MSFC

NASA
Technical
Paper
2685

January 1987

Space Shuttle Main Engine High Pressure Fuel Pump Aft Platform Seal Cavity Flow Analysis

S. A. Lowry and
L. W. Keeton

NASA

NASA Technical Paper 2685

1987

Space Shuttle Main Engine High Pressure Fuel Pump Aft Platform Seal Cavity Flow Analysis

S. A. Lowry
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

L. W. Keeton
CHAM (NA) Inc.
Huntsville, Alabama

National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

ACKNOWLEDGMENTS

The authors would like to thank Jeff Ingram, EP23, NASA MSFC for his work in compiling the property curves in Appendix B. Special thanks also go to Gene Teal of Lockheed Inc., Huntsville, Alabama, for his assistance in providing the boundary conditions used in the model. The authors would also like to thank Loren Gross, EP23, NASA MSFC; Dwayne McCay, EP26, NASA MSFC; and Ashok Singhal and colleagues at CHAM (N.A.), Huntsville, Alabama, for their technical advice and support.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Concluded)

NASA TECHNICAL PAPER

SPACE SHUTTLE MAIN ENGINE HIGH PRESSURE FUEL PUMP AFT PLATFORM SEAL CAVITY FLOW ANALYSIS

I. INTRODUCTION

In working to improve the performance of the Space Shuttle Main Engine (SSME), the engineer is confronted with the difficult task of analyzing a complex engine system running under extreme operating conditions. The temperatures in the Shuttle's engines range from 37°R up to 5000°R, pressure vary from 20 to 8000 psi, and the engines' pumps rotate at speeds up to 37,000 rpm. Direct measurement of the engine environment is often impractical. Indeed, the particular area under consideration may be virtually inaccessible to instrumentation. Fortunately, the capability of modeling heat and mass transfer using computers has advanced to the point where computational fluid dynamics (CFD) can provide an alternate method of analyzing the engine. When used to model the various components and processes in the engine, numerical analysis can provide the engineer with valuable insight by allowing him or her to examine a wide range of operating conditions. The effect of a change in geometry, of a change in flowrate, or of a change in any parameter can be examined. Even a simple numerical model can demonstrate the sensitivity of the engine system to such changes, and a sophisticated numerical model, especially when used in conjunction with measured data, is a highly effective analytical tool.

In the current application, a general-purpose CFD code named PHOENICS, developed by CHAM Inc., is used to model the temperatures, pressures, and velocities in the SSME's High Pressure Fuel Turbopump (HPFTP) aft-platform seal cavity for a variety of boundary conditions and geometries. This cavity is located downstream of the fuel pump's second turbine disk, between the disk and the aft platform seal (Figs. 1 to 4). It is an annular cavity where 1400°R combustion products and 150°R coolant hydrogen mix in a complex flow pattern and then are vented into the pump's turbine exhaust. An understanding of the flow field in this cavity is critical since there are at least two known problems in the High Pressure Fuel Pump which may be linked to the environment in this region. Specifically, these problems are (1) cracking of the second stage turbine blade shanks, and (2) hot gas leakage into the stack behind the aft platform seal (Fig. 4). The first problem, blade cracking, can severely limit the time a pump can operate before it must be rebuilt. The second problem, that of hot gas leakage, is potentially more severe since, in the extreme, it may cause the pump to shut down prematurely if the temperatures or pressures in the coolant liner behind the aft-platform seal exceed certain redlines.

Accordingly, the primary purpose of the present analysis is to investigate the two problem areas mentioned above. In doing so, the study addresses the following questions:

1) How severe is the temperature gradient in the region where the turbine blades are cracking?

2) What would be the temperature of any fluid which leaked from the cavity into the coolant liner?

The analysis addresses these questions, not only for the pump operating under normal conditions, but also for a range of off-design conditions since even a slight departure from the norm might have a radical effect on the flow pattern and temperatures in the aft-platform seal cavity. As such, the broad objective of this study is to develop a model flexible enough that it can examine the effect that boundary parameters such as clearances, pressures, and flowrates have on the flow pattern and temperatures in the cavity. Such a model must be general enough that it can support future analytical and experimental investigations of the HPFTP aft-platform seal cavity.

ORIGINAL PAGE IS
OF POOR QUALITY

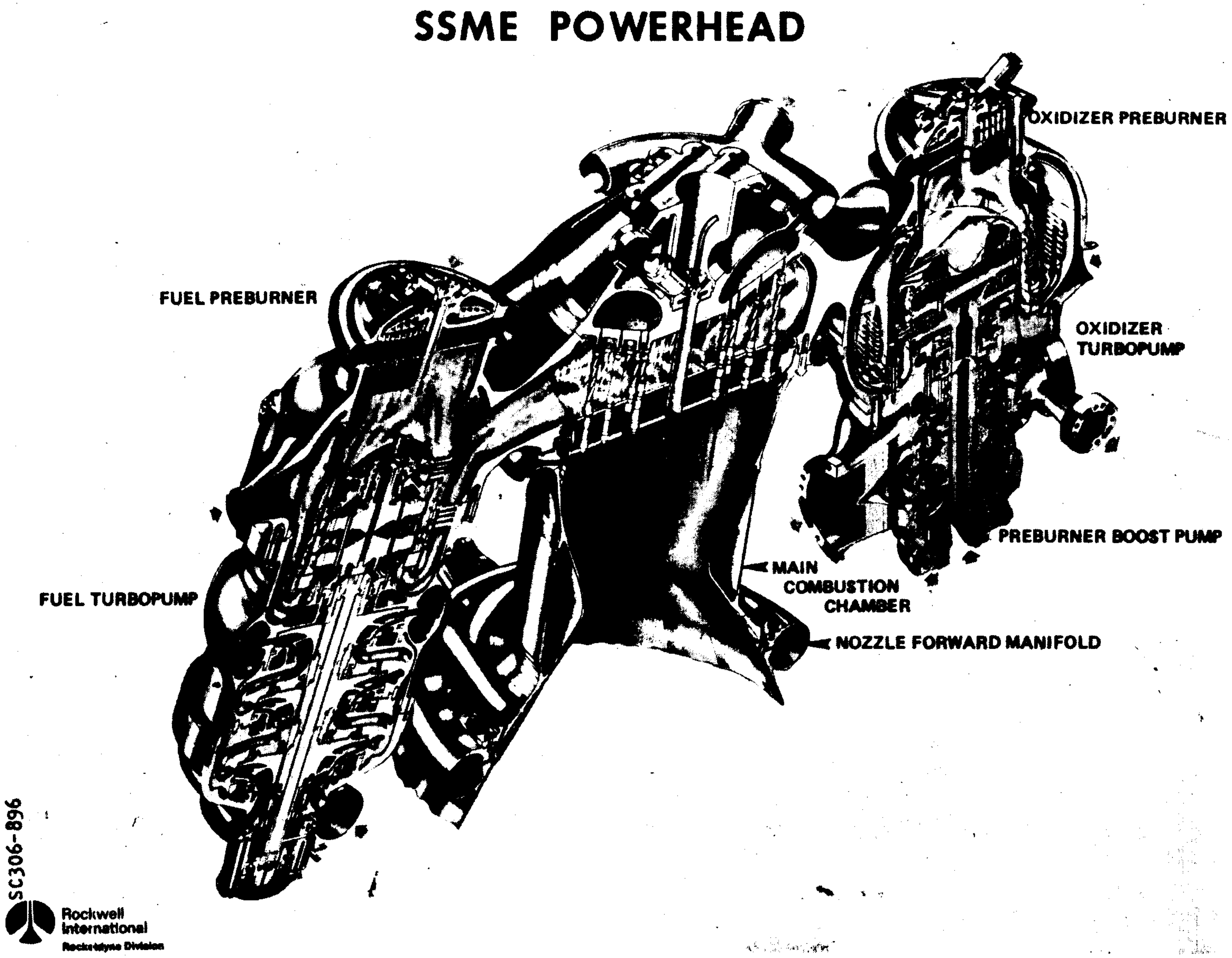


Figure 1. The Space Shuttle Main Engine.

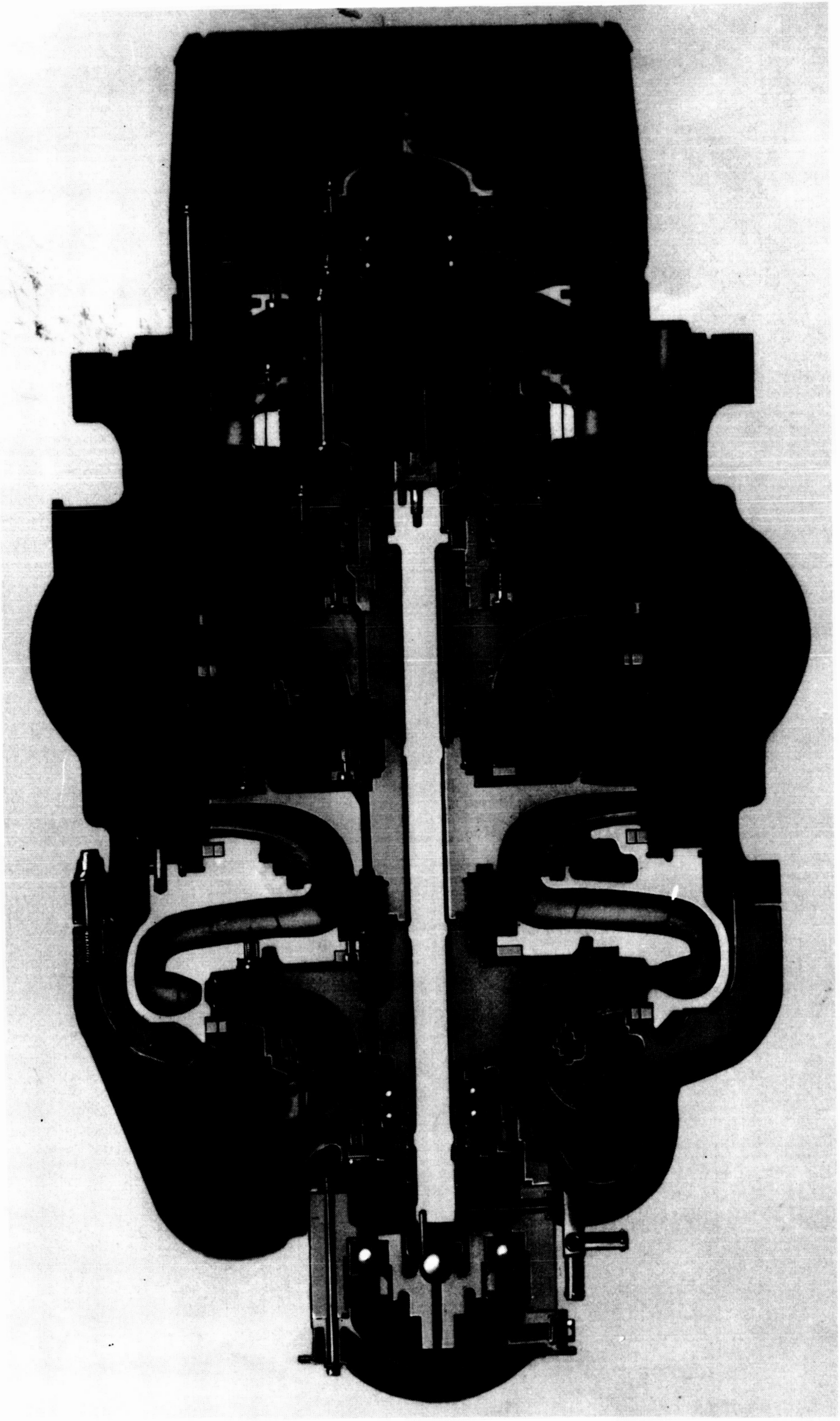


Figure 2. SSME High Pressure Fuel Turbopump.

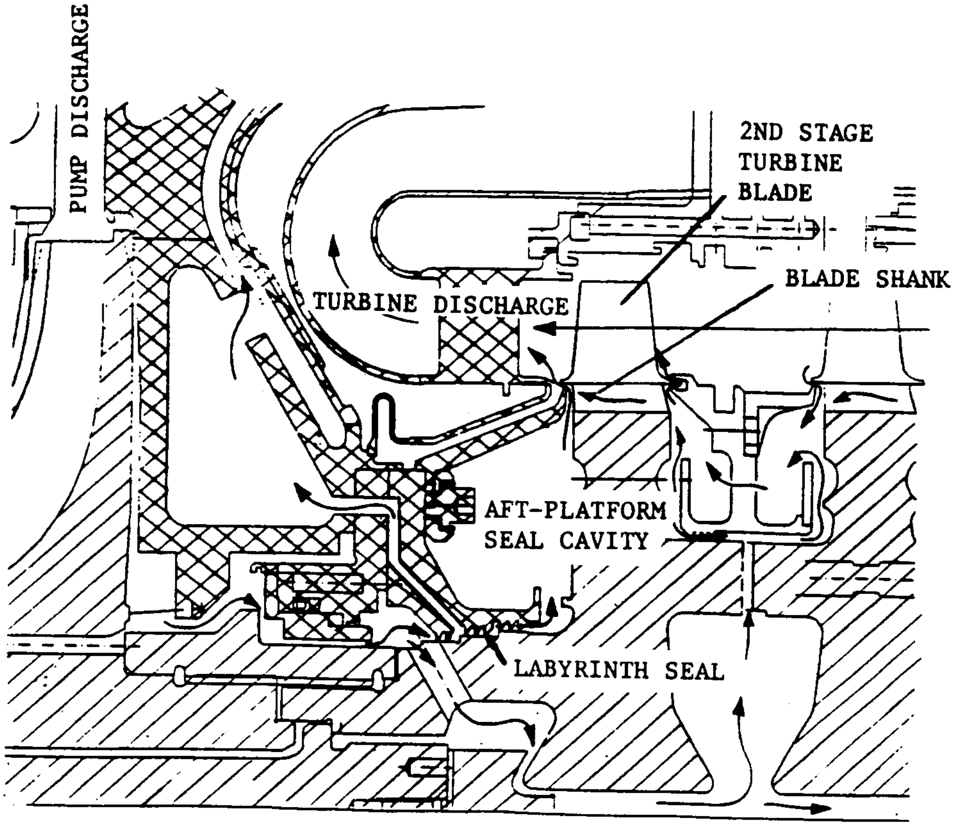


Figure 3. HPFTP turbine flow paths.

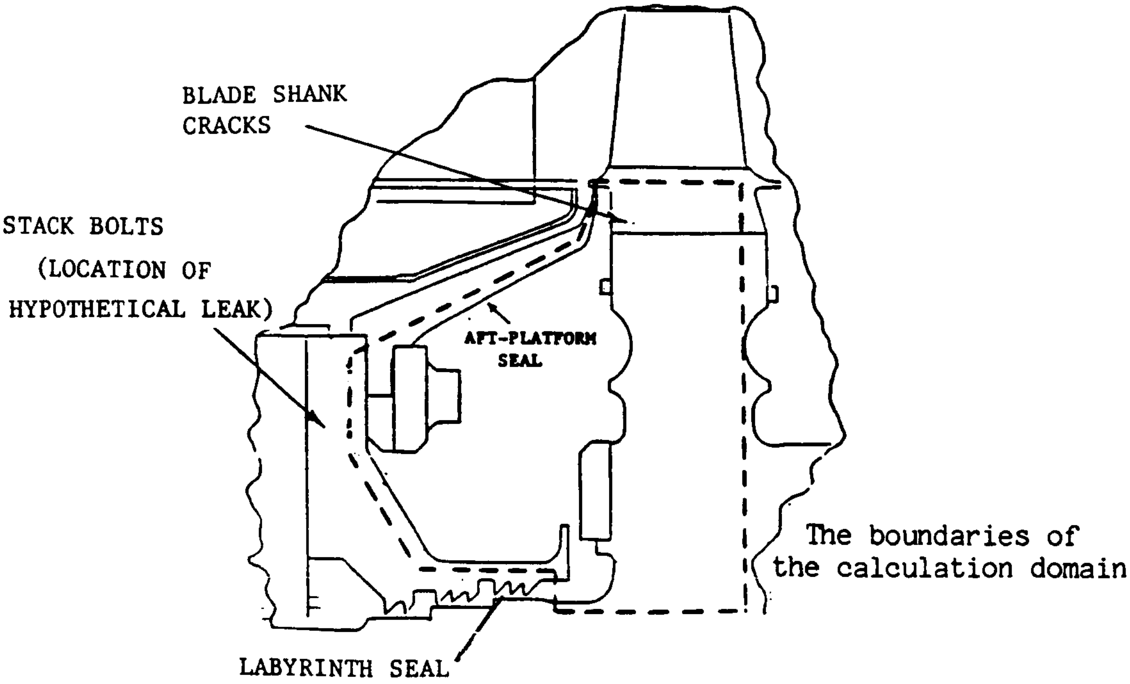


Figure 4. Aft-platform seal cavity.

II. PROBLEM DESCRIPTION

The region being modeled is the aft-platform seal cavity downstream of the second-stage turbine in the Space Shuttle's HPFTP. A close-up of the aft-platform seal cavity is provided in Figure 4. General views of the shuttle engine, the fuel pump, and the fuel pump turbine section are given in Figures 1 to 3. The dashed lines in the close-up view, Figure 4, represent the limits of the problem as specified in the model. The fluid properties and either the pressures or the flowrates at the boundaries must be input into the program. Unfortunately, the only available measurements of these parameters (i.e., temperature and pressure) are far removed from the inlets and exits of the aft-platform seal cavity. As such, the boundary conditions chosen as inputs rely heavily on an existing one-dimensional analysis of the HPFTP and must be used with caution [1].

Inspection of Figure 2, the HPFTP, will show that the aft-platform seal cavity is an axisymmetric annular cavity defined by stationary walls on one side and a rotating disk on the other. Flow enters the cavity through two inlets, one at the inner radius of the cavity and one near the outer radius of the disk. The flow leaves the region via the gap between the outer radius of the aft-platform seal and the blade lips. At high rpm (up to 37,000) the flow is a turbulent mixture of hydrogen and water at temperatures ranging from approximately 140°R to possibly as high as the turbine exhaust at 1700°R. Flowrates are on the order of 1 lbm/sec and pressures are in the range of 4000 psi.

The inlets and exits of the aft-platform seal cavity are described qualitatively below. The specific numbers used in this study, e.g., flowrates, pressures, etc., and the assumptions used in defining these numbers, can be found in the section on numerical model set-up.

A. Inlets

1. Coolant Inlets

At the inner radius of the cavity, approximately 0.3 lbm/sec of liquid hydrogen flows into the aft-platform seal cavity through a labyrinth seal. The source of this hydrogen is the coolant circuit which is fed by the discharge of the HPFTP (Fig. 3). In the two-dimensional model, this flowrate is calculated implicitly based on the pressure drop through the labyrinth seal. In the three-dimensional model, in the interest of computational economy, the coolant flowrate through the labyrinth seal is not calculated internally, but is simply set to the value predicted by the two-dimensional model operating with the same average clearances and flowrate through the blade shanks.

2. Hot Gas Inlet at the Blade Shanks

One wall of the aft-platform seal cavity is formed by the rotating disk upon which are mounted the second stage turbine blades. At the periphery of this disk, a mixture of coolant hydrogen and combustion products enters the cavity through the gap between the shank of one turbine blade and the next (Fig. 5). Since there are 58 blades in the second stage disk, there are, accordingly, 58 holes available for this hot gas mixture to flow through into the aft platform seal cavity from the high pressure side of the turbine disk. The flow pattern of the fluid entering through these holes is complex since the shanks of the blades are curved and the disk itself is rotating at up to 37,000 rpm.

In modeling this inlet, the 58 separate streams entering through the disk have been "smeared" in the circumferential direction into a single, continuous axisymmetric source. The flowrate and fluid properties at this inlet are prescribed based on predicted values, and the angular velocity of the fluid entering the cavity through these passages is assumed to have the same angular velocity as the disk.

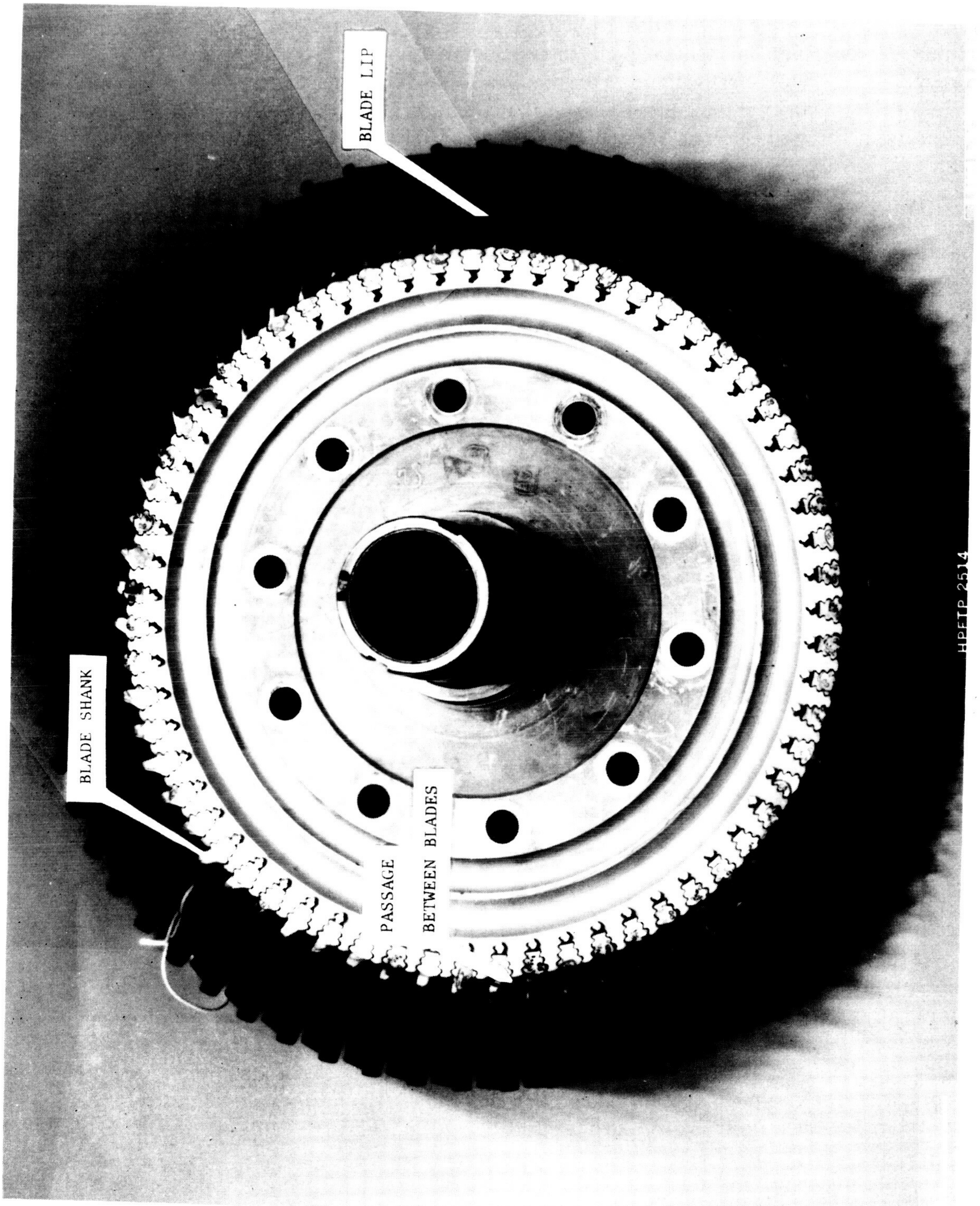


Figure 5. HPFTP second stage turbine disk with blades.

B. Exits

1. Exit Gap Between the Outer Diameter of the Aft-Platform Seal and the Blades

In this study, the single most important parameter which affects the flow pattern in the aft-platform seal cavity is the gap between the outer diameter of the aft platform seal and the lip of the turbine blades. This gap is very small, on the order of one hundredth of an inch, and it supports a high pressure drop of over 500 psi between the aft-platform seal cavity and the turbine exhaust. Any slight variation in this gap clearance will have a strong effect on the total flow and overall flow pattern in the cavity. In general, the actual flow exiting through this gap at a given location will respond to changes in the overall turbine discharge pressure, the circumferential variation in turbine discharge pressure, and any changes in the width of the gap. The latter could be due to a number of different causes, including: sideloads, dynamics, machining tolerances, eccentricity, or thermal expansion.

In the model, the exit pressure outside the gap is fixed at the best estimate for the turbine discharge pressure. The pressure drop across this exit is then related to the flowrate based on a loss coefficient times the local dynamic head.

2. Secondary Exit Hole

In one of the test runs discussed in this report, the aft-platform seal is modeled with a second exit in order to simulate a postulated leak. The leak was assumed to be around the bolts which secure the aft-platform seal to the lift-off seal stack (Fig. 4). The hole size, loss coefficient, and exit pressure of this second exit were chosen such that the resulting calculated flowrate would be approximately 0.2 lbm/sec. The leak rate of 0.2 lbm/sec was chosen because it is the maximum flowrate which could be leaking past the bolts. This last conclusion is based on experimental measurements of the pressure drops in the coolant liner cavity which is downstream of the postulated leak.

III. NUMERICAL MODEL SET-UP

CHAM Inc.'s general purpose computational fluid dynamics code, PHOENICS [2], has been employed for all the numerical studies described herein. To use PHOENICS, special purpose "satellite" and "ground station" sub-programs must be formulated whereby the built-in features can either be turned on or off or modified, as necessary. One set of the sub-programs adapted specifically for the HPFTP aft-platform seal cavity three-dimensional studies is listed, in full, in Appendix A. Full listings of the other adapted sub-programs used in this study are given in a separate CHAM report [3]. All of these sets of sub-programs are extensively annotated (via built-in "COMMENT" statements) so as to make then self-explanatory when read in conjunction with the PHOENICS User's Manual [4]. Consequently, no detailed line-by-line description is given here; however, the most relevant features are described below.

