400), BCHGS(1000), BCHGP(100)

Z BCTXS(1000), BCTGP(1000), BCGS(1000), BCQGP(1000)

Z BCPGS(1000), BCQGP(1000), THUN(400), THUN(80)

Z QHUBIN(400), QHUB(80), TTPIN(400), TTIP(80)

Z QTPIN(80), QTIP(80), PTO(50), WPEN

Z WSST(50), AKCTBL(20), AKWTBL(20), NBCS, NBCP

C COMMON /GAAS/ GS(200), NG

C COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80)

Z CPC(80), GMAC(80), DUMR1(80), DUMR2(80)

C COMMON /RIL/ APLN(15), DLPLN(15), RIN(15), ROUT(15)

Z PIN(15), TIN(15), W(15), WS

C COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000)

Z IPLOT, MD1, MD2, MD3, IADJIN, IWRITE

C COMMON /TCO/ ADUMP, BTX, CD, CP

Z GAM, PIN, R, SPAN, TIC, TGO

Z WDUMP, WIR, AKC(15, 80), AKW(15, 80)

Z A(400), AJET(80), AM2(80), CNUM(80)

Z DH(80), DHF(80), DJH(80)

Z DLX(400), FF(80), HC(80), HG(80)

Z P(2, 15, 80), PEXIT(15), PUMP(80), QG(80)

Z QSNK(80), BR(80), S(15), T(2, 15, 400)

Z TG(80), TAU(400), WPC(80)

Z WJ(15, 80), WECOS(2, 15, 80), XN(80)

Z ICON, IFILM, IHUB, ITIP

Z ISBLK, ISLICE, NBLKSZ, NSLICE

Z NFWD, NSTA, INC(80)

C
C INITIALIZE TEMPERATURE DISTRIBUTION (DEGREES R)

C

I = ICHNL
NODSF = 5*NPWD
NODSTM = 5*NSTA - 4
NODST = 5*NSTA

DO 830 I1 = 5, NODSF, 5
LO = I1-4
LJ = I1-3
L = I1-2
LI = I1-1
T(2,1,LO) = .9*TG(IS)
T(2,1,LI) = T(2,1,LO)/1.08

T(2,1,LJ) = T(2,1,LO) - (T(2,1,LO) - T(2,1,LI)) * TAU(LO)/(TAU(LO) + TAU(L))
T(2,1,L) = T(2,1,LO) - (T(2,1,LO) - T(2,1,LI)) * (TAU(LO) + TAU(L)/2.) / (TAU(LO) + TAU(L))

930 T(2,1,11) = TTI0(1) + 460.

DO 860 I1 = ISTRT, NODSTM, 10
T(2,1,I1) = T(2,1,NODSF)
T(2,1,I1+5) = T(2,1,I1)

DO 865 J = 1,4
IPJ = I1 + J
IMJ = I1 + J - 5
IUPP = NODSF + J - 5
T(2,1,IPJ) = T(2,1,IUPP)
T(2,1,IMJ) = T(2,1,IUPP)

860 DO 865 J = 1, NODST
865 T(1,1,J) = T(2,1,J)

C

IF (ICHNL.GT.1) GO TO 94
WRITE (6, 408)
408 FORMAT (1H1, ///, 20X, 'PROPERTY TABLES' ///)
WRITE (6, 410)
410 FORMAT (1H, 'OUTER COATING EFFECTIVE THERMAL CONDUCTIVITY')

C

IF (IUNITS.EQ.1) GO TO 420
WRITE (6, 412) (AKCTBL(I), I=1,19,2)
WRITE (6, 414) (AKCTBL(I), I=2,20,2)
412 FORMAT(//5X,'T, (F) ',10X,10(F9.1))  NINPRTT 2101
414 FORMAT(5X,'K, (BTU/HR/FT/R) ',10(F9.3))  NINPRTT 2102
WHITE(6,416)  NINPRTT 2103
416 FORMAT(//"WALL METAL THERMAL CONDUCTIVITY")  NINPRTT 2104
WRITE(6,412) (AKWTBL(I),I=1,19,2)  NINPRTT 2105
WHITE(6,414) (AKWTBL(I),I=2,20,2)  NINPRTT 2106
C GO TO 445  NINPRTT 2107
C  NINPRTT 2108
420 CONTINUE  NINPRTT 2109
DO 418 I = 1,19,2  NINPRTT 2110
AKCTBL(I) = (AKCTBL(I)+460.)/1.8  NINPRTT 2111
AKWTBL(I) = (AKWTBL(I)+460.)/1.8  NINPRTT 2112
AKCTBL(I+1) = AKCTBL(I+1)/CTCON(1)  NINPRTT 2113
C WRITE(6,422) (AKCTBL(I),I=1,19,2)  NINPRTT 2114
WRITE(6,424) (AKCTBL(I),I=2,20,2)  NINPRTT 2115
422 CONTINUE  NINPRTT 2116
424 FORMAT(5X,'K, (W/M/K)' )  NINPRTT 2117
WRITE(6,448) I = 1,19,2  NINPRTT 2118
AKCTBL(I) = 1.8*AKCTBL(I) - 460.  NINPRTT 2119
AKWTBL(I) = 1.8*AKWTBL(I) - 460.  NINPRTT 2120
AKCTBL(I+1) = AKCTBL(I+1)*CTCON(1)  NINPRTT 2121
C WRITE(6,450) (GS(J),J=1,NGS)  NINPRTT 2122
450 NGS = NG  NINPRTT 2123
IF (NG.GT.10) NGS = 10  NINPRTT 2124
C IF (IUNITS.EQ.1) GO TO 470  NINPRTT 2125
WRITE(6,452) (GS(J),J=1,NGS)  NINPRTT 2126
452 FORMAT(5X,'TEMPERATURE (F)' ,10(F9.1))  NINPRTT 2127
L = NG + 1  NINPRTT 2128
LE = L + 1  NINPRTT 2129
WRITE(6,454) (GS(J),J=L,LE)  NINPRTT 2130
454 FORMAT(5X,'K, (BTU/HR/FT/R)' ,10(F9.5))  NINPRTT 2131
L = 2*NG + 1  NINPRTT 2132
LE = 2*NG + LE  NINPRTT 2133
WRITE(6,456) (GS(J),J=L,LE)  NINPRTT 2134
456 FORMAT(5X,'CP, (BTU/LBM/HR)' ,10(F9.5))  NINPRTT 2135
L = 3*NG + 1  NINPRTT 2136
LE = 3*NG + LE  NINPRTT 2137
WRITE(6,458) (GS(J),J=L,LE)  NINPRTT 2138
458 FORMAT(5X,'PRANDTL NUMBER',10(F9.5))  NINPRTT 2139
L = 4*NG + 1  NINPRTT 2140
LE = 4*NG + LE  NINPRTT 2141
WRITE(6,460) (GS(J),J=L,LE)  NINPRTT 2142
460 FORMAT(5X,'VIS. (LBM/FT/HR)' ,10(F9.5))  NINPRTT 2143
C GO TO 90  NINPRTT 2144
C  NINPRTT 2145
470 CONTINUE  NINPRTT 2146
472 FORMAT(5X,'RALL METAL THERMAL CONDUCTIVITY')  NINPRTT 2147
WRITE(6,470) (AKCTBL(I),I=1,19,2)  NINPRTT 2148
WRITE(6,472) (AKWTBL(I),I=2,20,2)  NINPRTT 2149
472 CONTINUE  NINPRTT 2150
C C  NINPRTT 2151
474 CONTINUE  NINPRTT 2152
476 CONTINUE  NINPRTT 2153
478 CONTINUE  NINPRTT 2154
480 CONTINUE  NINPRTT 2155
88
CONTINUE
DO 471 J = 1, NGS
WRITE(6,472) (DUM1(J), J=1, NGS)
FORMAT(' //5X,' TEMPERATURE (K) ',10 (F9.1))
L = NG + 1
LE = NG + NGS
JI = 0
DO 473 J = L, LE
JI = JI+1
DUM1(JI) = GS(J)/CTCON(1)
WRITE(6,474) (DUM1(J), J=1, JI)
FORMAT(5X,'K, (W/M/K)',10 (F9.5))
L = 2*NG + 1
LE = 2*NG + NGS
JI = 0
DO 475 J = L, LE
JI = JI+1
DUM1(JI) = GS(J)/CSPHT(1)
WRITE(6,476) (DUM1(J), J=1, JI)
FORMAT(5X,'CP, (J/KG/K)',10 (F9.2))
L = 3*NG + 1
LE = 3*NG + NGS
WRITE(6,478) (GS(J), J=JL, LE)
FORMAT(5X,'PRANDTL NUMBER',10 (F9.5))
L = 4*NG + 1
LE = 4*NG + NGS
JI = 0
DO 479 J = L, LE
JI = JI+1
DUM1(JI) = GS(J)/CVISC(1)
WRITE(6,480) (DUM1(J), J=1, JI)
FORMAT(5X,'VIS. (N S/M**2)',10 (F9.5))
C
CONTINUE
IF (INEDIT.EQ.0) GO TO 350
C-- LIST OUT THE INPUT HOT GAS BOUNDARY CONDITIONS--
C--MNBC IS THE MAX OF NBCS & NBCP
C
MNBC = NBCS
IF (MNBC.LT.NBCP) MNBC=NBCP
NTIMES = 1
NTIMES IS THE NUMBER OF TIME STEPS IN BC TABLES
IF (BCTIME(NTIMES+1) .LE.0.0) GO TO 482
NTIMES = NTIMES + 1
GO TO 481
C
CONTINUE
WRITE(6,4820)
FORMAT(1H1,40X,'HOT GAS BOUNDARY CONDITIONS')/n
WRITE(6,483)
FORMAT('  ***************SUCTION SIDE***************',22X,
' ***************PRESSURE SIDE***************')
C
-- SET THE NO. OF POINTS PER TIME STEP IN S&P BC ARRAYS
NPRTS = NSLICE*MNBC
NPRTP = NSLICE*NBCP

C

NL = 3
DO 499 IT = 1,NTIMES
C-- SET THE NO. OF POINTS THAT PRECEDED TIME STEP 'IT'
NPECS = NPRTS*(IT-1)
NPRCP = NPRTP*(IT-1)
C
C--START THE LOOP THROUGH ALL SLICES
DO 499 ISL = 1,NSLICE
C--SET THE NO. OF POINTS PRECEDING THIS SLICE
NBFRS = NPRCS + NBCS*(ISL-1)
NBFRP = NPRCP + NBCP*(ISL-1)
C
NL = NL + 3 + MBBC
IF (NL.LT.60) GO TO 4860
NL = 3 + MBBC
WRITE(6,4820)
WRITE (6,483)
C
C--HERE WE LOOP WITHIN A SLICE
DO 499 IBC = 1,MNBC
IF (IBC.GT.NBCS) GO TO 487
J = NBFRS + IBC
JXS = (ISL-1)*NBCS + IBC
TBCXS = BCXS(JXS)/CINCH(IUNITS)
TBCHGS = BCHGS(J)/CHTC(IUNITS)
TBCTGS = BCTGS(J)
IF (IUNITS.EQ.1) TBCTGS = (BCTGS(J)+60.)/1.8
TBCQGS = BCQGS(J)/CHFLX(IUNITS)
TBCPGS = BCPGS(J)/CPRSR(IUNITS)
WRITE(6,489) TBCXS,TBCHGS,TBCTGS,TBCQGS,TBCPGS
C
IF (IBC.GT.NBCP) GO TO 499
J = NBFRP + IBC
JXP = (ISL-1)*NBCP + IBC
TBCXP = BCXP(JXP)/CINCH(IUNITS)
TBCHGP = BCHGP(J)/CHTC(IUNITS)
TBCTGP = BCTGP(J)
IF (IUNITS.EQ.1) TBCTGP = (BCTGP(J)+460.)/1.8
TBCQGP = BCQGP(J)/CHFLX(IUNITS)
TBCPGP = BCPGP(J)/CPRSR(IUNITS)
IF (IBC.LE.NBCS) WRITE (6,488) TBCXP,TBCHGP,TBCTGP,TBCQGP,TBCPGP
IF (IBC.GT.NBCS) WRITE (6,490) TBCXP,TBCHGP,TBCTGP,TBCQGP,TBCPGP
CONTINUE
C
C
488 CONTINUE
489 FORMAT(2X,F6.2,2F8.1,F12.1,F8.1)
490 FORMAT(2X,F6.2,2F8.1,F12.1,F8.1)
499 CONTINUE
CONTINUE

DO 95 I = 1,200
NFLUID(I) = 0

WRITE(6,150) ICHNL
IF (ICHNL.GT.1) GO TO 101
IF (IHUB.EQ.1) WRITE(6,142)
IF (IHUB.EQ.2) WRITE(6,144)
IF (IHUB.EQ.3) WRITE(6,146)

CONTINUE

IF (ICHNL.LT.NSLICE) GO TO 102
IF (ITIP.EQ.1) WRITE(6,147)
IF (ITIP.EQ.2) WRITE(6,148)
IF (ITIP.EQ.3) WRITE(6,149)
CONTINUE

TRIN = RIN(ICHNL)/CINCH(IUNITS)
TROUT = ROUT(ICHNL)/CINCH(IUNITS)
TAPLN = APLN(ICHNL)/(CINCH(IUNITS)*CINCH(IUNITS))
WRITE(6,103) TRIN, UL(IUNITS),TROUT,UL(IUNITS),
Z

IF (IUNITS.EQ.1) GO TO 500

WRITE(6,153) NFWD,NSTB,SPAN
WRITE(6,155) CD,ADUMP
TEM = TTOI(1) + 460.
WRITE(6,157) TEM,P1O(I),PEX(ICHNL),WPLEN
ITPBG = NFWD + 2
WRITE(6,154) ICHNL,ITRBG
DO 118 I = 1,NSTAB,20

IF (IP18.GT.NSTA) IP18 = NSTA
IF (I.EQ.1) WRITE(6,156) (J,J=I,IP18,2)
IF (I.GT.1) WRITE(6,159) (J,J=I,IP18,2)

ID = 0
DO 104 J = I,IP18,2

ID = ID + 1
DUM(ID) = RP(J)
NFLUID(J) = 5*J
CONTINUE

WRITE(6,158) (NFLUID(J),J=I,IP18,2)

ID = 0
DO 116 J = I,IP18,2

ID = ID + 1
NOS = NFLUID(J) - 4
IF (NOS.GT.1) GO TO 106

XOS = 0.0
XJN = 0.0
XMM = 0.0
XIS = 0.0
XCC = 0.0

GO TO 106

GOS = GOS + DLX(NOS)
XJN = XJN + DLX(NOS+1)
XMM = XMM + DLX(NOS+2)
XIS = XIS + DLX(NOS+3)
XCC = XCC + DLX(NOS+4)

CONTINUE
DUM2(ID) = XOS
DUM25(ID) = XJN
DUM3(ID) = XMM
DUM4(ID) = XIS
DUM5(ID) = XCC
DUM55(ID) = TAU(NOS)
DUM6(ID) = TAU(NOS+2)
NOS = NFLUID(J)
DUM7(ID) = TAU(NOS)
DUM8(ID) = A(NOS)
DUM9(ID) = DH(J)
DUM10(ID) = DHJ(J)
DUM11(ID) = CNUM(J)
DUM12(ID) = AJET(J)
DUM16(ID) = THUBIN(J) - 460.
DUM17(ID) = QHUBIN(J)
DUM18(ID) = TTIPIN(J) - 460.
DUM19(ID) = QTIPIN(J)
NOS = NFLUID(J) - 4
DUM13(ID) = TG(J) - 460.
DUM14(ID) = HG(J)
JHCAL = IHC(J)
DUM15(ID) = HCAT(JHCAL)
IF (BTA.GT.0.0) DUM14(ID) = 0G(J)
116 CONTINUE
WRITE (6,160) (DUM1(J),J=1,ID)
WRITE (6,162) (DUM2(J),J=1,ID)
WRITE (6,163) (DUM25(J),J=1,ID)
WRITE (6,164) (DUM3(J),J=1,ID)
WRITE (6,166) (DUM4(J),J=1,ID)
WRITE (6,168) (DUM5(J),J=1,ID)
WRITE (6,169) (DUM55(J),J=1,ID)
WRITE (6,170) (DUM6(J),J=1,ID)
WRITE (6,172) (DUM7(J),J=1,ID)
WRITE (6,174) (DUM8(J),J=1,ID)
WRITE (6,176) (DUM9(J),J=1,ID)
WRITE (6,178) (DUM10(J),J=1,ID)
WRITE (6,180) (DUM11(J),J=1,ID)
WRITE (6,182) (DUM12(J),J=1,ID)
WRITE (6,183) (DUM15(J),J=1,ID)
WRITE (6,184) (DUM13(J),J=1,ID)
IF (BTA.LT.0.0) WRITE (6,186) (DUM14(J),J=1,ID)
IF (ICHNL.EQ.1) GO TO 118
IF (ICHCNLT.GT.1) WRITE (6,196) (DUM16(J),J=1,ID)
IF (ICHCNLT.EQ.3) WRITE (6,198) (DUM17(J),J=1,ID)
IF (ITIP.EQ.1) WRITE (6,202) (DUM18(J),J=1,ID)
IF (ITIP.EQ.3) WRITE (6,204) (DUM19(J),J=1,ID)
116 CONTINUE
ITRBG = NFWD + 1
WRITE (6,190) ICHNL,ITRBG
XOS = 0.0
XJN = 0.0
XMM = 0.0
XIS = 0.0
XCC = 0.0
DO 140 I = 2,NSTA,20
IP18 = I * 19
IF (IP18.GT.NSTA) IP18 = NSTA - 1
IF (I.EQ.2) WRITE(6,156) (J,J=I/IP18,2)
IF (I.GT. 2) WRITE(6,159) (J, J=I, IP18, 2)
ID = 0
DO 122 J = I, IP18, 2
ID = ID + 1
DUM1(ID) = RR(J)
NFLUID(J) = 5*J
CONTINUE
WRITE(6,158) (NFLUID(J), J=I, IP18, 2)
ID = 0
DO 130 J = I, IP18, 2
ID = ID + 1
NOS = NFLUID(J) - 4
XOS = XOS + DLX(NOS)
XJN = XJN + DLX(NOS+1)
XMM = XMM + DLX(NOS+2)
XIS = XIS + DLX(NOS+3)
XCC = XCC + DLX(NOS+4)
DUM2(ID) = XOS
DUM3(ID) = XMM
DUM4(ID) = XIS
DUM5(ID) = XCC
DUM55(ID) = TAU(NOS)
DUM6(ID) = TAU(NOS+2)
NOS = NFLUID(J)
DUM7(ID) = TAU(NOS)
DUM3(ID) = A(NOS)
DUM9(ID) = DH(J)
DUM10(ID) = DHJ(J)
DUM11(ID) = CNUM(J)
DUM12(ID) = AJET(J)
DUM16(ID) = THUBIN(J) - 460.
DUM17(ID) = QHUBIN(J)
DUM18(ID) = TTPIN(J) - 460.
DUM19(ID) = QTIPIN(J)
NOS = NFLUID(J) - 4
DUM13(ID) = TG(J) - 460.
DUM14(ID) = HG(J)
JHCAL = IHC(J)
DUM15(ID) = HCAL(JHCAL)
IF (BTA.GT.0.01) DUM14(ID) = QG(J)
CONTINUE
WRITE(6,160) (DUM1(J), J=1, ID)
WRITE(6,162) (DUM2(J), J=1, ID)
WRITE(6,163) (DUM25(J), J=1, ID)
WRITE(6,164) (DUM3(J), J=1, ID)
WRITE(6,166) (DUM4(J), J=1, ID)
WRITE(6,168) (DUM5(J), J=1, ID)
WRITE(6,169) (DUM55(J), J=1, ID)
WRITE(6,170) (DUM6(J), J=1, ID)
WRITE(6,172) (DUM7(J), J=1, ID)
WRITE(6,174) (DUM8(J), J=1, ID)
WRITE(6,176) (DUM9(J), J=1, ID)
WRITE(6,178) (DUM10(J), J=1, ID)
WRITE(6,180) (DUM11(J), J=1, ID)
WRITE(6,182) (DUM12(J), J=1, ID)
WRITE(6,183) (DUM15(J), J=1, ID)
WRITE(6,184) (DUM13(J), J=1, ID)
IF (BTA.LT.0.01) WRITE(6,186) (DUM14(J), J=1, ID)
IF (BTA.GT.0.01) WRITE(6,188) (DUM14(J), J=1, ID)
IF (ICHNL.GT.1) GO TO 140
IF (IHUB.EQ.1) WRITE(6,196) (DUM16(J),J=1,ID)
IF (IHUB.EQ.3) WRITE(6,198) (DUM17(J),J=1,ID)
IF (ITIP.EQ.1) WRITE(6,202) (DUM18(J),J=1,ID)
IF (ITIP.EQ.3) WRITE(6,204) (DUM19(J),J=1,ID)
140 CONTINUE

150 FORMAT(1H1,46X,'INPUT FOR SLICE NUMBER',I3)
152 FORMAT(21X,'HUB TEMPERATURES ARE SPECIFIED')
154 FORMAT(21X,'ADIABATIC HUB SPECIFIED')
156 FORMAT(21X,'HUB HEAT FLUX IS SPECIFIED')
158 FORMAT(21X,'TIP TEMPERATURES ARE SPECIFIED')
160 FORMAT(21X,'ADIABATIC TIP SPECIFIED')
162 FORMAT(1H1,46X,'INPUT FOR SLICE NUMBER',I3)
164 FORMAT(21X,'TOTAL NUMBER OF STATIONS IS',I3)
166 FORMAT(21X,'SPAN OF THIS SLICE IS',F6.3,' IN')
168 FORMAT(21X,'IMPELLER HOLES DISCHARGE COEF.=',F6.3)
170 FORMAT(21X,'COOLANT INLET TEMP.=',F7.1,' R, COOLANT INLET PRESSURE=',F8.5)
PTIOC = PTIO(1)/CPBR(1)
PEXC = PEX(ICHNL)/CPBR(1)
WPLENC = WLEN/CMSFL(1)
WRITE(6,557) TEM,PTIOC,PEXC,WPLENC
ITRBG = NFYD + 2
WRITE(6,154) ICHNL,ITRBG
DO 518 I = 1,NSTA,20
IP18 = I + 18
IF (IP18.GT.NSTA) IP18 = NSTA
IF (I.EQ.1) WRITE(6,556) (J,J=I,IP18,2)
IF (I.GT.1) WRITE(6,559) (J,J=I,IP18,2)
ID = 0
DO 504 J = I,IP18,2
ID = ID + 1
DUM1(ID) = RR(J)/CINCH(1)
NFLUID(J) = 5*J
504 CONTINUE
WRITE(6,558) (NFLUID(J),J=I,IP18,2)
ID = 0
DO 516 J = I,IP18,2
ID = ID + 1
NOS = NFLUID(J) - 4
IF (NOS.GT.1) GO TO 506
XOS = 0.0
XJN = 0.0
XMM = 0.0
XIS = 0.0
XCC = 0.0
GO TO 508
506 XOS = XOS + DLX(NOS)/CINCH(1)
XJN = XJN + DLX(NOS+1)/CINCH(1)
XMM = XMM + DLX(NOS+2)/CINCH(1)
XIS = XIS + DLX(NOS+3)/CINCH(1)
XCC = XCC + DLX(NOS+4)/CINCH(1)
508 CONTINUE
DUM2(ID) = XOS
DUM25(ID) = XJN
DUM3(ID) = XMM
DUM4(ID) = XIS
DUM5(ID) = XCC
DUM55(ID) = TAU(NOS)/CINCH(1)
DUM6(ID) = TAU(NOS+2)/CINCH(1)
NOS = NFLUID(J)
DUM7(ID) = TAU(NOS)/CINCH(1)
DUM8(ID) = A(NOS)/(CINCH(1)**2)
DUM9(ID) = DH(J)/CINCH(1)
DUM10(ID) = DHJ(J)/CINCH(1)
DUM11(ID) = CNUM(J)
DUM12(ID) = AJET(J)/(CINCH(1)**2)
DUM16(ID) = THUBIN(J)/1.8
DUM17(ID) = QHUBIN(J)/CHFLX(1)
DUM18(ID) = TTIPIN(J)/1.8
DUM19(ID) = QTIPIN(J)/CHFLX(1)
NOS = NFLUID(J) - 4
DUM13(ID) = TG(J)/1.8
DUM14(ID) = HG(J)/CHTC(1)
JHCAL = IHC(J)
DUM15(ID) = HCAL(JHCAL)
IF (BTA.GT..01) DUM14(ID) = QG(J)/CHFLX(1)
WRITE (6,560) (DUM1(J),J=1,ID) WRITE (6,561) (DUM2(J),J=1,ID) WRITE (6,562) (DUM25(J),J=1,ID) WRITE (6,563) (DUM3(J),J=1,ID) WRITE (6,564) (DUM4(J),J=1,ID) WRITE (6,565) (DUM5(J),J=1,ID) WRITE (6,566) (DUM6(J),J=1,ID) WRITE (6,569) (DUM55(J),J=1,ID) WRITE (6,570) (DUM6(J),J=1,ID) WRITE (6,571) (DUM7(J),J=1,ID) WRITE (6,572) (DUM8(J),J=1,ID) WRITE (6,573) (DUM9(J),J=1,ID) WRITE (6,574) (DUM10(J),J=1,ID) WRITE (6,575) (DUM11(J),J=1,ID) WRITE (6,576) (DUM12(J),J=1,ID) WRITE (6,577) (DUM13(J),J=1,ID) WRITE (6,578) (DUM14(J),J=1,ID) WRITE (6,579) (DUM15(J),J=1,ID) WRITE (6,580) (DUM16(J),J=1,ID) WRITE (6,581) (DUM17(J),J=1,ID) WRITE (6,582) (DUM18(J),J=1,ID) WRITE (6,583) (DUM19(J),J=1,ID) WRITE (6,584) (DUM20(J),J=1,ID) WRITE (6,585) (DUM21(J),J=1,ID)

IF (BTA.LT. .01) WRITE(6,587) (DUM14 (J) ,J=1,ID) IF (BTA.GT. .01) WRITE(6,588) (DUM14 (J) ,J=1,ID)

IF (ICHNL.GT.1) GO TO 518

IF (ICHNL .EQ. 1) WRITE (6,596) (DUM16 (J) ,J=1,ID)

IF (ICHNL .EQ. 3) WRITE (6,597) (DUM17 (J) ,J=1,ID)

IF (ITIP.EQ.1) WRITE (6,600) (DUM18 (J) ,J=1,ID)

IF (ITIP.EQ.3) WRITE (6,601) (DUM19 (J) ,J=1,ID)

IF (ITIP.EQ.3) WRITE (6,602) (DUM18 (J) ,J=1,ID)

IF (ITIP.EQ.3) WRITE (6,603) (DUM19 (J) ,J=1,ID)

IF (ITIP.EQ.1) WRITE (6,604) (DUM20 (J) ,J=1,ID)

IF (ITIP.EQ.3) WRITE (6,605) (DUM21 (J) ,J=1,ID)

CONTINUE

ITRBG = NFWD + 1
WRITE(6,190) ICHNL,ITRBG
XOS = 0.0
XJN = 0.0
XMM = 0.0
XIS = 0.0
XCC = 0.0
DO 540 I = 2,NSTA,2

IP18 = I + 18

IF (IP18.GT.NSTA) IP18 = NSTA-1

IF (I.EQ.2) WRITE(6,556) (J,J=I,IP18,2)

IF (I.EQ.2) WRITE(6,557) (J,J=I,IP18,2)

ID = 0
DO 522 J = I,IP18,2

ID = ID + 1

DUM1(ID) = RR(J)/CINCH(1)

NFLUID(J) = 5*J

CONTINUE

WRITE (6,558) (NFLUID(J),J=I,IP18,2)

ID = 0

DO 530 J = I,IP18,2

ID = ID + 1

NOS = NFLUID(J) - 4

XOS = XOS + DLX(NOS)/CINCH(1)

XJN = XJN + DLX(NOS+1)/CINCH(1)

XMM = XMM + DLX(NOS+2)/CINCH(1)

XIS = XIS + DLX(NOS+3)/CINCH(1)

XCC = XCC + DLX(NOS+4)/CINCH(1)

DUM2(ID) = XOS

DUM25(ID) = XJN

DUM3(ID) = XMM

DUM4(ID) = XIS

DUM5(ID) = XCC

DUM55(ID) = TAU(NOS)/CINCH(1)

DUM6(ID) = TAU(NOS+2)/CINCH(1)

NOS = NFLUID(J)
DUM7(ID) = TAU(NOS)/CINCH(1)
DUM8(ID) = A(NOS)/CINCH(1)**2
DUM9(ID) = DH(J)/CINCH(1)
DUM10(ID) = DHJ(J)/CINCH(1)
DUM11(ID) = CNUM(J)
DUM12(ID) = AJET(J)/CINCH(1)**2
DUM16(ID) = THUBN(J)/1.8
DUM17(ID) = QHUBN(J)/CHFLX(1)
DUM18(ID) = TTIPIN(J)/1.8
DUM19(ID) = QTIPIN(J)/CHFLX(1)
NOS = NFLUID(J) - 4
DUM13(ID) = TG(J)/1.8
DUM14(ID) = AG(J)/CHTC(1)
JHCAL = IHC(J)
DUM15(ID) = HCAL(JHCAL)
IF (BTA.GT.0.01) DUM14(ID) = QG(J)/CHFLX(1)
CONTINUE
WRITE(6,560) (DUM1(J),J=1,ID)
WRITE(6,562) (DUM2(J),J=1,ID)
WRITE(6,563) (DUM25(J),J=1,ID)
WRITE(6,564) (DUM3(J),J=1,ID)
WRITE(6,566) (DUM4(J),J=1,ID)
WRITE(6,568) (DUM5(J),J=1,ID)
WRITE(6,569) (DUM55(J),J=1,ID)
WRITE(6,570) (DUM6(J),J=1,ID)
WRITE(6,572) (DUM7(J),J=1,ID)
WRITE(6,574) (DUM8(J),J=1,ID)
WRITE(6,576) (DUM9(J),J=1,ID)
WRITE(6,578) (DUM10(J),J=1,ID)
WRITE(6,580) (DUM11(J),J=1,ID)
WRITE(6,582) (DUM12(J),J=1,ID)
WRITE(6,583) (DUM15(J),J=1,ID)
WRITE(6,584) (DUM13(J),J=1,ID)
IF (BTA.LT.0.01) WRITE(6,586) (DUM14(J),J=1,ID)
IF (BTA.GT.0.01) WRITE(6,588) (DUM14(J),J=1,ID)
IF (IHUB.EQ.1) GO TO 540
IF (IHUB.EQ.3) WRITE(6,598) (DUM17(J),J=1,ID)
IF (IITIP.EQ.1) WRITE(6,602) (DUM18(J),J=1,ID)
IF (IITIP.EQ.3) WRITE(6,604) (DUM19(J),J=1,ID)
CONTINUE
FORMAT('/' NUMBER OF STATIONS IN IMPINGEMENT REGION IS',I3,
Z ' , TOTAL NUMBER OF STATIONS IS',I3,
Z ' , SPAN OF THIS SLICE IS',F6.3,' CM'
553 FORMAT(' IMPINGEMENT HOLE DISCHARGE COEF.=',F6.3,
Z ' , AREA OF DUMP TO TRAILING EDGE =',F8.5,' CM**2')
554 FORMAT(' COOLANT INLET TEMP.=',F7.1,' K, COOLANT INLET',
Z ' , PRESSURE =',F7.1,' KPA, EXIT PRESSURE =',
Z ' , F7.1,' KPA, Kg/Hr')
555 FORMAT(' STATION NUMBER',5X,10(6X,I4))
556 FORMAT(' COOLANT NODE NUMBER',10,6X,I4)
557 FORMAT(' RADIAL LOCATION(CM)',10F10.3)
558 FORMAT(' X, OUTSIDE SUR. (CM)',10F10.5)
559 FORMAT(' X, INTERFACE (CM)',10F10.5)
560 FORMAT(' X, MID-METAL (CM)',10F10.5)
561 FORMAT(' X, INSIDE SURF. (CM)',10F10.5)
562 FORMAT(' X, MID.COOL.CH. (CM)',10F10.5)
563 FORMAT(' COATING THKNSS (CM)',10F10.5)
564 FORMAT(' WALL THICKNESS (CM)',10F10.5)
C----SOURCE.NPARAYT
SUBROUTINE PARRAY(JS,JSENS,ICHOKE)

REAL*8 TCOP

COMMON /MATRX/ TCOF(400,30)

COMMON /PRPS/ CPC(80), GAMO, DP(80), SP(80), RE(80),
Z CPC(80), GAMC(80), DUMR1(80), DUMR2(80)

COMMON /TCO/ ADUMP, BTA, CD, CP, 
Z GAM, PIM, R, SPAN, TOG, 
Z WDUMP, WIM, AKC(15,80), AKW(15,80),
Z A(400), AJET(80), AN2(80), CNUM(80),
Z DH(80), DHP(80), DJH(80),
Z DLX(400), FF(80), HC(80), HG(80),
Z P(2,15,80), PEXIT(15), PUMP(80), QG(80),
Z QSNK(80), RR(80), S(15), T(2,15,400),
Z TG(60), TAU(400), WFC(80),
Z WJ(15,80), WCROS(2,15,80), XN(80),
Z IOR, IFILM, IHUB, ITIP, 
Z ISBLOK, ISLICE, NBLKSZ, NSLICE, 
Z NFWD, NSTA, IHC(80)

COMMON /TRANST/ RHOC, RHOM, SPHTC, SPHTM,
Z DLTIME, TYME, TEEPS, TEEPSMAX

DIMENSION POLD(80), PSAV(5)

COPUTE NEW PressURES

C IFNL = THE NUMBER OF FLOW CHANNEL NODES
C TREFS = 1.0
IF (TIME.GE.0.) TREFS = TEEPS
800 IFNL = NSTA - 3
NODST = 5*NSTA
NODSF = 5*NFWD
C INITIALIZE COEFFICIENT ARRAY TO 0.0
C DO 810 I = 1,IPNL
DO 810 J = 1,30
810 TCOF(I,J) = 0.0
C COMPUTE THE COEFFICIENT VALUES
C DO 900 I = 1,IPNL
FILN = 0.0
820 TCOF(I,J) = 0.0
C FOR THE IMPINGEMENT REGION:
C ICHK = 0 IMPLIES I IS EVEN AND STATION IS ON SUCTION SIDE
C = 1 IMPLIES THAT I IS ODD AND STATION IS ON PRESSURE SIDE
C DEFINE THE REAL NODE NUMBER IN TERMS OF I
C WHERE IRL IS THE PIVOTAL ELEMENT = COOLANT NODE NUMBER, LC00L
C IDN = DOWNSTREAM COOLANT NODE
C IUP = UPSTREAM COOLANT NODE
C IF (I.LT.NFWD) GO TO 840
IF (I.EQ.NFWD) GO TO 890
C FOR I=NFWD, THE NODE IS THE ENTRANCE TO THE TRAILING EDGE AND IS
C TREATED SEPARATELY AT (890)
C FOR I>NFWD, THE NODE IS IN THE TRAILING EDGE AND IRL IS DEFINED AS:
C IF (ICHK.GT.0) GO TO 885
IRL = 5*I
IDN = IRL + 10
IDNS = I+2
IUP = IRL
IUPS = IDNS - 2
ITC = 10
ITCP = 12
830 CONTINUE
GO TO 860
C 840 CONTINUE
IRL = 5*I
IF (I.GT.JS) GO TO 843
C IF (I.LT.JS) GO TO 852
C IF (ICHK.GT.0) GO TO 849
GO TO 855
C 843 IF (ICHK.GT.0) GO TO 849
C STATION I IS SUCTION SIDE, DOWNSTREAM OF SPLIT POINT
C IUPS = I - 2
IDNS = I
IUP = IRL - 10
IDN = IRL
ITC = 8
ITCP = 10
IF (I.GT.2) GO TO 860
IUPS = 1
IUP = 5
ITC = 9
GO TO 860
C
C STATION I IS PRESSURE SIDE, DOWNSTREAM OF SPLIT POINT
C
849 CONTINUE
IUPS = I
IDNS = I + 2
IUP = IRL
IDN = IRL + 10
ITC = 10
ITCP = 8
GO TO 860
852 CONTINUE
IF (ICHK.GT.0) GO TO 858
IF (1CHK.NE.JSENS) GO TO 846
855 CONTINUE
C
C I IS ON SUCTION SIDE, FORWARD OF SPLIT POINT
C
IUPS = I
IDNS = I - 2
IUP = IRL
IDN = IRL - 10
IDX = IUP
ITC = 10
ITCP = 12
GO TO 860
858 CONTINUE
IF (ICHK.NE.JSENS) GO TO 849
IDNS = I
IUPS = I + 2
IUP = IRL
IDN = IRL + 10
IDX = IUP
ITC = 12
ITCP = 10
GO TO 860
C
860 CONTINUE
C
TRTRM = 0.0
IF (DLYME.GT.0.0.AND.TYME.GE.0.) TRTRM = 12.*DLX(IDN) *
Z ((WCRos(2,ISLICE,IDNS) - WCRos(1,ISLICE,IDNS))/DLYME*A(IUP)*32.2)
WPCDUM = WFC(IDNS)
IF (I.GT.NFWD) WPCDUM = WPCDUM + WFC(IDNS+1)
IF(WCRos(2,ISLICE,IDNS).NE.0.0) FILM = WPCDUM/WCRos(2,ISLICE,IDNS)
TCOP(I,ITC) = TREPS*
((1.0 + GANC(IUPS)*AM2(IUPS)) + (A(IDN)-A(IUP))/(2.*A(IUP)))
TCOP(I,ITCP) = TREPS*(-(1.0 + 0.5*GANC(IDNS)*AM2(IDNS)) *
+ (A(IDN)-A(IUP))/(2.*A(IUP)))
Z = (4.*FF(IDNS)*DLX(IDX)/DH(IDNS)) * FILM) * A(IDN)/A(IUP)
PRINT 2801
ROOT = SQRT(32.2*GAMC(IDNS)*R*T(2,ISLICE,IDN)*AM2(IDNS))

IF (ROOT.NE.0.0) PUMTRM = (3.14159265*WS/30.)***2
Z = RR(IDNS)*(RR(IDNS)-RR(IUPS))*WCROS(2,ISLICE,IDNS)
Z = (A(IUP)*ROOT*144.*32.2)

TCOF(I,20) = -PUMTRM + TRTRM - (1.-TREPS)*
Z = (P(I,ISLICE,IUPS)*TCOF(I,ITC) + P(I,ISLICE,IDNS)*TCOF(I,ITCP))

CONTINUE

IF (IDNS.NE.ICH0KE) GO TO 880

TCOF(I,20) = -PEXIT(ISLICE)*TCOF(I,12) + TCOF(I,20)

CONTINUE

C FOR TRAILING EDGE CHANNELS:
C IF (I.LT.IFNL) GO TO 900
C
TCOF(I,20) IS NON-ZERO ONLY FOR I=IFNL
C
IF (ICH0KE.EQ.NSTA-1) GO TO 900
TCOF(I,20) = -PEXIT(ISLICE)*TCOF(I,12) + TCOF(I,20)
TCOF(I,12) = 0.0
GO TO 900

CONTINUE

C FOR A PRESSURE SIDE, TRAILING EDGE REGION STATION, COOLANT NODE
C IS IDENTICAL TO SUCTION SIDE NODE.

TCOF(I,10) = 1.0
TCOF(I,9) = -1.0
TCOF(I,20) = 0.0
GO TO 900

CONTINUE

C FOR THE SPECIAL NODE AT THE ENTRANCE TO THE TRAILING EDGE:
C ALLOWING FOR THE POSSIBILITY OF ADDITION OF EXTRA COOLING AIR
C INTO TRAILING EDGE,
C
TRTRM = 0.0

IF (DLTYNE.GT.0.0.AND.TYNE.GE.0.) TRTRM=12.*DLX(NFWD+1) *
Z = (WCROS(2,ISLICE,NFWD+1)-WCROS(1,ISLICE,NFWD+1))/
Z = (DLTYNE=A(NODSF+5)*32.2)
AVRGA = (A(NODSF-5) + A(NODSF) - A(NODSF+5))/(3.*A(NODSF+5))
TCOF(I,9) = TREPS*((1. + GAMC(NFWD-1)*AM2(NFWD-1))*
Z = A(NODSF-5)/A(NODSF+5) - AVRGA)
TCOF(I,10) = TREPS*((1. + GAMC(NFWD)*AM2(NFWD))*
Z = A(NODSF)/A(NODSF+5) - AVRGA)
IF (WCROS(2,ISLICE,NFWD+1).NE.0.0) PUMTRM =
Z = (WFC(NFWD+1)+WFC(NFWD+2))/WCROS(2,ISLICE,NFWD+1)
TCOF(I,11) = TREPS*(-1. - GAMC(NFWD+1)*AM2(NFWD+1))
Z = (1. + 2.*FF(NFWD+1)*DLX(NODSF+5)/DH(NFWD +1)*PUMTRM - AVRGA)

PUMP(NFWD+1) = (3.14159265*WS/30.)***2
Z = RR(NFWD+1)*(RR(NFWD+1)-RR(NFWD))
ROOT = SQRT(32.2*GAMC(NFWD+1)*R*
Z = T(2,ISLICE,NODSF+5)*AM2(NFWD+1))

PUMTRM = 0.0

IF (ROOT.NE.0.0) PUMTRM = (3.14159265*WS/30.)***2*RR(NFWD+1)*
Z = (RR(NFWD+1)-RR(NFWD))*WCROS(2,ISLICE,NFWD+1)
Z /(A(NODS*5)*ROOT*144.*32.2)

C PUMP HAS UNITS OF (IN**2/SEC**2): ROOT HAS UNITS OF (FT/SEC)

C DUMNER = 0.0
IF (ADUMP.GT.0.) DUMNER = - WDUMP**2*R*
Z (T(2,ISLICE,NODSF-5)+T(2,ISLICE,NODSF))/
Z (P(1,ISLICE,NFWD-1)+P(1,ISLICE,NFWD))*ADUMP*A(NODS+5)*32.2)
TCOP(I,20) = - PUMTRM + DUMTR + TRTR
Z - (1.0-TRPS)*P(1,ISLICE,NFWD-1)*TCOP(I,9)
Z + P(1,ISLICE,NFWD)*TCOP(I,10)+P(1,ISLICE,NFWD+1)*TCOP(I,11))
900 CONTINUE
C RETURN
END

C---- SOURCE.NPLENMP

SUBROUTINE PLNUM(WXX,PXX,PTEXIT,TXX,TTEXIT)

C SUBROUTINE TO COMPUTE PRESSURE DROP IN THE CENTRAL COOLANT PLENUM

ARGUMENTS FOR THIS SUBROUTINE ARE:

WXX = MASS FLOW RATE INTO THIS SLICE, LBM/HR
PXX = AVERAGE STATIC PRESSURE FOR THIS PLENUM SLICE, PSIA,
CALCULATED IN THIS SUBROUTINE.
PTEXIT = TOTAL PRESSURE IN AND TOTAL PRESSURE OUT, PSIA
TXX = AVERAGE STATIC TEMPERATURE FOR THIS PLENUM SLICE, (F),
CALCULATED IN THIS SUBROUTINE.
TTEXIT = TOTAL TEMP. IN AND TOTAL TEMP. OUT, (F)

COMMON /RADL/ APLN(15), DPLN(15), RIN(15), ROUT(15),
Z FIN(15), TIN(15), W(15), WS

COMMON /TCO/ ADUMP, ETA, CD, CP,
Z GAM, PIM, R, SPAN, TOG,
Z WDIUMP, WIM, AKC(15,80), AKW(15,80),
Z A(400), AJET(80), AM2(80), CNUM(80),
Z DH(80), DHF(80), DJW(80),
Z DLX(400), FF(80), HC(80), HG(80),
Z P(2,15,80), PEXIT(15), PUMP(80), QG(80),
Z QSNK(80), RR(80), S(15), T(2,15,400),
Z TG(80), TAU(400), WFC(80),
Z WJ(15,80), WCROS(2,15,80), XM(80),
Z ICOR, IFILM, IHUB, ITIP,
Z ISBLK, ISLICE, NBLKsz, NSLICE,
Z NFWD, NSTA, IH(80)

COMMON /TRNSNT/ RHOC, RHON, SPHTC, SPHTM,
Z DLYTIME, TME, TESS, TMMAX

COMMON /UNITS/ CINCH(2), CHTC(2), CHFLX(2), CPRSR(2), CNPSFL(2),

102
DIMENSION BETTA (20), P (20), AMC (20), SIGMA (20), TT1 (20), F1 (20), Z1 (15), CH (15)

C

C---PUNP(PP) IS THE EQUATION FOR DELTA P OVER LENGTH DX

C

FUNP(PP) = (D1*RRP*(PP/R/TP-V1)/144.0 - 2.*PP*G2*TP/PP/AA/AA/DD)*Z
          /[1.-TP*G2/(PP*AA)**2+G1*CP]
          *(778.161*R*TP*V1/(PP*AA)**2/PP)*DX

C

C---PUNT(XK) IS THE EQUATION FOR DELTA T OVER LENGTH DX

C

FUNT(XK) = (DZ*REP/CP+G1*R*(TP/PP/AA)**2*(XK/PP/DX))
          /[1.+G1*R*(TP/PP/AA)**2]*DX

C

INITIALIZE

1 CONTINUE

DIFTOL=0.005

ACH=1.