The two-dimensional calculations described herein have been performed by using the two-dimensional y/z, polar coordinate option of the code. Figure 6 shows the selected two-dimensional grid distribution. There are 1120 control cells, with 40 and 28 cells in the radial (IY) and (IZ) directions, respectively. Due to the (initially) assumed cyclic symmetry of the problem, only one control cell is required in the circumferential (IX) direction. However, to enable correct account to be taken of the wall shear stresses acting on the fluid entering between the blade shanks, the circumferential extent of the calculation domain is taken to be equal to the space between 2 consecutive blades (i.e., an angle of $1/58 \times 2\pi$ deg, where 58 = total number of blades).

In the three-dimensional calculation, the full three-dimensional x/y/z coordinate capabilities of PHOENICS were employed. The identical y/z grid distribution of the 2-dimensional calculations was retained with, in addition, 8 cells in the circumferential (IX) direction, such that a total of $8 \times 40 \times 28 = 8960$ control cells is used.

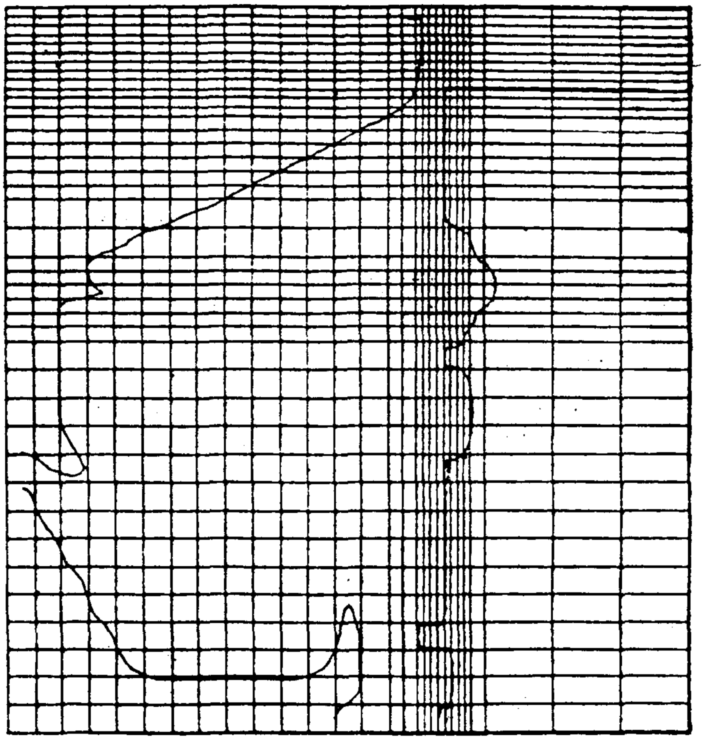


Figure 6. Computational grid.

As depicted in Figure 3, the "cold" liquid hydrogen coolant enters axially, through the labyrinth seal, at the inner radius of the cavity. This "cold" hydrogen then joins with the mixture of "hot" hydrogen and water that flows into the cavity from between the blade shanks located at the outer radius of the rotating disk. The combined streams of fluid then exit beneath the blade lips, as also shown in Figure 3.

A. Assumptions/Model Details

The major assumptions and salient features of the physical models and the boundary conditions employed are described below.

1) All boundary surfaces (both stationary and rotating) have been assumed to be adiabatic.

2) The hydrogen and water mixture are treated as a single homogeneous fluid with mixture properties (density and laminar viscosity) and temperature deduced from the calculated mixture enthalpy and specified hydrogen and water property curve fit data as described in Appendix B.

3) The turbulence effects are presented by way of the two-equation (k-ϵ) model of turbulence. In this model, two parameters, viz: the turbulence kinetic energy, k, and its dissipation rate, ϵ, are computed from differential transport equations. Thus, it has the capability of representing both the local and history effects. The effective viscosity is expressed as:

$$\mu_{eff} = \mu_{\ell} + C_{\mu}\,\rho\,k^2/\epsilon$$

where μ_ℓ is the laminar viscosity, C_μ is an empirical constant and ρ is the local mixture density. In addition, four other empirical constants are assigned the values as recommended in original publications [4].

4) All boundary surfaces of irregular shape are accommodated in the present calculations by use of "cell porosities." In this approach, each control cell is characterized by a set of fractions, in the range from 0 to 1. These fractions determine the proportion of the cell volume which is available for flow from the cell to its neighbor in a given direction. This practice is much more rigorous and accurate than the practice of using rectangular steps.

5) The wall shear stress is calculated by using the conventional wall functions which are based on the assumption of the logarithmic law of the wall. For partially blocked control cells, the wall stress is calculated for the projected surfaces parallel to the velocity components.

It should be noted that the PHOENICS (1981 version) built-in process for determining wall shear stress is restricted to a finite number of special regions, to be set via the satellite subroutine. For the complex aft-platform seal geometry, many such special regions would be necessary, in excess of the built-in maximum, and a special PHOENICS user subroutine program was written for the current problem to overcome this restriction. This user sub-program (GWALL) performs the identical job as the built-in PHOENICS "WALL" subroutine but is used via the PHOENICS ground station. A listing of GWALL is included in Appendix A.

6) In PHOENICS, an iterative finite-difference solution procedure is employed to solve the governing differential equations together with the above mentioned relations. The method is based on a fully implicit, conservative formulation. As a result there is no restriction on the selection of the grid and the magnitude of the time steps.

The variables calculated and/or solved for (and printed) in the seal cavity flow calculation include the following:

a. The fluid velocities in the 3 coordinate directions

b. The mixture enthalpy and deduced temperature

c. The (mass) concentration of water vapor

d. The turbulent kinetic energy and its dissipation rate

e. The static and total pressures

f. The mixture density and separate densities of both the hydrogen and water

g. The effective viscosity.

7) Boundary conditions are:

a. Prescribed mass flowrate, velocities, enthalpy, mixture ratio, and turbulence parameters at all inlets except for the two-dimensional solutions, in which case the flowrate through the labyrinth seal was computed based on a prescribed inlet pressure.

b. Prescribed exit pressure at all outlets, with the pressure drop related to the flowrate based on a specified loss coefficient times the dynamic head.

c. The incoming fluid enclosed between the blade shanks is assumed to rotate at the same speed as the adjacent disk surface.

8) The (phase change) freezing of the water is not accounted for; any water at temperatures below freezing is given the properties (density, etc.) of liquid water at freezing.

9) The effects of viscous heating have been ignored.

IV. TWO-DIMENSIONAL TEST RUNS

Three different two-dimensional test cases were run. The first of these was considered to be the basecase using the best estimate of the average conditions for the pump operating at the full power level (FPL). A second test run was made with a reduced amount of coolant entering through the labyrinth seal in order to determine the sensitivity of the solution to the ratio of hot gas flowing in at the blades relative to the hydrogen entering at the labyrinth seal. Finally, a third two-dimensional test run was made in order to see what effect a postulated leak through the stack bolts would have on the calculated cavity temperatures and flows. These three test runs and results are described in more detail below.

A. Two-Dimensional Test Runs: Boundary Conditions

1. Basecase 2-D

The basecase two-dimensional run uses boundary conditions and operating clearances taken from a one-dimensional flow analysis provided by Lockheed, Inc. [1]. These boundary conditions are tabulated in Table 1. It should be noted that for this particular run, the boundary condition specified at the labyrinth seal is that of a prescribed pressure boundary from which the flowrate is then deduced based on the following relationship [5]:

```
MASSFLOW = FC * AREA * SQRT((RHO*PO(1-(PN/PO)**2))/
                          (NUMBER OF TEETH + ALOG(PO/PN)))

WHERE PO= UPSTREAM PRESSURE; PN=DOWNSTREAM PRESSURE; FC=FLOW COEFF.
```

(Note that for the basecase test run, the above equation when coupled with the PHOENICS two-dimensional model predicts a slightly lower flowrate through the labyrinth seal (0.26 lbm/sec versus 0.36 lbm/sec) as compared to the Lockheed one-dimensional model predictions.)

2. Reduced Coolant (Labyrinth) Flow

In the second run, the basecase two-dimensional model was modified by reducing the clearance at the outer diameter of the aft-platform seal while leaving all the other boundary conditions, including the hot gas flowrate, the same. When the gap size is reduced, the pressure in the cavity goes up and the coolant through the labyrinth seal decreases. The purpose here was to determine the effect that a reduction in coolant flow would have on the temperature field in the cavity.

3. Leak Through the Stack Bolts

The final two-dimensional run of the current study simulated a 0.2 lbm/sec leak through the stack bolts. The boundary conditions for this run were the same as the basecase but with a "hole" at the location shown in Figure 4. The loss coefficient at this hole and the hole size were chosen such that they dictated a leak rate of approximately 0.2 lbm/sec.

TABLE 1. TWO-DIMENSIONAL BOUNDARY CONDITIONS

Variable	Basecase	Reduced Coolant	Leak
Rotational speed of the disk (RPM)	37,000	37,000	37,000
Gap size at the labyrinth seal (in.)	0.1069	0.1069	0.1069
Total flow area (360°) between the blade shanks (in.^{2})	3.877	3.877	3.877
Clearance between the aft-platform seal and blades (in.)	0.0108	0.0102	0.0108
Loss coefficient at the exit near the blade shanks	1.5	1.5	1.5
Enthalpy of the H_2 upstream of the labyrinth seal (Btu/lbm) (Resultant calculated temperature – degrees Rankine)	278.3 (145°R)	278.3 (145°R)	278.3 (145°R)
Enthalpy of H_2 and H_2O entering through the blades (Btu/lbm) (Resultant calculated temperature – degrees Rankine)	3558 (1466°R)	3558 (1466°R)	3558 (1466°R)
Density of the H_2 upstream of the labyrinth seal (lbm/ft^3)	3.574	3.574	3.574
Density of H_2 and H_2O entering through the blades (lbm/ft^3)	0.931	0.931	0.931
Mass flowrate of H_2 and H_2O entering past the blades (lbm/s)	3.649	3.649	3.649
Mass fraction of H_2O entering through the blades	0.474	0.474	0.474
Pressure at the turbine discharge (psi)	3582	3582	3582
Pressure at the labyrinth seal inlet (psi)	4254	4254	4254
Loss coefficient for the second (leak) exit	1.5	1.5	1.5
Total flow area at the second (leak) exit (in.^{2})	0	0	0.00019

B. Two-Dimensional Test Runs: Results and Observations

1. Basecase

According to the model, when the HPFTP is operating at full power, centrifugal force dominates the flow field and pressure field in the aft-platform seal cavity. This is not surprising when one considers that at 37,000 rpm and a radius of 4.5 in., the hot gas exits the blade shanks with a centrifugal force equal to approximately 175,000 g's. Figures 7 and 8 show that there is virtually no penetration of the hot gas down into the aft-platform seal cavity and, as a result, the temperature in the cavity remains cold, at approximately 375°R (-85°F). The flow pattern in the main cavity consists of two large co-rotating vortices which maintain the cavity at a relatively uniform temperature. The liquid hydrogen which enters through the labyrinth seal at the inner radius of the cavity flows radially outward along the face of the disk and then abruptly merges with the hot fluid stream exiting from between the shanks. While it appears from the drawing that the cold flow then recirculates, in fact all of the coolant which enters through the labyrinth seal must, by continuity, mix and then exit with the hot stream, resulting in the sharp temperature gradient at the blade shanks especially evident in the close-up view provided in Figure 8. The actual local gradients that the blade shanks would see would be more severe than predicted here since, in the model, the hot gas flow is treated as an axisymmetric source which would tend to smooth out the temperature gradients at the trailing edge of the blade shanks. In reality, there are 58 blades shanks between which the hot gas flows into the cavity. The cold fluid which is slung off the disk, up behind the trailing edge of the blade shanks, will be sheltered from the hot flow entering from between the blade shanks. As a result, the local mixing of hot and cold fluid will be delayed, and the local temperature gradient will be even more severe than shown here.

2. Reduced Coolant (Labyrinth) Flow

During the course of the study the question was raised as to what would happen if the proportion of coolant to hot gas flow were different that that predicted by the one-dimensional model used to define the boundary conditions [1]. In order to answer this question, the boundary conditions in the model are manipulated in a somewhat contrived manner in order to change the proportion of hot gas flow to coolant flow, viz: the coolant flowrate is reduced by slightly reducing the clearance between the aft-platform seal and the blade lips. The result is that a reduction of only six ten-thousandths of an inch (6 percent) of this clearance reduces the coolant flow by over half. This change, however, has little effect on the flow field in the cavity. As with the basecase, the flow in the cavity remains dominated by the centrifugal force. As shown in Figures 9 and 10, the temperature in the cavity has risen by only approximately 150 deg, up to 525°R, which is a moderate increase when compared with the hot gas inlet temperature of 1466°R. The conclusion is that the flow field and temperature field of the aft-platform seal cavity is relatively insensitive to the amount of coolant entering at the labyrinth seal relative to the amount of hot gas mixture entering through the blade shanks. However, the pressure and coolant flowrate are extremely sensitive to the exit clearance at the outer diameter of the aft-platform seal for a fixed hot gas inlet flow.

3. Leak Through the Stack Bolts

For the third two-dimensional test run, a "hole" is simulated underneath the bolts which secure the lift-off stack. The rationale behind such a study is that a flow leaking past these bolts into the coolant liner might be one explanation for the erratic temperatures and pressures sometimes recorded in the coolant liner. The exit area and loss coefficient at this hole are adjusted so that the calculated leakage rate is 0.2 lbm/sec. The flowrate of 0.2 lbm/sec comes from the best estimate of the upper limit of what the leak rate could be, based on the known temperature and pressure measurements in the liner [6].

Figures 11 and 12 show that a leak of 0.2 lbm/sec through the stack bolts does not dramatically change the flow field or temperature field as compared with the no-leak, baseline case. The temperature of the main cavity and the fluid leaking out past the bolts remains relatively unchanged at around 375°R (-85°F).

ORIGINAL PAGE IS
OF POOR QUALITY

BASECASE

DSK 2D BC

O.D. GAP =
.0103"

2-D SOLUTION

EXIT PRESSURE = 3558 PSI

FLOWRATE THROUGH THE BLADE SHANKS = 3.65 lbm/s

RESULTANT LABYRINTH FLOWRATE = .26 lbm/s

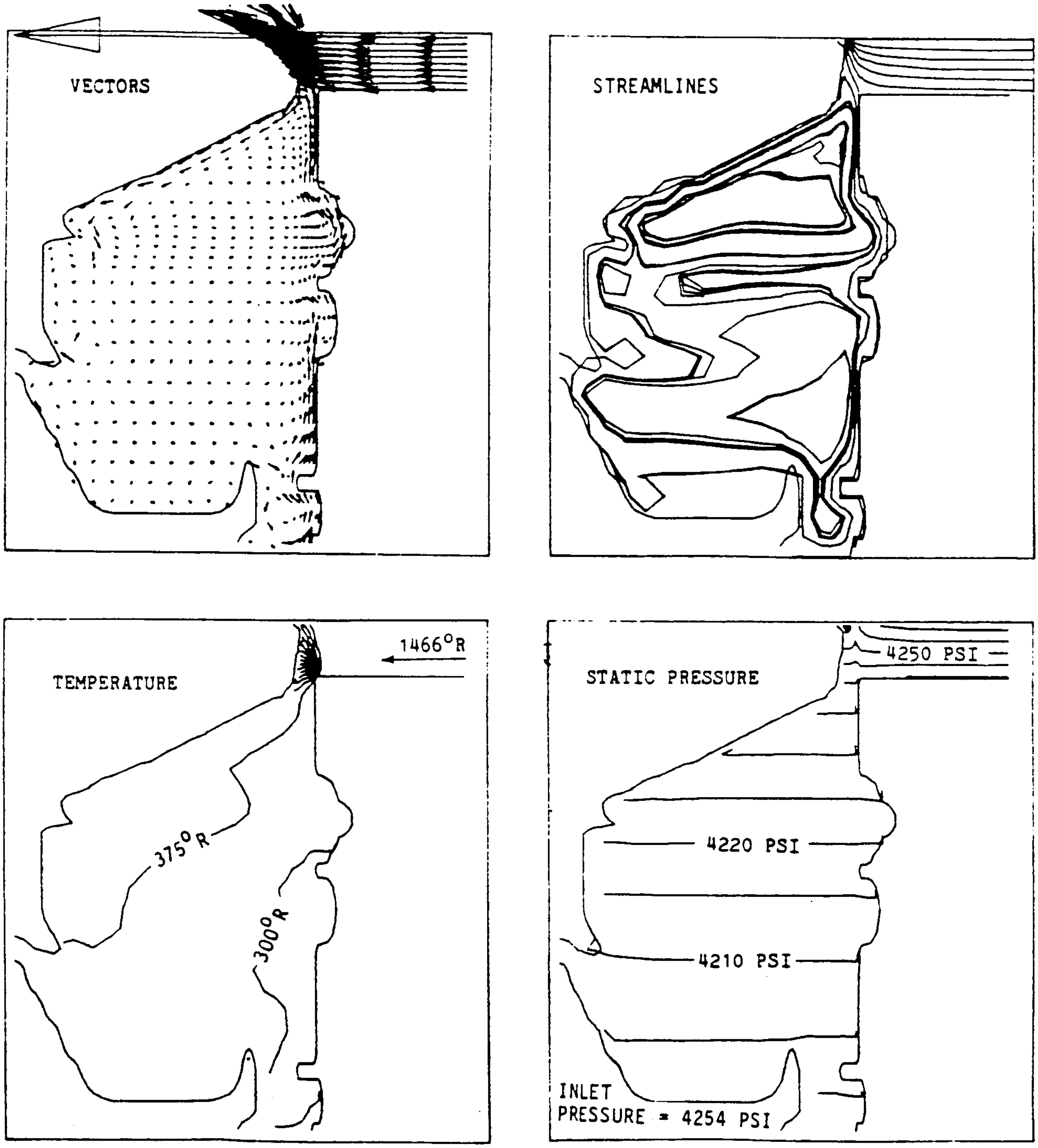


Figure 7. Two-dimensional basecase results.

BASECASE

DSK 2D BC O.D. GAP = .0108"

2-D SOLUTION

EXIT PRESSURE = 3558 PSI

FLOWRATE THROUGH THE BLADE SHANKS = 3.65 lbm/s

RESULTANT LABYRINTH FLOWRATE = .26 lbm/s

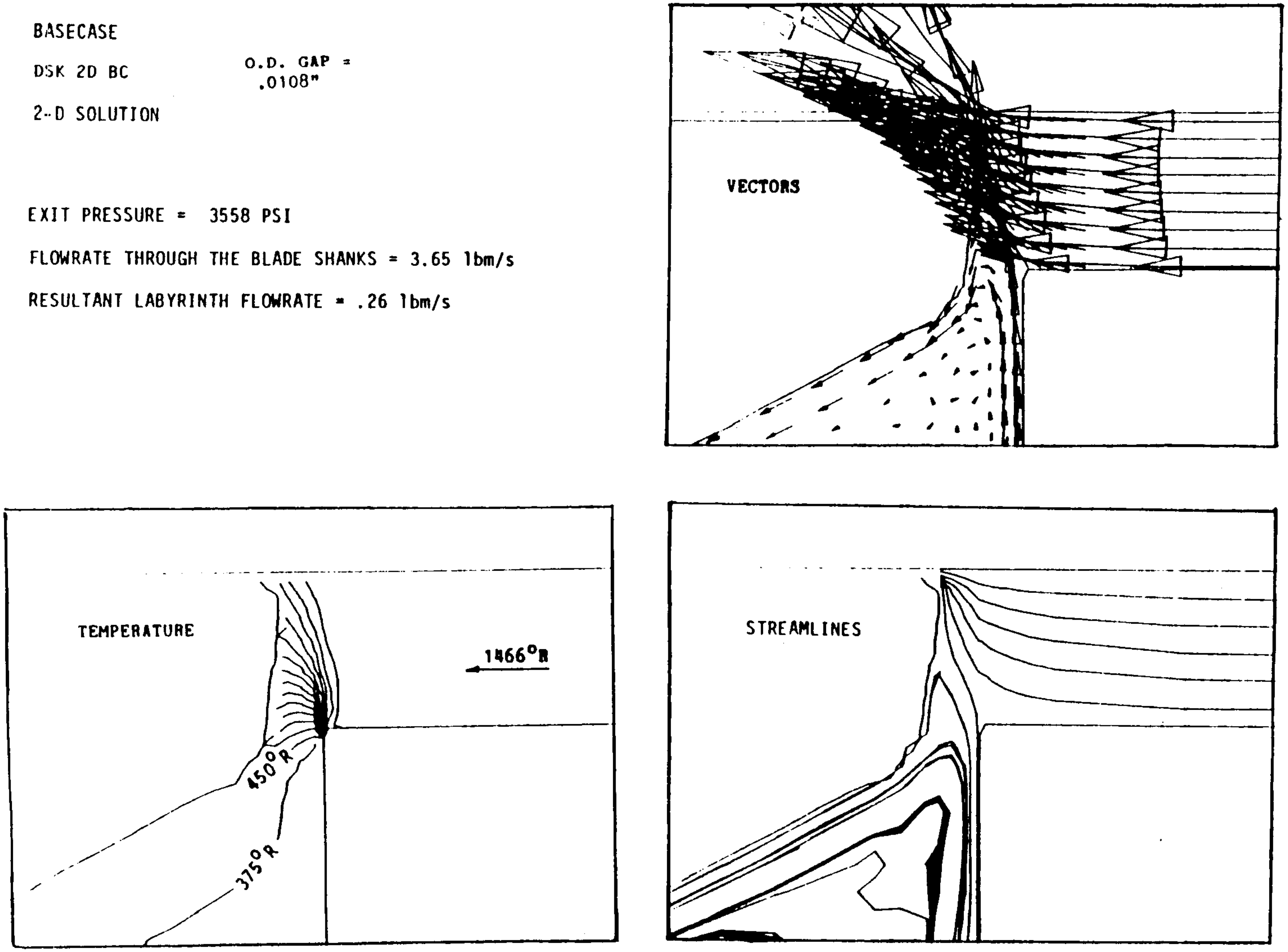


Figure 8. Two-dimensional basecase results, expanded view.

DSK 2D B

2-D SOLUTION

O.D. GAP = .0102"

(.0108" - .0006")

EXIT PRESSURE = 3558 PSI

FLOWRATE THROUGH THE BLADE SHANKS = 3.65 lbm/s

RESULTANT LABYRINTH FLOWRATE = .12 lbm/s

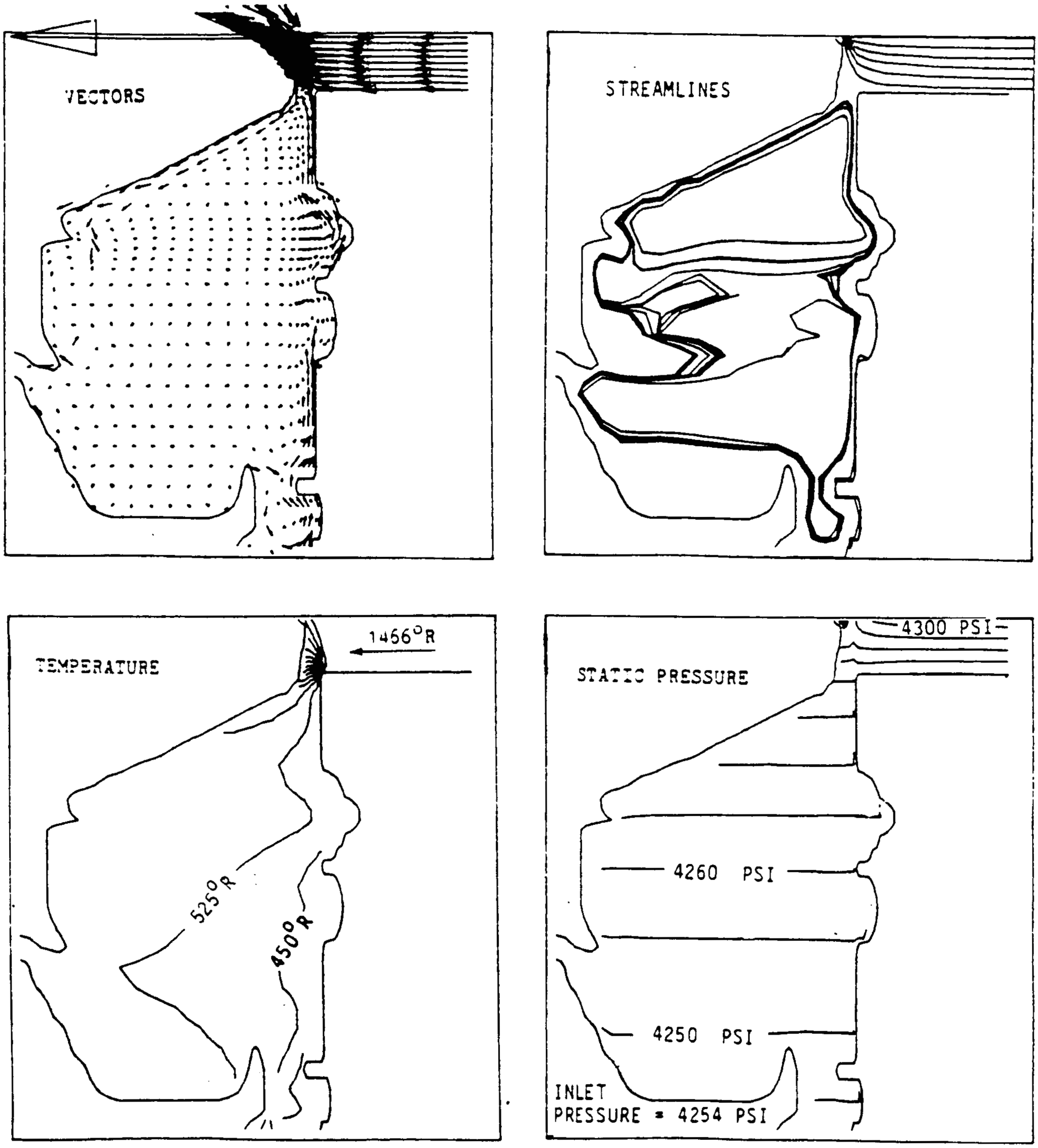


Figure 9. Two-dimensional reduced coolant flow results.