KSIG=0

NCC=1

IS=0

KTR1=0

W(ISLICE) = WXX

C

SAVE INLET TOTAL PRESSURE (PSIA) IN PIN AND INLET TOTAL TEMPERATURE (F) IN TIN

C

PIN(ISLICE) = PTEXIT

TIN(ISLICE) = TTEXIT

CH(ISLICE) = 0.0

ZED=Z1(ISLICE)*1.01

IF (ZED.EQ.0.) ZED=.001

IF (TIN(ISLICE).GT.-430.0) GO TO 5

3 TIN(ISLICE)=50.0

SIGB=0.0

C

7 NSTNS= 4

SEGMTS=NSTNS-1

C

T1=TIN(ISLICE)+460.0

B(1)=T1

BETA1=PIN(ISLICE)**2

BETA(1)=BETA1

DX=S(ISLICE)/SEGMTS

DXTMP=DX

XXN=NSTNS

C

13 DR=(ROUT(ISLICE)-RIN(ISLICE))/SEGMTS

C

CCOMPUTE CONSTANT TERMS-C1-C8

17 CONTINUE

TTX=B(1)
CALL GASTBL (TTX, C, CP, GAM, PD, R, XMU)

J=1
C6=.5*(GAM-1.0)
C1=GAM/C6

IF (WS) 21,19,21
C NO PUMPING
19 C3=0.0
GO TO 23
C PUMPING
21 C3=2.36695E-6*(WS**2)/(C1*R)
23 C8=32.17*GAM*R
C5=1.0/SQRT(C8)
C7=1.0/(32.17*C1*R)
IF (J.GT.1) GO TO 33
25 CONTINUE
C COMPUTE CHANNEL REYNOLDS NO. IF J = 1
C
REY = 12.0*W(ISLICE)/XMU*DPLN(ISLICE)/APLN(ISLICE)
C COMPUTE FRICTION FACTOR
C COMPUTE 2 TERMS
33 CONTINUE
Z3=12.0*W(ISLICE)/XMU
Z4=(R*W(ISLICE)/3600.0)**2
IF (J.GT.1) GO TO 77
C DETERMINE INLET CONDITIONS
35 CONTINUE
C INITIAL STATION COMPUTATIONS -
C BALANCING ON TOTAL PRESSURE -
39 NAG=-1
41 SIGC=(B(J)/APLN(ISLICE))**2*Z4/BETTA(J)
IF (ABS(SIGB-SIGC).LE.0.00001*SIGC) GO TO 57
C SIGMA NOT CONVERGED
43 IS=IS+1
SV(IS)=SIGC
IF (IS.EQ.3) GO TO 135
45 B(J)=T1-C7*SIGC
IF (B(J).LT.50.0) GO TO 159
C TEMP OK
47 SIGB=SIGC
BETTA(J)=BETAI*(B(J)/T1)**C1
GO TO 41
C SIGMA CONVERGED
57 B(J)=T1-C7*SIGC
AMC(1)=SQRT(SIGC/B(J))*C5
IF (B(J).LE.0.0) GO TO 159
KTRBZ=0
63 BETTA(J)=BETAI/(1.0+C6*AMC(1)**2) **C1
IF (BETTA(J).LE.0.) GO TO 159
65 B(1)=T1/(1.0+C6*AMC(1)**2)
SIGMA(1) = (B(J)/APLN(ISLICE))**2*Z4/BETTA(J)
SIGC = SQRT(SIGMA(1)/B(1))*C5
IF (ABS(SIGC-AMC(1)).LE.0.01) GO TO 71
AMC(1) = SIGC
KTRBZ = KTRBZ + 1
IF (KTRBZ.LT.20) GO TO 63
TT1(1) = TIN(ISLICE)

C CHANNEL PRESSURE DROP -

73 NAG = 1
F1(1) = REY
SIGMA(2) = SIGMA(1)*.95
J = 2
IS = 0
CONTINUE
AJ = J - 1
DBR2 = DBR*(2.0*(RIN(ISLICE)*AJ*DBR) - DBR)
TT1(J) = TT1(J-1) + C3*DR2
KSIG = 0
AZ = DPLN(ISLICE)/APLN(ISLICE)
REY = Z3*AZ
PP = SQRT(BETTA(1))
SIGMA(1) = SQRT(SIGMA(1))
BETTA(1) = PP
TP = B(1)
G2 = (W(ISLICE)/3600.0)**2*R/32.174
G1 = G2/CP/778.161
D1 = (WS*3.1415927/30.0)**2*DR/DX/32.174
D2 = D1/778.161/144.0
RRP = RIN(ISLICE)
DO 97 J = 2, NSTNS
AZ = DPLN(ISLICE)/APLN(ISLICE)
REY = Z3*AZ
F1(J) = .079*REY**(-.25)
IF (REY.LT.2300.) F1(J) = 16.0/REY
PTEMP = BETTA(J-1)
TTTEMP = D(J-1)
RTEMP = RRP
MACH1 = 1
XNN = 2.0
DD = DPLN(ISLICE)
AA = APLN(ISLICE)
GO TO 85
81 MACH1 = XNN
WRITE(6,83) ISLICE, J

83 FORMAT(5X,8H***,50HDECREASED INCREMENT DERIVATIVE CHANGING 1 TOO FAST,3X,'BRANCH NO. 1,12,' STATION NO. 1,12/)
XNN = XNN*2.0
DX = DX/2.0
PTEMP = BETTA(J-1)
TTTEMP = D(J-1)
PP = PTEMP
TP = TTTEMP
RRP = RTEMP
DD = DPLN(ISLICE)
AA = APLN(ISLICE)
DO 91 L = 1, 4
V1 = G1/PP/AA**2/(1.0+G1*R*TP/(PP*AA)**2)
TERM1 = TP*G2/(PP*AA)**2
TERM2 = G1 * CP * 778.161 * R * TP * TP * V1 / (PP * AA) ** 2 / PP

TESTMA = 1.0 - TERM1 + TERM2

IF (TESTMA .LE. 0.0) GO TO 159

XK (L) = FUNP (F1 (J))

IF (L .EQ. 1) GO TO 89

DO 87 LL = 2, L

XTEST = ABS ( (XK (L) - XK (LL - 1)) / PP )

IF (XTEST .GT. DIFTOL) GO TO 81

CONTINUE

XL (L) = PUNP (PI (J))

IF (L .EQ. 1) GO TO 89

DO 87 LL = Z, L

XTEMP = ABS ( (XK (L) - XK (LL - 1)) / PP )

IF (XTEST .GT. DIFTOL) GO TO 81

CONTINUE

XL (L) = PUNP (PI (J))

IF (L .EQ. 4) GO TO 93

PP = PTEMP + XK (L) / 2.0

TP = TTEMP + XL (L) / 2.0

IF (L .EQ. 2) GO TO 91

RRP = RRP + DR / XNN

IF (L .NE. 3) GO TO 91

PP = PTEMP + XK (L)

TP = TTEMP + XL (L)

CONTINUE

IF (PP .LE. 0.0 .OR. TP .LE. 0.0) GO TO 159

VL = G1 / PP / AA ** 2 / (1.0 + G1 * R * TP / (PP * AA) ** 2)

TERM1 = TP * G2 / (PP * AA) ** 2

TERM2 = G1 * CP * 778.161 * R * TP * TP * VL / (PP * AA) ** 2 / PP

TESTMA = 1.0 - TERM1 + TERM2

IF (TESTMA .LE. 0.0) GO TO 159

BETTA (J) = PTEMP + (XK (1) + 2.0 * (XK (2) + XK (3)) + XK (4)) / 6.0

B (J) = TTEMP + (XL (1) + 2.0 * (XL (2) + XL (3)) + XL (4)) / 6.0

PP = BETTA (J)

IF (PP .LE. 0.0 .OR. BETTA (J) .LE. 0.0) GO TO 159

MACH1 = MACH1 - 1

PTEMP = PP

TTEMP = TP

GO TO 85

95 XNN = 2.0

DX = DXTEMP

SIGMA (J) = B (J) / APLN (ISLICE) / BETTA (J) * SQRT (Z4)

AMC (J) = SIGMA (J) * SQRT (1.0 / B (J)) * C5

IF (AMC (J) .GE. 1.0) GO TO 159

F1 (J) = REY

TT1 (J) = B (J) * (1.0 + C6 * AMC (J) ** 2) - 460.0

CONTINUE

C ALL STATIONS COMPUTED

99 SIGC = 1.0

AMC (NSTNS) = AMC (NSTNS) / SIGC

IF (AMC (NSTNS) .GT. 1.0) GO TO 159

BETTA (NSTNS) = BETTA (NSTNS) * SIGC ** 2

C RESTART CHOKE BRANCH IF M.LT. .6

IF (CH (ISLICE) .EQ. 0.0 .OR. CH (ISLICE) .EQ. 1.0) GO TO 113

IF (ACH .EQ. (-1.0)) GO TO 113

ACH = -1.

AB = 0.

DO 109 J = 1, NSTNS

IF (AMC (J) .GT. 8) GO TO 113

IF (AMC (J) .LT. AB) GO TO 109

AB = AMC (J)

CONTINUE
CH(ISLICE) = 0.0
AJ = (GAM+1.0)/2.0
CX = -(GAM+1.00)/(GAM-1.0)/2.0
AZ = .95/AB*(1.0+C6*.90)**(CX)*(1.0+C6*AB**2)**(-CX)
WCHOKE=W(ISLICE)
W(ISLICE) = AZ*W(ISLICE)
WRITE(6,111) ISLICE, W(ISLICE), WCHOKE
111 FORMAT(8X,6H***,23H RESTART CHOKED BRANCH,5H PLOM RATE 1NC
GO TO 177
113 BETA1 = PIN(ISLICE)**2
C CALCULATE THE CHOKING FLOW RATE
AB = 0.0
DO 115 J = 1, NSTNS
IF (AHC(J).LT.AB) GO TO 115
AB = AUC(J)
C 115 CONTINUE
AJ = (GAM+1.0)/2.0
CX = -(GAM+1.0)/(GAM-1.0)/2.0
AZ = .95/AB*(1.0+C6*.90)**(CX)*(1.0+C6*AB**2)**(-CX)
C COMPUTE RESISTANCE EQUATION FOR BALANCE
PT1 = (1.0+C6*AMC(1)**2)**(C1/2.0)*BETTA(1)
IF (C3.NE.0.0) GO TO 117
GO TO 119
117 DR2 = ROUT(ISLICE)**2 - RIN(ISLICE)**2
119 PEXT = PT1*(1.0+C3/T1*DR2)**(C1/2.0)
121 CONTINUE
Z1(ISLICE) = (PEXT**2-BETTA(NSTNS)**2)/W(ISLICE)**2
IF (Z1(ISLICE).GT.0.0) GO TO 129
WHITE(6,125) ISLICE, Z1(ISLICE)
125 FORMAT(///5X,'PASSAGE ',13,5X,' HAS NEGATIVE OR NO RESISTANCE'
Z
, F12.4//)
Z1(ISLICE) = ZED
129 CONTINUE
PP = BETTA(NSTNS)*(1.0+C6*AMC(NSTNS)**2)**(C1/2.0)
PTEXIT = PP
DIFTOL = 0.005
KTR = 0
C COMPUTE AVERAGE STATIC PRESSURE AND STATIC TEMPERATURE
C
PXX = 0.0
TXX = 0.0
DO 134 I = 1, NSTNS
TXX = TXX + B(I)
134 PXX = PXX + BETTA(I)
TXX = TXX/XN - 460.
PXX = PXX/XN
TTEXIT = TT1(NSTNS)
RETURN
C COMPUTE ACCELERATION
C
135 D = SV(2) - SV(1)
137 D = (SV(3) - SV(2))/D
E = ABS(D) - 1.0
IF (ABS(D).GT..6) GO TO 143
139 E = D/(D-1.0)
141 SIGB = E*SV(2) + (1.0 - E)*SV(3)
143 IS = 0
145 SIGC=SIGB
147  IF (SIGC.LE.0.) GO TO 159
149  KSIG=KSIG+1
151  IF (KSIG.LT.50) GO TO 155
153  WRITE (6, 151) ISLICE, J
155  CONTINUE
157  STOP
C
CCOKING ADJUSTMENT
C
159  WCHOK= W(ISLICE)
161  IF (CH(ISLICE).EQ.0.) GO TO 165
163  W(ISLICE)=.98*W(ISLICE)
165  AB=0.
167  CONTINUE
AJ=(GAM+1.)/2.
CX=- (GAM+1.00)/(GAM-1.0) /2.0
AZ=.95/AB*(1.0+C6*.90)**(CX)*(1.0+C6*AB**2)**(-CX)
W(ISLICE)=AZ*W(ISLICE)
169  CONTINUE
W(ISLICE)=1600.0*APLN(ISLICE)*SQRT((32.17*BETA1*GAM)/(K*T1))
171  CH(ISLICE)=1.0
173  WRITE (6, 175) ISLICE, W(ISLICE), WCHOK
175  FORMAT (3X, 12H*** PASSAGE ,15,23H HAS CHOKED
177  134H AND THE FLOW HAS BEEN REDUCED TO ,F10.4,6H FROM ,F10.4,4H ***)
179  GO TO 17
181  WRITE (6, 183) ISLICE
183  FORMAT (/2X,16H**FLOW IN BRANCH ,16.
185  Z HAS BEEN REDUCED 5C TIMES BECAUSE OF CHOKING*)
C
C CCOKING ADJUSTMENT
C
C---SOURCE.NPLOTIT
SUBROUTINE PLOTMF (ALPH2)
C
C SOURCE. NPLOTIT
C
NPLENMP 3356
NPLENMP 3357
NPLENMP 3358
NPLENMP 3359
NPLENMP 3360
108
ALABL(7) = PLTYME(2)

MD2 > 0 INDICATES JOB IS COMPLETE.--NOW DO SUMMARY PLOTS.

CONTINUE
IF (MD2.GT.0) GO TO 80
NSTAPS = NSTA/2 + 1
NLBLS = NSTAPS/5

SET UP TIME AND DATE LABEL FOR PLOT IDENTIFICATION

ALABL(1) = ALPH2(3)
ALABL(2) = ALPH2(4)
ALABL(3) = ALPH2(1)
ALABL(4) = ALPH2(2)

SET UP TITLE

DO 45 I = 1,30
IF (I.LE.21) TLABL1(I) = TITLE(I)
IF (I.GT.21) TLABL2(I-21) = TITLE(I)
45 CONTINUE

PRESSURE SIDE

IF (MD3.GT.1) GO TO 55
XP(1) = 0.0
IX = 1
DO 50 I = 3,NSTA,2
NMM IS THE MIDMETAL NODE NUMBER (L)
NMM = 5*I - 2
IX = IX + 1
50 XP(IX) = XP(IX-1) + DLX(NMM)/CINCH(IUNITS)
XPL = XP(NSTAPS)
DO 51 I = 2,NSTAPS
XP(I) = XP(I)/XPL
55 CONTINUE
IY = 0
ITP = NSTAPS*(ISLICE-1)
DO 60 I = 1,NSTA,2
II = IY + 1
NOS = 5*I - 4
Y(IY) = T(2,ISLICE,NOS)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IY) = Y(IY) - 460.
ITP = ITP + 1
TPO(ITP) = Y(IY)
IYP = IY + NSTAPS
Y(IYP) = T(2,ISLICE,NOS+1)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
IYP = IY + 2*NSTAPS
Y(IYP) = T(2,ISLICE,NOS+2)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
TMP(ITP) = Y(IYP)
IYP = IY + 3*NSTAPS
Y(IYP) = T(2,ISLICE,NOS+3)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
IYP = IY + 4*NSTAPS
NCOOL = NOS + 4
Y(IYP) = T(2, ISLICE, NC0OL)/CTMPF(IUNITS)

IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.

XLABL(16) = VARIB(1)
XLABL(17) = VARIB(2)

IF (IUNITS.EQ.2) GO TO 601
XLABL(12) = VARIB(9)
XLABL(13) = VARIB(10)
XLABL(28) = VARIB(11)
XLABL(29) = VARIB(12)

IF (IUNITS.EQ.2) GOTO 601

XLABL(12) = VARIB(16)
XLABL(13) = VARIB(17)

XLABL(12) = VARIB(16)
XLABL(13) = VARIB(17)

YLABL(5) = VARIB(13)
YLABL(6) = VARIB(10)

YPLABL(10) = VARIB(11)
YPLABL(11) = VARIB(13)

CONTINUE
DO 611 I = 1, 15

C

IF (IPLOT.EQ.3) GO TO 63

C

CPIM = PM/CPRSR(IUNITS)
CALL NUMEEB(l, ISLICE,4,P,XLABL2(6))
CALL NUMEER(4, CPIM,8, 1,XLABL2(11))
CALL NUMBER(4 ,XPL,8,4,XLABL(10))
CALL CHARS (84, YLABL, 90., 6.2, 9.3, 12)
CALL CHARS (56 , YPLABL, 90., 6.2, 9.3, 12)

C--- TITLES ARE DONE, NOW SET UP AXES FOR TEMPERATURE PLOTS

C

NPTS = NSTAPS
CALL SCALE(IXAX, NPTS, XP)
NPTS = 5*NSTAPS
CALL SCLBAK(IYAX, NPTS, Y, RTNHFR)
CALL GINTVL(RTNARR(1) ,F.TIARR(2) ,lo, 1,YSIN,YRAX)

C

VARS(1) = 7.0
VARS(2) = 9.0
VARS(3) = 0.0
VARS(4) = 0.0
VARS(5) = 1.0
VARS(6) = .5
VARS(7) = 1.0

C

CALL YAXIS(.8,.6,VARS)
VARS(2) = 8.9
VARS(3) = 3.0
VARS(4) = YMIN
VARS(5) = YMAX

C--- AXES ARE SET. NOW PLOT THE FIVE TEMPERATURE CURVES, USING

C DIFFERENT SYMBOLS FOR EACH.

C

IVARS(1) = 4
IVARS(2) = NSTAPS
IVARS(3) = 2
DO 603 I = 1, 5

IVARS(4) = ISYM(I)
YST = 1 + (I-1)*NSTAPS
YEN = I*NSTAPS
II = 0
DO 602 II = YST,YEN
II = II + 1
602 YTEM(II) = Y(II)
CALL GNPLOT(XP,YTEM,IVARS)
603 CONTINUE
CALL DISPLA(1)
C
C PRESSURE SIDE COOLANT PRESSURE DISTRIBUTION
C
I = 0
DO 61 I = 1,NSTA,2
IY = IY + 1
61 Y(IY) = F(2,ISLICE,I)*
2 (1.+(GAMC(I)-1.)*AM2(I)/2.)**((GAMC(I)/(GAMC(I)-1.)))/PESR(IUNITS)
CALL CHARS(84,TLABL1,0.0,0.15,9.85,12)
CALL CHARS(36,TLABL2,0.0,0.15,9.85,12)
CALL CHARS(60,XLABL,0.0,1.5,25,12)
CALL CHARS(56,XLABL2,0.0,1.5,.05,12)
CALL CHARS(40,YLABEL,90,.25,2.8,12)
MD3 = MD3 + 1
CALL NUMER (1,MD3,4,0,ALABL(5))
CALL CHARS (28,ALABL,0.0,6.2,SA3,12)
NPTS = NSTAPS
CALL SCALE (IXAX, NPTS, XP)
CALL SCLBAK (IYBX, NPTS, Y, RTNARR)
CALL GINTVL( RTNARR(1), RTNARR(2), 10, 1, YMIN, YMAX)
VARS(1) = 7.0
VARS(2) = 9.0
VARS(3) = 0.0
VARS(4) = 0.0
VARS(5) = 1.0
VARS(6) = .5
VARS(7) = 1.0
CALL YAXIS (.8,.6,VARS)
VARS(2) = 8.9
VARS(3) = 90.
VARS(4) = YMIN
VARS(5) = YMAX
CALL YAXIS (.8,.6,VARS)
C--- AXES ARE SET, NOW PLOT THE PRESSURE
C
IVARS(1) = 4
IVARS(2) = NSTAPS
IVARS(3) = 2
IVARS(4) = 65
CALL GNPLOT(XP,Y,IVARS)
CALL DISPLA(1)
C
C SUCTION SIDE
C
IF (MD3.GT.2) GO TO 69
63 XS(1) = 0.0
IX = 1
DO 65 I = 2,NSTA,2
NMM = 5*I - 2  
IX = IX + 1  
65 XS(IX) = XS(IX-1) + DLX(NMM)/CINCH(IUNITS)  
XSL = XS(NSTAPS)  
DO 66 I = 2,NSTAPS  
66 XS(I) = XS(I)/XSL  
69 CONTINUE  
C  
IY = 1  
ITS = NSTAPS*(ISLICE-1) + 1  
Y(IY) = T(2,ISLICE,1)/CTMPF(IUNITS)  
IF (IUNITS.EQ.2) Y(IY) = Y(IY) - 460.  
TSO(ITS) = Y(IY)  
IYP = IY + NSTAPS  
Y(IYP) = T(2,ISLICE,2)/CTMPF(IUNITS)  
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.  
IYP = IYP + NSTAPS  
Y(IYP) = T(2,ISLICE,3)/CTMPF(IUNITS)  
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.  
TSM(ITS) = Y(IYP)  
IYP = IYP + NSTAPS  
Y(IYP) = T(2,ISLICE,4)/CTMPF(IUNITS)  
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.  
IYP = IYP + NSTAPS  
Y(IYP) = T(2,ISLICE,5)/CTMPF(IUNITS)  
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.  
DO 70 I = 2,NSTA,2  
66 IY = IY + 1  
NOS = 5*I - 4  
Y(IY) = T(2,ISLICE,NOS)/CTMPF(IUNITS)  
IF (IUNITS.EQ.2) Y(IY) = Y(IY) - 460.  
ITS = ITS + 1  
TSO(ITS) = Y(IY)  
IYP = IY + NSTAPS  
Y(IYP) = T(2,ISLICE,NOS+1)/CTMPF(IUNITS)  
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.  
IYP = IYP + NSTAPS  
Y(IYP) = T(2,ISLICE,NOS+2)/CTMPF(IUNITS)  
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.  
TSM(ITS) = Y(IYP)  
IYP = IYP + NSTAPS  
Y(IYP) = T(2,ISLICE,NOS+3)/CTMPF(IUNITS)  
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.  
IYP = IYP + NSTAPS  
NCOL = NOS + 4  
Y(IYP) = T(2,ISLICE,NCOL)/CTMPF(IUNITS)  
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.  
XLABL2(1) = VARIB(3)  
XLABL2(2) = VARIB(4)  
C  
IF (IPLT.EQ.3) GO TO 90  
C  
CALL NUMBER(1,ISLICE,4,0,XLABL2(6))  
CALL NUMBER(4,CPIM,8,1,XLABL2(11))  
CALL NUMBER(4,XSL,8,4,XLABL(10))  
IX = 2 + NFWD/2  
CALL CHARS(S4,TLABL1,0.0,.0.15,9.85,12)  
CALL CHARS(36,TLABL2,0.0,.0.15,9.65,12)  
CALL CHARS(60,XLABL,0.0,1.5,.25,12)  
CALL CHARS(56,XLABL2,0.0,1.5,.05,12)  
C

113
CALL CHARS (28, YLABL, 90..25, 3.3, 12)
MD3 = MD3 + 1
CALL NUMBER (1, MD3, 4, 0, YLABL (5))
CALL CHARS (28, YLABL, 0.0, 6.2, 9.3, 12)

C--- TITLES ARE DONE, NOW SET UP AXES FOR TEMPERATURE PLOTS

NPTS = NSTAPS
CALL SCALE (IXAX, NPTS, XS)

CALL SCLBAK (IXAX, NPTS, Y, RTNARR)
CALL GINTVL (RTNARR (1), RTNARR (2), 10, 1, YMIN, YMAX)

VARS (1) = 7.0
VARS (2) = 9.0
VARS (3) = 0.0
VARS (4) = 0.0
VARS (5) = 1.0
VARS (6) = 0.5
VARS (7) = 1.0
CALL XAXIS (.8, 6, VARS)

CALL YA XIS (.8, 6, VARS)

CALL DISPLA (1)

C--- AXES ARE SET. NOW PLOT THE FIVE TEMPERATURE CURVES, USING

DIFFERENT SYMBOLS FOR EACH.

IVARS(1) = 4
IVARS(2) = NSTAPS
IVARS(3) = 2
DO 703 I = 1, 5
IVARS(4) = ISYM(I)
IYST = 1 + (I-1)*NSTAPS
IYEN = I*NSTAPS
III = 0
DO 702 II = IYST, IYEN
III = III + 1

702 YTEM(III) = Y(II)
CALL G PLOT (XS, YTEM, IVARS)

703 CONTINUE
CALL DISPLA (1)

C SUCTION SIDE COOLANT PRESSURE DISTRIBUTION

Y (1) = P (2, ISLICE, 1) *
Z (1.+(GAMC (1)-1.)*AM2 (1)/2.)**(GAMC (1)/(GAMC (1)-1.)) / CPRSR (IUNITS)

DO 75 I = 2, NSTA, 2

75 Y (IY) = P (2, ISLICE, I) *(1.+ (GAMC (I)-1.)
Z*AM2 (I)/2.)**(GAMC (I)/(GAMC (I)-1.)) / CPRSR (IUNITS)

CALL CHARS (84, TLABL1, 0.0, 0.15, 9.85, 12)
CALL CHARS (36, TLABL2, 0.0, 0.15, 9.65, 12)
CALL CHARS (60, YLABL0, 0.0, 1.5, 25, 12)
CALL CHARS (56, YLABL2, 0.0, 1.5, 0.05, 12)
CALL CHARS (40, YLABL90, 0.0, 25, 2.8, 12)
MD3 = MD3 + 1
CALL NUMBER(1, MD3, 4, 0, ALABL(5))
CALL CHARS(28, ALABL, 0, 0, 6.2, 9.3, 12)
NPTS = NSTAPS
CALL SCALE(IYAX, NPTS, XS)
CALL SCLBAK(IYAX, NPTS, Y, RTNARR)
CALL GINTVL(RTNARR(1), RTNARR(2), 10, 1, YMIN, YMAX)
VARS(1) = 7.0
VARS(2) = 9.0
VARS(3) = 0.0
VARS(4) = 0.0
VARS(5) = 1.0
VARS(6) = .5
VARS(7) = 1.0
CALL XAXIS(.8, .6, VARS)
VARS(2) = 8.9
VARS(3) = 90.
VARS(4) = YMIN
VARS(5) = YMAX
CALL YAXIS(-8, -1.6, VARS)

AXES ARE SET, NOW PLOT THE PRESSURE

IVARS(1) = 4
IVARS(2) = NSTAPS
IVARS(3) = 2
IVARS(4) = 65
CALL GPOINT(XS, Y, IVARS)
GO TO 150

THE FOLLOWING SECTION PUTS OUT PLOTS CONTAINING TEMPERATURES FROM ALL SLICES ON ONE FRAME

CONTINUE
IF (ISLICE.LT.NSLICE) GO TO 150

THE FOLLOWING PUTS TWO PLOTS ON ONE FRAME OF FILM

FIRST PLOT THE OUTSIDE SURFACE TEMPERATURES FOR EACH SLICE ON THE SAME PLOT

NPTS = NSTAPS*NSLICE
CALL SCLBAK(IYAX, NPTS, TPC, RTNARR)
TMXP = RTNARR(2)
TMXP = RTNARR(1)
CALL SCLBAK(IYAX, NPTS, TSO, RTNARR)
TMXP = RTNARR(2)
TMINS = RTNARR(1)
IF (TMXP.GT.TMAXP) TMAXP = TMXP
IF (TMINS.LT.TIMNP) TIMNP = TMINS
NINTRV = (TMAXP-TMINP)/100 + 2
CALL GINTVL(TIMNP, TMAXP, NINTRV, 0, ATMINP, ATMAXP)
AINTRV = NINTRV
CALL CHARS(84, TLAB1, 0.0, 0.15, 9.85, 12)
CALL CHARS(36, TLAB2, 0.0, 0.15, 9.65, 12)
YLABL2(5) = VARIB(5)
YLABL2(6) = VARIB(6)
CALL CHARS(44, YLABL2, 90, -25, 1.6, 12)
VARS(1) = 8.
VARS(2) = 8.5

NPLOIT 3721
NPLOIT 3722
NPLOIT 3723
NPLOIT 3724
NPLOIT 3725
NPLOIT 3726
NPLOIT 3727
NPLOIT 3728
NPLOIT 3729
NPLOIT 3730
NPLOIT 3731
NPLOIT 3732
NPLOIT 3733
NPLOIT 3734
NPLOIT 3735
NPLOIT 3736
NPLOIT 3737
NPLOIT 3738
NPLOIT 3739
NPLOIT 3740
NPLOIT 3741
NPLOIT 3742
NPLOIT 3743
NPLOIT 3744
NPLOIT 3745
NPLOIT 3746
NPLOIT 3747
NPLOIT 3748
NPLOIT 3749
NPLOIT 3750
NPLOIT 3751
NPLOIT 3752
NPLOIT 3753
NPLOIT 3754
NPLOIT 3755
NPLOIT 3756
NPLOIT 3757
NPLOIT 3758
NPLOIT 3759
NPLOIT 3760
NPLOIT 3761
NPLOIT 3762
NPLOIT 3763
NPLOIT 3764
NPLOIT 3765
NPLOIT 3766
NPLOIT 3767
NPLOIT 3768
NPLOIT 3769
NPLOIT 3770
NPLOIT 3771
NPLOIT 3772
NPLOIT 3773
NPLOIT 3774
NPLOIT 3775
NPLOIT 3776
NPLOIT 3777
NPLOIT 3778
NPLOIT 3779
NPLOIT 3780

115
VARS(3) = 0.
VARS(4) = 0.0
VARS(5) = 1.0
VARS(6) = .5
VARS(7) = 1.0
VARS(8) = 0.0
CALL XAXIS(1.2,5.5,VARS)
MD3 = MD3 + 1
CALL NUMBER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
CALL CHARS(20,PLEGN,0.0,6.0,9.5,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMXXP
VARS(6) = AINTRV
VARS(7) = 1.
CALL YAXIS(1.2,.5,VARS)
DO 100 I = 1,NSLICE
  JST = NSTAPS*(I-1)
  DO 95 J = 1,NSTAPS
    Y(J) = TPO(JST+J)
    SYMBOL = SYMBL(1)
    KS = 0
C C--- LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO
C IDENTIFY THE CURVES.
KSTART = I + 1
DO 98 K = KSTART,NSTAPS,10
  KS = KS + 1
  XLBL(KS) = XP(K)
  YLBL(KS) = Y(K)
IVARS(1) = 6
IVARS(2) = NLBLS
IVARS(3) = 3
IVARS(4) = 240+I
IVARS(5) = 1
IVARS(6) = 8
CALL GPLOT(XLBL,YLBL,IVARS)
IVARS(1) = 3
IVARS(2) = NSTAPS
IVARS(3) = 0
100 CALL GPLOT(XP,Y,IVARS)
C
VARS(1) = 7.
VARS(2) = 8.5
VARS(3) = 0.
VARS(4) = 0.0
VARS(5) = 1.0
VARS(6) = .5
VARS(7) = 1.0
CALL XAXIS(1.2,.5,VARS)
CALL CHARS(42,XLABL,0.0,3.0,.05,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMXXP
VARS(6) = 0.
VARS(7) = 0.0
VARS(8) = 1.0
CALL XAXIS(1.2,5.5,VARS)
MD3 = MD3 + 1
CALL NUMBER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMXXP
VARS(6) = 0.
VARS(7) = 0.0
VARS(8) = 1.0
CALL XAXIS(1.2,5.5,VARS)
MD3 = MD3 + 1
CALL NUMBER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMXXP
VARS(6) = 0.
VARS(7) = 0.0
VARS(8) = 1.0
CALL XAXIS(1.2,5.5,VARS)
MD3 = MD3 + 1
CALL NUMBER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMXXP
VARS(6) = 0.
VARS(7) = 0.0
VARS(8) = 1.0
CALL XAXIS(1.2,5.5,VARS)
MD3 = MD3 + 1
CALL NUMBER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMXXP
VARS(6) = 0.
VARS(7) = 0.0
VARS(8) = 1.0
CALL XAXIS(1.2,5.5,VARS)
MD3 = MD3 + 1
CALL NUMBER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMXXP
VARS(6) = 0.
VARS(7) = 0.0
VARS(8) = 1.0
CALL XAXIS(1.2,5.5,VARS)
MD3 = MD3 + 1
CALL NUMBER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMXXP
VARS(6) = 0.
VARS(7) = 0.0
VARS(8) = 1.0
CALL XAXIS(1.2,5.5,VARS)
MD3 = MD3 + 1
CALL NUMBER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMXXP
VARS(6) = 0.
VARS(7) = 0.0
VARS(8) = 1.0
CALL XAXIS(1.2,5.5,VARS)
MD3 = MD3 + 1
CALL NUMBER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMXXP
VARS(6) = 0.
VARS(7) = 0.0
VARS(8) = 1.0
CALL XAXIS(1.2,5.5,VARS)
MD3 = MD3 + 1
CALL NUMBER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMXXP
VARS(6) = 0.
VARS(7) = 0.0
VARS(8) = 1.0
CALL XAXIS(1.2,5.5,VARS)
MD3 = MD3 + 1
CALL NUMBER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMXXP
VARS(6) = 0.
VARS(7) = 0.0
VARS(8) = 1.0
CALL XAXIS(1.2,5.5,VARS)
MD3 = MD3 + 1
CALL NUMBER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMXXP
VARS(6) = 0.
VARS(7) = 0.0
VARS(8) = 1.0
CALL XAXIS(1.2,5.5,VARS)
MD3 = MD3 + 1
CALL NUMBER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMXXP
VARS(6) = 0.
VARS(7) = 0.0
VARS(8) = 1.0
CALL XAXIS(1.2,5.5,VARS)
MD3 = MD3 + 1
CALL NUMBER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMXXP

VAES(6) = AINTV
VARS(7) = 1.
CALL YAXIS(1.2, 5, VARS)
CALL CHARS(20, SLEGN, 0.0, 6.0, 4.5, 12)
DO 110 I = 1, NSLICE
JST = NSTAPS*(I-1)
DO 105 J = 1, NSTAPS
Y(J) = TSO(JST+J)
SYMBOL = SYMB(L(I))
KS = 0

C--- LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO
C IDENTIFY THE CURVES.
KSTART = I + 1
DO 108 K = KSTART, NSTAPS, 10
KS = KS + 1
XLBL(KS) = XS(K)
YLBL(KS) = Y(K)
IVARS(1) = 6
IVARS(2) = NLBS
IVARS(3) = 3
IVARS(4) = 240+I
IVARS(5) = 1
IVARS(6) = 8
CALL GPLOT(XLBL, YLBL, IVARS)
IVARS(1) = 3
IVARS(2) = NSTAPS
IVARS(3) = 0
CALL GPLOT(XS, Y, IVARS)
CALL DISPLA(1)

C NOW PLOT THE MID-METAL TEMPERATURES FOR EACH SLICE, ALL ON ONE PLOT

112 CONTINUE
CALL SCLBAK(IYAX, NPTS, TFM, RTNARR)
TMAXP = RTNARR(2)
TMINP = RTNARR(1)
CALL SCLBAK(IYAX, NPTS, TSM, RTNARR)
TMAXS = RTNARR(2)
TMINS = RTNARR(1)
IF (TMAXS.GT.TMAXP) TMAXP = TMAXS
IF (TMINS.LT.TMINP) TMINP = TMINS
NINTV = (TMAXP-TMINP)/100. + 2
CALL GINTVL(TMINP, TMAXP, NINTV, 0, ATMINS, ATMAXP)
ATMINS = TMINP
CALL CHARS(84, TLABL1, 0.0, 0.15, 9.85, 12)
CALL CHARS(36, TLABL2, 0.0, 0.15, 3.65, 12)
YLABL2(5) = VARS(7)
YLABL2(6) = VARS(8)
CALL CHARS(44, YLABL2, 90., 25, 1.6, 12)
VARS(1) = 8.
VARS(2) = 8.5
VARS(3) = 0.
VARS(4) = 0.0
VARS(5) = 1.0
VARS(6) = .5
VARS(7) = 1.0
VARS(8) = 0.0
CALL XAXIS(1.2,5.5,VARS)
MD3 = MD3 + 1
CALL NUMBER(1,MD3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)
CALL CHARS(20,PLEGN,0.0,6.0,9.5,12)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMAXP
VARS(6) = AINTRV
VARS(7) = 1.
CALL YAXIS(1.2,5.5,VARS)
CONTINUE
DO 120 I = 1,NSLICE
JST = NSTAPS*(I-1)
DO 115 J = 1,NSTAPS
Y(J) = TPfl(JST+J)
SYRBOL = SYMBL(1)
KS = 0
LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO IDENTIFY THE CURVES.
KSTART = I + 1
DO 118 K = KSTART,NSTAPS,10
KS = KS + 1
XLBL(KS) = XP(K)
YLBL(KS) = Y(K)
IVARS(1) = 6
IVARS(2) = NLBLS
IVARS(3) = 3
IVARS(4) = 240*I
IVARS(5) = 1
IVARS(6) = 8
CALL GPLOT(XLBL,YLBL,IVARS)
IVARS(1) = 3
IVARS(2) = NSTAPS
IVARS(3) = 0
CALL GPLOT(XP,Y,IVARS)
VARS(1) = 7.
VARS(2) = 8.5
VARS(3) = 0.
VARS(4) = 0.0
VARS(5) = 1.0
VARS(6) = .5
VARS(7) = 1.0
CALL XAXIS(1.2,.5,VARS)
CALL CHARS(42,XLABL,0.0,.35,.05,15)
VARS(1) = 7.
VARS(2) = 3.8
VARS(3) = 90.
VARS(4) = ATMINP
VARS(5) = ATMAXP
VARS(6) = AINTRV
VARS(7) = 1.
CALL XAXIS(1.2,.5,VARS)
CALL CHARS(20,SLEGN,0.0,6.0,4.5,12)
DO 130 I = 1,NSLICE
JST = NSTAPS*(I-1)
DO 125 J = 1,NSTAPS
125 Y(J) = TSM(JST+J)
SYMBOL = SYMBOL(I)
KS = 0
C
C--- LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO
C IDENTIFY THE CURVES.
KSTART = I + 1
DO 128 K = KSTART,NSTAPS,10
KS = KS + 1
XLBL(KS) = XS(K)
YLBL(KS) = Y(K)
IVARS(1) = 6
IVARS(2) = NLBLS
IVARS(3) = 3
IVARS(4) = 240+I
IVARS(5) = 8
CALL GPLOT(XLBL,YLBL,IVARS)
IVARS(1) = 3
IVARS(2) = NSTAPS
IVARS(3) = 0
CALL GPLOT(XS,Y,IVARS)
CALL DISPL.4 (1)

CONTINUE
RETURN
END

C---- SOURCE.NPREPAT
SUBROUTINE PREP(ICHNL,NTTG)

COMMON /BOUND/ BCXS(400), BCXP(400), BCHGS(1000), BCHGP(1000), NPREPAT 3991
Z BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000), NPREPAT 3992
Z BCPS(1000), BCQPS(1000), THUBIN(400), THUB(80), NPREPAT 3993
Z QHUBIN (400), QHUB(80), TQIPIN(400), TTIP(80), NPREPAT 3994
Z QTIMIN(400), QTIP(80), RHOGV(400), PEX(400), NPREPAT 3995
Z BTIME(50), TTIO(50), PTO(50), WPLEN, NPREPAT 3996
Z WVEST (50), AKCTBL(20), AKWTBL(20), NBCS, NBCP NPREPAT 3997
Z

COMMON /FLMCOL/ BHOGA(80), PG(80), XFC(80), PLMEFF(80), NPREPAT 3998
Z XMUC(80), LMES(80), REF(80), NFCSUP(80), NPREPAT 3999
Z

COMMON /FRIC/ ALPHA, BETA, DELTA, EPS
Z

COMMON /FRPS/ CPO, GAMO, DP(80), SP(80), RE(80), NPREPAT 4000
Z CPC(80), GAMC(80), DUMB1(80), DUMB2(80), NPREPAT 4001
Z

COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000), NPREPAT 4002
Z IPILOT, MD1, MD2, MD3, IADJIN, INRITE
Z

COMMON /TCC/ ADUMP, ETA, CD, CP, NPREPAT 4003
Z GAM, PIM, R, SPAN, TOG, NPREPAT 4004
Z WDUMP, WIM, AKG(15,80), AKW(15,80), NPREPAT 4005
Z A(400), AJET(80), AM2(80), CNUM(80), NPREPAT 4006
Z DH(80), DHF(80), DHJ(80), NPREPAT 4007
Z DLX(400), FF(80), HC(80), HG(80), NPREPAT 4008
Z
I.