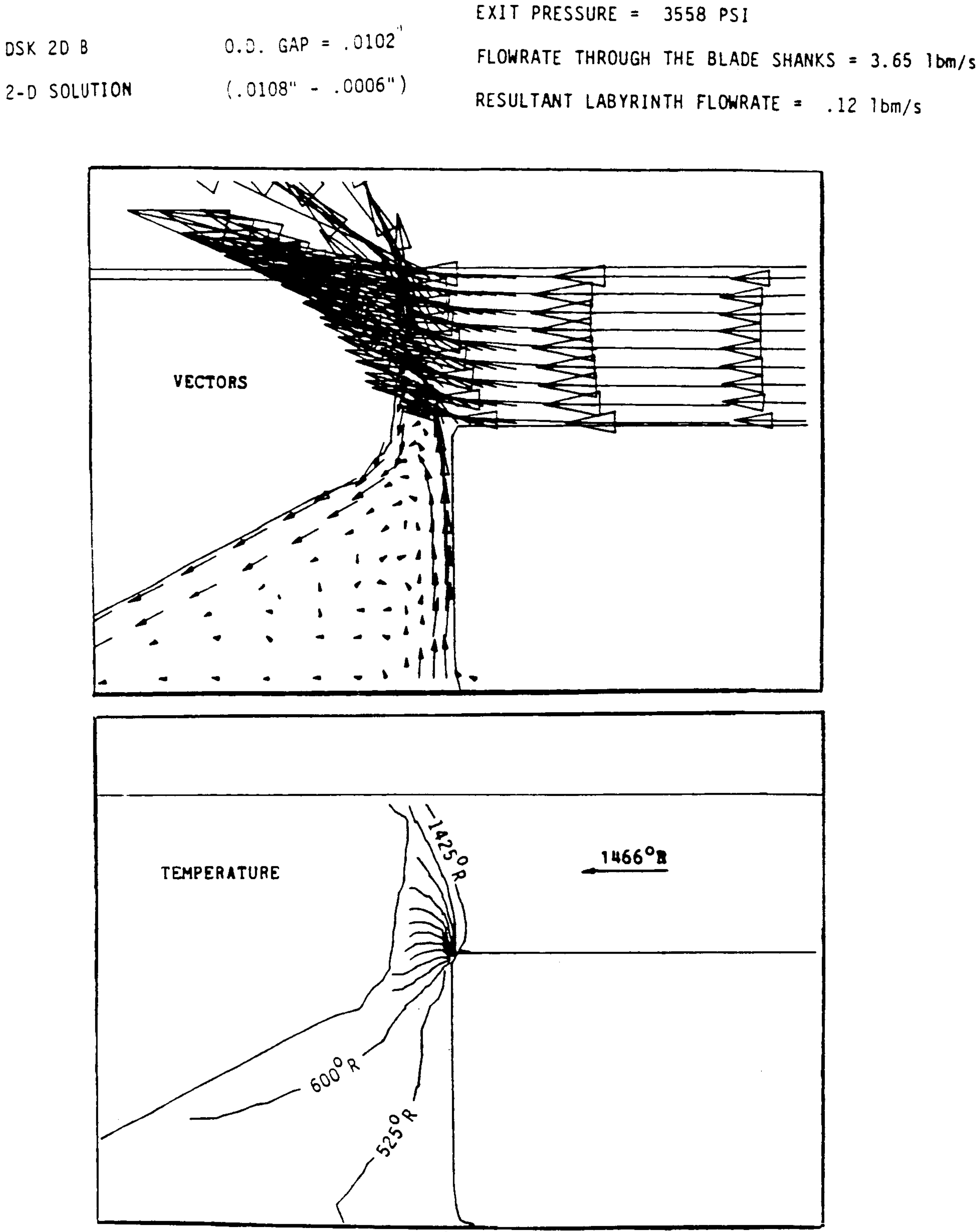


Figure 10. Two-dimensional reduced coolant flow, expanded view.

SECOND EXIT

DSK 2D 2H

2-D SOLUTION

O.D. GAP = .0108"

EXIT PRESSURE = 3558 PSI

FLOWRATE THROUGH THE BLADE SHANKS = 3.65 lbm/s

RESULTANT LABYRINTH FLOWRATE = .34 lbm/s

SECOND EXIT HOLE FLOWRATE = .20 lbm/s

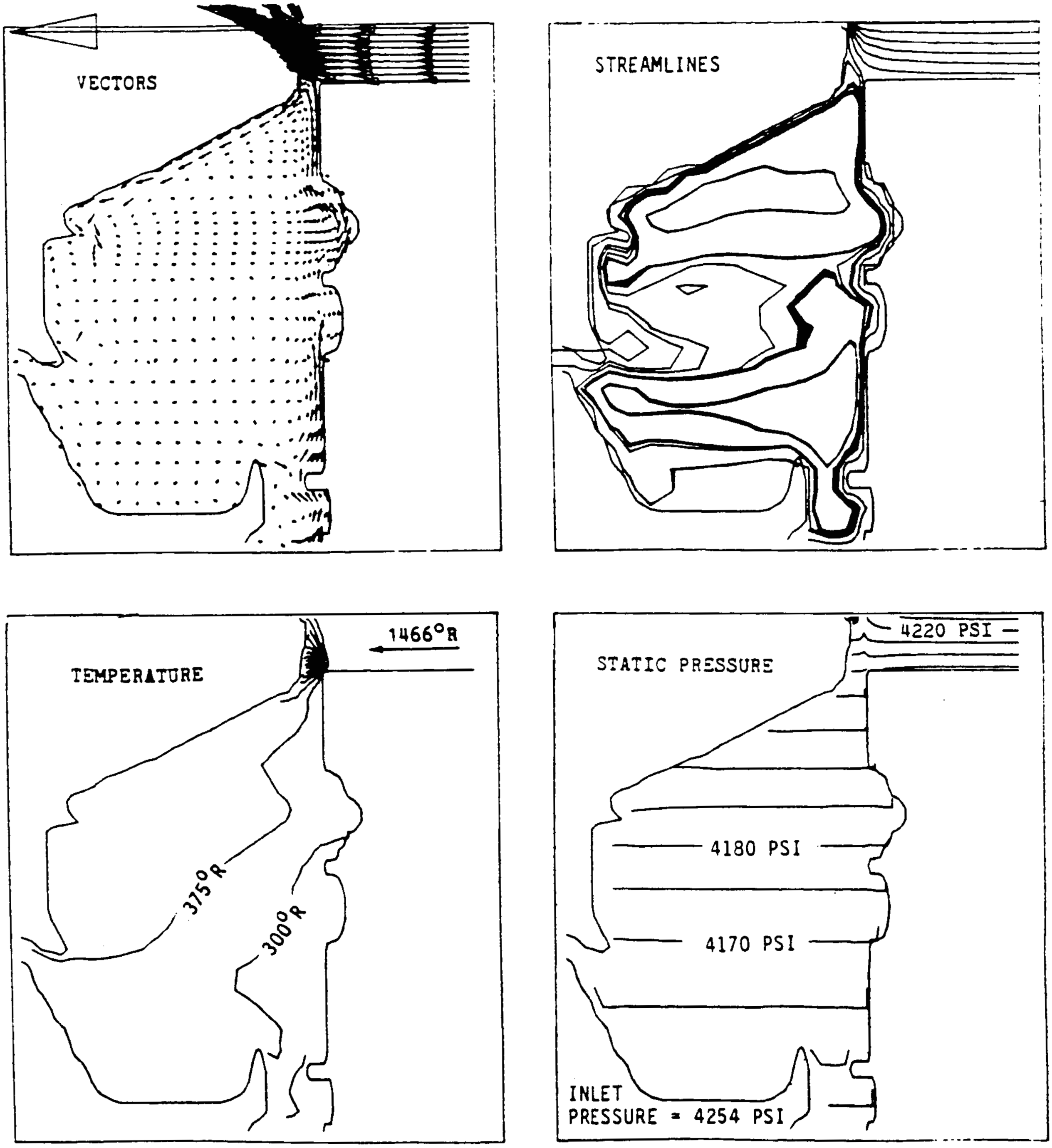


Figure 11. Two-dimensional 0.2 lbm/s leak flow results.

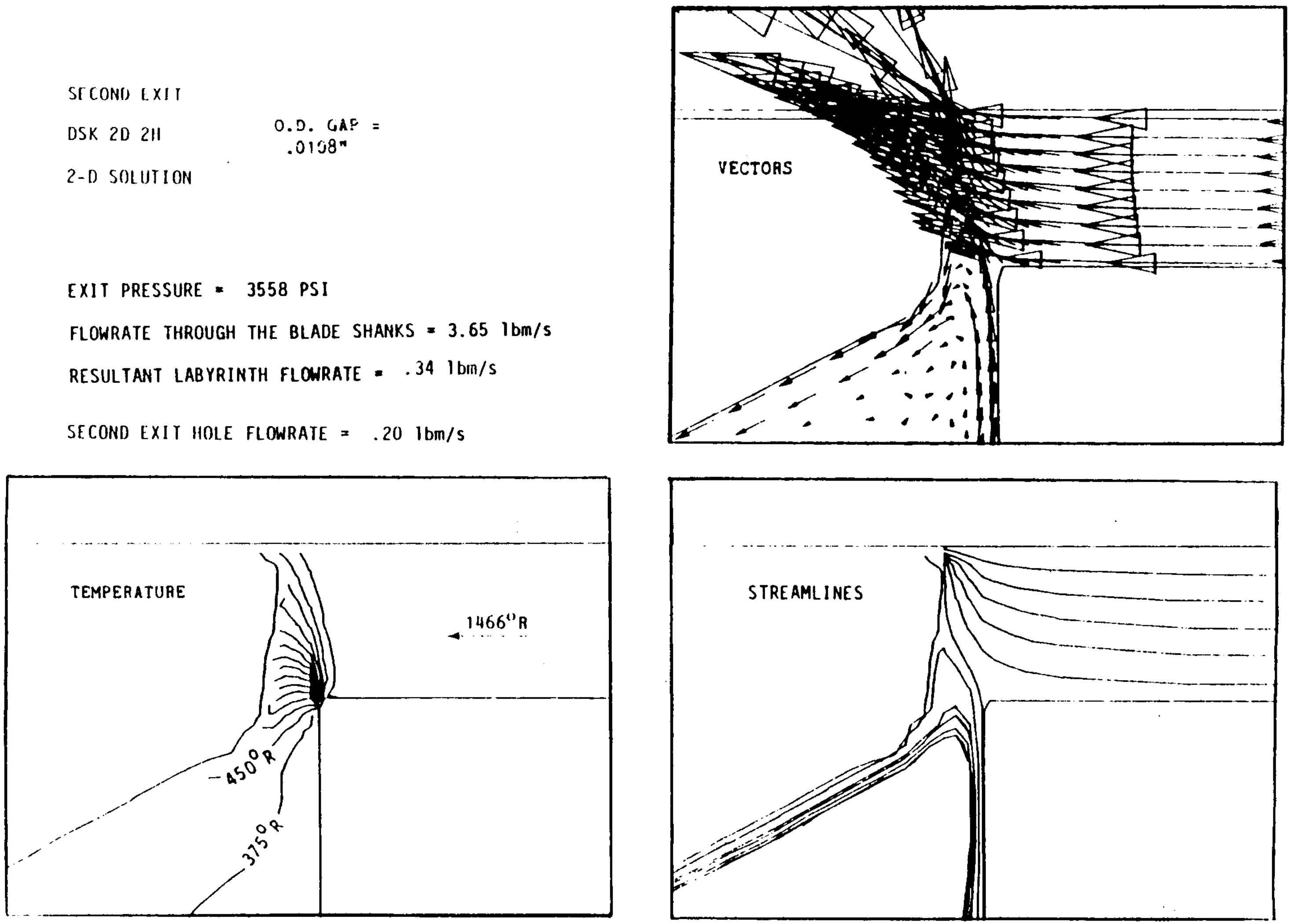


Figure 12. Two-dimensional 0.2 lbm/s leak flow, expanded view.

C. Convergence Characteristics and Computer Time

Numerical solutions of flows involving rotating boundaries are notoriously slow to converge for a variety of reasons (not to be discussed here) and so it was deemed essential that careful checks be made to ensure that the PHOENICS solutions being obtained were meaningful. Thus, before the two-dimensional production runs described above were fully completed, a series of test calculations were performed to ensure that the solutions were converged to an acceptable degree. To this end, various runs were made for the basecase setup with different initial guess/starting solutions that were quite extensive. The results of these investigations are presented and discussed in Appendix C. As shown in the latter, the PHOENICS solutions are clearly converging to an identical solution in each case, as should (and must) be expected.

As depicted in Appendix C, the two-dimensional basecase was run for a total of 500 sweeps, at which time all calculated monitor flow variables had settled to an acceptable degree (Figs. C-1 to C-8). All the other 2-dimensional runs reported here were restarted from this basecase solution (i.e., the initial fields for the starting of the iterative calculation procedure were taken to be the basecase solution, rather than some simple initial guess) and then run on until, again, the solution monitor values were suitably settled. This usually required another 150 to 200 sweeps, at most. Computer times for these restart runs were approximately 35 CPU minutes on CHAM's Perkin Elmer 3251 mini-computer.

All the 3-dimensional calculations described in the next section were also restarted from the 2-dimensional basecase solution which was symmetrically duplicated in the circumferential IX-direction. Again, converged solutions then took approximately 150 to 200 more sweeps and required approximately 5 CPU hours of computer time on the Perkin-Elmer 3251 machine.

V. THREE-DIMENSIONAL TEST RUNS

The disadvantage of the preceding axisymmetric analysis is that, by definition, it does not include the three-dimensional effects either known or suspected to exist in the pump. One of the most important of these asymmetries left unaccounted for by the two-dimensional analysis is the circumferential variation in pressure which has been measured downstream of the exit of the fuel turbine. This exit pressure serves as one of the boundary pressures which regulates the flow in the aft-platform seal cavity. In addition to this known pressure variation, there may be variations in clearances or other parameters which could radically alter the flow pattern in the cavity. As such, a three-dimensional model is an essential tool for a proper study of this cavity. As a starting point, three different three-dimensional cases were run and are presented here. The first is the basecase which uses the same set of flowrates, fluid properties, and clearances as used in the two-dimensional basecase. The only difference between the two is that the three-dimensional basecase also includes a prescribed asymmetrical turbine exit pressure based on pressure measurements taken during a full scale test of the shuttle engine. The second three-dimensional case was set-up to simulate a 0.003 in. shift in the rotor position with a corresponding change in the clearance at the labyrinth seal and at the exit gap between the aft-platform seal and the blade lip. This shift is relative to the average labyrinth seal clearance of 0.003 in. and the average exit gap of 0.0108 in. The last three-dimensional run presented here simulates a relatively large eccentricity of the aft-platform seal alone, such that the exit clearance is skewed to one side by 0.0081 in., which is 75 percent of its average clearance.

A. Three-Dimensional Test Runs: Boundary Conditions

1. Basecase (Geometrically Axisymmetric with Asymmetric Exit Pressures)

As its boundary conditions, the basecase three-dimensional run uses the same operating clearances, flowrates, pressures, mixture ratios, and enthalpies, etc., as used by the two-dimensional basecase analysis. The only exception is that the exit pressure of the turbine is no longer uniform but varies circumferentially based on data taken during Rocketdyne's engine test 902-279 [7]. These boundary conditions are the best estimate of the operating conditions in the fuel pump at full power (109 percent). The specific numbers used for this run, and for the subsequent three-dimensional runs, are listed in Table 2.

2. Eccentric Rotor (Rotor Shift of 0.003 in.)

The Eccentric Rotor (0.003 in.) case was set up to simulate the effect that a rotor shift of 0.003 in. would have on the flow field in the cavity. The shift of 0.003 in. was chosen because it is an upper limit on the distance the rotor can shift before the shaft starts rubbing against the labyrinth seal. Such a rotor shift in a given direction would open up the exit clearance between the aft-platform seal and the blade lips, while at the same time it would close down the clearance at the labyrinth seal. This effect is simulated in the model by, on the one hand, directly adjusting the clearances at the outer diameter of the aft-platform seal and, on the other, by adjusting the flow rate at the labyrinth seal. All the other inputs remain the same as for the three-dimensional basecase.

TABLE 2. THREE-DIMENSIONAL BOUNDARY CONDITIONS

Variable		Basecase	Rotor Eccentricity = 0.003 in.	Aft-Platform Eccentricity = 0.0081 in.
Rotational speed of the disk (RPM)		37,000	37,000	37,000
Flowrate at the labyrinth seal (lbm/sec)	1:00	0.0323	0.0095	0.0323
	2:30	0.0323	0.0323	0.0323
	4:00	0.0323	0.0551	0.0323
	5:30	0.0323	0.0646	0.0323
	7:00	0.0323	0.0551	0.0323
	8:30	0.0323	0.0323	0.0323
	10:00	0.0323	0.0095	0.0323
	11:30	0.0323	0.0000	0.0323
Total Mass Flowrate		0.258	0.258	0.258
Total flow area (360°) between the blade shanks (in.^{2})		3.877	3.877	3.877
Clearance between the aft-platform seal and blades (in.)	1:00	0.0108	0.0129	0.0165
	2:30	0.0108	0.0108	0.0108
	4:00	0.0108	0.0087	0.0051
	5:30	0.0108	0.0078	0.0027
	7:00	0.0108	0.0087	0.0051
	8:30	0.0108	0.0108	0.0108
	10:00	0.0108	0.0129	0.0165
	11:30	0.0108	0.0138	0.1800
Total Area		0.307	0.307	0.307
Loss coefficient at the exit near the blade shanks		1.5	1.5	1.5
Enthalpy of the H_2 entering at the labyrinth seal (Btu/lbm) (Resultant calculated temperature – degrees Rankine)		278.3 (145°R)	278.3 (145°R)	278.3 (145°R)
Enthalpy of H_2 and H_2O entering through the blades (Btu/lbm) (Resultant calculated temperature – degrees Rankine)		3380 (1466°R)	3380 (1466°R)	2280 (1466°R)
Density of the H_2 entering at the labyrinth seal (lbm/ft^3)		3.574	3.574	3.574
Density of H_2 and H_2O entering through the blades (lbm/ft^3)		0.931	0.931	0.931
Mass flowrate of H_2 and H_2O entering past the blades (lbm/s)		3.649	3.649	3.649
Mass fraction of H_2O entering through the blades		0.474	0.474	0.474
Pressure at the turbine discharge (psi)	1:00	3451	3451	3451
	2:30	3541	3541	3541
	4:00	3697	3697	3697
	5:30	3622	3622	3622
	7:00	3606	3606	3606
	8:30	3592	3592	3592
	10:00	3476	3476	3476
	11:30	3481	3481	3481
Average Exit Pressure		3558	3558	3558

3. Eccentric Aft-Platform Seal (Aft-Platform Seal Shift of 0.0081 in.)

The third three-dimensional test run models the flow field for a highly eccentric (75 percent) aft-platform seal. In this run the clearance at the gap between the aft-platform seal and the blade lips is adjusted so that it models what the gap would be if the aft-platform seal had moved laterally 0.0081 in. in the 11:30 direction. Note that the rotor itself has not moved but is still concentric with the labyrinth seal so that the gap between the labyrinth seal and the rotor axle remains at a uniform 0.003 in. (In general, the clocking positions used in this report correspond to the convention adopted by Rocketdyne in Reference 7, however, in this particular test run the decision to move the aft-platform seal in the 11:30 direction is arbitrary, and is based on convenience rather than any physical justification.) The choice of the magnitude of the eccentricity is also somewhat arbitrary but the reasoning behind the shift of 0.0081 in. was the desire to choose a large aft-platform eccentricity in order to observe extreme effects. An aft-platform shift of 0.0081 in. is 75 percent of the total aft-platform seal clearance.

B. Three-Dimensional Test Runs: Results and Observations

1. Basecase

A comparison of the three-dimensional basecase results (Figs. 13 to 21) with the two-dimensional basecase results (Figs. 7 and 8) shows that the addition of an asymmetric pressure distribution at the exit of the turbine has had little effect on the flow pattern in the aft-platform seal cavity. While some evidence of the influence of the external pressure distribution can be seen at the outer diameter of the disk near the blade shanks (e.g., Fig. 16), this effect is small; toward the center of the cavity the results are nearly identical to the two-dimensional solution. At the flowrates and small clearances of the aft-platform seal cavity running at full power, a circumferential pressure difference of 220 psi as modeled here represents only a fraction of the over 600 psi pressure drop between the aft-platform seal cavity and the turbine exhaust. As a result, the 220 psi circumferential variation on the outside of the cavity has little effect on the flow pattern inside. In addition, even with the circumferential variation in turbine exhaust pressure, the centrifugal force in the aft-platform seal cavity still dominates the flow such that the influence that is felt due to the pressure variation is confined to the periphery of the cavity. For an example of this effect, examine the lines of constant temperature given in the close-up view in Figure 17.

2. Eccentric Rotor (Rotor Shift of 0.003 in.)

Perhaps the most notable feature of the aft-platform seal cavity flow field (Figs. 22 to 30) with a 0.003 in. eccentric rotor is the small change as compared to the three-dimensional basecase with its centered rotor. Even with an eccentric rotor, the temperatures in the cavity have risen just 75°R, indicating only a slight increase in the heat transferred down into the cavity. Again, the only significant effect is felt at the outer diameter of the turbine disk where, at the 5:30 clock position, the hot gas actually flows down into the cavity causing a local hot spot. This hot spot will be felt by the blade shanks once per revolution, with a corresponding cooling in between. In general, therefore, a rotor shift of 0.003 in. results in a slight warming of the average cavity temperature, and a cyclical variation of temperature at the outer diameter of the disk of approximately 600°R.

3. Eccentric Aft-Platform Seal (Aft-Platform Seal Shift of 0.0081 in.)

Of the six different two-dimensional and three-dimensional test runs investigated during this study, the most dramatic results (Figs. 31 to 39) come from running the model with an aft-platform seal that has been shifted to one side by 3/4 of the exit clearance (i.e., by 0.0081 in.). With the exit gap substantially closed down on one side, the hot gas which would normally exit through that gap must, instead, exit at a different location. The centrifugal force

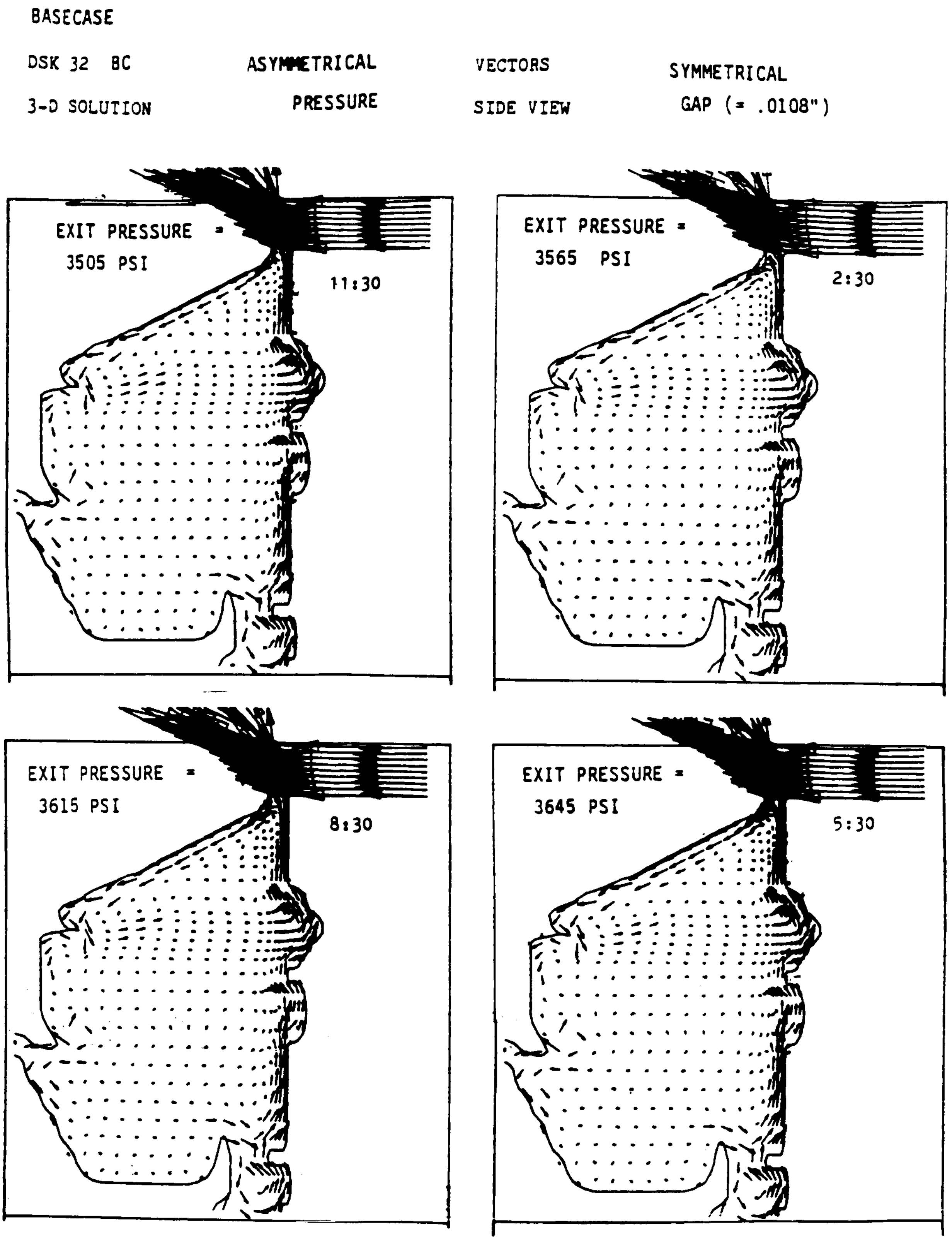


Figure 13. Three-dimensional basecase results: vectors.

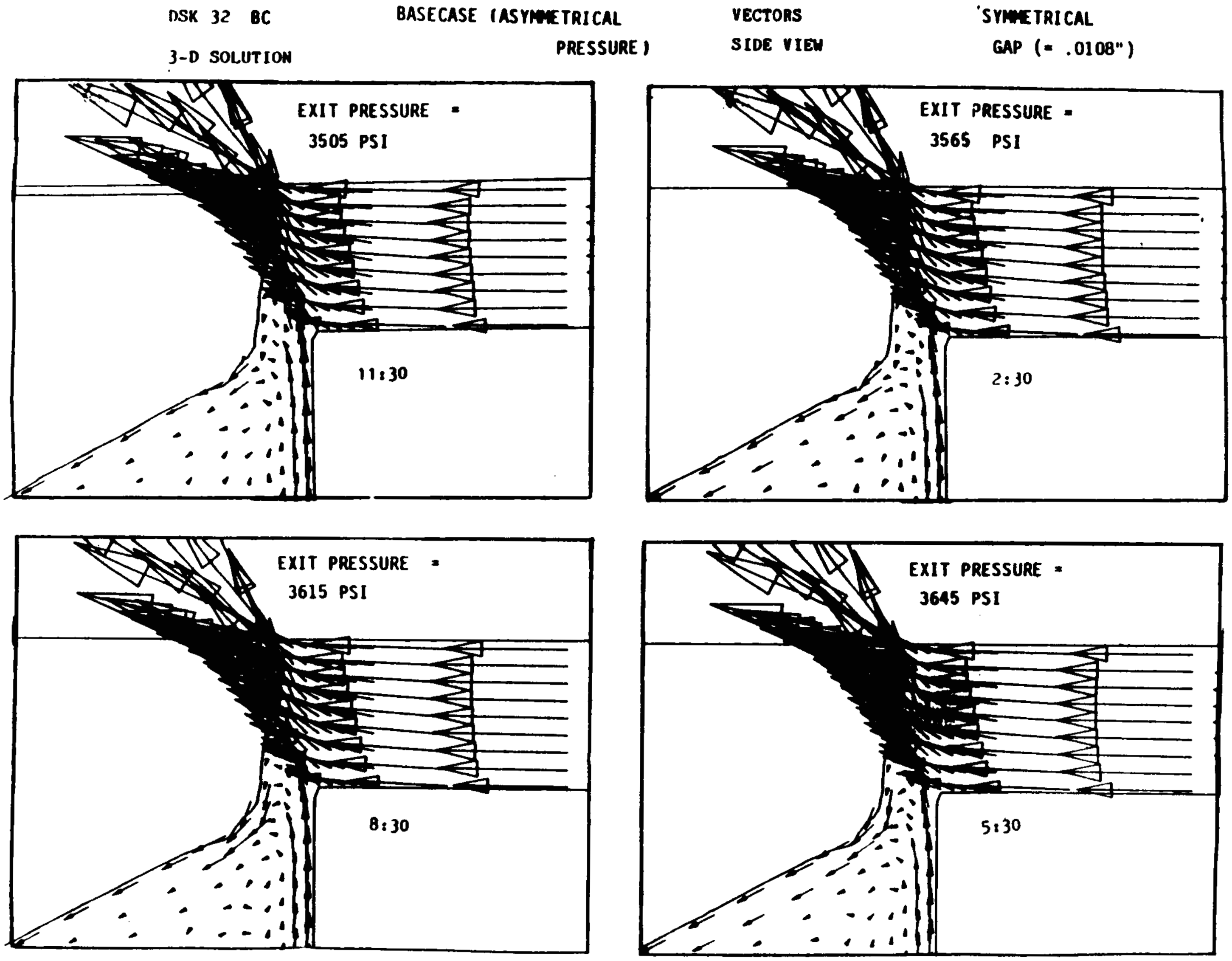


Figure 14. Three-dimensional basecase results: vectors (close-up).

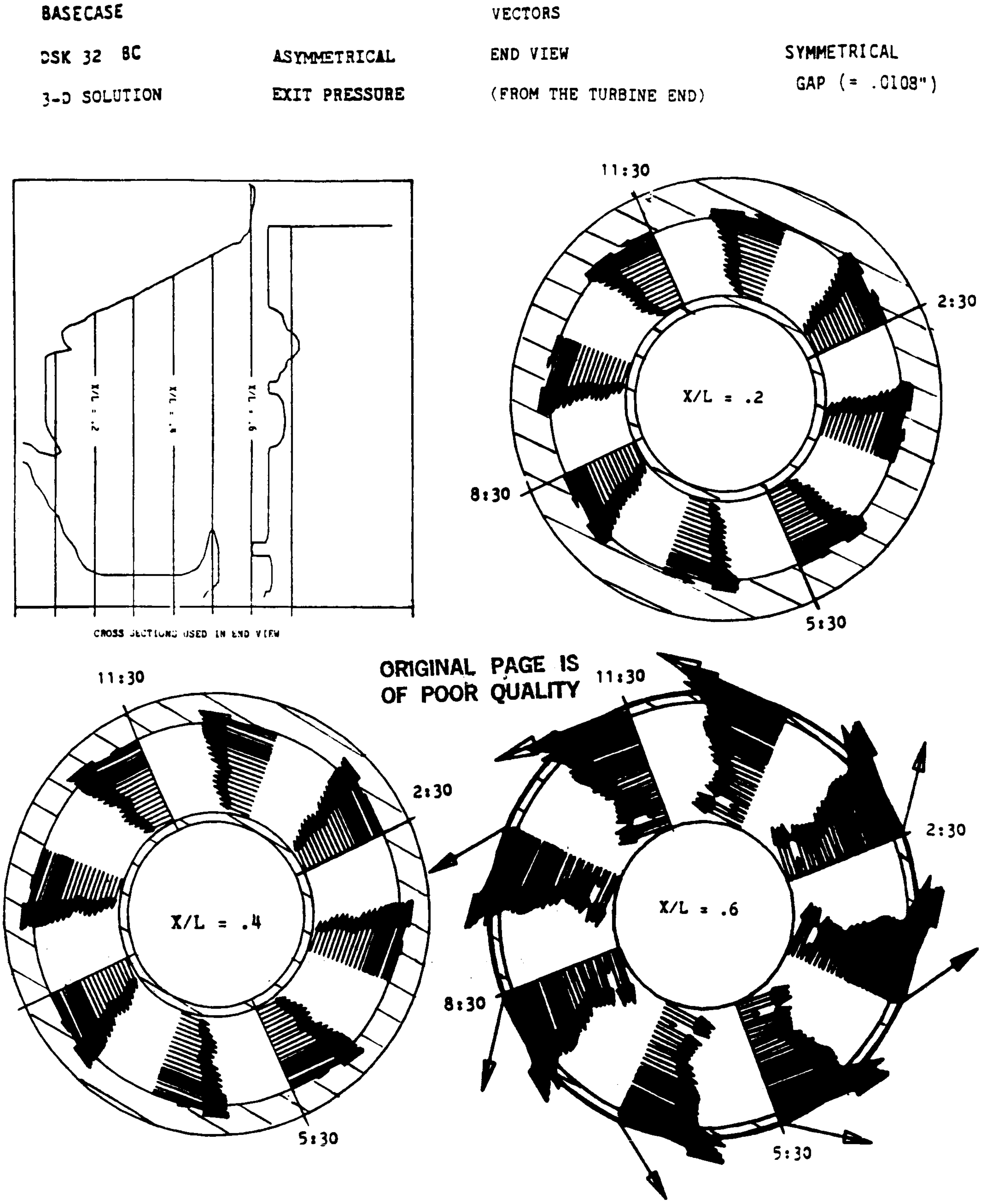


Figure 15. Three-dimensional basecase results: vectors (end view).

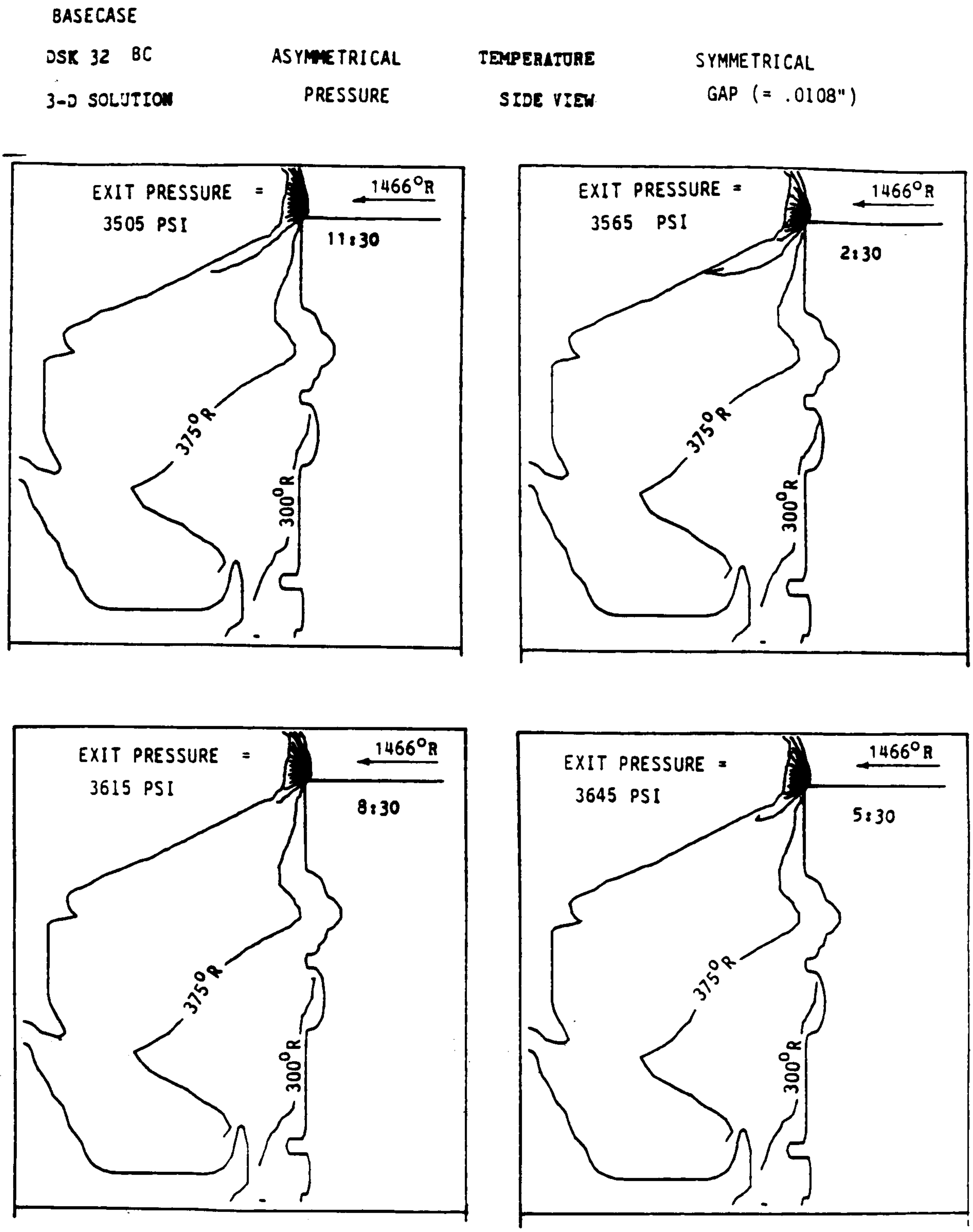


Figure 16. Three-dimensional basecase results: temperature.

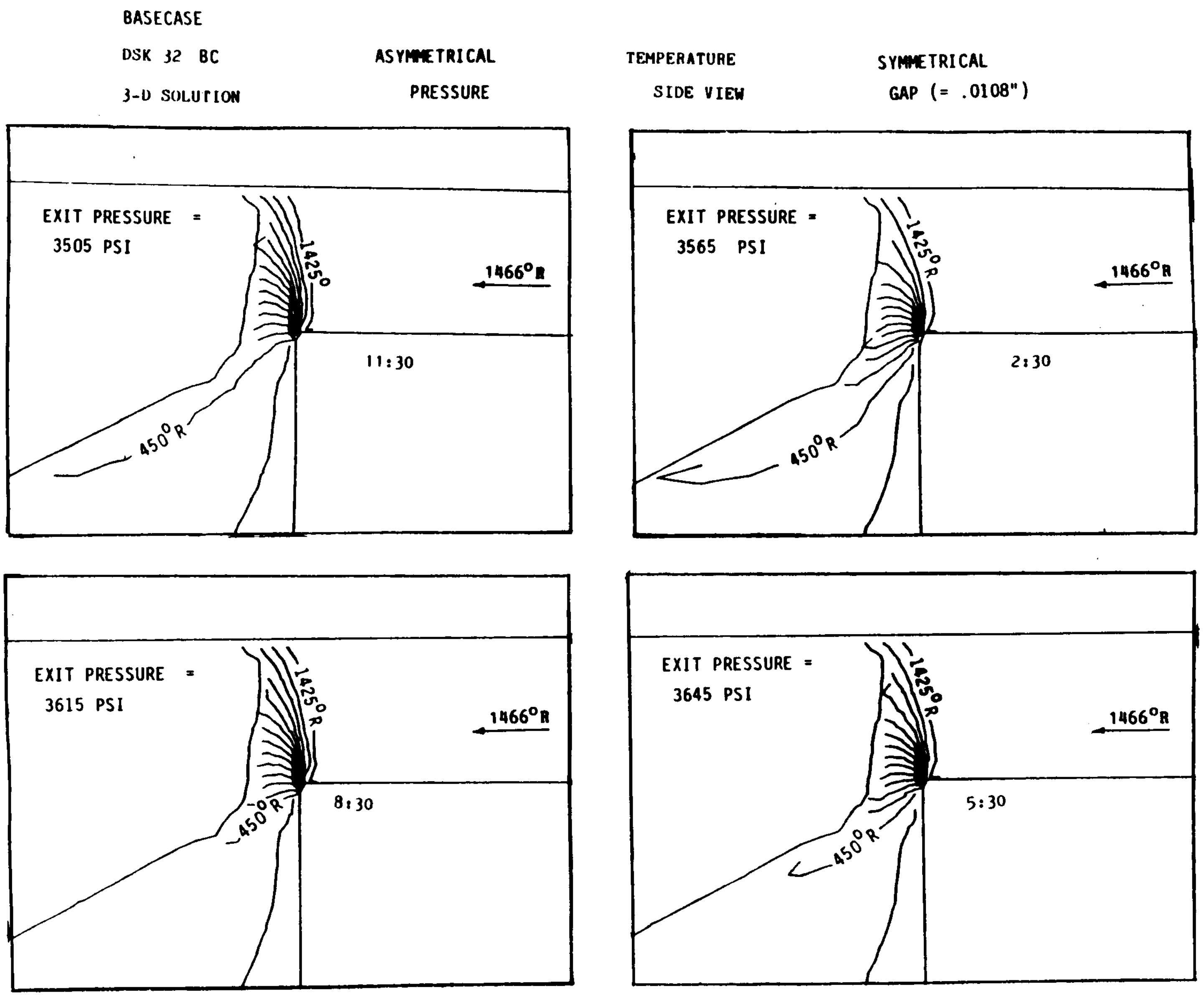


Figure 17. Three-dimensional basecase results: temperature (close-up).

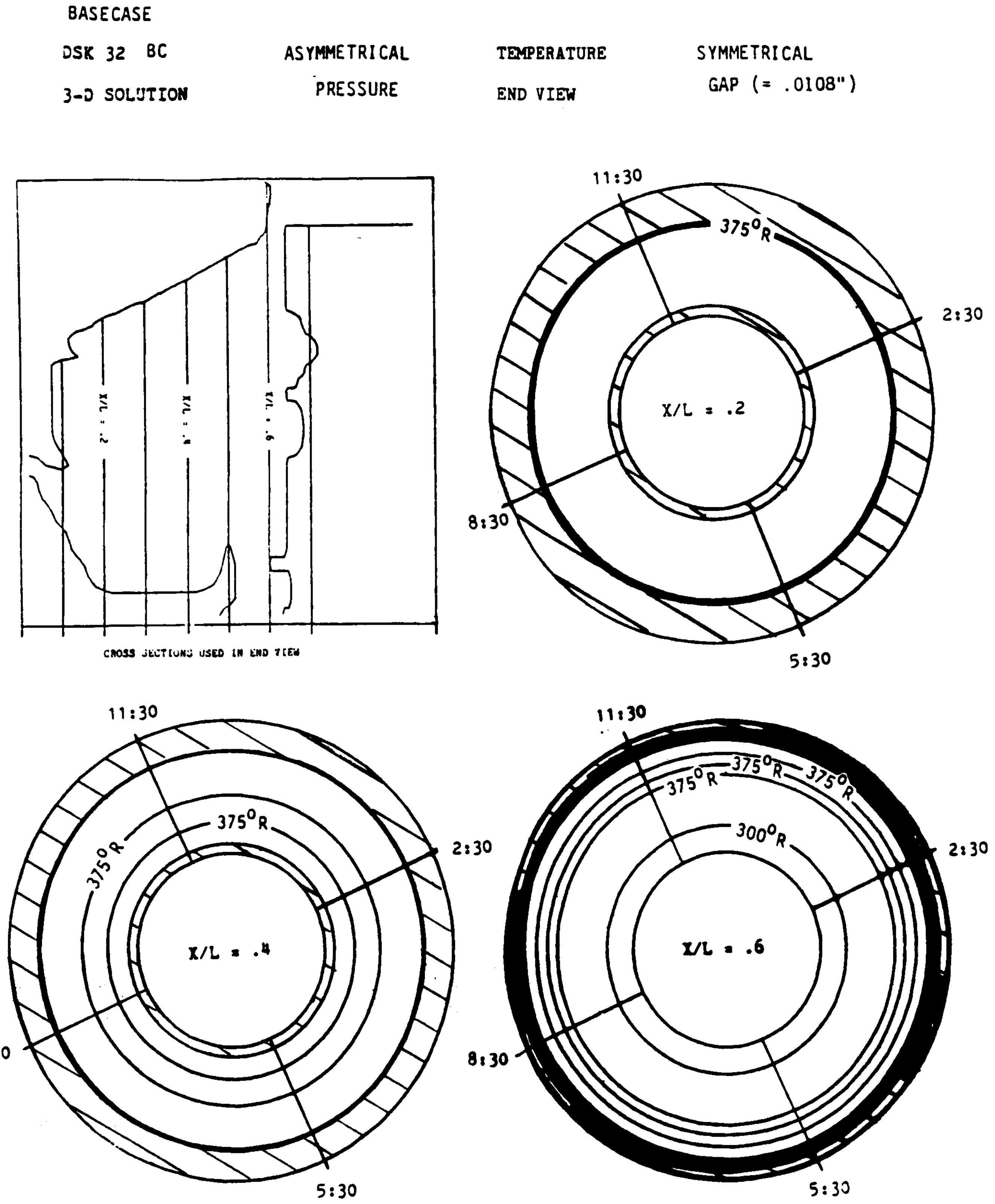


Figure 18. Three-dimensional basecase results: temperature (end view).

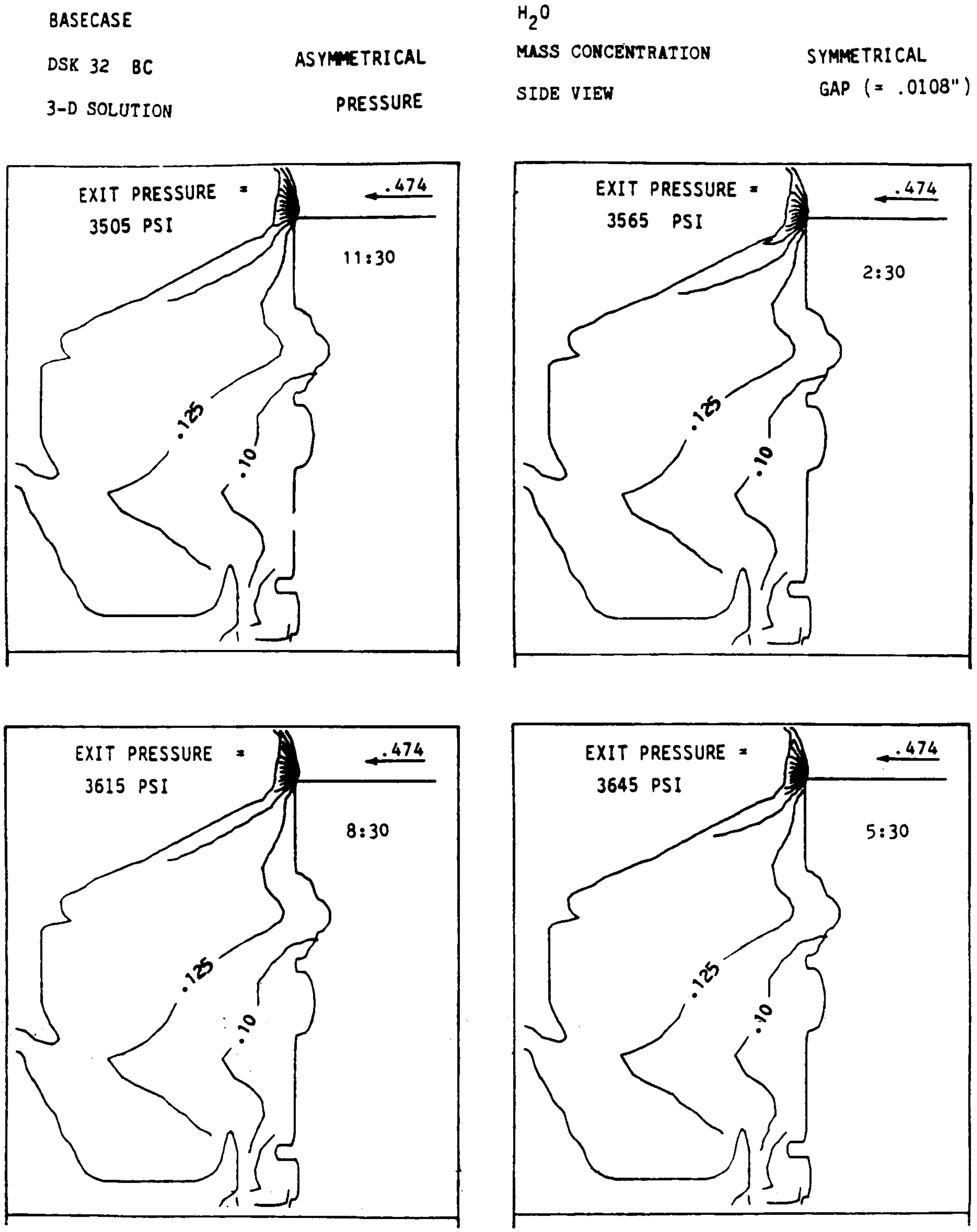


Figure 19. Three-dimensional basecase results: mass concentration.

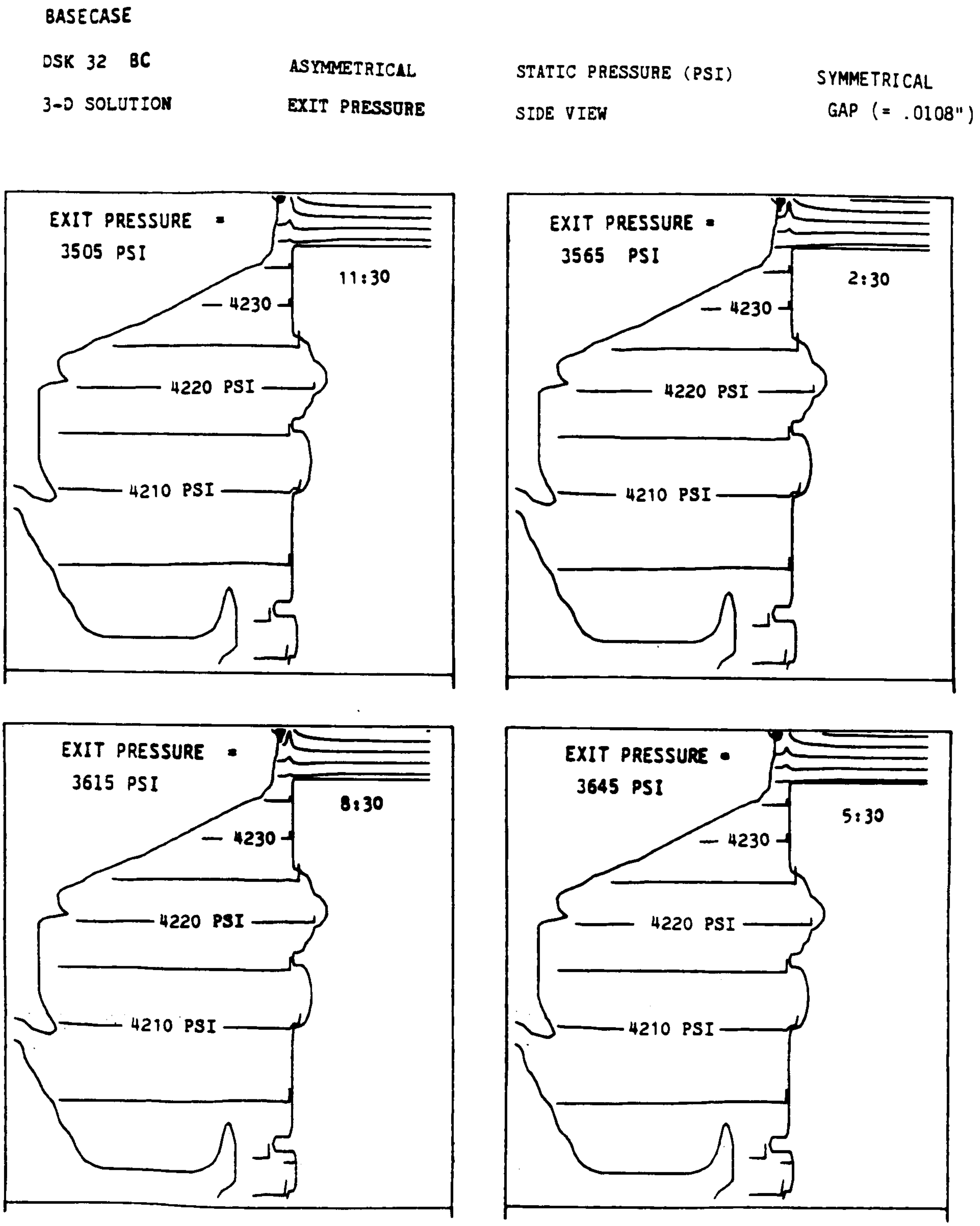


Figure 20. Three-dimensional basecase results: static pressure.