P(2,15,80), PEXIT(15), PUMP(80), QG(80),

QSNK(80), RR(80), S(15), T(2,15,400),

WJ(15,80), TA(400), WFC(80),

WCR(2,15,80), XN(80),

ICOR, IPAIN, IML, IHUB, ITIP,

ISBLOK, ISLICE, NBLKSZ, NSLICE,

NFWD, NSTA, IHC(80),

C

COMMON TRN, /RHOC, PHOM, SPHTC, SPHTN,

DLTYME, Tyme, TEP, TYM,

NPREPAT 4021

NPREPAT 4022

NPREPAT 4023

NPREPAT 4024

NPREPAT 4025

NPREPAT 4026

NPREPAT 4027

NPREPAT 4028

NPREPAT 4029

NPREPAT 4030

NPREPAT 4031

NPREPAT 4032

NPREPAT 4033

NPREPAT 4034

NPREPAT 4035

NPREPAT 4036

NPREPAT 4037

NPREPAT 4038

NPREPAT 4039

NPREPAT 4040

NPREPAT 4041

NPREPAT 4042

NPREPAT 4043

NPREPAT 4044

NPREPAT 4045

NPREPAT 4046

NPREPAT 4047

NPREPAT 4048

NPREPAT 4049

NPREPAT 4050

NPREPAT 4051

NPREPAT 4052

NPREPAT 4053

NPREPAT 4054

NPREPAT 4055

NPREPAT 4056

NPREPAT 4057

NPREPAT 4058

NPREPAT 4059

NPREPAT 4060

NPREPAT 4061

NPREPAT 4062

NPREPAT 4063

NPREPAT 4064

NPREPAT 4065

NPREPAT 4066

NPREPAT 4067

NPREPAT 4068

NPREPAT 4069

NPREPAT 4070

NPREPAT 4071

NPREPAT 4072

NPREPAT 4073

NPREPAT 4074

NPREPAT 4075

NPREPAT 4076

NPREPAT 4077

NPREPAT 4078

NPREPAT 4079

NPREPAT 4080

120
IM = IM + NODST
DLX(I) = CHANL(IM)
DO 215 I = 1, NSTA
IFLU = 5*I
DH(I) = 4.0*A(IFLU)/(2.*(SPAN*TAU(IFLU)))
IM = IM + NODST
DHJ(I) = CHANL(IM)
IM = IM + NODST
XN(I) = CHANL(IM)
14 CONTINUE
IF (DHJ(I) .GT. 0.0 .AND. XN(I) .GT. 0.0) GO TO 202
AJet(I) = 0.0
GO TO 212
202 CONTINUE
XOD = XN(I)/DHJ(I)
IF(XOD .LT. 3.1 .OR. XOD .GT. 12.5) WRITE (6,527)
527 FORUAT(1H0,' WARNING, RATIO OF JET HOLE SPACING TO JET DIAMETER
FOR JET ',I2,' IS ',F10.4,' WHICH IS OUT OF'
CNUM(I) = SPAN/XN(I)
C NUM IS THE NUMBER OF IMPINGEMENT JETS AT CHANNEL NODE I
C TOTAL JET AREA IS (AREA OF ONE JET)*(NUMBER OF JETS)
C
54 AJet(I) = 3.14159*DHJ(I)**2/4.*CNUM(I)
212 CONTINUE
IM = IM + NODST
PR(I) = CHANL(IM)
IM = IM + NODST
DP(I) = CHANL(IM)
IM = IM + NODST
SP(I) = CHANL(IM)
IM = IM + NODST
IM = IM + NODST
IIN = IM1 + I
IHC(I) = INDCHN(INN)
215 CONTINUE
C*****
C-- NOW, GIVEN SLICE, ICHNL, EVALUATE B.C. AT METAL NODE POINTS. IN THE
C-- FOLLOWING:
C-- XS & XP ARE DISTANCE FROM LEADING EDGE, ALONG OUTSIDE SURFACE
C-- (INCHES), FOR SUCTION & PRESSURE SIDES.
C-- THE CONVENTIONS USED IN THE FOLLOWING ARE: INDEX BEGINING WITH -I-
C-- IS A SLICE INDEX, INDEX BEGINING WITH -N- IS A TIME INDEX, INDEX
C-- BEGINING WITH -L- IS A N X INDEX, AND AN INDEX BEGINING WITH -J- IS
C-- A PROPERTY INDEX I.E. HG,QG,TG,PG.
C
C-- FIRST, CHECK THAT THIS IS A TRANSIENT CASE, AND DETERMINE THE MAX.
C-- BCTIME INDEX, NMX.
NMX = 1
310 IF (BCTIME(NMX+1).LE.0.0) GO TO 315
NMX = NMX + 1
GO TO 310
315 CONTINUE
C-- NOW, IF THIS IS A TRANSIENT, FIND THE LOCATION IN THE BCTIME ARRAY
C-- OF THE CURRENT TIME, AND CALCULATE THE VALUE OF THE INTERPOLATING
C-- PARAMETER, TMPFRAC.
TMFRAC = 0.0
NLST = 1
IF (NMX.EQ.1.0.OR.TYME.LE.0.0) GO TO 330
C-- THE ABOVE TRANSFER OCCURS IF THIS IS A STEADY STATE PROBLEM
C-- THE FOLLOWING TRANSFER OCCURS IF WE ARE BEYOND THE LAST BCTIME ENTRY
NLST = NMX
IF (TYME.GE.BCTIME(NMX)) GO TO 330
NMXM1 = NMX - 1
DO 320 N = 1,NMXM1
NLST = N
IF (TYME.GE.BCTIME(N).AND.TYME.LT.BCTIME(N+1)) GO TO 325
320 CONTINUE
325 TMFRAC = (TYME-BCTIME(NLST))/(BCTIME(NLST+1)-BCTIME(NLST))
C
C-- NEXT, SEARCH THE BCXS & BCXP ARRAYS TO FIND THE X INTERPOLATING FACTORS, XSF & XPP, FOR POSITIONS XS & XP, SLICE ICHNL.
C-- THE BRACKETING INDICES ARE LBLOWS & LABOVS AND LBLOYP & LABOVP.
C-- THE STARTING POINTS IN THE BCXS & BCXP ARRAYS FOR THIS SLICE ARE:
330 LSS = (ICHNL-1)*NBCS
LSP = (ICHNL-1)*NBCP
XS = 0.0
XP = 0.0
C-- THE STARTING POINTS IN THE PROPERTY ARRAYS, FOR THE LATEST TIME STEP
JS1S = NSLICE*(NLST-1)*NBCS + LSS
JS1P = NSLICE*(NLST-1)*NBCP + LSP
IF (NMX.EQ.NLST) GO TO 335
JS2S = JS1S + NSLICE*NBCS
JS2P = JS1P + NSLICE*NBCP
335 HG(1) = BCHGS(JS1S+1) + TMFRAC*(BCHGS(JS2S+1)-BCHGS(JS1S+1))
TG(1) = BCTGS(JS1S+1) + TMFRAC*(BCTGS(JS2S+1)-BCTGS(JS1S+1)) + 460.0
QG(1) = BCQGS(JS1S+1) + TMFRAC*(BCQGS(JS2S+1)-BCQGS(JS1S+1))
PG(1) = BCPGS(JS1S+1) + TMFRAC*(BCPGS(JS2S+1)-BCPGS(JS1S+1))
C
C-- THE OUTSIDE SURFACE NODE NUMBERS FOR S & P SIDES ARE:
NNS = 5*K - 4
NNP = 5*K + 1
XS = XS + DLX(NNS)
XP = XP + DLX(NNP)
C
C-- INSERT ERROR MESSAGE HERE-- EXTRAPOLATING BEYOND THE BCXS TABLE
C
C-- INSERT ERROR MESSAGE HERE-- EXTRAPOLATING BEYOND THE BCXS TABLE
C
C-- INSERT ERROR MESSAGE HERE-- EXTRAPOLATING BEYOND THE BCXS TABLE
C
DO 340 L = 1,NBCS
LBlOWS = LSS + L - 1
LABOVS = LSS + L
IF (BCXS(LABOVS).GT.XS) GO TO 342
340 CONTINUE
C
DO 345 L = 1,NBCP
LBLOWP = LSP + L - 1
LABOVP = LSP + L
IF (BCXP(LABOVP).GT.XP) GO TO 347
345 CONTINUE
C-- INSERT ERROR MESSAGE HERE-- EXTRAPOLATING BEYOND THE BCXP TABLE

C-- NOW THE FRACTIONS ARE KNOWN, CALCULATE THE INTERPOLATED PROPERTIES.

C-- FIRST, FOR THE STEADY STATE OR FOR TIMES BEYOND THE LAST BCXI:

C

347 XPP = (XP-BCXP(LBLOWP)) / (BCXP(LBLOWP)-BCXP(LBLOWP))

C

C-- NOY THE FRACTIONS ARE KNOWN, CALCULATE THE INTERPOLATED PROPERTIES.

C-- FIRST, FOR THE STEADY STATE OR FOR TIMES BEYOND THE LAST BCXI:

C

JBlS = NSLICE*(NLST-1)*NBCS + LBLOW

JBlP = NSLICE*(NLST-1)*NBCP + LBLOWP

HG(K) = BCGS(JBlS) + XSF*(BCGS(JBlS+1) - BCGS(JBlS))

HG(K+1) = BCGP(JBlP) + XPF*(BCGP(JBlP+1) - BCGP(JBlP))

QG(K) = BCQGS(JBlS) + XSF*(BCQGS(JBlS+1) - BCQGS(JBlS))

QG(K+1) = BCQGP(JBlP) + XPF*(BCQGP(JBlP+1) - BCQGP(JBlP))

TG(K) = BCTGS(JBlS) + XSF*(BCTGS(JBlS+1) - BCTGS(JBlS)) + 460.

TG(K+1) = BCTGP(JBlP) + XPF*(BCTGP(JBlP+1) - BCTGP(JBlP)) + 460.

PG(K) = BCPGS(JBlS) + XSF*(BCPGS(JBlS+1) - BCPGS(JBlS))

PG(K+1) = BCPGP(JBlP) + XPF*(BCPGP(JBlP+1) - BCPGP(JBlP))

IF (NMX.EQ.1 OR TYP. GE. BCXI(NNX) OR TYP. LE. 0.0) GO TO 350

JB2S = NSLICE*NLST*NBCS + LBLOW

JB2P = NSLICE*NLST*NBCP + LBLOWP

AHG = BCGS(JB2S) + XSF*(BCGS(JB2S+1) - BCGS(JB2S))

AHG = BCHGP(JB2P) + XPF*(BCHGP(JB2P+1) - BCHGP(JB2P))

AQG = BCQGS(JB2S) + XSF*(BCQGS(JB2S+1) - BCQGS(JB2S))

AQG = BCQGP(JB2P) + XPF*(BCQGP(JB2P+1) - BCQGP(JB2P))

ATG = BCTGS(JB2S) + XSF*(BCTGS(JB2S+1) - BCTGS(JB2S)) + 460.

ATG = BCTGP(JB2P) + XPF*(BCTGP(JB2P+1) - BCTGP(JB2P)) + 460.

APG = BCPGS(JB2S) + XSF*(BCPGS(JB2S+1) - BCPGS(JB2S))

APG = BCPGP(JB2P) + XPF*(BCPGP(JB2P+1) - BCPGP(JB2P))

350 CONTINUE

RETURN

END

C----SOURCE.NTARAY

SUBROUTINE 'NTARAY(JS, JSENS, DELTAN)

C

C++++++ A SUBROUTINE TO SET UP THE COEFFICIENT ARREY TO SOLVE FOR

C BLADE TEMPERATURES, TRANSIENT CALCULATIONS.

C FIRST PUT TOGETHER FROM STEADY STATE PROGRAM, NOVEMBER 24, 1975

C

C***************************************************************

C******************************************************************
**Node Names for Station IS Are:**

- **LCOOL** = Coolant Node at IS.
- **LIN** = Inside Wall Node.
- **L** = Mid Wall Node.
- **LOUT** = Outside Wall Node.
- **LCUP** = Adjacent Upstream Coolant Node.
- **LCUPS** = Adjacent Inside Wall Node.
- **LUP** = Adjacent Upstream Mid-Wall Node.
- **LDN** = Adjacent Downstream Mid-Wall Node.
- **LJ** = Junction of Coating and Wall Metal.

---

**FLOW**

---

- **LCUP**—**LCOOL**
- **LUP**—**LDN**
- **LCUPS**—**LTDN**

---

**Wall**

---

**Pressure Side Wall or Plenum**

---

**REAL*8** **TCOF**

**COMMON** * /BOUND/

- **BCXS(400)**
- **BCXP(400)**
- **BCHGS(1000)**
- **BCHGP(1000)**

**COMMON** * /PLMCOL/

- **RHOGVA(80)**
- **PG(80)**
- **XFC(80)**
- **FLMEFF(80)**
- **XMUC(80)**
- **EMES(80)**
- **REFC(80)**
- **NFCSUP(80)**

**COMMON** * /PRPS/

- **CPC(80)**
- **GAMC(80)**
- **DUMR1(80)**
- **DUMR2(80)**

**COMMON** * /TCO/

- **ADUMP**
- **BTA**
- **CD**
- **CP**
- **DUMP**
- **WIN**
- **AKC(15,80)**
- **AKM(15,80)**
- **A(400)**
- **AJET(80)**
- **AM2(80)**
- **CNUM(80)**
- **DH(80)**
- **DHP(80)**
- **DHJ(80)**
- **DLX(400)**
- **FP(80)**
- **HC(80)**
- **HG(60)**
- **P(2,15,80)**
- **PEXIT(15)**
- **PUMP(80)**
- **QG(80)**
- **QSNK(80)**
- **BR(80)**
- **S(15)**
- **T(2,15,400)**
- **TG(80)**
- **TAU(400)**
- **WFC(80)**
- **XN(80)**
- **ICOR**
- **IFILH**
- **IHUB**
- **ITIP**
C

C** BEGIN OVERALL LOOP, WHERE LOOP VARIABLE (IS) IS THE STATION NUMBER

C

DO 440 IS = 1,NSTA
ISUP = IS - 2
YIMP = 0.0
YFINS = 0.0
YCONV = 0.0
YIMP = 0.0
YFINSU = 0.0
YCONVU = 0.0
YIMP = 0.0
YFNSUU = 0.0
YCNVUU = 0.0
IF (IHC(IS).EQ.1) YIMP=1.0
IF (IHC(IS).EQ.2) YCONV= (1.0+RCVRY*AM2(IS)*GAMC(IS)-1.)/2.
IF (IHC(IS).EQ.3) YFINS=1.0
FACTOR = 1.0
IF (IS.EQ.ICOMS.OR.IS.EQ.ICOMP) FACTOR = .5
ISENS = 0
ISEN = IS - 2*(IS/2)

C

C IF (IS.GT.NFWD+1)-- IN TRAILING EDGE REGION, GO TO 380

C

IF(IS.GT.NFWD+1) GO TO 380

C

JS = 16
LCOOL = 5*IS
LIN =LCOOL-1
L =LCOOL-2
LJ = LCOOL - 3
LOUT = LCOOL - 4
IF (IHC(IS).NE.2) GO TO 320
C
C FORCED CONVECTION HC:
TMP = (T(2, ISLICE, LCOOL) + T(2, ISLICE, LIN)) / 2.
CALL GASTBL(TMP, C, CP, GAM, PD, R, XMU)
RE(IS) = 12. * 3600. * ABS(WCROSS(2, ISLICE, IS)) * DH(IS) / (A(LCOOL) * XMU)
HC(IS) = 0.023 * 12. * (C/DH(IS)) * (RE(IS)**0.8) * (PD**0.333)
C
320 CONTINUE
IF (IS.GE.JS) GO TO 322
C
C SPECIAL CASE FOR STATION NUMBER 1:
IF(IS.EQ.1) ISUP = 2
C
C IF STATION IS IS FORWARD OF FLOW SPLIT, AND ON SAME SIDE, GO TO 370
IF (ISEN.EQ.JSENS) GO TO 370
C
322 CONTINUE
IF (ISUP.LT.1) ISUP = 1
IF (IHC(ISUP).EQ.1) YIHPU = 1.0
IF (IHC(ISUP).EQ.2) YCONVU = 1.0 + RCVRY * AM2(ISUP) * (GAMC(ISUP) - 1) / 2.
IF (IHC(ISUP).XQ.3) YFINSU = 1.0
LCUP = LCOOL - 10
LCUPS = LCOOL - 11
LUP = LCOOL - 12
LDN = LCOOL + 8
IF (IS.NE.2) GO TO 324
C
C IF THIS IS STATION NUMBER 2:
LCUP = 5
LCUPS = 4
LUP = 3
324 CONTINUE
IF (IS.GT.1) GO TO 326
C
C IF THIS IS STATION NUMBER 1:
LCUP = 10
LCUPS = 9
LUP = 8
326 CONTINUE
C
IS = 1, STATION NO. 1, LEADING EDGE NODES
C
IF (IS.NE.JS) GO TO 330
C
** THIS BLOCK COMPUTES TCOf ELEMENTS FOR THE STATION AT WHICH THE FLOW SPLITS **
C
328 CONTINUE
DX1 = DLX(LUP)
IF (DX1.EQ.0.0) DX1 = DLX(L)
DX2 = DLX(LDN)
IF (DX2.EQ.0.0) DX2 = DLX(L)
DX3 = DLX(LUP-2)
IF (DX3.EQ.0.0) DX3 = DLX(LOUT)

DX4 = DLX(LDN-2)
IF (DX4.EQ.0.0) DX4 = DLX(LOUT)

DX5 = DLX(LUP+1)
IF (DX5.EQ.0.0) DX5 = DLX(LIN)

DX6 = DLX(LDN+1)
IF (DX6.EQ.0.0) DX6 = DLX(LIN)

DX9 = DLX(LUP-1)
IF (DX9.EQ.0.0) DX9 = DLX(LOUT)

DX10 = DLX(LDN-1)
IF (DX10.EQ.0.0) DX10 = DLX(LIN)

DX11 = DLX(LJ)
IF (DX11.EQ.0.0) DX11 = DLX(LOUT)

TRTRMC = 0.0
IF (DLTY.EQ.0.0 AND TYNE.GE.0.) TRTRMC=(3600./144.)*RHOC*SPHTC*
Z (TAU(LOUT)**2)/(4.*AKC(ISLICE,IS)*DLTYNE)

TCOF(LOUT,13) = -TREPS + TRTRMC*CURV

TCOF(LOUT,12) = TRTRMC*CURV + TREPS*(1.0 + (1.0 - BTA)*HG(IS)*TAU(LOUT))
Z = (12.*AKC(ISLICE,IS))

Z = TCOF(LOUT,9) = -(1.-BTA)*TREPS*FLMEPF(IS)*HG(IS)*TAU(LOUT) /

Z = (12.*AKC(ISLICE,IS))

TCOF(LOUT,24) = (BTA*GJ(IS) + (1.0 - BTA)*HG(IS)*TG(IS))
Z = (1.0 - FLMEPF(IS))

Z = TCOF(LOUT,13) = TAU(LOUT)/(2.*DX1)*2.*TAU(L)/(DX1+DX2+DX9+DX10)

Z = TCOF(LJ,12) = - TCOF(LJ,11) - TCOF(LJ,13)

Z = TCOF(LJ,24) = (1.-TREPS)*(T(1,ISLICE,LJ) - T(1,ISLICE,L))/

Z = TCOF(LJ,13) = TAU(L)/(12.*AKC(ISLICE,IS))

J1 = 12 - L + LUP
J2 = 12 - L + LDN

THETA1 = (DX1+DX2+DX5+DX6)/(DX1+DX2+DX9+DX10)

THETA2 = ((TAU(L)+TAU(LUP))/(2.*DX1))**2.*TAU(L)/(DX1+DX2+DX9+DX10)

THETA3 = ((TAU(L)+TAU(LDN))/(2.*DX2))**2.*TAU(L)/(DX1+DX2+DX9+DX10)

THETA6 = 24.*TAU(L)/(AKW(ISLICE,IS)*S(ISLICE)*(DX1+DX2+DX9+DX10))

THETA4 = 0.0

THETA5 = 0.0

HUB1 = 0.0

HUB3 = 0.0

TIP1 = 0.0

TIP3 = 0.0

IF (ISLICE.EQ.1) GO TO 3290

THETA4 = (TAU(L)/S(ISLICE)) * (TAU(L)/(S(ISLICE)+S(ISLICE-1)))

127
* \[ \frac{(2. + \text{DRX} + \text{DXZ} + \text{DXI})}{(\text{DX1} + \text{DX2} + \text{DX9} + \text{DX10})} \]

IF (ISLICE.EQ.ISLICE) GO TO 3292
THETA5 = THETA4*{(S(ISLICE) + S(ISLICE-1)))/(S(ISLICE) + S(ISLICE+1))}
TBELOW = T(1, ISLICE-1, L)
TABOVE = T(1, ISLICE+1, L)
GO TO 3294