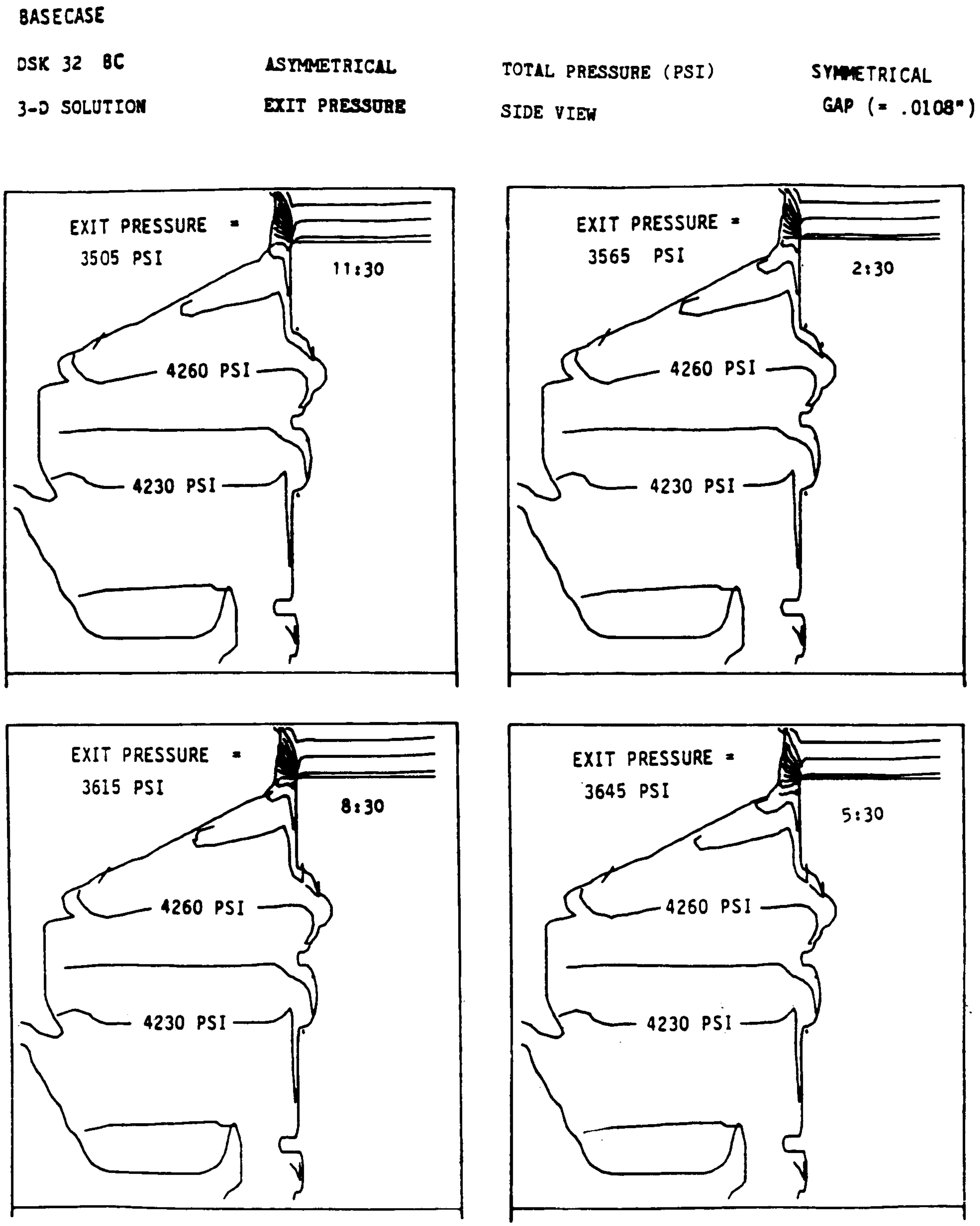


Figure 21. Three-dimensional basecase results: total pressure.

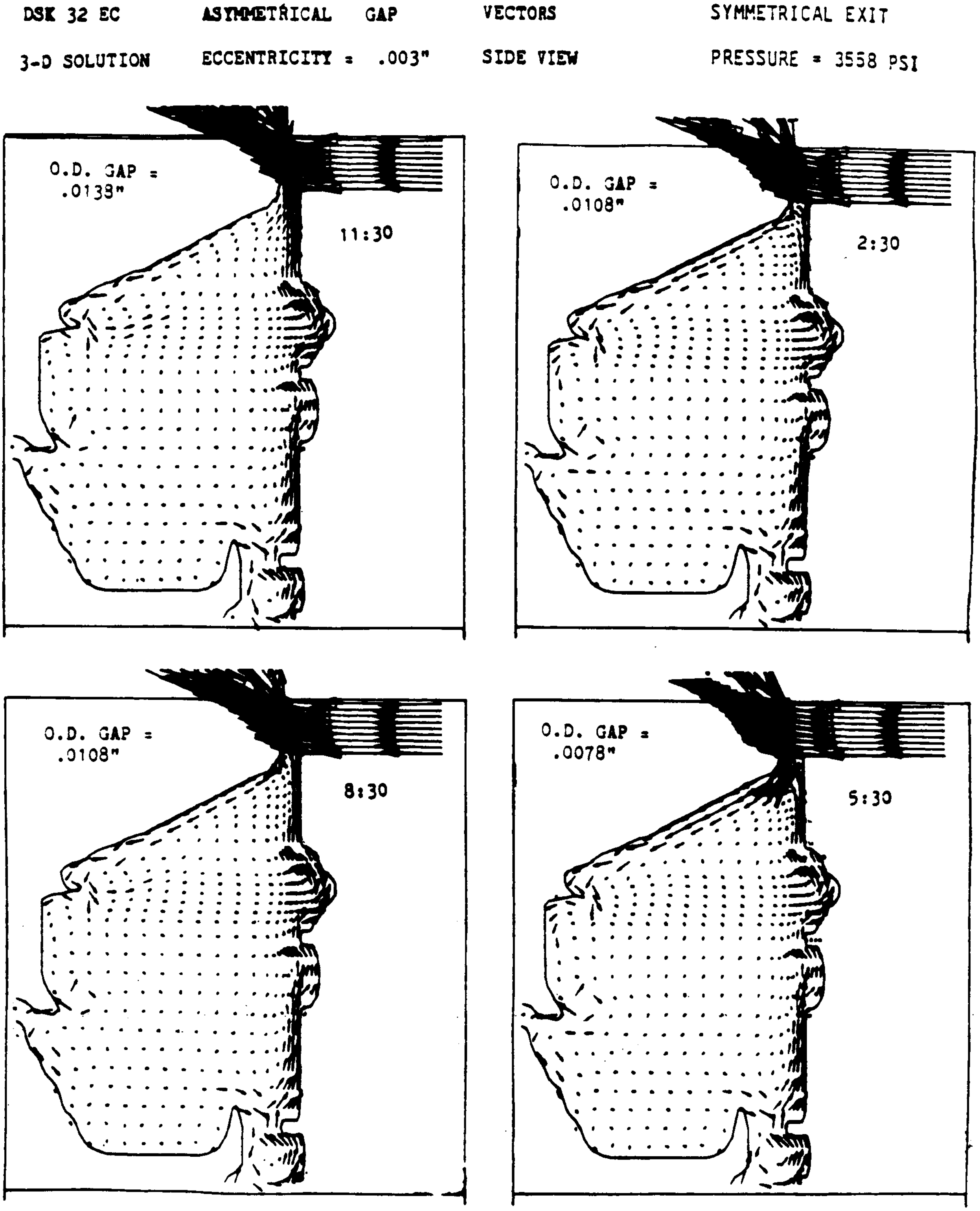


Figure 22. Three-dimensional eccentric (0.003 in.) rotor: vectors.

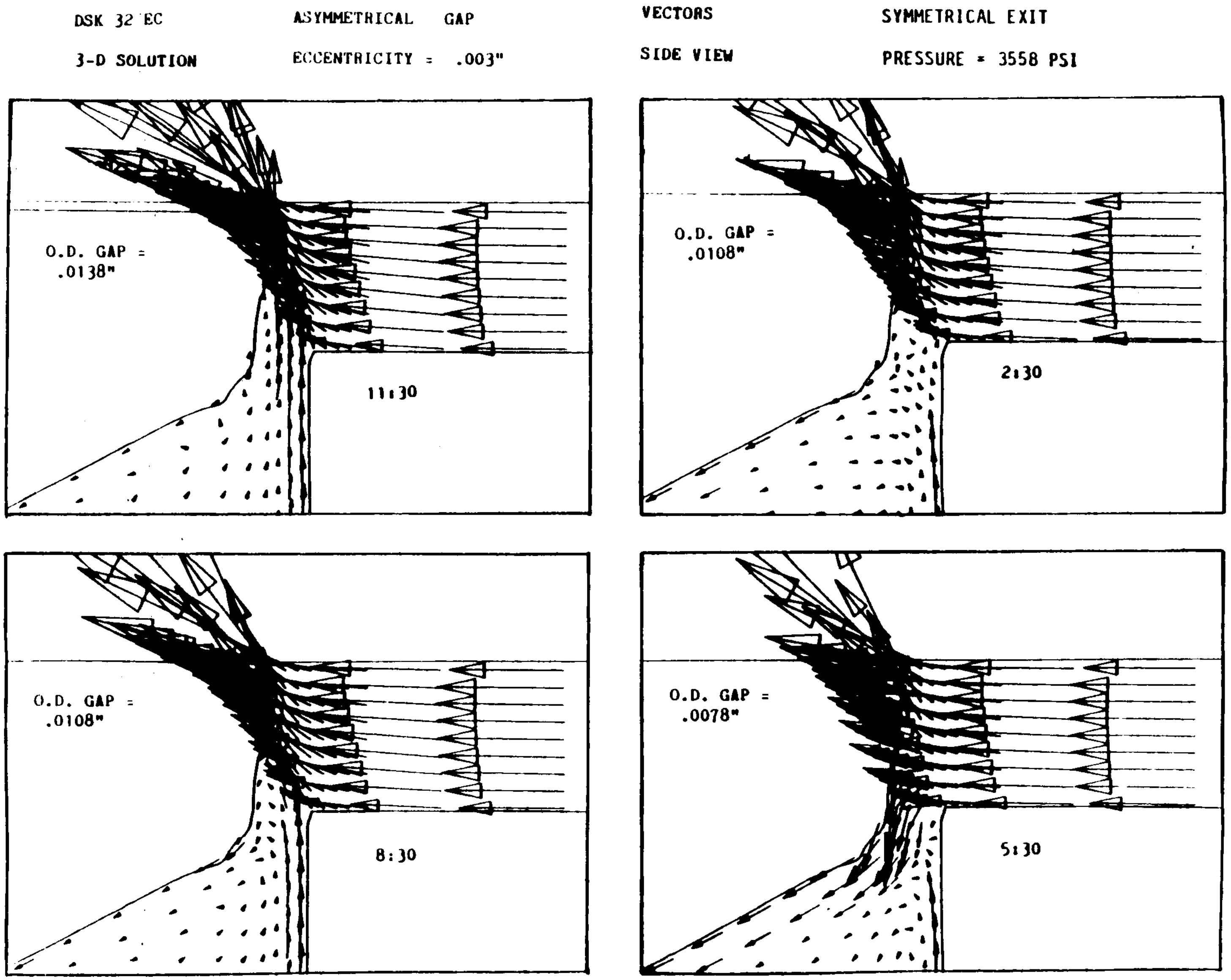


Figure 23. Three-dimensional eccentric (0.003 in.) rotor: vectors (close-up).

VECTORS

DSK 32 EC   ASYMMETRICAL GAP   END VIEW   SYMMETRICAL EXIT

3-D SOLUTION   ECCENTRICITY = .003"   (FROM THE TURBINE END)   PRESSURE = 3558 PSI

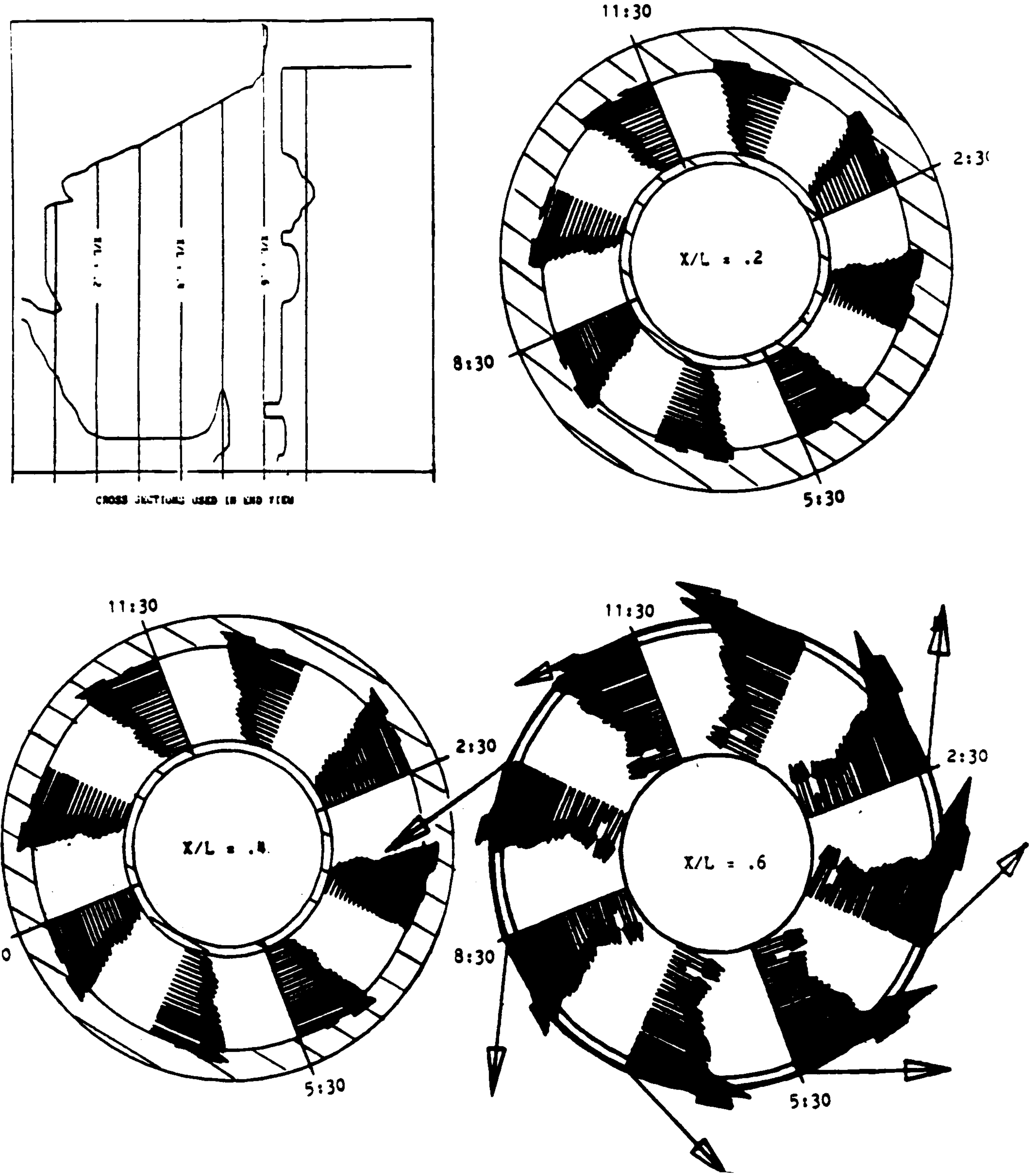


Figure 24. Three-dimensional eccentric (0.003 in.) rotor: vectors (end view).

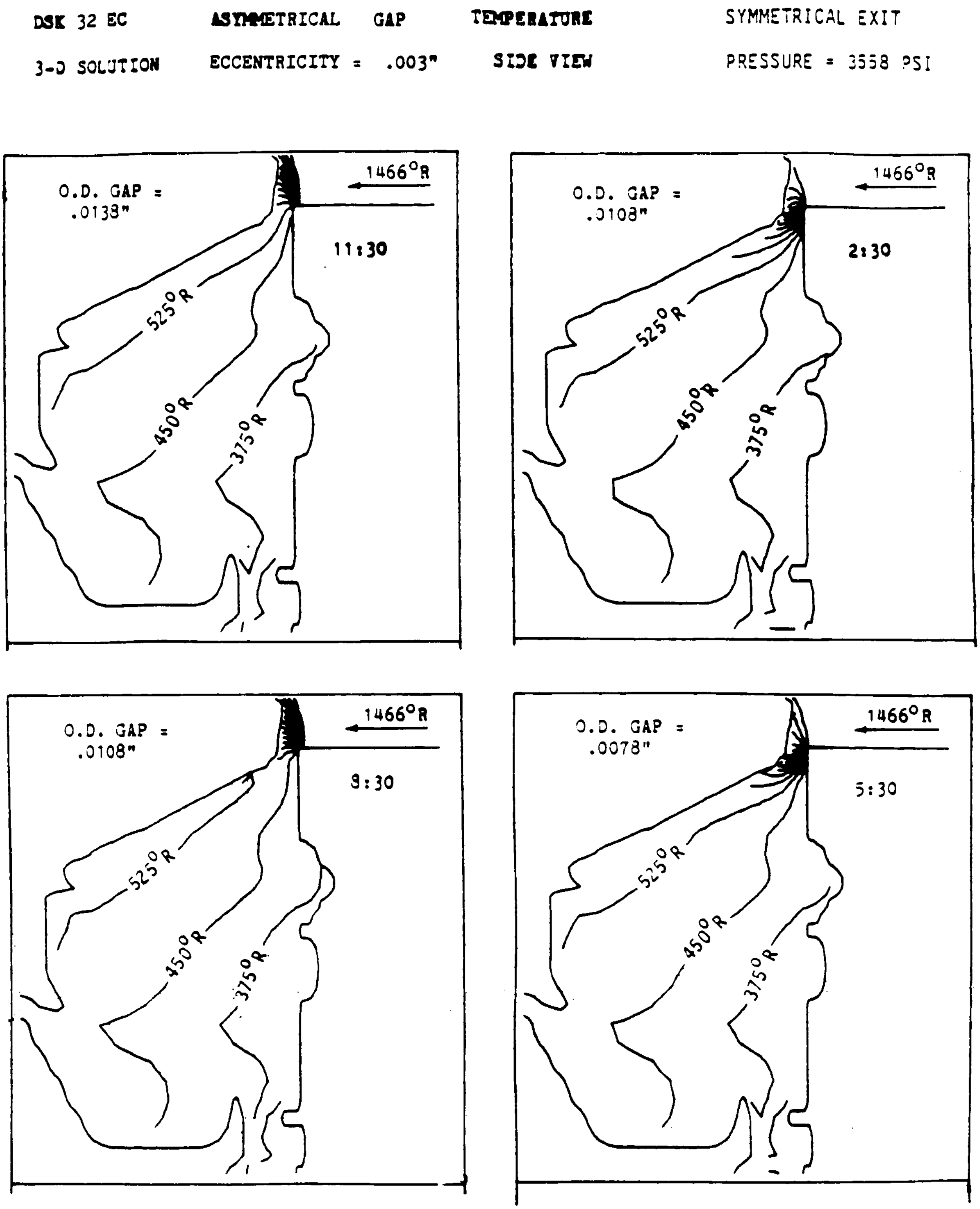


Figure 25. Three-dimensional eccentric (0.003 in.) rotor: temperature.

ORIGINAL PAGE IS
OF POOR QUALITY

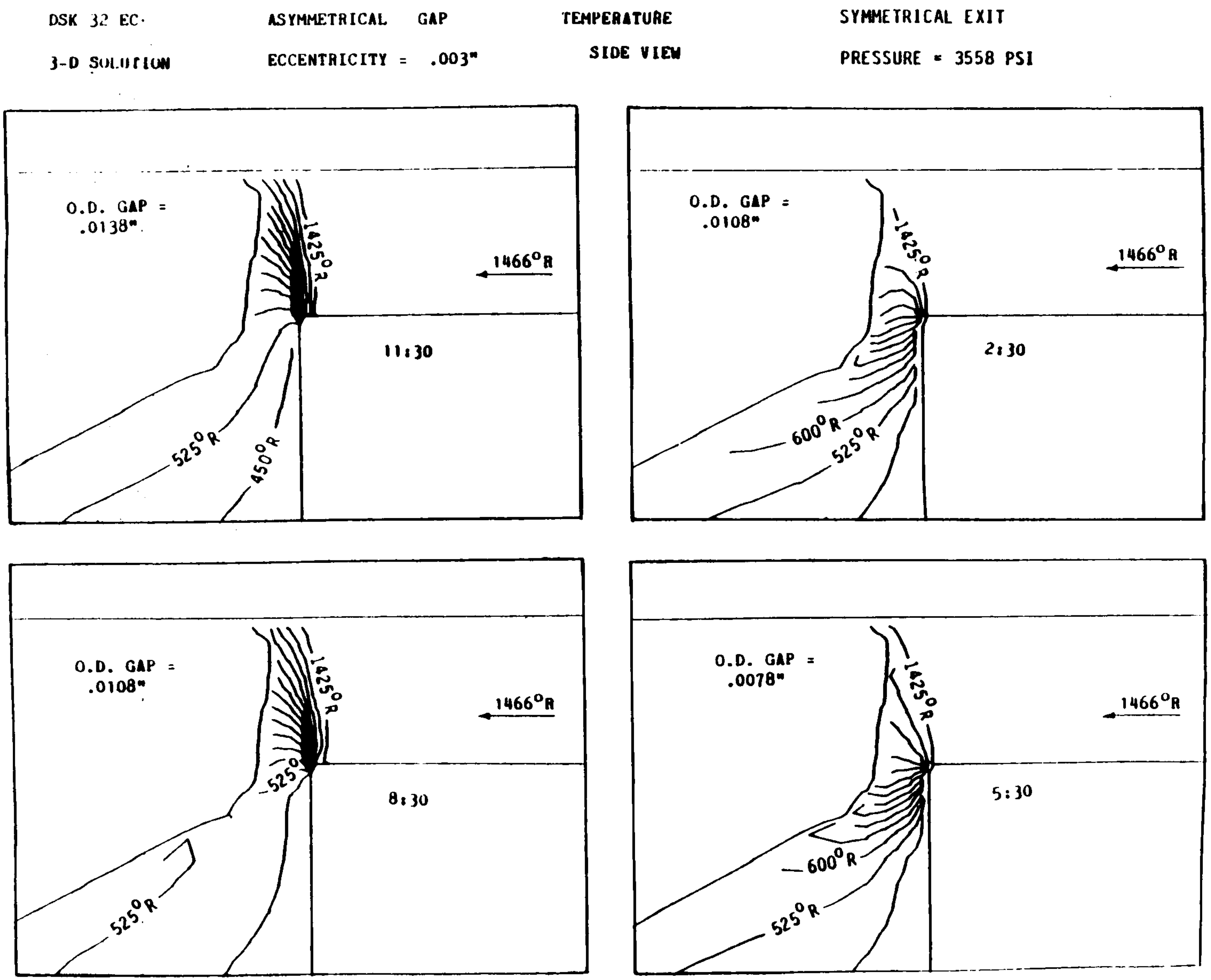


Figure 26. Three-dimensional eccentric (0.003 in.) rotor: temperature (close-up).

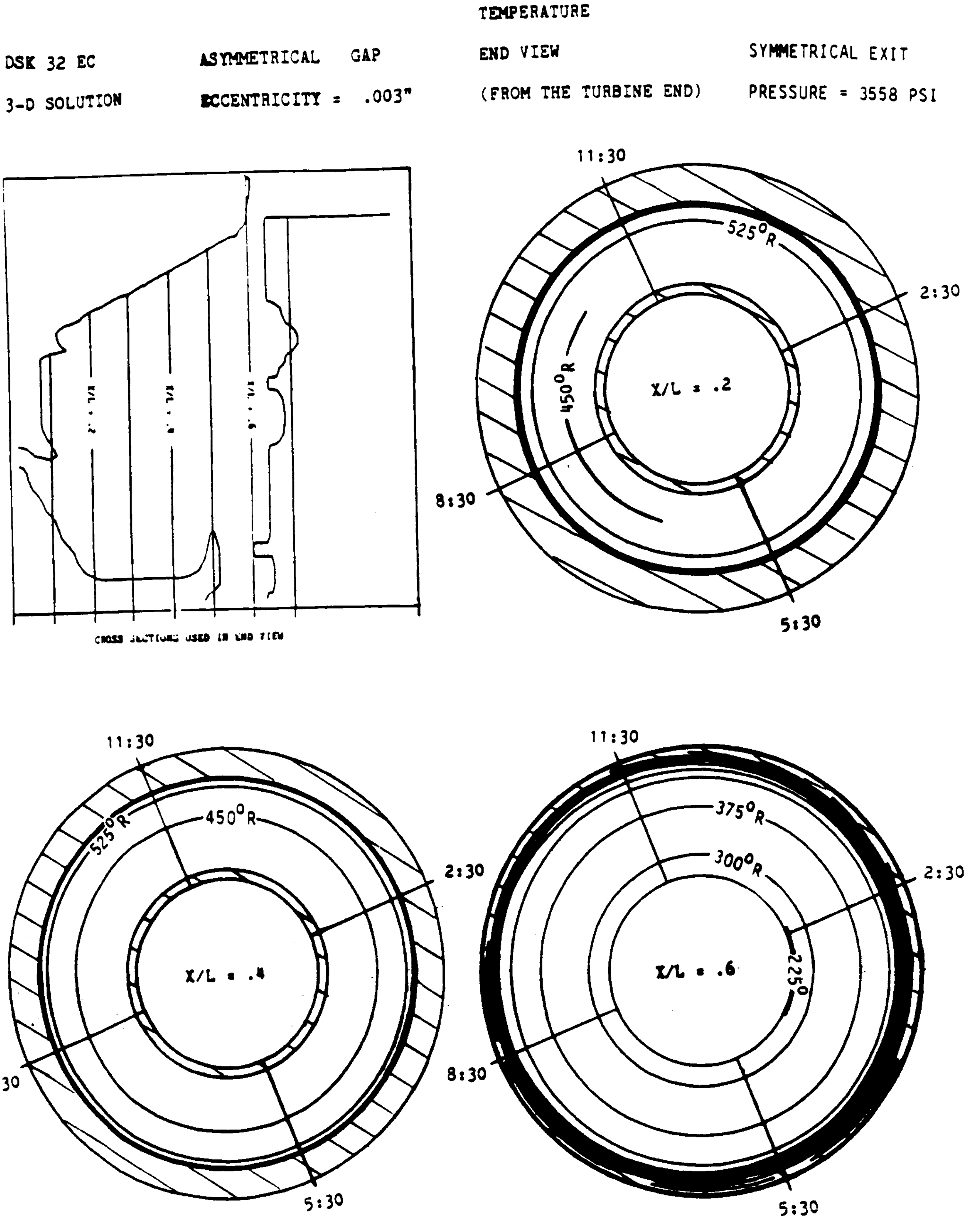


Figure 27. Three-dimensional eccentric (0.003 in.) rotor: temperature (end view).

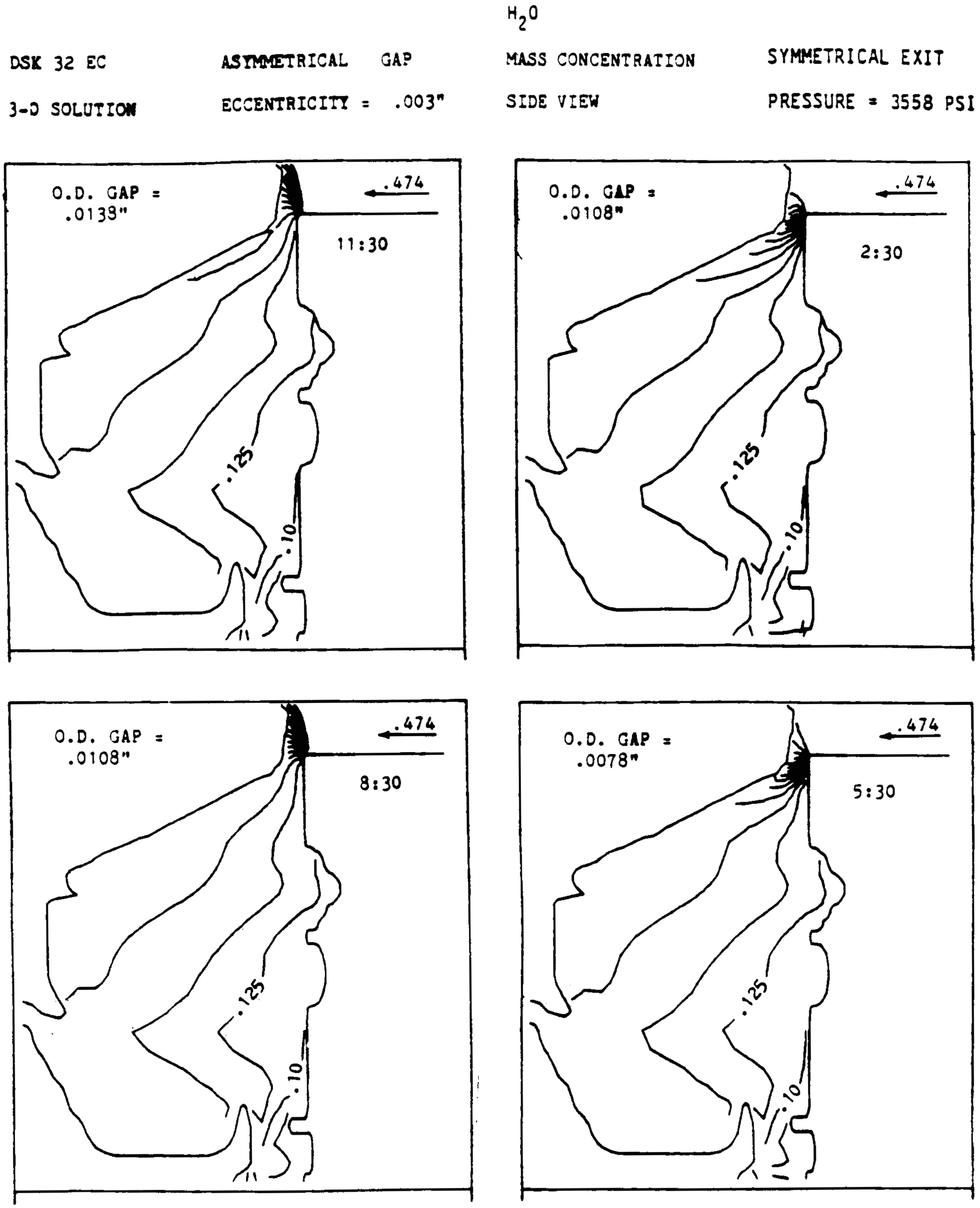


Figure 28. Three-dimensional eccentric (0.003 in.) rotor: Mass concentration.

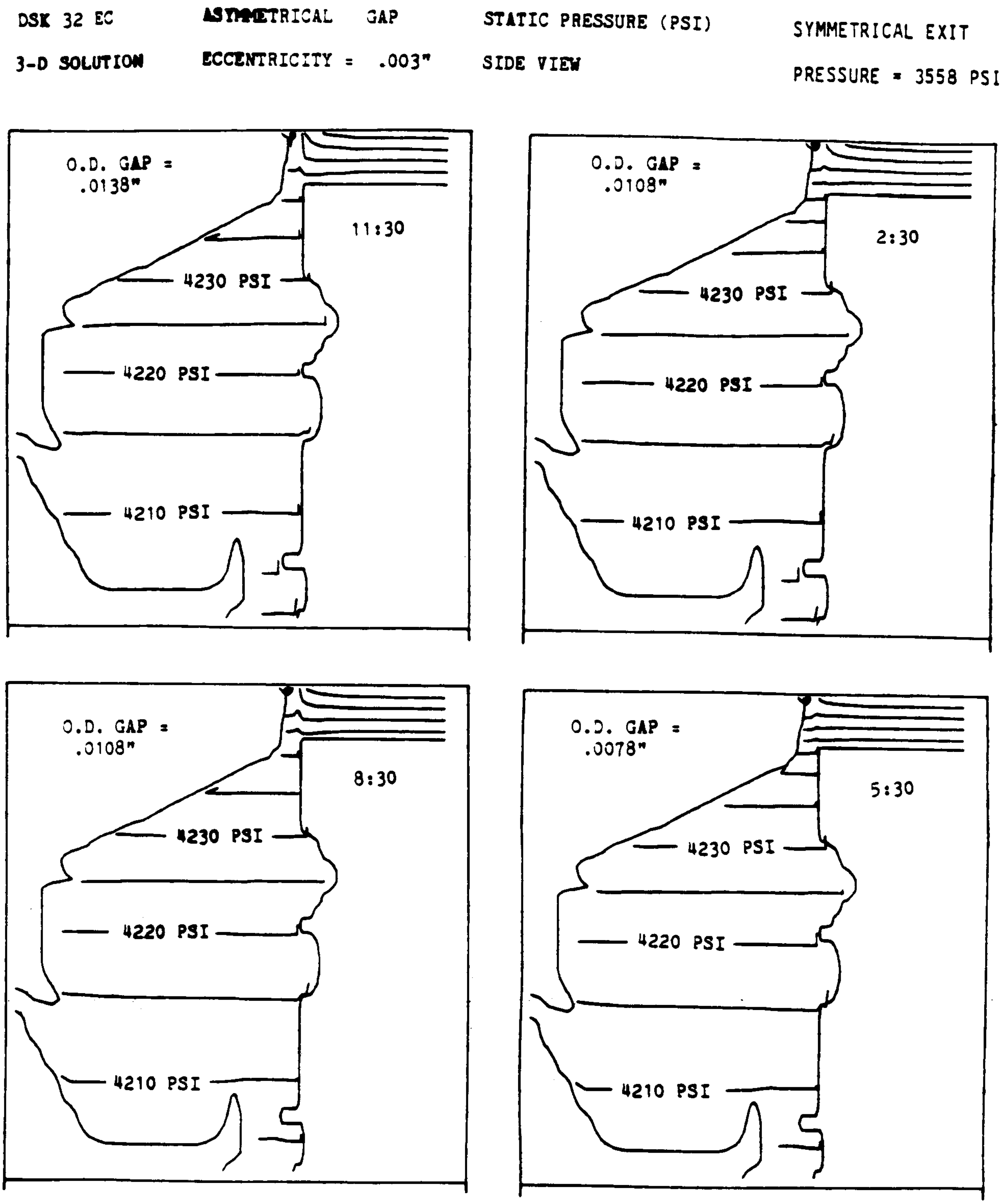


Figure 29. Three dimensional eccentric (0.003 in.) rotor: static pressure.

ORIGINAL PAGE IS
OF POOR QUALITY

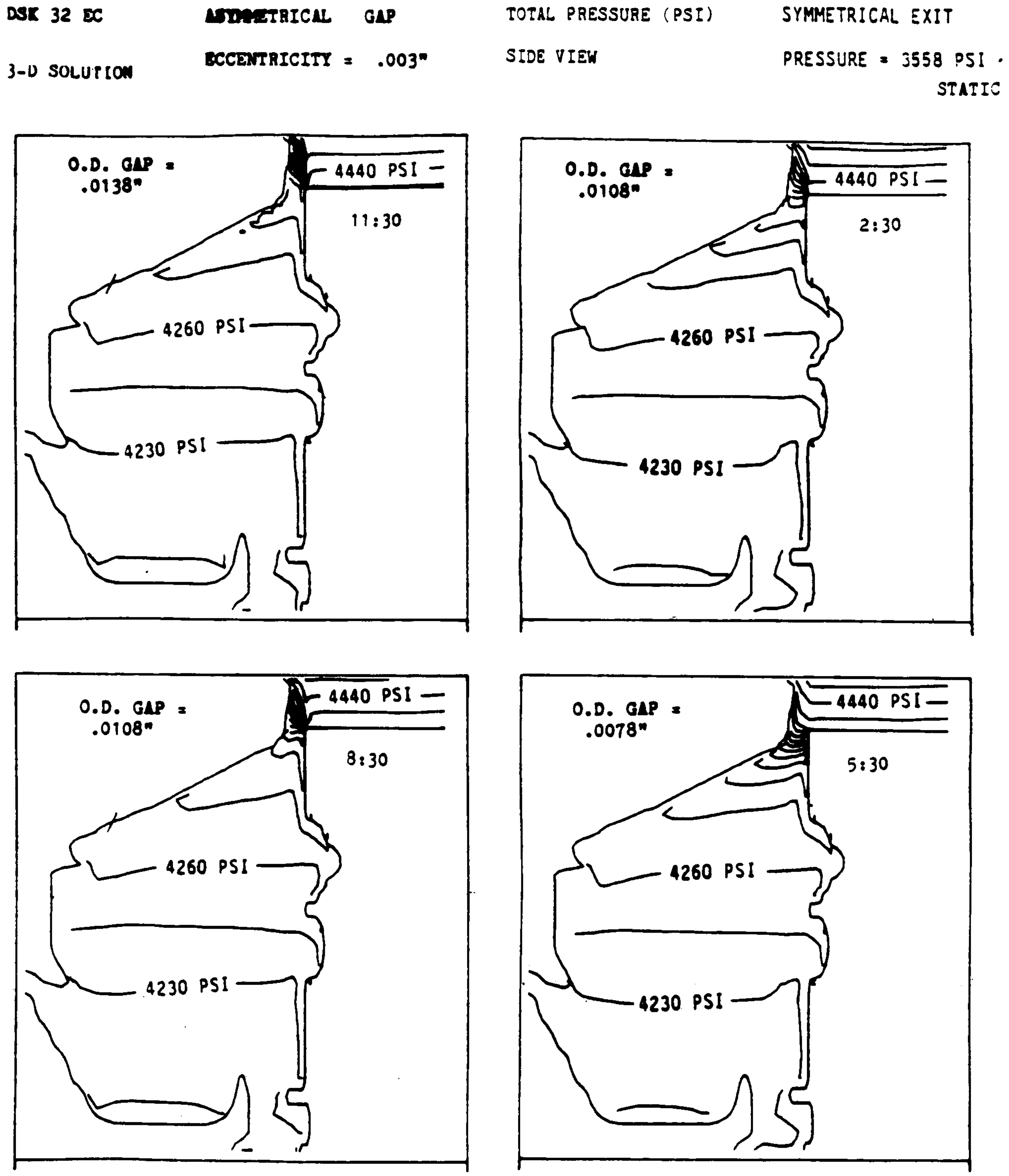


Figure 30. Three-dimensional eccentric (0.003 in.) rotor: total pressure.

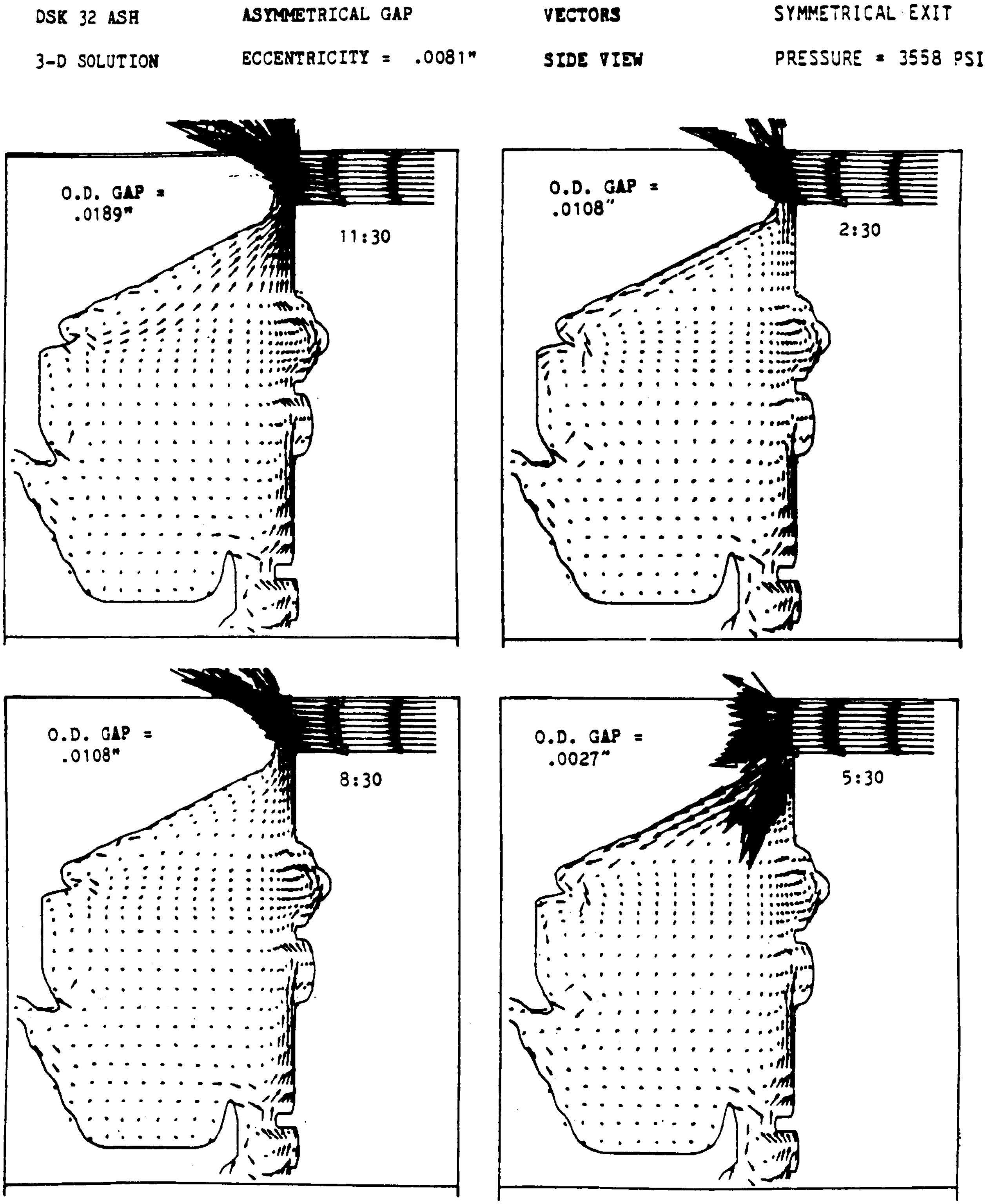


Figure 31. Three-dimensional eccentric (0.0081 in.) aft-platform seal: vectors.

ORIGINAL PAGE IS
OF POOR QUALITY

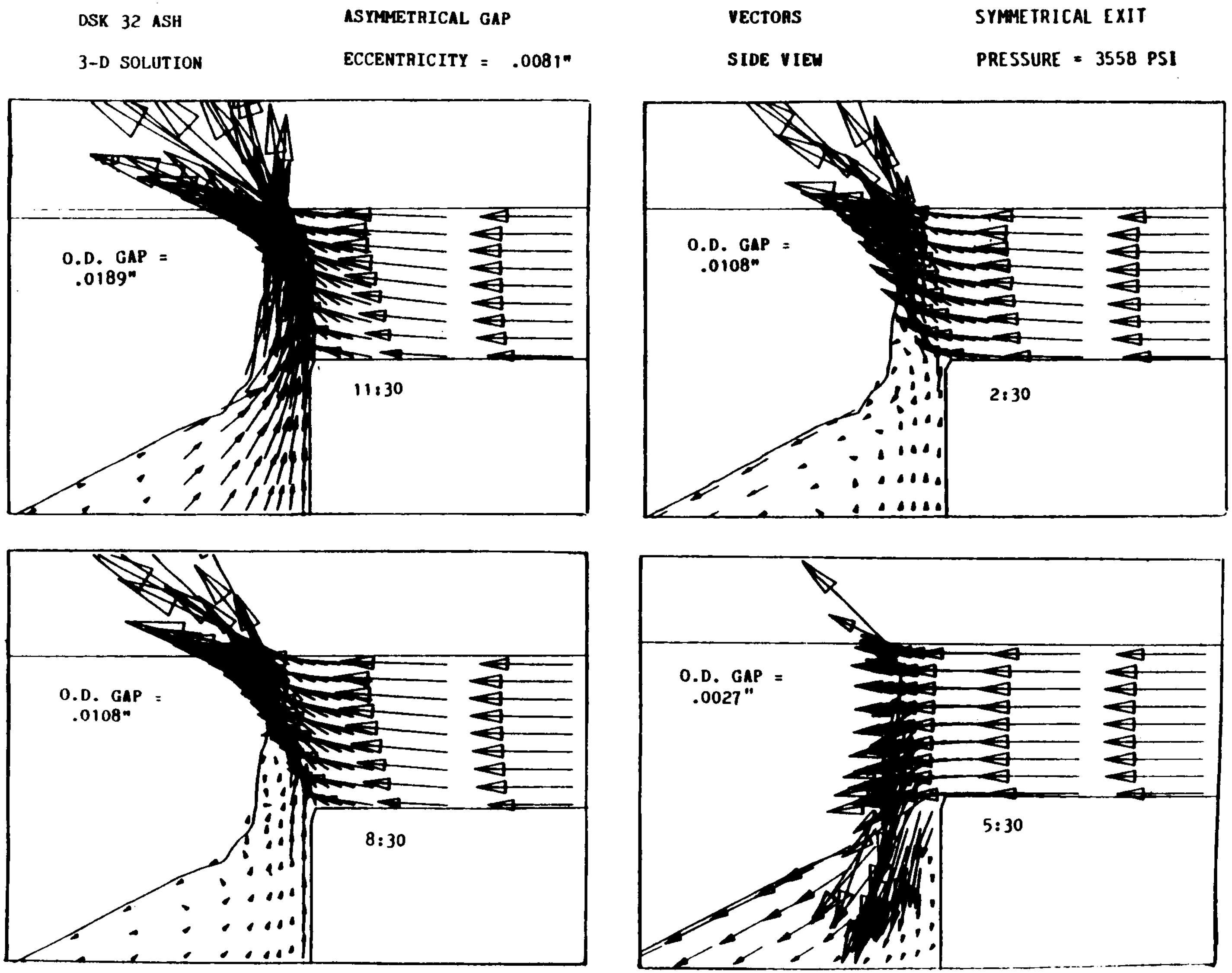


Figure 32. Three-dimensional eccentric (0.0081 in.) aft-platform seal: vectors (close-up).

VECTORS

DSK 32 ASH ASYMMETRICAL GAP END VIEW SYMMETRICAL EXIT

3-D SOLUTION ECCENTRICITY = .0081" (FROM THE TURBINE END) PRESSURE = 3558 PSI

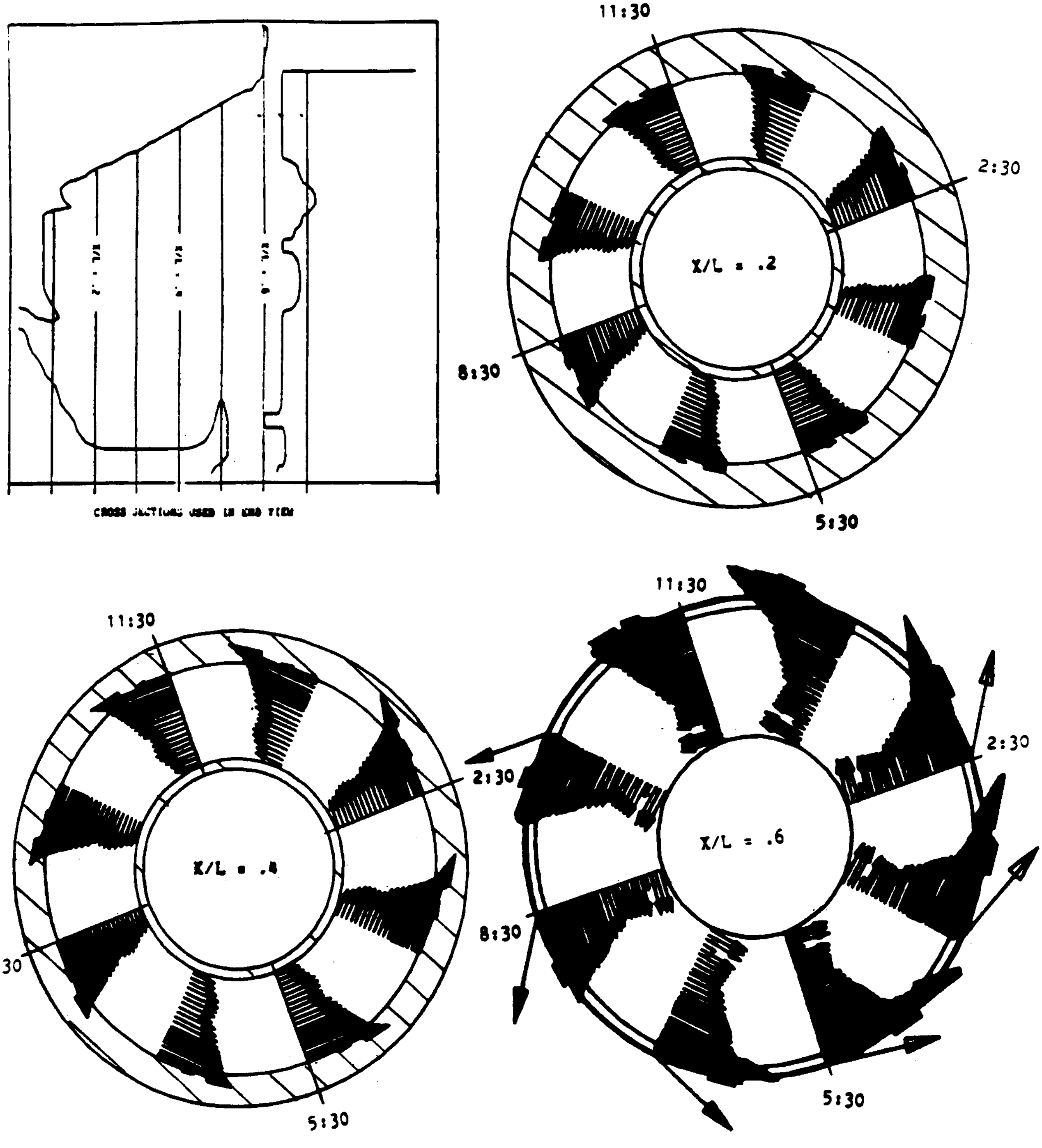


Figure 33. Three-dimensional eccentric (0.0081 in.) aft-platform seal: vectors (end view).