C FOR THE SLICE AT THE HUB END OF THE BLADE:
3290 CONTINUE
IF (IHUB.EQ.1) HUB1 = \((\text{DX1} + \text{DX2})/(\text{DX1} + \text{DX2} + \text{DX9} + \text{DX10})\) * 
\((2. \times (\text{T(Log} / S(1))^{**2})\)
THETA5 = 0.0
IF (NSLICE.GT.1) THETA5 = \((\text{T(Log} / S(1)) \times (\text{T(Log} / (S(1) + S(2))))^**2) /
\((\text{DX1} + \text{DX2} + \text{DX9} + \text{DX10})\) 
Z = \((\text{AKW(1,IS) * 12.} \times S(ISLICE))\)
3291 CONTINUE
IF (ITIP.EQ.1) TIP1 = \((\text{DX1} + \text{DX2})/(\text{DX1} + \text{DX2} + \text{DX9} + \text{DX10})\) * 
\((2. \times (\text{T(Log} / S(NSLICE)))^{**2})\)
TABOVE = T(1, ISLICE+1, L)
Z = \((\text{AKW(NSLICE, IS) * 12.} \times S(NSLICE))\)
3292 CONTINUE
THETA9 = 0.0
IF (DLTYME.GT.0.0 AND TYME.GE.0.) THETA9 = 2.3600.*HOM*SPHMT* 
\((\text{DX1} + \text{DX2}) \times (\text{T(Log}) ** 2) / (144.*\text{AKW(ISLICE, IS)} \times 
\((\text{DX1} + \text{DX2} + \text{DX9} + \text{DX10}) \times \text{DLTYME})\)
TCOF(L, 11) = 1.0*TCRPS
TCOF(L, 13) = THETA1*TCRPS
TCOF(L, 11) = THETA2*TCRPS
TCOF(L, 12) = THETA3*TCRPS
TCOF(L, 12) = (-1.0 - THETA1 - THETA2 - THETA3 - THETA4 - THETA5 - 
\((\text{HUB1} + \text{TIP1}) \times \text{TCRPS} - \text{THETA9}\)
TCOF(L, 24) = QSNK(IS) * THETA6 - (THETA4*HUB1) \times \text{TBELOW}
Z = \((\text{T(Log} + \text{TIP1}) \times \text{TABOVE} - \text{QHUB(IS) * HUB3 + QTIP(IS) * TIP3}) \times 
\((\text{HUB1} + \text{TIP1}) \times \text{TABOVE} + \text{THETA2} + (1.0 \text{THETA3} \times \text{THETA4} - \text{THETA5} - 
\((\text{HUB1} + \text{TIP1}) \times \text{TABOVE} - \text{THETA9})\)
Z = \((\text{DX5} + \text{DX6}) \times S(ISLICE)/2.\)
AHTRN1 = \((\text{DX5} + \text{DX6}) \times S(ISLICE)/2.\)
THETA8 = 2.*HX*HC(IS) * AHTRN1 * TAU(L) / (12.*AKW(ISLICE, IS) * S(ISLICE))

128
Z = *(DX1+DX2+DX5+DX6))
TCOF(LIN,11) = TREP

TCOF(LIN,24) = -THETA*TOG - (1.0 - TREP)*(T(1,ISLICE,L) - T(1,ISLICE,LIN)*(1.0 + THETA))

TCOF(LIN,12) = (-1.0 - THETA)*TREP

TCOF(LCOOL,12) = 1.0
TCOF(LCOOL,24) = TOG

IF (TMYE.GT.0.) RCHRD = (144./3600.)*AKW(ISLICE,IS)*DLTYME/

Z = (RHOH*SPHTM*(((DX1+DX2)/2.)**2))

IF (TMYE.GT.0.) RTRNV = (144./3600.)*AKW(ISLICE,IS)*DLTYME/

Z = (RHOH*SPHTM*(TAU(L)**2))

IF (RCHRD.GT.RCHRD) RCHRD = RCHRD
IF (RTRNV.GT.RTRNV) RTRNV = RTRNV

GO TO 440

C

CONTINUE

DX1 = DLX(L)
DX2 = DLX(LDN)
DX3 = DLX(LOUT)
DX4 = DLX(LDN-2)
DX5 = DLX(LIN)
DX6 = DLX(LDN+1)
DX7 = DLX(LCOOL)
DX9 = DLX(LJ)
DX10 = DLX(LDN-1)
IF (IS.GT.1) GO TO 332

IF THIS IS STATION NUMBER 1:

C

CONTINUE

LCUPP = LCOOL - 5
LCOOLP = LCOOL + 4
J1 = 12 - L + LUP
J2 = 12 - L + LDN
J4 = 12 - LCOOL + LCUPS
J5 = 12 - LCOOL + LCUP
J6 = 12 - LCOOL + LCUPP
J8 = 12 - LCOOL + LCOOLP

C

CONTINUE
\[ A_{\text{TRR}} = (D_{5} + D_{6})S(\text{ISLICE})/2. \]

\[ A_{1} = \text{SPAN}D_{5}/2. \]

\[ A_{2} = A_{1} \]

\[ A_{3} = 0.0 \]

\[ A_{4} = 0.0 \]

\[ \text{C IF } (\text{IHC(IS)) .EQ. 3. AND. IS .LE. NFWD} \]

\[ \text{CALL HCPINS(IS,DLTAN,LCOOL,LCUP,LIN,LCOOLP,PINS,EPAREA)} \]

\[ \text{IF (IHC(IS-2) .EQ. 3, A1 = EPAREA(IS-2)/2.} \]

\[ \text{IF (IHC(IS) .EQ. 3, A2 = EPAREA(IS)/2.} \]

\[ \text{IF (IHC(IS) .EQ. 3, A1 = EPAREA(IS)} \]

\[ I_{3} = 12 - \text{LIN} + \text{LCOOL} \]

\[ \text{CONTINUE} \]

\[ \text{C CURV} = 1.0 + (D_{9} + D_{10})/(D_{3} + D_{4}) \]

\[ \text{C***********************************************************************} \]

\[ \text{C FOR MID-METAL NODE (L):} \]

\[ \text{C FOR THE COOLANT NODE (LCOOL):} \]

\[ \text{C FOR OUTSIDE NODE:} \]

\[ \text{C AT JUNCTION OF COATING AND METAL, NODE LJ:} \]

\[ \text{TCOF(LJ,11) = TREPS} \]

\[ \text{TCOF(LJ,13) = TREPS*AK(C(ISLICE,IS)/AK(C(ISLICE,IS)))} \]

\[ \text{Z} \]

\[ \text{---TCOF(LJ,12) = -(1.0 - BATA)*TBEPS*FLP(EF(IS))**2)/(12.*AK(C(ISLICE,IS)))} \]

\[ \text{Z} \]

\[ \text{---TAU(LJ)}/(12.*AK(C(ISLICE,IS))) \]

\[ \text{C} \]

\[ \text{C} \]

\[ \text{C} \]

\[ \text{C} \]

\[ \text{C} \]

\[ \text{C} \]

\[ \text{C} \]

\[ \text{C} \]

\[ \text{C} \]

\[ \text{C} \]

\[ \text{C} \]

\[ \text{C} \]

\[ \text{C} \]

\[ \text{C} \]

\[ \text{C} \]

\[ \text{C} \]
TCOP(LJ,24) = (1.-TREPS)*(((T(1,ISLICE,LJ)-T(1,ISLICE,LOUT)) + (T(1,ISLICE,LJ)-T(1,ISLICE,L))*TCOP(LJ,13)/TREPS)

C FOR MID-METAL NODE:

C

`TCOP(LJ,24) = (1.-TREPS)*(((T(1,ISLICE,LJ)-T(1,ISLICE,LOUT)) + (T(1,ISLICE,LJ)-T(1,ISLICE,L))*TCOP(LJ,13)/TREPS)`

C

\[
\text{THETA1} = \frac{(DX1+DX2+DX5+DX6)/(DX1+DX2+DX9+DX10)}{((TAU(L)+TAU(LUP))/(2.*DX1))*2.*TAU(L)/(DX1+DX2+DX9+DX10)}
\]

\[
\text{THETA2} = \frac{((TAU(L)+TAU(LUP))/(2.*DX1))*2.*TAU(L)/(DX1+DX2+DX9+DX10)}{((TAU(L)+TAU(LUP))/(2.*DX1))*2.*TAU(L)/(DX1+DX2+DX9+DX10)}
\]

\[
\text{THETA3} = 0.0
\]

\[
\text{THETA4} = 0.0
\]

\[
\text{THETA5} = 0.0
\]

\[
\text{HUB1} = 0.0
\]

\[
\text{HUB3} = 0.0
\]

\[
\text{TIP1} = 0.0
\]

\[
\text{TIP3} = 0.0
\]

\[
\text{IF (ISLICE.EQ.1) GO TO 3410}
\]

C FOR A SLICE THAT IS NOT AT THE HUB OF THE BLADE:

C

\[
\text{THETA4} = \frac{(TAU(L)/S(ISLICE))*((TAU(L)/S(ISLICE)+S(ISLICE-1)))*2.*(DX1+DX2)/(DX1+DX2+DX9+DX10)}{(AKY(ISLICE,IS)*S(ISLICE))*(DX1+DX2+DX9+DX10)}
\]

C FOR THE SLICE AT THE HUB END OF THE BLADE:

C

\[
\text{TBELOW} = T(1,ISLICE-1,L)
\]

\[
\text{TABOVE} = T(1,ISLICE+1,L)
\]

\[
\text{GO TO 3414}
\]

C

\[
\text{THETA5} = 0.0
\]

C FOR THE SLICE AT THE TIP OF THE BLADE:

C

\[
\text{TBELOW} = T(1,1,L)
\]

\[
\text{TABOVE} = T(1,1,L)
\]

\[
\text{IF (NSLICE.GT.1) GO TO 3414}
\]

C FOR THE SLICE AT THE BLADE TIP, (IF THERE ARE MORE THAN 1 SLICES BEING CONSIDERED) : ************

C

\[
\text{TBELOW} = T(1,ISLICE-1,L)
\]

\[
\text{IF (TIP.EQ.1) TIP1 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*2.*(TAU(L)/S(NSLICE)**2)}
\]

\[
\text{IF (NSLICE.GT.1) TBELOW = T(1,ISLICE-1,L)}
\]

131
IF (ITIP.EQ.3) TIP3 = ((DX1+DX2)/(DX1+DX2+DX9+DX10)) * (TAU(L)**2) / (AKW(NSLICE,IS) * 12. * S(NSLICE))

Z
TABOVE = T(1,ISLICE,L)
IF (ITIP.EQ.1) TABOVE = TTIP(IS)

3414 CONTINUE

THETA9 = 0.0
IF (DLTYGE.0.0.AND.TYME.0.) THETA9 = 2.*3600.*RHOM*SPHTM*
Z (DX1+DX2)*(TAU(L)**2)/(14.*AKW(ISLICE,IS))
Z (DX1+DX2+DX9+DX10)*DLTYGE)

3416 CONTINUE

TCOF(L,11) = 1.0*TREPS
TCOF(L,13) = THETA1*TREPS
TCOF(L,J1) = THETA2*TREPS
TCOF(L,J2) = 0.0
TCOF(L,12) = (-1.0 - THETA1 - THETA2 - THETA3 - THETA4 - THETA5)*ENDEFF + HUB1 - TIP1)*TREPS - THETA9
TCOF(L,24) = QSNK(IS)*THETA6 - (THETA4+HUB1)*TBELOW -
Z (THETA5*THETA6 - (THETA4+HUB1)*TBELOW -
Z - (1.-TREPS)*T(1,ISLICE,LUP)*THETA1 - T(1,ISLICE,LJ) +
Z T(1,ISLICE,LIN)*THETA1 + T(1,ISLICE,LDN)*THETA3 +
Z T(1,ISLICE,L)*(((1.0-TREPS)*T(1.0-TREPS)*T(1.0-TREPS)*T(1.0-TREPS))*T(1.0-TREPS)*T(1.0-TREPS))
Z THETA4+THETA5+HUB1+TIP1+ENDEFF + THETA9 - TG(IS)*ENDEFF - ENDFLX
IF (IS.LT.NSTA-1) TCOF(L,J2) = THETA3*TREPS

PUMP(IS) = (.1047198*WS)**2*RR(IS)*(RR(IS)-RR(IS-2))

342 THETA8 = 2.*HX*HC(IS)*AHTRN1*TAU(L)/(12.*AKW(ISLICE,IS)*
Z S(ISLICE)*(DX1+DX2+DX5+DX6))

TCOF(LIN,11) = TREPS
TCOF(LIN,12) = (-1.0 - THETA8)*TREPS
TCOF(LIN,13) = THETA8*(YCONV + YFINS)*TREPS
TCOF(LIN,24) = -YIMP*THETA8*T0G
Z - (1.-TREPS)*T(1,ISLICE,L) - T(1,ISLICE,LIN) +
Z + THETA8*(YCONV+YFINS)*T(1,ISLICE,LCOOL))

343 CONTINUE

C IF THIS IS A TRAILING EDGE, PRESSURE SIDE, STATION, COOLANT NODE
C COINCIDES WITH SUCTION SIDE COOLANT NODE.
IF (ISENS.EQ.0) GO TO 343
TCOF(LCOOL,7) = -1.0
TCOF(LCOOL,12) = 1.0
TCOF(LCOOL,24) = 0.0
IF (ISENS.EQ.1) GO TO 430

CONTINUE
C FOR COOLANT NODE:
C FOR THE SPECIAL CASE OF IS = NFWD+1, GO TO 350

FILMW = WFC(IS)
IF (IS.GT.NFWD) FILMW = FILMW + WFC(IS+1)
IF (IS.EQ.NFWD+1) GO TO 350

WXCP = WCROS(2,ISLICE,IS)*144.*CPC(IS)*3600.
IF (IS.EQ.ICONF.OR.IS.EQ.ICOMS) WXCP = WJ(ISLICE,JS)*144.*CPC(IS)*3600./Z.

DEFINE A COOLANT SIDE TRANSIENT TERM, TRTRMG

TRTRMG = 0.0
IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRMG = (1.+CPC(IS-2)/CPC(IS))
Z (P(1,ISLICE,IS-2)/T(1,ISLICE,LCOUP)) reinforcements
Z * (A(LCUP)*A(LCOOL)*DLX(LCOOL)/(16.*DLTYME**2))
Z WCRS(2,ISLICE,IS)*12.)

TCOF(LCOOL,J4) = HX*TREPS*HC(ISUP)*A1/WXCP
TCOF(LCOOL,J5) = 144.*3600.*TrEps*WCROS(2,ISLICE,ISUP)*12.0
Z (1.+AM2(ISUP)*(GAMC(ISUP)-1.)/2.)*CPC(ISUP)/WXCP
Z - HX*TREPS*HC(ISUP)*(YCONV+YFINSU)*(A1+A3)/WXCP - TRTRMG
TCOF(LCOOL,11) = HX*TREPS*HC(IS)*A2/WXCP
TCOF(LCOOL,12) = TREPS*((-FILMW*144.*3600.*CPC(IS)/WXCP - TRTRMG
Z - (GAMC(IS)-1.)/2.)*AM2(IS)
Z - HX*HC(IS)*(A2+A4)*(YCONV+YFINS)/WXCP) - TRTRMG
TREDGE = 0.0
IF (IS.GT.NFWD) TREDGE = ((1. -TREPS)*HX/WXCP) *(T(1,ISLICE,LCOUP)
Z HC(ISUP)*A3 + T(1,ISLICE,LCOOP)*HC(IS)*A4)

*** OF THE TERMS YIMP,YFINS,YCONV, ONLY ONE CAN BE NON-ZERO AT A TIME

IF (TYME.GT.0.) RCHRD = (144./3600.)*AKW(ISLICE,IS)*DLTYME/
Z (RHOM*SPHTM*(((DX1+DX2)/2.)**2))
IF (TYME.GT.0.) RTRNV = (144./3600.)*AKW(ISLICE,IS)*DLTYME/
Z (RHOM*SPHTM*(TAU(L)**2))
IF (RCHRD.GT.RCHRDH) RCHRDH = RCHRD
IF (RTRNV.GT.RTRNVH) RTRNVH = RTRNV
GO TO 430

GO TO 4801
GO TO 4802
GO TO 4803
GO TO 4804
GO TO 4805
GO TO 4806
GO TO 4807
GO TO 4808
GO TO 4809
GO TO 4810
GO TO 4811
GO TO 4812
GO TO 4813
GO TO 4814
GO TO 4815
GO TO 4816
GO TO 4817
GO TO 4818
GO TO 4819
GO TO 4820
GO TO 4821
GO TO 4822
GO TO 4823
GO TO 4824
GO TO 4825
GO TO 4826
GO TO 4827
GO TO 4828
GO TO 4829
GO TO 4830
GO TO 4831
GO TO 4832
GO TO 4833
GO TO 4834
GO TO 4835
GO TO 4836
GO TO 4837
GO TO 4838
GO TO 4839
GO TO 4840
GO TO 4841
GO TO 4842
GO TO 4843
GO TO 4844
GO TO 4845
GO TO 4846
GO TO 4847
GO TO 4848
GO TO 4849
GO TO 4850
GO TO 4851
GO TO 4852
GO TO 4853
GO TO 4854
GO TO 4855
GO TO 4856
GO TO 4857
GO TO 4858
GO TO 4859
GO TO 4860

133
CONTINUE
IF (IHC(IS).EQ.3) CALL HCPINS(IS,DELTAN,LCOOL,LCUP,LIN,LCOOLP,PINS)
            ,EFAEPA)
            AHTRN1 = EFAEPA(IS)
            IF (IHC(IS).EQ.3) A2 = EFAEPA(IS)/2.
            A3 = SPAN*DLX(LCOOL)/2.
            IF (IHC(IS-1).EQ.3) A3 = EFAEPA(IS-1)/2.
            A4 = SPAN*DLX(LCOOL)/2.
            IF (IHC(IS).EQ.3) A4 = EFAEPA(IS+1)/2.
            IF (IHC(IS).EQ.3) GO TO 360
            IF (IHC(IS).EQ.2) HC(IS) = HCPS(IS,LCOOL,LIN)
            AHTRN1 = (DX5*DX6)*S(ISLICE)/2.
CONTINUE
IF (IHC(IS-1).EQ.1) YIMPPU = 1.0
IF (IHC(IS-1).EQ.2) YCNVUU=1.0*CPC(IS-1)*GAMC(IS-1-1)/2.
IF (IHC(IS-1).EQ.3) YFNSUU = 1.0
IF (IHC(IS-1).EQ.4) CALL ACPINS(IS,DELTAN,LCOOL,LCUPP,LIN)
            EFAEPA)
            AHTRN1 = EFAEPA(IS)
            IF (IHC(IS).EQ.3) A2 = EFAEPA(IS)/2.
            A3 = SPAN*DLX(LCOOL)/2.
            IF (IHC(IS-1).EQ.3) A3 = EFAEPA(IS-1)/2.
            A4 = SPAN*DLX(LCOOL)/2.
            IF (IHC(IS).EQ.3) A4 = EFAEPA(IS+1)/2.
            IF (IHC(IS).EQ.3) GO TO 360
            IF (IHC(IS).EQ.2) HC(IS) = HCPS(IS,LCOOL,LIN)
            AHTRN1 = (DX5*DX6)*S(ISLICE)/2.
CONTINUE
IF (IHC(IS-1).EQ.1) YIMPPU = 1.0
IF (IHC(IS-1).EQ.2) YCNVUU=1.0*CPC(IS-1)*GAMC(IS-1-1)/2.
IF (IHC(IS-1).EQ.3) YFNSUU = 1.0
IF (IHC(IS-1).EQ.4) CALL ACPINS(IS,DELTAN,LCOOL,LCUPP,LIN)
            EFAEPA)
            AHTRN1 = EFAEPA(IS)
            IF (IHC(IS).EQ.3) A2 = EFAEPA(IS)/2.
            A3 = SPAN*DLX(LCOOL)/2.
            IF (IHC(IS-1).EQ.3) A3 = EFAEPA(IS-1)/2.
            A4 = SPAN*DLX(LCOOL)/2.
            IF (IHC(IS).EQ.3) A4 = EFAEPA(IS+1)/2.
            IF (IHC(IS).EQ.3) GO TO 360
            IF (IHC(IS).EQ.2) HC(IS) = HCPS(IS,LCOOL,LIN)
            AHTRN1 = (DX5*DX6)*S(ISLICE)/2.
C 370 CONTINUE
Z ISUP = IS + 2
Z LCUP = LCOOL + 10
Z LCUPS = LCOOL + 9
Z LUP = LCOOL + 8
Z LDN = LCOOL - 12
Z IF (IS.EQ.2) LDN = 3
Z IF (IS.EQ.1) LDN = 8
Z IF (IHC(ISUP).EQ.1) YIMPU = 1.0
Z IF (IHC(ISUP).EQ.2) YCONVU = 1.0 + RCVRY * AM2(ISUP) * (GAMC(ISUP) - 1.)/2.
Z IF (IHC(ISUP).EQ.3) YFINSU = 1.0
Z DX1 = DLX(LUP)
Z DX2 = DLX(L)
Z DX3 = DLX(LUP-2)
Z DX4 = DLX(LOUT)
Z DX5 = DLX(LCUPS)
Z DX6 = DLX(LIN)
Z DX7 = DLX(LCUP)
Z DX9 = DLX(LUP-1)
Z DX10 = DLX(LJ)
Z IF (IS.GT.1) GO TO 332
C
Z DX2 = DLX(8)
Z DX4 = DLX(6)
Z DX6 = DLX(9)
Z DX10 = DLX(7)
Z J1 = 22
Z J2 = 17
Z J4 = 21
Z J5 = 22
Z J6 = 8
Z J8 = 15
Z GO TO 333
C
C ISENS = 0 MEANS IS IS EVEN AND STATION IS ON SUCTION SIDE
C ISENS = 1 MEANS IS IS ODD AND STATION IS ON PRESSURE SIDE

LCOOL = 5*IS
LIN = LCOOL - 1
L = LCOOL - 2
LJ = LCOOL - 3
LOUT = LCOOL - 4
LUP = L - 10
LDN = L + 10
LCUP = LCOOL - 10
LCUPS = LCUP - 1
LCUPP = LCOOL - 6
LCOOLP = LCOOL + 4

I3 = 12 - LIN + LCOOL
J1 = 12 - L + LUP
J2 = 12 - L + LDN
J4 = 12 - LCOOL + LCUPS
J5 = 12 - LCOOL + LCUP
J6 = 12 - LCOOL + LCUPP
J8 = 12 - LCOOL + LCOOLP
J9 = 16

A1 = SPAN*DLX (LIN)/2.
A2 = A1
A3 = SPAN*DLX (LCOOLP)/2.
A4 = A3

IF (IHC(IS-2).EQ.3) A1 = EFAREA(IS-2)/2.
IF (IHC(IS-2).EQ.3) A3 = EFAREA(IS-1)/2.