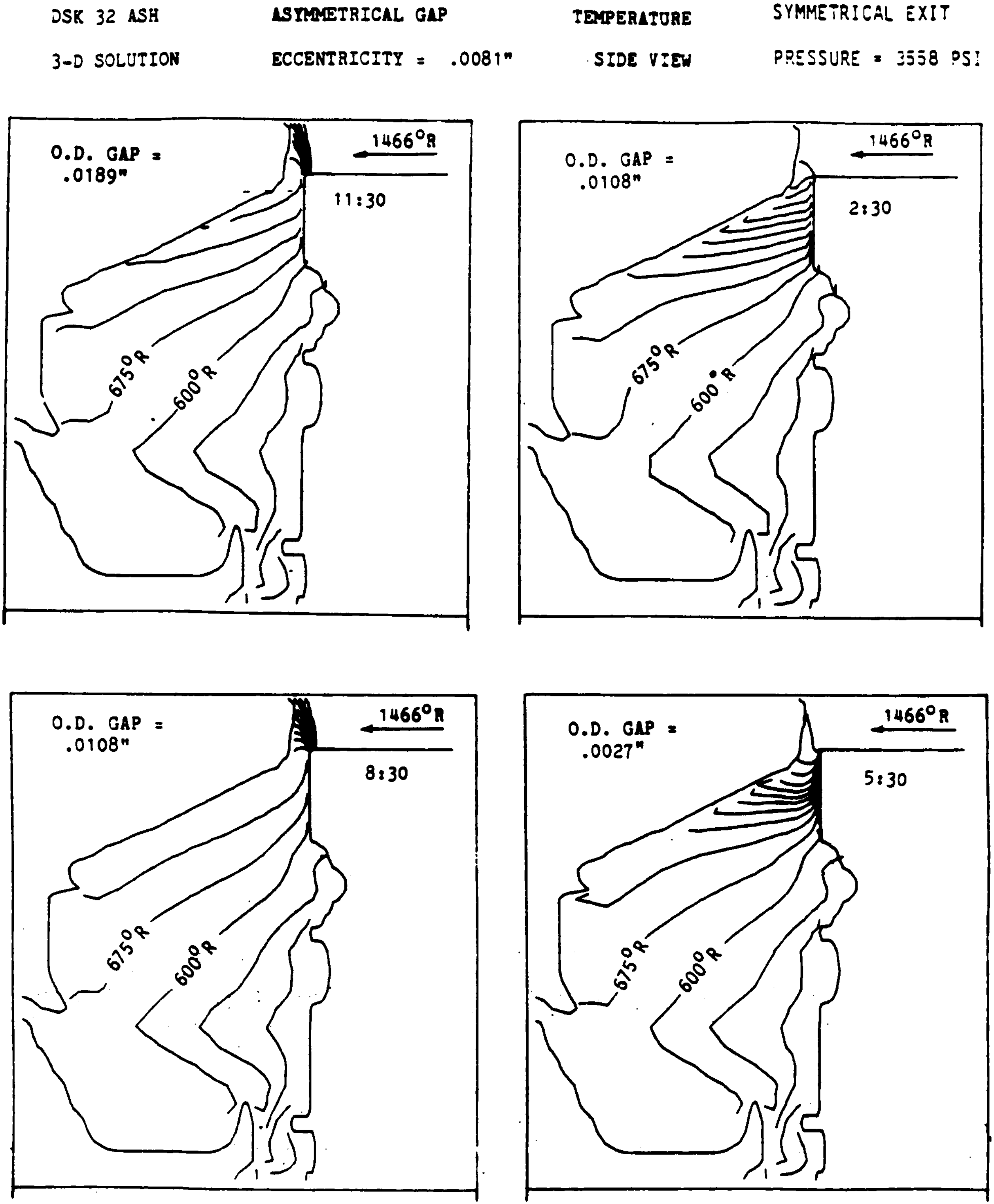


Figure 34. Three-dimensional eccentric (0.0081 in.) aft-platform seal: temperature.

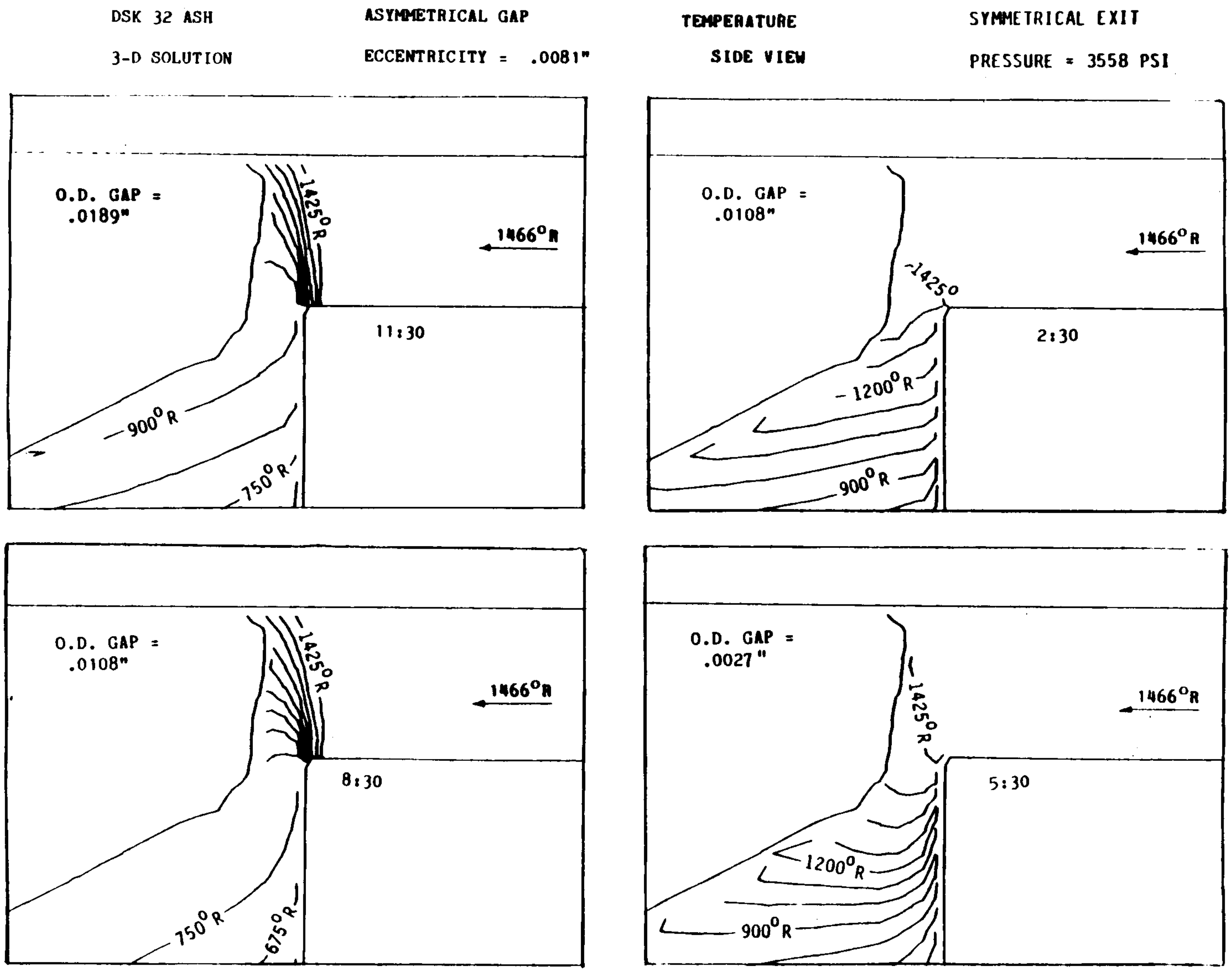


Figure 35. Three-dimensional eccentric (0.0081 in.) aft-platform seal: temperature (close-up).

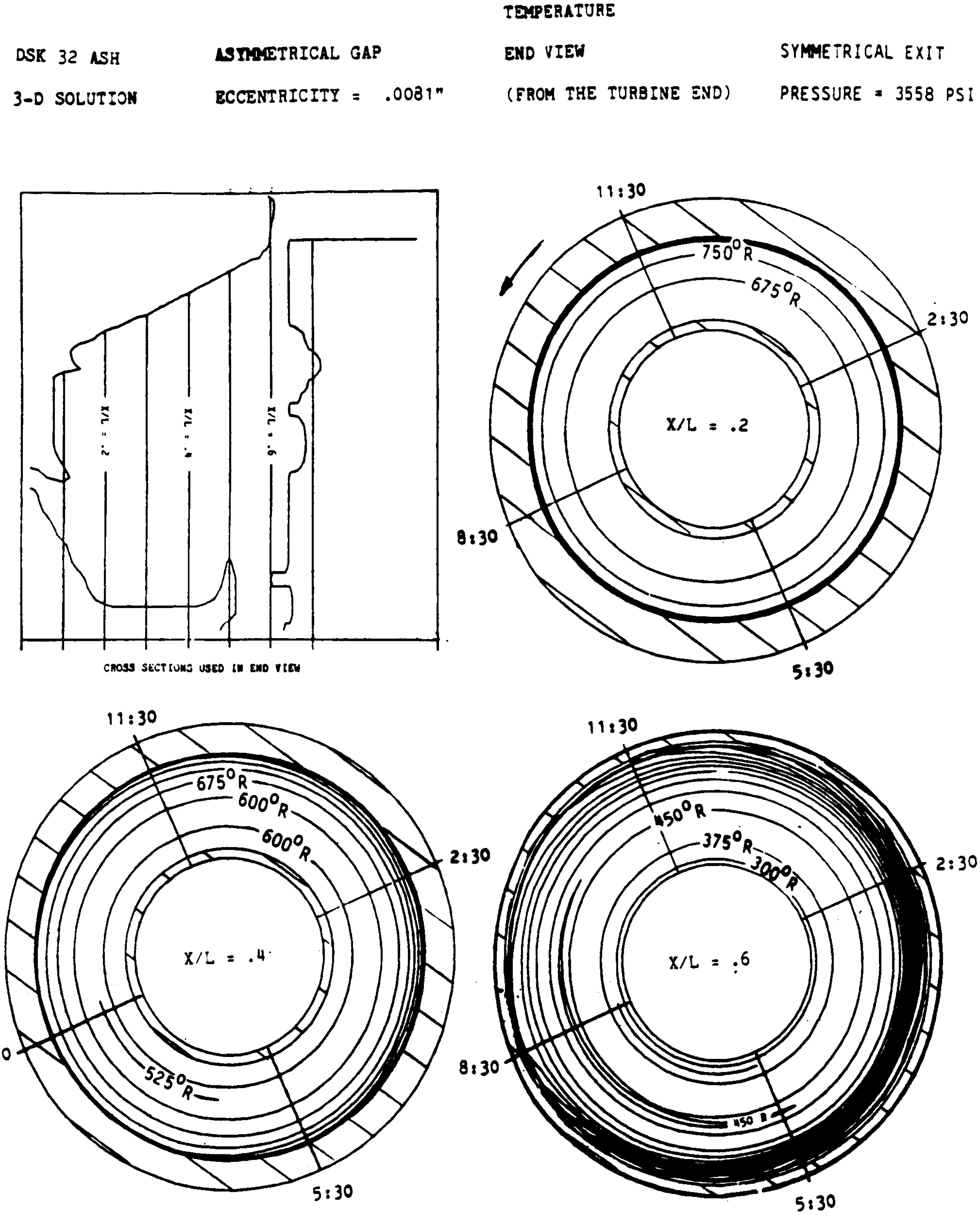


Figure 36. Three-dimensional eccentric (0.0081 in.) aft-platform seal: temperature (end view).

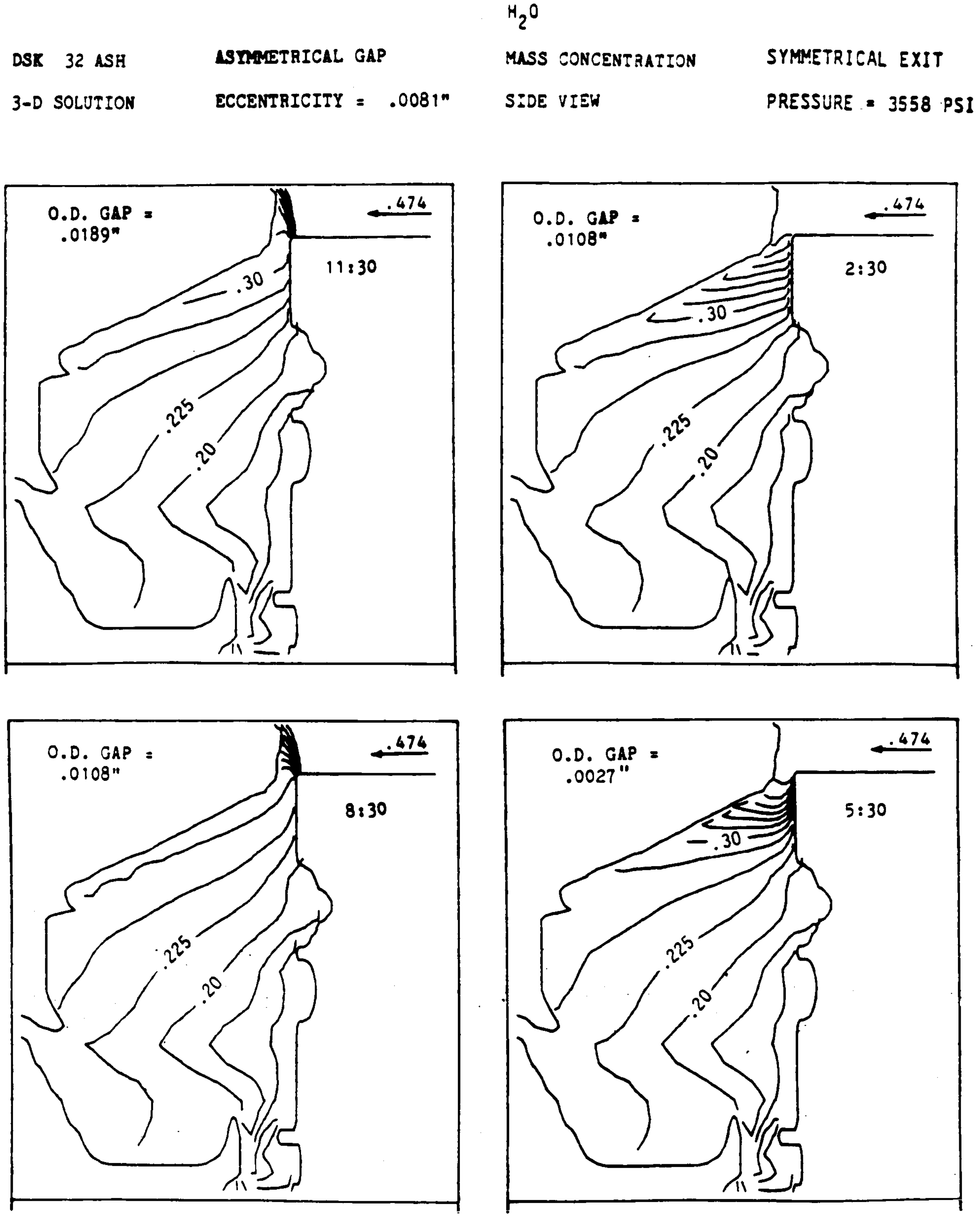


Figure 37. Three-dimensional eccentric (0.0081 in.) aft-platform seal: mass concentration.

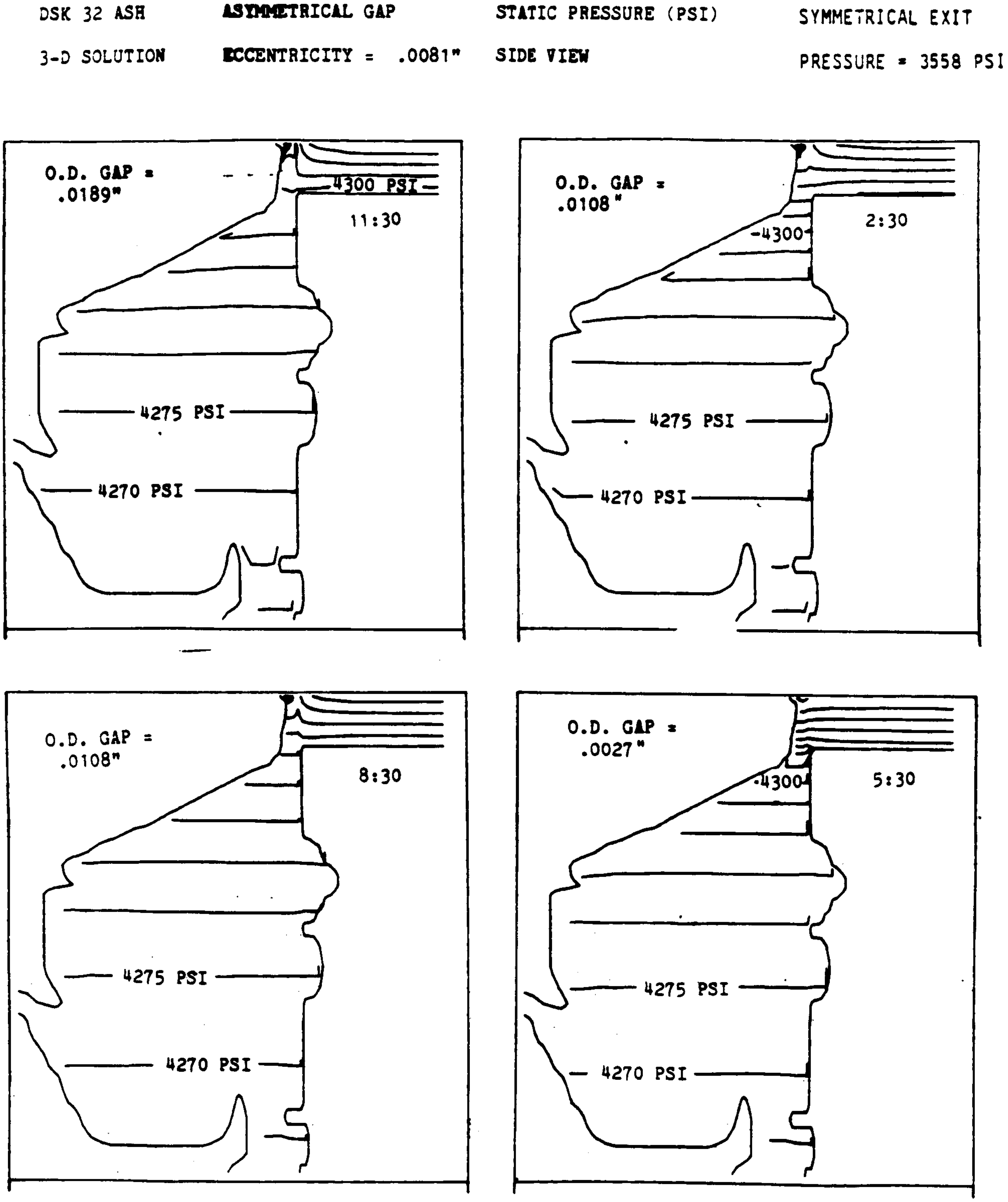


Figure 38. Three-dimensional eccentric (0.0081 in.) aft-platform seal: static pressure.

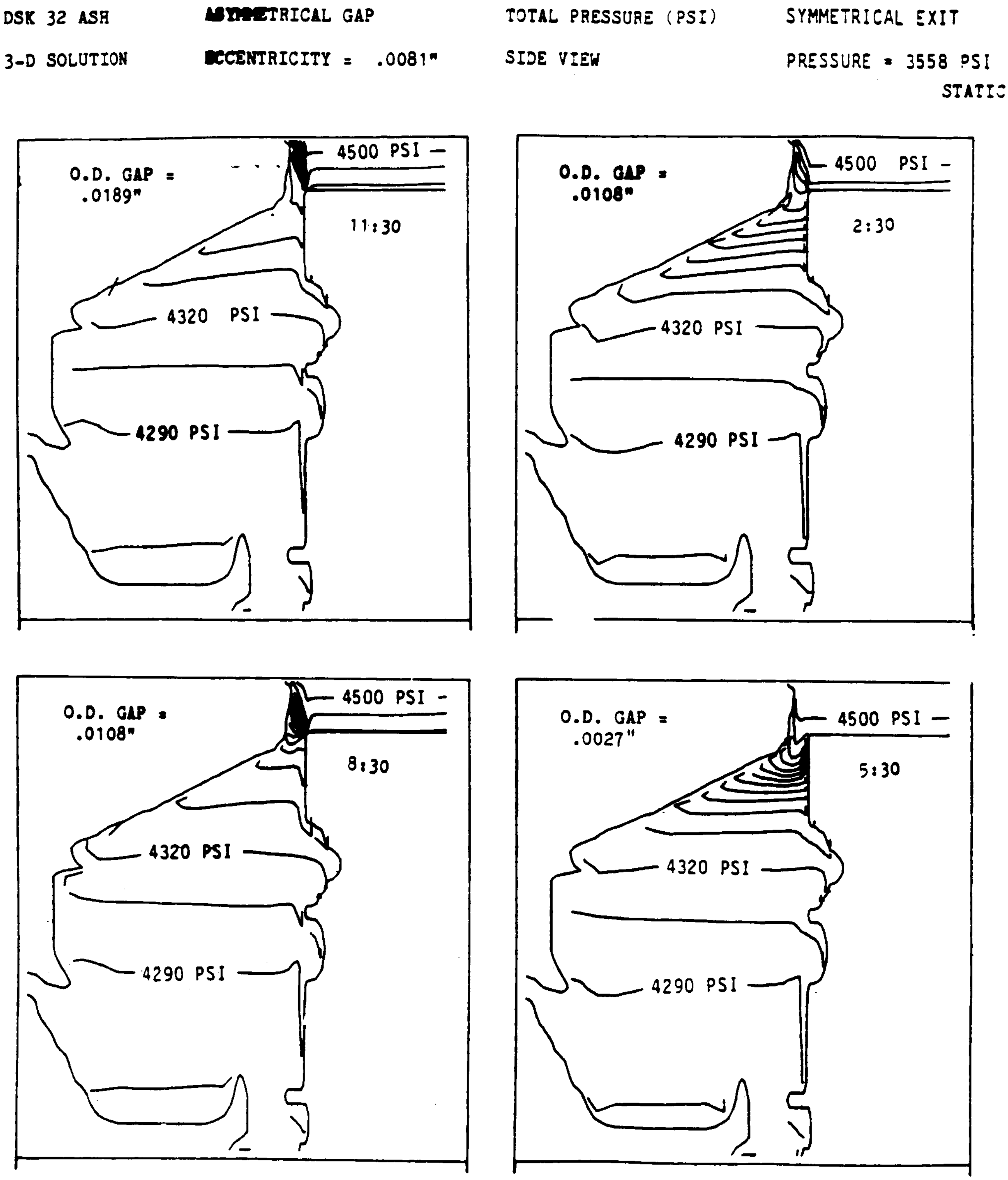


Figure 39. Three-dimensional eccentric (0.0081 in.) aft-platform seal: total pressure.

on this hot gas still works to confine it to the outer radius of the cavity. However, the pressure differences in the radial direction are, in this case, becoming large enough that they are forcing more and more hot gas down into the cavity. This is clearly evident in the velocity diagrams (Figs. 31 and 32) where, at the 5:30 clock position, there is a strong inward flow of hot gas down into the cavity. The temperature profiles also indicate a dramatic increase in hot gas in the cavity. The temperatures at the center of the cavity are now up to 675°R (215°) which is 300° warmer than for the basecase. As with the other three-dimensional runs, the most pronounced effects can still be seen at the outer radius of the cavity. Here at the outer radius of the disk near the blade shanks, there is a large circumferential variation in both temperature and pressure with the temperature cycling 600°R and the pressure varying by 20 psi.

One further observation on the results from this test run has to do with the static pressure. The observation is that for a 0.0081 in. eccentric aft-platform seal, the pressure in the cavity has gone up by 80 psi relative to the three-dimensional basecase. The implications of this pressure rise are not simple to determine. The difficulty lies in the fact that such a pressure in the cavity would reduce the flow rate through the labyrinth seal, which, for the three-dimensional model is (numerically) fixed based on the flowrates calculated earlier in the two-dimensional basecase. Since the total exit areas and average turbine discharge pressure for all the three-dimensional runs are the same as for the two-dimensional basecase, this is a reasonable assumption. In this final three-dimensional case, however, the assumption leads to a contradiction. For the eccentric aft-platform seal case, using the flowrates from the two-dimensional basecase, the pressure at the exit of the labyrinth seal is calculated as being higher than the pressure at the labyrinth inlet. In other words, if in the three-dimensional model this boundary had been specified as a fixed pressure instead of a fixed flowrate, the model would have predicted reverse flow through the labyrinth seal. That there could actually be reverse flow through the labyrinth seal is considered extremely unlikely. What would more likely occur is that an eccentricity in the aft-platform seal would raise the pressure in the aft-platform seal cavity, reducing both the hot gas flow past the blade shanks and the labyrinth seal flow.

VI. SUMMARY OF THE CURRENT TEST RUN RESULTS AND OBSERVATIONS

The axisymmetric computer model of the aft-platform seal cavity indicates that at 37,000 rpm the flow in the aft-platform seal cavity is dominated by the centrifugal force caused by the rotating turbine disk. The disk drives a recirculating flow in the central region of the cavity, creating a core of nearly uniform temperature. In general, the temperature field throughout the aft-platform seal cavity is dictated primarily by convection (as opposed to conduction) as indicated by the fact that little heat from the hot gas at the periphery of the cavity is conducted down into the isothermal core. As a result, the core stays relatively cold, even when the coolant flowrate is reduced by over 50 percent.

The most severe temperature gradient in the aft-platform seal cavity occurs at the outer diameter of the turbine disk, near the blade shanks. At this location, the hot gas entering from between the blade shanks mixes with the coolant flow that is being slung off the face of the disk. The temperature difference between the two streams is over 1000°R.