DX1 = DLX(L)
DX2 = DLX(LDN)
DX3 = DLX(LOUT)
DX4 = DLX(LDN-2)
DX5 = DLX(LIN)
DX6 = DLX(LDN-1)
DX7 = DLX(LCOOL)
DX9 = DLX(LJ)
DX10 = DLX(LDN-1)

IF (IHC(ISUP).EQ.1) YIMFU = 1.0
IF (IHC(ISUP).EQ.2) YCONVU = 1.0 + 1.0 + AM2(ISUP)*(GAMC(ISUP) - 1.)/2.
IF (IHC(ISUP).EQ.3) YFINSU = 1.0
IF (IS.LT.NSTA-1) GO TO 390

C FOR THE LAST STATIONS IN THE TRAILING EDGE:

DX2 = 0.0
DX4 = 0.0
DX6 = 0.0
DX10 = 0.0

390 CONTINUE
IF (IHC(IS).EQ.3) GO TO 420
IF (IHC(IS).EQ.2) GO TO 410
GO TO 340

C*** HCPRCD COMPUTES HC FOR FORCED CONVECTION

C DX2 = 0.0
DX4 = 0.0
DX6 = 0.0
DX10 = 0.0

CONTINUE
IF (IHC(IS).EQ.3) GO TO 420
IF (IHC(IS).EQ.2) GO TO 410
GO TO 340

C**** HCPTNS COMPUTES HC FOR A PIN FIN SURFACE OR FOR
TURBULENT FORCED CONVECTION CHANNEL FLOW

410 CONTINUE
TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.
CALL GASTB(TMP,C,CP,GAM,PD,8,XMU)
RE(IS) = 12.*3600.*ABS(WCRS(2,ISLICE,IS))*DH(IS)/(A(LCOOL)*XMU)

HC(IS) = .023*12.*(C/DH(IS))*(RE(IS)**.8)*(PD**.333)
AHTRN1 = (DX5 + DX6)*SPAN/2.
GO TO 340

C**** SUBROUTINE HCPTNS COMPUTES HC FOR A PIN FIN SURFACE OR FOR
TURBULENT FORCED CONVECTION CHANNEL FLOW

420 CONTINUE
IF(ISENS.EQ.0) GO TO 424
AHTRN1 = EFAREA(IS)
HC(IS) = HC(IS-1)
GO TO 340

424 CALL HCPTNS(IS,DELTAN,LCOOL,LCUP,LIN,LCOOLP,PINS,EFAREA)
AHTRN1 = EFAREA(IS)
IF (IS.GE.NSTA-1) AHTRN1 = AHTRN1/2.
A2 = EFAREA(IS)/2.
A4 = EFAREA(IS+1)/2.
GO TO 340

C
C
C L30 CONTINUE
C 440 CONTINUE
C 450 CONTINUE
C C RETURN
C END

C---- SOURCE.NTCOFTT
SUBROUTINE TCOEF(IWRITE,WS,NIT,IPLLOT,ALPH2)

C
C SOURCE.NTCOFTT

DIMENSION POLD(15,80), PSAV(5), X(80), ALPH2(4), DELTAN(15),
Z, TOTC(80), JSO(15)
DIMENSION PEXOLD(15), PIMOLD(15)
REAL*8 TCOP
COMMON /MATRIX/ TCOP(400,30)

COMMON /PRPS/, CPO, GAMO, DP(80), SP(80), RE(80), TE

137
BTA = 0. INDICATES THAT A HEAT TRANSFER COEFFICIENT BOUNDARY CONDITIONS IS SPECIFIED
1. INDICATES THAT A HEAT FLUX IS SPECIFIED ON THE GAS BOUNDARY
UNITS EXPECTED ON THE FOLLOWING INPUT DATA ARE: LENGTHS ARE ALL IN INCHES--DLX, TAU, SPAN, DH, DPX
TEMPERATURES ARE ABSOLUTE (R)
MASS FLOWS ARE ALL IN (LBM/SEC)--WCROS, WJ, WFC, WDUMP
HEAT TRANSFER COEFFICIENTS ARE IN BTU/(HR*FT**2*R)--HG, HC--GAS SIDE AND COOLANT SIDE
HEAT FLUX QG IS IN BTU/(AR*FT**2)
HEAT SINK, QSNK, IS IN BTU/(HR)
THERMAL PROPERTIES ARE: CONDUCTIVITY-AKW-BTU/(HR*FT*R)
HEAT CAPACITY-CP-BTU/(LBM*R)

SET UP FIRST GUESS AT TEMPERATURE DISTRIBUTION
ASSUME COOLANT TEMPERATURE IS CONSTANT, = PLENUM STATIC TEMPERATURE
ASSUME METAL TEMPERATURE IS CONSTANT, = 2200. R

CONTINUE
C-- FOR TRANSIENT CASES, ADJUST INITIAL GUESS OF PRESSURE DISTRIBUTION
C-- BASED ON THE VARIATION OF SUPPLY AND EXIT PRESSURES.
C IF (TYME.GT.0.0) GO TO 234
C
C-- FOR STEADY STATE, ONLY INITIALIZE P'S ON FIRST OVERALL LOOP.
C IF (NIY.GT.1) GO TO 255
PEXOLD(ISLICE) = PEXIT(ISLICE)
PIMOLD(ISLICE) = PIM
WRITE(6,165) (I,T(1,ISLICE,I),I=1,NODST)
165 FORMAT(1H1, ' ASSUMED INITIAL TEMPERATURE DISTRIBUTION, (NODE NO. ',NTCOPTT 5172
Z 'I',T) '/7 ('I3',',F8.2',')
170 CONTINUE
C---- PRESSURE INITIALIZATION, GIVEN PIM (PLENUM STATIC PRESSURE) AND
C---- PEXIT (GAS SIDE STATIC PRESSURE AT TRAILING EDGE), FIT PRESSURE TO
C A
C---- CUBIC EQN. OF THE FORM A+B*X**3 , ASSUMING 85% OF THE PRESSURE DROP
C---- OCCURS IN THE TRAILING EDGE CHANNEL
C
X(1) = 0.0
X(2) = DLX(10)
DO 180 I = 3,NFWD
X(I) = DLX(5*I) + X(I-2)
180 CONTINUE
P(1,ISLICE,1) = PIM - (PIM-PEXIT(ISLICE))*15
P(1,ISLICE,N) = PEXIT(ISLICE)
P(1,ISLICE,N+1) = P(1,ISLICE,N)
P(1,ISLICE,NFWD-1) = P(1,ISLICE,1) - 2*(P(1,ISLICE,1)-PEXIT(ISLICE))
190 P(1,ISLICE,NFWD) = P(1,ISLICE,NFWD-1)
DO 200 I = 3,NFWD,2
P(1,ISLICE,I) = P(1,ISLICE,1) - (P(1,ISLICE,1)-P(1,ISLICE,NFWD-1))
Z (X(I)/X(NFWD-1))**3 NTCOPTT 5194
Z ISTRT = NFWD+1
IFNL = N
DO 232 I = ISTRT,IFNL,2
P(1,ISLICE,I) = P(1,ISLICE,1) - (P(1,ISLICE,1)-P(1,ISLICE,NFWD-1))
Z (X(I-1)/X(NFWD-1))**3 NTCOPTT 5196
GO TO 240
C C---- FOR TRAILING EDGE CHANNEL, X VALUES ARE RELATIVE TO END OF
C IMPINGEMENT CHANNEL
C
X(NFWD+1) = (DLX(NODSF) + DLX(NODSF-5))/2.
210 ITEM = NFWD+3
DO 220 I = ITEM,N,2
LCOOL = 5*I
X(I) = X(I-2) + DLX(LCOOL)
220 CONTINUE
DO 230 I = ISTRT,IFNL,2
P(1,ISLICE,I) = P(1,ISLICE,1) - (P(1,ISLICE,1)-P(1,ISLICE,NFWD-1))
Z (X(I)/X(N))**3 NTCOPTT 5210
Z DLTAPC = .84*(PIM-PIMOLD(ISLICE))
PIMOLD(ISLICE) = PIM
139
DLTAPC = PEXIT(ISLICE) - PEXOLD(ISLICE)  
PEXOLD(ISLICE) = PEXIT(ISLICE)  
DO 235 I = 1, NFWD  
235  
P(2,ISLICE,I) = P(2,ISLICE,I) + DLTAPC  
ISTRAT = NFWD + 1  
IFNL = NSTA - 1  
DO 236 I = ISTRAT, IFNL, 2  
236  
P(2,ISLICE,I) = P(2,ISLICE,I) + DLTAPC* (1.0 - X(I)/X(IFNL)) + Z  
   DLTAPC*X(I)/X(IFNL)  
GO TO 255  
C  
C----INITIALLY, THE FLOW SPLIT AT THE LEADING EDGE IS ASSUMED  
C TO BE 50/50 (DELTA=.5)  
C----IDEALT COUNTS THE NUMBER OF FLOW SPLIT ITERATIONS. IF NO  
C CONVERGENCE, IDELT IS SET NEGATIVE.  
C----DELTAN IS THE FRACTION OF FLOW TO THE SUCTION SIDE (EVEN  
C NUMBERED STATIONS)  
C----IVERGE COUNTS THE NUMBER OF ITERATIONS A GIVEN FLOW SPLIT  
C IFNL = THE NUMBER OF FLOW CHANNEL NODES, USED IN PRESSURE CALCULATIONS  
C IF (NIT. EQ. 1. AND. TNYME.LT.0.0) DELTAN(ISLICE) = .5  
275  
   IFNL = NSTA - 3  
   IVERGE = 0  
   IDELT = 1  
   JS = 1  
   IF (NIT.CT.1) JS = JSO(ISLICE)  
290  
   IVERGE = IVERGE + 1  
   CONTINUE  
   JSENS = JS - 2*(JS/2)  
C  
C----SUBROUTINE FLOWS COMPUTES JET FLOW RATES, CROSSFLOW RATES, AND  
C THE SQUARE OF THE MACH NUMBER  
C----SUBROUTINE HCQOL COMPUTES IMPINGEMENT REGION HEAT TRANSFER COEFF'NTS
CALL HCOOL(JS)

CONTINUE

CALL THERCON

SUBROUTINE THERCON EXTRACTS THERMAL CONDUCTIVITIES FROM INPUT TABLES AKCTBL AND AKWTBL.

CONTINUE

C----> CHECK TO MAKE SURE STAGNATION HC IS LESS THAN MAXIMUM PHYSICALLY POSSIBLE VALUE.

IF (JS.EQ.1) ASTG = (DLX(9)+DLX(14))*SPAN/2.0
IF (JS.GT.1) ASTG = (DLX(5*JS-1)+DLX(5*JS+9))*SPAN/2.0
HSTGMAX = WJ(ISLICE,JS)*CF0*144.*3600./ASTG
IF (HC(JS).LE.HSTGMAX) HC(JS) = HSTGMAX

CONTINUE

CALL TAPF4Y (JS,JSENS,DLTX)

CONTINUE

CALL GAUSS(NODST,23)
DO 350 I = 1,NODST
T(2,ISLICE,I) = TCOF(I,24)
IF (T(2,ISLICE,I).LE.0.0) T(2,ISLICE,I) = TOG
350 CONTINUE

CONTINUE

IF (ABS((T(2,ISLICE,NSAVE)-TSAVE)/TSAVE).GT.0.05)
Z CALL FLOWS (JS,DELTAH,ICHOK,AMCHOK)

CONTINUE

IF (ICHOK.EQ.0) GO TO 370
WRITE (8,365) ISLICE,IVERGE,IDELT,NIT,ICHOK,AMCHOK
365 FORMAT(/10X,'SLICE ',',I2,' IS CHOKE FOR IVERGE = ',',I3,' IDELT = ',',I3,' NIT = ',',I3,' ICHOK = ',',I4,' M**2 = ',F6.3)
370 CONTINUE

CALL Compute New Pressures

CALL PARMAY (JS,JSENS,ICHOK)

CALL Solve The TCOF Array and Compute New Pressures

CONTINUE

CALL GAUSS(IFNL,19)

CONTINUE

DO 460 I = 1,IFNL
450 P(2,ISLICE,I) = TCOF(I,20)
460 CONTINUE

P(2,ISLICE,IFNL+1) = P(2,ISLICE,IFNL)
P(2,ISLICE,NSTA-1) = PEXIT(ISLICE)
P(2,ISLICE,NSTA) = PEXIT(ISLICE)
470 CONTINUE

IF (IWRITE.EQ.2) CALL WROUT (IDELT,JS,DELTAH,IVERGE)
IF (IPLOT.EQ.2) CALL PLOTHP(ALPH2)

CONTINUE

CONTINUE

141
C CHECK OVERALL CONVERGENCE
C CALCULATIONS ARE REPEATED UNTIL THE PRESSURE AT STATION 1 (NODE 5) HAS STABILIZED FOR FOUR ITERATIONS. THEN WE GO TO CHECK THE FLOW SPLIT.
C
DO 490 I=1,3
K=5-I
490 PSAV(K)=PSAV(K-1)
PSAV(I)=P(2,ISLICE,JS)
C
DIFO=0.0
DO 500 I=1,3
JJ=I+1
DO 500 K=JJ,4
DIFN=ABS(PSAV(I)-PSAV(K))
IF(DIFO.LT.DIFN) DIFO=DIFN
500 CONTINUE
C
510 DIFO=DIFO/(PIM-PEXIT(ISLICE))
IF (NT.EQ.1.AND.ISLICE.GT.1) PCNVRG = .05
EPSN = (P(2,ISLICE,NFWD-1)-P(2,ISLICE,NFWD-1))/(P(2,ISLICE,NFWD-1))
IF (IDELT.EQ.1.AND.IVERGE.LT.3.AND.TYFIE.GT.0) GO TO 516
IF (DIFO.LE.PCNVRG.AND.IVERGE.GT.4) GO TO 560
516 CONTINUE
IF (IVERGE.LT.10) GO TO 520
IF (ABS(PSAV(1)-PSAV(2)).GT.ABS(PSAV(3)-PSAV(4))) V=1.-((1.-V)/2.)
520 CONTINUE
DO 530 I = 1,NSTA
P(2,ISLICE,I) = P(2,ISLICE,I) + V*(POLD(ISLICE,I) - P(2,ISLICE,I))
P(2,ISLICE,I) = .999*PIM
T3TOT(I) = T(2,ISLICE,5*I)*(1.+GAMC(I)-1.)*AM2(I)/2.
530 CONTINUE
C
540 CONTINUE
IF (IVERGE.GT.30.OR.V.GT.95) WRITE(8,550) IVERGE,V,
550 FORMAT('*************** CONVERGENCE PROBLEMS ***************')
Z = IVERGE='1,3,'; V = 'F6.4,'; PSAV = '4(F10.2)
GO TO 290
C
C*********** ONCE PRESSURE-TEMPERATURE ITERATION HAS CONVERGED, CHECK
C THE FLOW SPLIT AND ADJUST IF NECESSARY.
C
560 CONTINUE
IF (WRITE.EQ.1) CALL WRROUT(IDELT,JS,DELTAN,IVERGE)
IF (ICHOE.GT.0) WRITE(6,565) ICHOE,AMCHOK
565 FORMAT('/10X,'MACH NO. > 1 AT STATION ','I4,',' **2 = ','F6.3')
CONTINUE
EPSN = (P(2,ISLICE,NFWD-1)-P(2,ISLICE,NFWD-1))/(P(2,ISLICE,NFWD-1))
575 CONTINUE
IF (TYSM.GT.0.0) GO TO 590
CALL FLSPLT(AJET,EPSN,ISLICE,NODSF,IDELT,JS,DELTAN,ICONV)
580 CONTINUE
IF (ICONV.EQ.1) CALL WRROUT(IDELT,JS,DELTAN,IVERGE)
IVERGE = 0
IF (ICONV.EQ.0) GO TO 290
CONTINUE
IF (TMYL.GT.0.0) CALL WRUTF((IDELT,JS,DELTAN,IVERGE)
JSCO(ISLICE) = JS
RETURN
END

C---- SOURCE. NTHRCNT
SUBROUTINE THRCON
C
C COMMON /BOUND/ BCXS(400), BCP(400), BCHGS(1000), BCHGP(1000)
Z BCTGS(1000), BCTGP(1000), THUBIN(400), THUB(80), NTHRCNT 5411
Z QHUBIN(400), QHUB(80), TTIPIN(400), TTIP(80), NTHRCNT 5413
Z QHUBIN(400), QHUB(80), TTIPIN(400), TTIP(80), NTHRCNT 5414
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5410
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5412
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5413
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5414
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5415
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5416
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5417
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5418
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5419
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5420
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5421
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5422
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5423
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5424
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5425
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5426
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5427
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5428
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5429
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5430
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5431
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5432
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5433
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5434
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5435
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5436
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5437
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5438
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5439
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5440
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5441
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5442
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5443
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5444
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5445
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5446
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5447
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5448
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5449
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5450
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5451
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5452
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5453
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5454
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5455
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5456
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5457
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5458
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5459
Z BCHGS(1000), BCHGP(1000), BCHGS(1000), BCHP(1000), NTHRCNT 5460
C  
DO 170 J = 3, JLSTM, 2
   IF (TC.GT.AKCTBL(J)) GO TO 160
C  
C FOUND LOCATION, NOW INTERPOLATE.
C  
RATIO = (TC - AKCTBL(J-2))/(AKCTBL(J) - AKCTBL(J-2))
AKC(ISLICE,I) = AKCTBL(J-1) * (AKCTBL(J+1) - AKCTBL(J-1)) * RATIO
GO TO 500
160 IF (J.LT.JLSTU) GO TO 170
C  
C TEMPERATURE IS ABOVE THE RANGE OF THE TABLE, SO EXTRAPOLATE UP.
C  
RATIO = (TC - AKCTBL(J-2))/(AKCTBL(J) - AKCTBL(J-2))
AKC(ISLICE,I) = AKCTBL(J-1) * (AKCTBL(J+1) - AKCTBL(J-1)) * RATIO
GO TO 500
170 CONTINUE
500 CONTINUE
C  
C NOW LOOK UP METAL CONDUCTIVITY IN TABLE AKWTBL.
C  
IF (TU.GT.AKWTBL(1)) GO TO 550
C  
C FOR A TEMPERATURE LOWER THAN THE BOTTOM OF THE TABLE, EXTRAPOLATE
C BELOW TABLE
RATIO = (TU - AKWTBL(1))/(AKWTBL(3) - AKWTBL(1))
AKW(ISLICE,I) = AKWTBL(2) * (AKWTBL(4) - AKWTBL(2)) * RATIO
GO TO 1000
550 CONTINUE
C  
C FIND SIZE OF TABLE
C  
DO 552 J = 3, 19, 2
   JLST = J-1
   IF (AKWTBL(J).LE.0.1) GO TO 554
552 CONTINUE
554 JLSTM = JLST-1
C  
C LOCATE WHERE TEMPERATURE FALLS IN THE TABLE AKWTBL.
C  
DO 570 J = 3, JLSTM, 2
   IF (TW.GT.AKWTBL(J)) GO TO 560
C  
C FOUND LOCATION, NOW INTERPOLATE.
C  
RATIO = (TW - AKWTBL(J-2))/(AKWTBL(J) - AKWTBL(J-2))
AKW(ISLICE,I) = AKWTBL(J-1) * (AKWTBL(J+1) - AKWTBL(J-1)) * RATIO
GO TO 1000
560 IF (J.LT.JLSTM) GO TO 570
C  
C TEMPERATURE IS ABOVE THE RANGE OF THE TABLE, SO EXTRAPOLATE UP.
C  
RATIO = (TW - AKWTBL(J-2))/(AKWTBL(J) - AKWTBL(J-2))
AKW(ISLICE,I) = AKWTBL(J-1) * (AKWTBL(J+1) - AKWTBL(J-1)) * RATIO
GO TO 1000
570 CONTINUE
1000 CONTINUE
RETURN
END
C- --- SOURCE.NWROTT
   SUBROUTINE NWROTT(IDELT, JS, DELTAN, IVERGE)

C- SOURCE.NWROTT----
C DUXR2 CFEtRIES THE IMPINGEMENT JET REYNOLDS NO. IN FROM HCcool.
C COMMON /CHKHOL/ WCHK(80), WCHKDM
C COMMON /FLMCOL/ RHOGA(60), FG(80), XFC(80), FLMEFF(80),
   XMUC(80), FME5(80), REF(E0), NFCSUP(80)
C COMMON /PPPS/ CPO, GAMO, DP(80), SP(80), BE(80),
   CPC(80), GMC(60), DUMR1(80), DUMR2(80)
C COMMON /RADL/ APLN(15), DPLN(15), RINC(15), ROUT(15),
   PIN(15), TINC(15), WS
C COMMON /TCO/ ADUMP, BTA, CD, CP, SPAN, TEG,
   GAM, VG, WHT(15,80), AKW(15,80), AK(15,80),
   HD(80), DHP(E0), DHJ(80),
   DLX(400), FP(80), HC(80), HG(80),
   P(2,15,80), PEXIT(15), PUMP(80), QG(80),
   QSNK(80), RR(80), S(15), T(2,15,400),
   TG(80), TAU(400), WFC(80),
   WJ(15,80), WSCOS(2,15,80), XN(80),
   ICOK, IFILM, IHub, ITIP,
   ISBLOK, ISLICE, NBLKSZ, NSLICE,
   NFWD, NSTA, IH(50)
C COMMON /TRNSST/ RAOC, RHOIT, SFHTC, SPHT3,
   DLTME, TYME, TEPF, TMMAX
C COMMON /UNITS/ CINCH(2), CHTC(2), CHFLX(2), CPRSR(2), CMSFL(2),
   CTMPF(2), CTGN(2), CDEN(2), CSPHT(2), CGASC(2),
   CVISC(2), CRHOVG(2), IUNITS
C DIMENSION DUM1(10), DUM2(10), DELTAN(15)
10 CONTINUE
   IF (ISLICE.EQ.1) TBLK = 0.0
   IF (ISLICE.EQ.1) TOTSEN = 0.0
   TTYME = TYME
   IF (TTYME.LT.0.) TTYME = 0.0
   WRITE(6,90) TTYME, DLTME, WFS
90 FORMAT(1H1,10X,' TIME = ',F6.2,' SEC., STEP SIZE = ',F6.3,
   ' SEC., WHEEL SPEED = ',F8.1,' RPM')
905 WRITE(6,8806) ISLICE,IDELT,JS,DElTAN(ISLICE),IVERGE
   ITPBG = NFWD + 2
   WRITE(6,160) ITRBG
C IF (IUNITS.EQ.2) WRITE(6,270)
   IF (IUNITS.EQ.1) WRITE(6,271)
   DO 210 I = 1,NSTA,2
   II = I
   LCIOOL = 5*II
   NOS = LCIOOL - 4
   DO 205 J = 1,4
   JM = NOS+J-1
DUM1(J) = T(2,ISLICE,JM) - 460.
IF (IUNITS.EQ.1) DUM1(J) = T(2,ISLICE,JM)/CTMPF(1)
CONTINUE
DUM1(5) = T(2,ISLICE,LCOOL) - 460.
IF (IUNITS.EQ.1) DUM1(5) = T(2,ISLICE,LCOOL)/CTMPF(1)
DUM1(6) = P(2,ISLICE,II)/CPH(F(IUNITS))
DUM1(7) = DUM1(6)*(1.+(GAMC(II)-1.)*AM2(II)/2.)**(GAMC(II)/
Z
DUM1(8) = HC(II)/CHTC(IUNITS)
DUM1(9) = HG(II)/CHTC(IUNITS)
DUM1(10) = TG(II) - 460.
IF (IUNITS.EQ.1) DUM1(10) = TG(II)/1.8
IF (BTA.GT.0.01) DUM1(10) = 5.E20
IF (BTA.GT.0.01) DUM1(9) = 4.E20
WRITE(6,274) (II,LCOOL, (DUM1(J),J=1,10))
IF (I.EQ.NFWD) WRITE(6,276)
CONTINUE
DUM2(1) = WJ(ISLICE,II)/CMSFL(IUNITS)
DUM2(2) = DUM2(2)(II)
DUM2(3) = WCROS(2,ISLICE,II)/CMSFL(IUNITS)
DUM2(4) = RE(II)
DUM2(5) = SQRT(AM2(II))
DUM2(6) = FF(II)
DUM2(7) = WFC(II)/CMSFL(IUNITS)
DUM2(8) = FMCPF(II)
WRITE(6,280) (II,LCOOL,DUM2(1),WCHK(II), (DUM2(J),J=2,8))
IF (I.EQ.NFWD) WRITE(6,276)
CONTINUE
DUM2(9) = WDUMP/CMSFL(IUNITS)
IF (ADUMP.GT.0.0.AND.IUNITS.EQ.2) WRITE(6,290) DUM2(9),WCHKDM
IF (ADUMP.GT.0.0.AND.IUNITS.EQ.1) WRITE(6,291) DUM2(9),WCHKDM
WRITE(6,124) ISLICE,ITRBG
DO 220 I = 1,NSTA,2
II = I
LCOOL = 5*II
NOS = LCOOL - 4
DUM2(1) = WJ(ISLICE,II)/CMSFL(IUNITS)
DUM2(2) = DUM2(2)(II)
DUM2(3) = WCROS(2,ISLICE,II)/CMSFL(IUNITS)
DUM2(4) = RE(II)
DUM2(5) = SQRT(AM2(II))
DUM2(6) = FF(II)
DUM2(7) = WFC(II)/CMSFL(IUNITS)
DUM2(8) = FMCPF(II)
WRITE(6,280) (II,LCOOL,DUM2(1),WCHK(II), (DUM2(J),J=2,8))
IF (I.EQ.NFWD) WRITE(6,276)
CONTINUE
DUM1(J) = T(2,ISLICE,JM) - 460.
IF (IUNITS.EQ.1) DUM1(J) = T(2,ISLICE,JM)/CTMPF(1)
CONTINUE
DUM1(6) = P(2,ISLICE,II)/CPH(F(IUNITS))
DUM1(7) = DUM1(6)*(1.+(GAMC(II)-1.)*AM2(II)/2.)**(GAMC(II)/
Z
DUM1(8) = HC(II)/CHTC(IUNITS)
DUM1(9) = HG(II)/CHTC(IUNITS)
DUM1(10) = TG(II) - 460.
IF (IUNITS.EQ.1) DUM1(10) = TG(II)/1.8
CONTINUE
...
IF (BTA.GT.0.01) DUM1(10) = 9.20E0
IF (BTA.GT.0.01) DUM1(9) = 9.20E0
WRITE(6,274) (II,LCOOL,(DUM1(J),J=1,10))
IF (I.EQ.NFYD) WRITE(6,276)
CONTINUE

IF (IUNITS.EQ.2) WRITE(6,278)
IF (IUNITS.EQ.1) WRITE(6,279)
DO 240 I = 1,NSTA Z
I1 = I
IF (I.GT.1) I1 = 1
LCOOL = 5*I1
NOS = LCOOL - 4
DUM2(1) = WJ(ISLICE,II)/CMSPL(IUNITS)
DUM2(2) = DUM2(II)
DUM2(3) = WCROS(2,ISLICE,II)/CMSPL(IUNITS)
DUM2(4) = RE(II)
DUM2(5) = SQRT(AM2(II))
DUM2(6) = FF(II)
DUM2(7) = WFC(II)/CMSPL(IUNITS)
DUM2(8) = FLMEFF(II)
WRITE(6,280) (II,LCOOL,DUM2(1),WCHK(II),(DUM2(J),J=2,8))
IF (I.EQ.NFYD) WRITE(6,276)
CONTINUE

DUM2(9) = WQUMP/CMSPL(IUNITS)
IF (ADUMP.GT.0.0.AND.IUNITS.EQ.2) WRITE(6,290) DUM2(9),WCHKDH
IF (ADUMP.GT.0.0.AND.IUNITS.EQ.1) WRITE(6,291) DUM2(9),WCHKDM
C
C 1000 CONTINUE
C
C TO DETERMINE THE MEAN OUTSIDE SURFACE TEMPERATURE FOR EACH
C SIDE OF THE BLADE, AND
C TO LOCATE THE EXTREME TEMPERATURE POINTS, BOTH HIGH AND LOW.
C
XTOT = 0.
XTOTMD = 0.0
HRAR = HC(I)*(DLX(2)+DLX(3))/2.
TBAR = T(2,ISLICE,1)*(DLX(5)+DLX(11))/2.
TBARMD = T(2,ISLICE,1)*(DLX(6)+DLX(13))/2.
ISTAT = 1
DO 1004 I = 2,NSTA
NODM = 5*I-2
ISTATD = ISTAT + 10
TBARMD = T(2,ISLICE,NODM)*(DLX(NODM)+DLX(NODM+10))/2.
XTOTMD = XTOTMD + DLX(NODM)
TBAR = T(2,ISLICE,ISTAT)*DLX(ISTAT)
XTOT = XTOT + DLX(ISTAT)
HBAR = HBAR + HC(I)*(DLX(ISTAT)+DLX(ISTATD))/2.
CONTINUE

IF (IUNITS.EQ.1) TBAR = TBAR/(1.8*XTOT)
IF (IUNITS.EQ.2) TBAR = TBAR/XTOT - 460.
IF (IUNITS.EQ.1) TBARMD = TBARMD/XTOTMD
IF (IUNITS.EQ.2) TBARMD = TBARMD/XTOTMD - 460.
TBULK = TBULK + TBARMD*S(ISLICE)
TOTSPT = TOTSPT + S(ISLICE)
HBAR = HBAR/(XTOT*CHTC(IUNITS))

1008 IF (IUNITS.EQ.2) WRITE(6,1115) TBAR,TBARMD,HBAR
IF (IUNITS.EQ.1) WRITE(6,1116) TBAR,TBARMD,HBAR
C
TSMAX = T(2,ISLICE,1) - 460.
TMIN = T(2, ISLICE, 1) - 460.
TPMAX = T(2, ISLICE, 1) - 460.
TMPIN = T(2, ISLICE, 1) - 460.
ISUCMX = 1
ISUCMN = 1
IPRSMX = 1
IPRSMN = 1
IPRES = 1
ISUCT = -4

C DO 1080 I = 3, NSTA, 2
IPRES = IPRES + 10
ISUCT = ISUCT + 10

C IF (T(2, ISLICE, IPRES) - 460. GT. TPMAX) GO TO 1030
IF (T(2, ISLICE, IPRES) - 460. LT. TMPIN) GO TO 1040
GO TO 1050

C 1030 TPMAX = T(2, ISLICE, IPRES) - 460.
IPRSMX = I
GO TO 1050

C 1040 TMPIN = T(2, ISLICE, IPRES) - 460.
IPRSMN = I
GO TO 1050

C 1050 IF (T(2, ISLICE, ISUCT) - 460. GT. TSMAX) GO TO 1060
IF (T(2, ISLICE, ISUCT) - 460. LT. TSMIN) GO TO 1070
GO TO 1080

C 1060 TSMAX = T(2, ISLICE, ISUCT) - 460.
ISUCMX = I - 1
GO TO 1080

C 1070 TSMIN = T(2, ISLICE, ISUCT) - 460.
ISUCMN = I - 1
GO TO 1080

C CONTINUE

C IF (ISLICE.LT.NSLICE) GO TO 1095
TBULK = TBULK/TOTSPN
IF (IUNITS.EQ.1) WRITE(6,1091) TYPYE, TBULK
IF (IUNITS.EQ.2) WRITE(6,1090) TYPYE, TBULK

C 1090 FORMAT(1H2, 30X,'TIME = ',F6.3,' SEC., OVERALL BULK TEMPERATURE = ',F7.1,' P')
1091 FORMAT(1H2, 30X,'TIME = ',F6.3,' SEC., OVERALL BULK TEMPERATURE = ',F7.1,' K')
1095 CONTINUE

C 1115 FORMAT(1H2, 'OVERALL AREA WEIGHTED AVERAGES--OUTSIDE T = ',F7.1,
Z ' F, MID-WALL T = ',F7.1,
Z ' F, COOLANT H = ',F6.1,' BTU/(HR/FT**2/F')
1116 FORMAT(1H2, 'OVERALL AREA WEIGHTED AVERAGES--OUTSIDE T = ',F7.1,
Z ' F, MID-WALL T = ',F7.1,
Z ' F, COOLANT H = ',F6.1,' WATTS/M**2/K')
IF (IUNITS.EQ.2) WRITE(6,1115) TPMAX,IPRSMX,TMPIN,IPRSMN, TSMAX,
Z ISUCMX,TSMIN,ISUCMN
1120 FORMAT(1H2, 'EXTREMES OF OUTER SURFACE TEMPERATURES (F)'/6X,
Z 'PRESSURE SIDE: ',
Z 'F7.1,' AT STATION ',I2,', ',F7.1,' AT STATION ',I2/6X,
Z 'SUCTION SIDE:',
Z F7.1, AT STATION ',I2', 'F7.1, AT STATION ',I2
IF (UNITS.EQ.2) GO TO 1130
TPMAX = (TMPAX*460.)/.8
TPMIN = (TMPIN*460.)/.8
TSMAX = (TSMAX*460.)/.8
TSMIN = (TSMIN*460.)/.8
WRITE(6,1125) TMAX,IPRSMX,TPMIN,IPRSNN,TSMAX,ISUCHX,TSUIN,ISUCHN
1125 FORRAT( /*EXTREMES OF OUTER SURFACE TEMPERATURES (K) */
Z 'PRESSURE SIDE:',
Z F7.1, AT STATION ',I2', 'F7.1, AT STATION ',I2
IF (IUNITS.EQ.2) GO TO 1130
TPHAX = (TPHAX+460.)/1.8
TPHIN = (TPHIN+460.) /1.8
TSHAX = (TSHAX+460.)/1.8
TSHIN = (TSHIN+460.) /1.8
WRITE (6,1125) TPHAX,IPRSMX,TPMIN,IPRSNN,TSMAX,ISUCHX,TSUIN,ISUCHN
1130 CONTINUE
120 FORMAT(30X,'PRESSURE SIDE, TRAILING EDGE REGION BEGINS AT ',
Z 'STATION-',I3)
124 FORMAT(H1,/* SLICE NO. ',I2,17X,'SUCTION SIDE, TRAILING EDGE ',
Z 'REGION BEGINS AT STATION-',I3)
8806 FORMAT(' SLICE NO.',I2,', FLOW SPLIT NO.',I3,', SPLIT AT ',
Z 'STATION',I3,
Z '; FRACTION SPLIT TO SUCTION SIDE IS',F7.4,16,
270 FORMAT(/)
271 FORMAT(/)
274 FORMAT(I6,2X,I6,1X,7F10.1,3F11.1)
276 FORMAT(47X,'BEGIN TRAILING EDGE REGION')
278 FORMAT(1H2,/* STATION * COOLANT * OUTSIDE * INTERFACE * MID-WALL * INSIDE */
Z 'COOLANT * STATIC P * TOTAL P * HC, BTU/HR* HG, BTU/HR* TG'
Z 'NUMBER * NODE NO. T (F) * T (F) * T (F) * T (F) * T (F)
Z 'PSIA) * (PSIA) /FT**2/R *FT**2/R * (F)
Z/119('*')/)
279 FORMAT(/)
281 FORMAT(7F10.1,/*F9.6,A1,F9.1, 'F9.6,2X,F9.1,4X,F9.6)
Z 'F9.6,A1, 'KB/SEC')
Z 'F10.6,A1, 'KG/SEC')
REFERENCES


<table>
<thead>
<tr>
<th>Subroutine name</th>
<th>Source module</th>
<th>COMMON blocks</th>
<th>Called subroutines</th>
<th>Calling subroutines</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLOCK DATA</td>
<td>NGASDAT</td>
<td>/GAAS/</td>
<td>NONE</td>
<td>NONE</td>
</tr>
<tr>
<td>FLOWS</td>
<td>NFLOEST</td>
<td>/CHKHOL/</td>
<td>GASTBL</td>
<td>TCOEF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/FLMCOL/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/PRPS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TCO/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TRANSNT/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLSPLT</td>
<td>NFLSPLP</td>
<td>NONE</td>
<td>NONE</td>
<td>TCOEF</td>
</tr>
<tr>
<td>GASTBL</td>
<td>NGASTB</td>
<td>/GAAS/</td>
<td>NONE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAUSS</td>
<td>NGAUS</td>
<td>/MATRIX/</td>
<td>NONE</td>
<td>TCOEF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GETIN</td>
<td>NGETINT</td>
<td>/BOUND/</td>
<td>INPRT</td>
<td>MAIN PROGRAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/FLMCOL/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/IMPCR/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/RADL/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/SPECCL/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TCO/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TRANSNT/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/UNITS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HCFRCD</td>
<td>NHCFRCT</td>
<td>/TCO/</td>
<td>GASTBL</td>
<td>HCOOL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>TARRAY</td>
</tr>
<tr>
<td>HCOOL</td>
<td>NHCOOLT</td>
<td>/IMPCR/</td>
<td>GASTBL</td>
<td>TCOEF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/PRPS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TCO/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HCPINS</td>
<td>NHCPINT</td>
<td>/PRPS/</td>
<td>GASTBL</td>
<td>TARRAY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TCO/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INPRT</td>
<td>NINPRTT</td>
<td>/BOUND/</td>
<td>PREP</td>
<td>GETIN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/GAAS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/PRPS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/RADL/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/SPECCL/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TCO/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TRANSNT/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/UNITS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAIN PROGRAM</td>
<td>NTTACT</td>
<td>/BOUND/</td>
<td>GETIN</td>
<td>NONE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/FLMCOL/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/GAAS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/RADL/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/SPECCL/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TCO/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TRANSNT/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/UNITS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subroutine name</th>
<th>Source module</th>
<th>COMMON blocks</th>
<th>Called subroutines</th>
<th>Calling subroutines</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARRAY</td>
<td>NPARAYT</td>
<td>/MATRIX/</td>
<td>NONE</td>
<td>TCOEF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/PRPS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TCO/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TRANSNT/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLNUM</td>
<td>NPLENMP</td>
<td>/RAID/</td>
<td>GASTBL</td>
<td>MAIN PROGRAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TCO/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TRANSNT/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/UNITS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PLOTMF</td>
<td>NPLOTTIT</td>
<td>/PRPS/</td>
<td>NONE</td>
<td>TCOEF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/SPECCL/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TCO/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TRANSNT/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/UNITS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PREP</td>
<td>NPREPAT</td>
<td>/BOUND/</td>
<td>NONE</td>
<td>INPRT</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/FLMCOL/</td>
<td></td>
<td>MAIN PROGRAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/PRPS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TCO/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TRANSNT/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/UNITS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARRAY</td>
<td>NTARRAYT</td>
<td>/BOUND/</td>
<td>GASTBL</td>
<td>HCFRCD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/FLMCOL/</td>
<td></td>
<td>HCPINS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/PRPS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TCO/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TRANSNT/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/UNITS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCOEF</td>
<td>NTCOFTT</td>
<td>/BOUND/</td>
<td>FLOWS</td>
<td>MAIN PROGRAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/FLMCOL/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/PRPS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TCO/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TRANSNT/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/UNITS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>THRCNT</td>
<td>NTHRCNT</td>
<td>/BOUND/</td>
<td>FROWS</td>
<td>TCOEF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TCO/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WROUT</td>
<td>NWROTTT</td>
<td>/CHKHOL/</td>
<td>NONE</td>
<td>TCOEF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/FLMCOL/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/PRPS/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/RADL/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TCO/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/TRANSNT/</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>/UNITS/</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### TABLE II. - COMMON-BLOCK CROSS-REFERENCE TABLE

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>COMMON block</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BOUND</td>
</tr>
<tr>
<td>BLOCK DATA</td>
<td></td>
</tr>
<tr>
<td>FLOWS</td>
<td>×</td>
</tr>
<tr>
<td>FLSPLT</td>
<td></td>
</tr>
<tr>
<td>GASTBL</td>
<td></td>
</tr>
<tr>
<td>GAUSS</td>
<td></td>
</tr>
<tr>
<td>GETIN</td>
<td>×</td>
</tr>
<tr>
<td>HCFRCD</td>
<td></td>
</tr>
<tr>
<td>HCOOL</td>
<td></td>
</tr>
<tr>
<td>HCPINS</td>
<td></td>
</tr>
<tr>
<td>INPRT</td>
<td>×</td>
</tr>
<tr>
<td>NTTACT</td>
<td>×</td>
</tr>
<tr>
<td>PARRAY</td>
<td></td>
</tr>
<tr>
<td>PLNUM</td>
<td></td>
</tr>
<tr>
<td>PLOTMF</td>
<td></td>
</tr>
<tr>
<td>PREP</td>
<td>×</td>
</tr>
<tr>
<td>TARRAY</td>
<td>×</td>
</tr>
<tr>
<td>TCOEF</td>
<td></td>
</tr>
<tr>
<td>THRCON</td>
<td>×</td>
</tr>
<tr>
<td>WROUT</td>
<td>×</td>
</tr>
</tbody>
</table>
Figure 1. Overall division of blade into slices.

Figure 2. Blade geometric model.
Set boundary conditions for current time

Calculate impingement-plenum conditions for this slice

Calculate flow rates, pressures, and temperatures

Pressure converged?

Yes

No

Does flow-split balance?

Yes

No

Adjust flow-split

Move to next slice

Solved all slices?

Yes

No

Adjust coolant-inlet flow; move back to first slice

Does total coolant match inlet?

Yes

No

Last time step?

Yes

No

Increment time; move back to first slice

Finished

Figure 3. Overall program procedure.
Figure 4. - Subroutine calling relations.
Interpolate in transient-boundary-condition tables: PTIN, TTIN, WS, PEXIT, RHOGA, QHUB, THUB, QTIP, TIP.

Begin marching up blade:
DO 1000. 1=1,NSLICE

Call PLNUM to calculate coolant conditions

Call PREP to prepare input data for current slice

Call TCOEF to calculate flows, temperatures, and pressures

Calculate amount of coolant remaining

1000

Check overall coolant balance

1100

Job complete

Figure 5. - Flow chart of main program.

Figure 6. - Coolant-channel mass balance.
Figure 7. Flow chart for subroutine FLSPLT.
Figure 7. - Continued.
Figure 7. - Concluded.

(a) Detail of station numbering at end of impingement insert.

(b) Detail of flow-split point.

Figure 8. - Details of flow-split parameters.
A computer program to calculate transient and steady-state temperatures, pressures, and coolant flows in a cooled axial-flow turbine blade or vane equipped with a coolant insert is described. Coolant-side heat-transfer coefficients are calculated internally in the program, with the user specifying either impingement or convection heat transfer at each internal flow station. Spent impingement air flows in a chordwise direction and is discharged through the trailing edge and through film-cooling holes. The ability of the program to handle film cooling is limited by the internal flow model. Input to the program includes a description of the blade geometry, coolant-supply conditions, outside thermal boundary conditions, and wheel speed. The blade wall can have two layers of different materials, such as a ceramic thermal-barrier coating over a metallic substrate. Program output includes the temperature at each node, the coolant pressures and flow rates, and the coolant-side heat-transfer coefficients.
12 11U.D. 121878 S00903DS
DEPT OF THE AIR FORCE
AF WEAPONS LABORATORY
ATTN: TECHNICAL LIBRARY (SUL)
KIRTLAND AFB NM 87117

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