The three-dimensional computer model of the aft-platform seal cavity shows that, for normal clearances and operating conditions, the flow field in the cavity is relatively insensitive to the circumferential pressure variation known to exist in the turbine discharge. The flow field is shown to be sensitive, however, to eccentricities of the exit gap between the aft-platform seal and the blade shanks. But in both cases, it is the centrifugal force which still dominates the flow pattern, such that any perturbation of the flow field or temperature field which results from either pressure changes or geometrical changes are, for the most part, confined by centrifugal force to the outer diameter of the cavity.

In addition to the above, the study also reveals that, for fixed flow through the blade shanks, the labyrinth flowrate is extremely sensitive to the exit area at the outer diameter of the aft-platform seal. While this result is somewhat misleading, since it is based on the unrealistic boundary condition of a fixed flowrate through the blade shanks, it nevertheless merits further consideration especially with regard to transient phenomena. Finally, as a related observation, the flowrate through the labyrinth seal is also sensitive to the eccentricity of the aft-platform seal clearance, even for a constant exit area. This sensitivity is something which has yet to be included in the current one-dimensional models of the flow through the pump's turbine section.

VII. CONCLUSIONS

The results of the study summarized above provide the following insight into the specific problems which initiated the study, i.e., (1) the cracking of the HPFTP second stage blades, and (2) the suspected hot gas leakage into the coolant cavity behind the aft-platform seal bolts.

As far as the blade cracking is concerned, the model has shown that the second stage blade shanks are subjected to varying degrees of thermal stress, both steady state and once per revolution. The severity of this gradient has been shown to be sensitive to asymmetries in the external pressure and to variations in the geometrical clearances. At the time of this writing, however, it is believed that the primary cause of cracking is not due to thermal effects but is the result of a very high mean mechanical stress coupled with the moderate thermal stress. The proposed solution to alleviate the cracking is to recontour the shank in the high stress area, to shot-peen the surface to reduce the surface mean operating stress, and to coat the shanks to reduce the thermal stress [8].

As for the variations in coolant liner pressure and temperature thought to be indicative of a leak into the coolant liner, they remain an enigma. In order to gain a clearer understanding of this problem, the fluid temperatures calculated in the current study will be used as an input to the thermal stress analysis of the hardware. Prior to this study it was believed that the temperatures in the cavity were on the order of 900°R hotter than predicted here [8]. With a better estimate of the fluid temperature, the thermal stress analysis will be better able to predict the deformation of the aft-platform seal and the other components neighboring the aft-platform seal cavity. This will, in turn, generate improved estimates of the clearances and flowrates in the region.

The new flowrates estimated from the above will be fed back into the PHOENICS model for an improved analysis of the flow and temperature field in the cavity. Other changes which could be incorporated into the model would be to include the effect of heat transfer into the cavity and the viscous heating of the fluid itself, both of which will result in increases in the cavity temperature.

In addition to further analytical studies and improvements, there are plans to build a fuel pump that has pressure and temperature measurements built into the aft-platform seal, the labyrinth seal, the lift-off seal stack, and the coolant liner [8]. The test data from this instrumented pump, in conjunction with the computer model predictions should greatly increase the level of understanding of the operating environment of the high pressure fuel pump aft-platform seal cavity.

REFERENCES

1. Lockheed Missiles and Space Company, Inc.: Fluid Flow Analysis of the SSME High Pressure Fuel Turbopump Operating at Full Power Level. LMSC-MSFC TR D697954, May 1980.

2. Spalding, D. B.: General Computer Program for Fluid Flow Heat Transfer and Chemical Reaction Process. International Finite Element Congress, Baden-Baden, West Germany, November 1980.

3. Keeton, L. W., Lowry, S. A., and Wayden, L.: Listings of Phoenics 2-D and 3-D Satellites and Ground Stations as Adapted for the SSME Aft-Platform Seal Cavity Flow Model. CHAM 4045/21, June 1985.

4. Rosten, H. I., Spalding, D. B., and Tatchell, D. G.: PHOENICS - An Instruction Manual. CHAM Report TR/75, January 1982.

5. Morrison, G. L., et al.: Labyrinth Seals for Incompressible Flow. NAS8-34536 Final Report, April 1983, p. 63.

6. Conversation with Garry Lyles, EP26, Marshall Space Flight Center, NASA, January 1985.

7. Wineland, D. L. and Cramer, K.: HPFTP Instrumented Turbine Test Data. Rockwell International, September 12, 1982, p. F3.

8. Conversation with Rick Ryan, EP23, Marshall Space Flight Center, NASA, June 1985.

APPENDIX A: PHOENICS COMPUTER CODE SATELLITE AND GROUND ADAPTATIONS FOR THE SPACE SHUTTLE MAIN ENGINE HPFTP AFT-PLATFORM SEAL CAVITY 3-D MODEL

```
RUNKLEBIN197*TPF$(O).SL
      $BATCH
      C$DIRECTIVE**SATLIT
      C ***
      C     *FILE NAME: DSK32SAT.FTN
      C ***
      C     *ABSTRACT: SATELLITE FOR SSME AFT-PLATFORM SEAL 3-D MODEL (2 EXITS)
      C ***
      C     *DOCUMENTATION: PHOENICS INSTRUCTION MANUAL (SPRING 1983).
      C     *AUXILIARY SUBROUTINES (TAPES, ETC.) ARE IN SATELLITE LIBRARY
      C      SERVICEU, WHICH MUST BE INCLUDED IN LINK EDIT TO RUN.
      CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 STARTS:
      C--------------------------------------------------------------------
      CHAPTER 1   COMMON BLOCKS AND USER'S DATA.
      C--------------------------------------------------------------------
      $INCLUDE 9,CMNGUSSI.FTN/G
      $INCLUDE 9,GUSSEQUI.FTN/G
      $INCLUDE 9,CMNGRFIC.FTN/G
            COMMON/CPI/IPWRIT,IDUM(243)
            DIMENSION GDTAPE(3),DFAULT(4)
            DIMENSION ARRAY1(309),ARRAY2(194),ARRAY3(421)
            LOGICAL ARRAY1,LSPDA,WRT,RD,NAMLST
            INTEGER ARRAY2,XPLANE,YPLANE,ZPLANE
            INTEGER P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,EP,H1,H2,H3,C1,C2,
           &C3,C4
            REAL NORTH,LOW
            EQUIVALENCE (ARRAY1(1),CARTES),(ARRAY2(1),NX)
            EQUIVALENCE (ARRAY3(1),SPARE1(1)),(M1,R1),(M2,R2)
            EQUIVALENCE (LSTRUN,INTGR(12)),(NAMLST,LOGIC(88))
      CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 ENDS.
      CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 STARTS:
      C     GRAFFIC ARRAYS DIMENSIONED AS NEEDED...
      C ***
            COMMON/GRAF1/PHI1(134500) /GRAF2/PHI2(239500)
      C     COMMON/GRAF1/PHI1(1) /GRAF2/PHI2(1)
      C ***
      C     POROSITY & SPECIAL DATA ARRAYS DIMENSIONED AS NEEDED...
      C ***
            DIMENSION PE(8,40,28),PN(8,40,28),PH(8,40,28),PC(8,40,28)
            DIMENSION LSPDA(1),ISPDA(1),RSPDA(37)
            DIMENSION PEXIT(8),GEXIT(8)
      C ***
      C     USER PLACES HIS VARIABLES, ARRAYS, EQUIVALENCES ETC. HERE.
      C ***
            DATA NLSP,NISP,NRSP/1,1,37/
            EQUIVALENCE (RSPDA(17),PEXIT(1)),(RSPDA(30),GEXIT(1))
      C ***
      C     USER PLACES HIS DATA STATEMENTS HERE.
      C ***
            DATA PI,G,TINY/3.1416,32.174,1.E-10/
      C ***
      CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 ENDS.
      CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 STARTS:
      C-------------------------------------------------------------------
      CHAPTER 2   SET CONSTANTS, AND ARRANGE FILE MANIPULATIONS.
      C-------------------------------------------------------------------
      C     PLEASE DO NOT ALTER, OR RE-SET, ANY OF THE REMAINING
      C     STATEMENTS OF THIS CHAPTER.
            DATA CELL,EAST,WEST,NORTH,SOUTH,HIGH,LOW,VOLUME/
           & 0.,1.,2.,3.,4.,5.,6.,7. /
```

```
      DATA P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,KE,EP,H1,H2,H3,C1,C2,
     &C3,C4/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20/
      DATA FIXFLU,FIXVAL,ONLYMS,WALL/1.E-10,1.E10,0.0,-10.0/
      DATA IPLANE,XPLANE,YPLANE,ZPLANE/0,1,2,3/
      DATA WRT,RD,DFAULT/.TRUE.,.FALSE.,4HDEFA,4HULT.,4HDTA/,1HG/
      DATA GDTAPE/4HGUSI,4HE1.D,2HTA/
      DATA NLDATA,NIDATA,NRDATA/309,194,421/
      DATA NLCREG,NTCVRG/60,350/
      CALL TAPES(10,GDTAPE,3,1,4*NRDATA)
C---------------------------READ DEFAULT FILE IF BLOCKDATA ABSENT
          IF(INTGR1(29).NE.10) GO TO 2
      CALL WRIT40(40HDATA ESTABLISHED IN BLOCK DATA.          )
      GO TO 3
    2 CALL TAPES(1,DFAULT,4,2,4*NRDATA)
      CALL DATAIO(RD,1)
      CALL WRIT40(40HDATA TAKEN FROM DEFAULT.DTA ON GROUP A/C)
    3 CALL WRIT40(40HFILE MODSTL.FTN IS THE SATLIT USED.      )
C------------------------------------------------------------
CHAPTER 3   DEFINE DATA FOR NRUN RUNS.
C-----------------------------------------------------------
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 STARTS:
C
      LOGIC(89)=.TRUE.
      DO 410 II=1,1
  410 RUN(II)=.TRUE.
C
CC   **********************************************************
CC
CC                    ****     INPUTS       ****
CC
CC   **********************************************************
CC
CC    ****        GEOMETRY
CC
CC ** SET GINC1 TO THE (LARGE) GAP HT (IN) AT THE COLD INLET
      GINC1 = .10693
CC ** SET GEXIT1 = THE AVERAGE GAP CLEARANCE AT THE EXIT (INCHES)
C     NB. SHOULD NOT BE LARGER THAN CELL WIDTH (=0.03333)
      GEXIT1 = .0108
CC ** SET ECCENT = THE RADIAL ECCENTRICITY (INCHES) OF THE ROTOR  IN
C     THE CELL 1 (11:30) DIRECTION. THIS ECCENTRICITY WOULD NORMALLY
C     BE LIMITED BY THE CLEARANCE OF THE LABYRINTH SEAL (GINC1S) AT
C     ITS NARROWEST (IE WHERE "TEETH" MEET SHAFT).
C     THE ECCENTRICITY EFFECTS BOTH THE EXIT GAP AT THE HOT GAS EXIT
C     AND THE DISTRIBUTION OF FLOW AT THE LABYRINTH SEAL INLET.
C
      GINC1S=0.003
C
      ECCENT = GINC1S
C
CC ** SET GEXIT ARRAY TO ACTUAL REQUIRED GAP CLEARANCE AT EXIT (FEET)
C     (SEE DESCRIPTION OF PEXIT ARRAY BELOW FOR CLOCKING CONVENTION)
C
C NB. GEXIT ARRAY CALCULATIONS BELOW ARE GRID DEPENDENT!!!
C
C     SET GEXIT(1) TO AVERAGE GAP CLEARANCE AT 11:30
      GEXIT(1)=(GEXIT1+COS(0.)*ECCENT)/12.
C     SET GEXIT(2) TO AVERAGE GAP CLEARANCE AT 10:00
      GEXIT(2)=(GEXIT1+COS(2.*PI/8.)*ECCENT)/12.
```

```
C     SET GEXIT(3) TO AVERAGE GAP CLEARANCE AT  8:30
      GEXIT(3)=(GEXIT1+COS(2.*PI/4.)*ECCENT)/12.
C     SET GEXIT(4) TO AVERAGE GAP CLEARANCE AT  7:00
      GEXIT(4)=(GEXIT1+COS(2.*PI*3./8.)*ECCENT)/12.
C     SET GEXIT(5) TO AVERAGE GAP CLEARANCE AT  5:30
      GEXIT(5)=(GEXIT1+COS(PI)*ECCENT)/12.
C     SET GEXIT(6) TO AVERAGE GAP CLEARANCE AT  4:00
      GEXIT(6)=(GEXIT1+COS(2.*PI*5./8.)*ECCENT)/12.
C     SET GEXIT(7) TO AVERAGE GAP CLEARANCE AT  2:30
      GEXIT(7)=(GEXIT1+COS(2.*PI*3./4.)*ECCENT)/12.
C     SET GEXIT(8) TO AVERAGE GAP CLEARANCE AT  1:00
      GEXIT(8)=(GEXIT1+COS(2.*PI*7./8.)*ECCENT)/12.
C
CC ** SET AINH1 TO THE AREA (SQ IN) AT THE HOT INLET
      AINH1 = 3.877
CC ** SET ARGRD1 = THE HIGH FACE GRID AREA (SQ IN) AT THE HOT INLET
      ARGRD1 = 8.143
CC ** 1/SLICES = THE FRACTION OF 360 DEGREES BEING MODELLED
      SLICES=1.0
CC
CC   ****      PROPERTIES
CC
CC ** SET H1INC1 TO THE ENTHALPY (BTU/LBM) AT THE COLD INLET
      H1INC1=278.3
CC ** SET H1INH1 TO THE ENTHALPY (BTU/LBM) AT THE HOT INLET
      H1INH1=3380.
CC ** SET HEXIT1 TO THE ENTHALPY (BTU/LBM) OF THE TURBINE EXIT
      HEXIT1=3895.4
CC ** SET ROINC1 TO THE DENSITY (LBM  /CU FT) AT THE COLD INLET
      ROINC1 =3.574
CC ** SET ROINH1 TO THE DENSITY ( LBM /CU FT) AT THE HOT INLET
      ROINH1 = .931
C *** NOTE: THE DIRECTION OF ROTATION OF THE TURBINE IS
C     COUNTERCLOCKWISE ACCORDING TO THE  CLOCKING CONVENTION
C     USED IN ROCKETDYNE HPFTP INSTRUMENTED TURBINE TEST
C     DATA REPORT P9-17-82
C     THE PRESSURE OF 3582 IS AN AVERAGE OF
C     THE 3D DATA (3505,3500,3615,3630,3645,3720,3565,3475) WHICH
C     COMES FROM TEST 902-279 FPL DATA - IT CORRESPONDS TO AN
C     AVERAGE COOLANT LINER PRESSURE OF 3800 PSI
C !!! NB. VALUES BELOW INCREMENTED IN ACCORDANCE WITH NEW DATA
C
CC ** SET PEXIT(1)  TO THE PRESSURE (PSF) AT 11:30
       PEXIT(1)= 144.0 * 3481.4
CC ** SET PEXIT(2)  TO THE PRESSURE (PSF) AT 10:00
       PEXIT(2)= 144.0 * 3476.4
CC ** SET PEXIT(3)  TO THE PRESSURE (PSF) AT 8:30
       PEXIT(3)= 144.0 * 3591.4
CC ** SET PEXIT(4)  TO THE PRESSURE (PSF) AT 7:00
      PEXIT(4)= 144.0 * 3606.4
CC ** SET PEXIT(5)  TO THE PRESSURE (PSF) AT 5:30
       PEXIT(5)= 144.0 * 3621.4
CC ** SET PEXIT(6)  TO THE PRESSURE (PSF) AT 4:00
       PEXIT(6)= 144.0 * 3696.4
CC ** SET PEXIT(7)  TO THE PRESSURE (PSF) AT 2:30
      PEXIT(7)= 144.0 * 3541.4
CC ** SET PEXIT(8)  TO THE PRESSURE (PSF) AT 1:00
       PEXIT(8)= 144.0 * 3451.4
C
CC ** SET PEXITA TO THE AVERAGE TURBINE DISCHARGE PRESSURE (PSF)
```

```
      PEXITA= 144.0 * 3558.4
C
C************************************************
C SET UP EXIT PRESSURES AS UNIFORM
      PEXIT(1)=PEXITA
      PEXIT(2)=PEXITA
      PEXIT(3)=PEXITA
      PEXIT(4)=PEXITA
      PEXIT(5)=PEXITA
      PEXIT(6)=PEXITA
      PEXIT(7)=PEXITA
      PEXIT(8)=PEXITA
C*************************************************
CC
CC   ****      BOUNDARY   CONDITIONS
CC
CC ** INPUT RPM
      RPM= 37000.
CC ** SET FEEDC1 TO THE TOTAL MASS FLOWRATE (LBM/S)
CC                                   AT THE COLD INLET
      FEEDC1 = .2582
CC ** SET FEEDH1 TO THE TOTAL MASS FLOWRATE (LBM/S)
CC                                   AT THE HOT  INLET
      FEEDH1 = 3.649
CC ** SET H2OINH TO THE H2O MASS FRACTION AT THE HOT INLET
      H2OINH = .474
CC ** SET H2OXIT TO THE H2O MASS FRACTION AT THE TURBINE EXIT
      H2OXIT = .5
CC ** SET GLOSK1 TO LOSS COEFFICIENT FOR  LOSSES AT EXIT
CC                                                  NEAR BLADE ROOTS
      GLOSK1=1.5 + TINY
C
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 3 STARTS:
        DO 10 IRUN=1,30
          IF(.NOT.RUN(IRUN)) GO TO 10
      NRUN=NRUN+1
      LSTRUN=IRUN
   10 CONTINUE
        DO 999 IRUN=1,LSTRUN
          IF(.NOT.RUN(IRUN)) GO TO 999
      INTGR(11) = IRUN
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 3 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 3 STARTS:
C---  ALL INTEGER VARIABLES ARE DEFAULTED TO 0, AND REAL VARIABLES
C     TO 0.0, UNLESS OTHERWISE INDICATED.
C     E.G. BY VARIABLE<10>, OR <10.0> AS APPROPRIATE.
C     THE DEFAULT SETTINGS OF ALL LOGICAL VARIABLES ARE ALWAYS
C     INDICATED, E.G. VARIABLE<.T.>, OR  VARIABLE<.F.>.
C
C---  RUN1
C------------------   ---------------   ------------------
C---  GROUP 1. FLOW TYPE :
C     PARAB<.F.>,CARTES<.T.>,ONEPHS<.T.>
      CARTES = .FALSE.
C------------------   ---------------   ------------------
C---  GROUP 2. TRANSIENCE :
C     STEADY<.T.>,ATIME,LSTEP<1>,FSTEP<1>
C     TLAST<1.E10>,TFRAC(1-30)<30*1.>
C     SERVICE SUBROUTINE FOR 'NT' POWER-LAW TIME STEPS:
```

```
C     CALL GRDPWR(0,NT,TLAST,POWER)
C------------------   --------------    ------------------
C---  GROUP 3. X-DIRECTION :
C     NX<1>,XULAST<1.0>,XFRAC(1-30)
C     SERVICE SUBROUTINE FOR POWER-LAW GRID:
C     CALL GRDPWR(1,NX,XULAST,POWER)
      NX = 8
      XFRAC(1)= -8.0
      XFRAC(2)= 1./8.0
      XULAST=2.*PI/SLICES
C------------------   --------------    ------------------
C---  GROUP 4. Y-DIRECTION :
C     NY<1>,YVLAST<1.0>,YFRAC(1-30),RINNER,SNALFA
C     SERVICE SUBROUTINE FOR POWER-LAW GRID:
C     CALL GRDPWR(2,NY,YVLAST,POWER)
      NY= 40
CC *** 2.6 = DISTANCE FROM THE INNER CAVITY RADIUS
C              TO THE OUTER RADIUS (INCHES)
      YVLAST= 2.6/12.
      YFRAC(1)= -14.0
      YFRAC(2)= 1./26.0
      YFRAC(3)= 6.0
      YFRAC(4)= 1./(2.0*26.0)
      YFRAC(5)= 2.0
      YFRAC(6)= 1./26.0
      YFRAC(7)= 6.0
      YFRAC(8)= 1.0/(2.0*26.0)
      YFRAC(9)= 12.0
      YFRAC(10)= 1.0/(3.0*26.0)
      RINNER= 1.87/12.
C------------------   --------------    ------------------
C---  GROUP 5. Z-DIRECTION :
C     NZ<1>,ZWLAST<1.0>,ZFRAC(1-30)
C     SERVICE SUBROUTINE FOR POWER-LAW GRID:
C     CALL GRDPWR(3,NZ,ZWLAST,POWER)
      NZ= 28
CC *** 2.5= DISTANCE FROM THE LEFT WALL OF THE CAVITY
C              TO THE RIGHT SIDE OF THE  GRID (INCHES).
      ZWLAST= 2.5/12.
      ZFRAC(1)= -14.0
      ZFRAC(2)= 2.0*1./50.0
      ZFRAC(3)= 2.0
      ZFRAC(4)= 1.0/50.0
      ZFRAC(5)= 8.0
      ZFRAC(6)= 1.0/(2.0*50.0)
      ZFRAC(7)= 1.0
      ZFRAC(8)= 1.0/50.0
      ZFRAC(9)= 3.0
      ZFRAC(10)= 5.0*1./50.0
C------------------   --------------    ------------------
C---  GROUP 6. MOVING GRID :
C     MGRID,IZW1,IZW2,AZW2,BZW2,CZW2,PINT,ZW2M1T
C------------------   --------------    ------------------
C---  GROUP 7. BLOCKAGE: BLOCK<.F.>,IPLANE,IPWRIT
C    *SET CONSTANT POROSITIES OVER SUB-DOMAINS USING:
C     CALL CONPOR(IR,TYPE,VALUE,IXF,IXL,IYF,IYL,IZF,IZL), WHERE:
C     IR=RUN SECTION NUMBER, E.G. 1 FOR RUN1 SECTION                'TYPE'= EAST,
C     WEST, NORTH, SOUTH, HIGH, LOW & CELL. 'VALUE'=WANTED POROSITY
C     OVER REGION IXF,...IZL.
C    *DIMENSION ARRAYS PE(NX,NY,NZ), PN(NX,NY,NZ), PH(NX,NY,NZ), &
```

```
C     PC(NX,NY,NZ) ABOVE.
C     FOR FULLY-BLOCKED CELLS (IE. 'VALUE'= 0.0) USER NEED SET ONLY
C     THE 'CELL' POROSITY (TO ZERO), AS CELL-FACE AREAS ARE THEN
C     AUTOMATICALLY ZEROED.
C    *FOR SATELLITE PRINTOUT OF ALL POROSITIES IN DOMAIN, 'IPLANE'=
C     XPLANE YPLANE OR ZPLANE, FOR DESIRED CROSS-SECTION DIRECTION.
C    *FOR EACH 'TYPE' A MAXIMUM OF 10 CALLS TO CONPOR IS ALLOWED,
C     BUT IF REQUIREMENTS EXCEED THIS PROVISION SET BLOCK=.T. &
C     IPWRIT=-1, AND SET POROSITY ARRAYS EXPLICITLY HERE AS WANTED.
C     IN THIS CASE, THE USER  M U S T   SET  A L L  ELEMENTS OF
C     ARRAYS PE, PN, PH, PC (MANY MAY BE 0.0 OR 1.0). HE MAY USE:
C     CALL CR(PARRAY,VALUE,IXF,IXL,IYF,IYL,IZF,IZL,NX,NY,NZ)
C     ANY NUMBER OF TIMES, TO SET 'PARRAY' (= PE, ETC.) TO
C     'VALUE' OVER RANGE IXF TO IXL, IYF TO IYL, IZF TO IZL.
C    *CONPOR  M U S T  N O T  BE USED IN CONJUNCTION WITH EXPLICIT
C     SETTINGS OF THE ARRAYS (INCLUDING SETTINGS VIA CR).
      BLOCK= .TRUE.
      IPWRIT= -1
C *** INITIALIZE ALL POROSITIES TO 1.0 (OPEN)
      DO 70 IX = 1,NX
      DO 70 IY = 1,NY
      DO 70 IZ = 1,NZ
      PE(IX,IY,IZ)= 1.0
      PN(IX,IY,IZ)= 1.0
      PH(IX,IY,IZ)= 1.0
 70   PC(IX,IY,IZ)= 1.0
CC
C  *** ROW 1 (BOTTOM)
      CALL CR(PC,0.0,1,NX, 1, 1, 1,12,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 1, 1, 1,13,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 1, 1, 1,12,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 1, 1, 1,12,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX, 1, 1,20,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 1, 1,20,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 1, 1,20,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 1, 1,19,28,NX,NY,NZ)
C *** ROW 2
      CALL CR(PC,0.0,1,NX, 2, 2, 1,13,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 2, 2, 1,13,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 2, 2, 1,13,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 2, 2, 1,13,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX, 2, 2,22,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 2, 2,22,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 2, 2,22,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 2, 2,21,28,NX,NY,NZ)
C
C *** ROW 3
      CALL CR(PC,0.0,1,NX, 3, 3, 1, 4,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 3, 3, 1, 4,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 3, 3, 1, 4,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 3, 3, 1, 4,NX,NY,NZ)
C
      CALL CR(PC,0.5,1,NX, 3, 3, 5, 5,NX,NY,NZ)
      CALL CR(PN,1.0,1,NX, 3, 3, 5, 5,NX,NY,NZ)
      CALL CR(PE,0.5,1,NX, 3, 3, 5, 5,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX, 3, 3, 5, 5,NX,NY,NZ)
C
      CALL CR(PC,0.5,1,NX, 3, 3,12,12,NX,NY,NZ)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CALL CR(PN,1.0,1,NX, 3, 3,12,12,NX,NY,NZ)
      CALL CR(PE,0.5,1,NX, 3, 3,12,12,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 3, 3,12,12,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX, 3, 3,13,13,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 3, 3,13,13,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 3, 3,13,13,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 3, 3,13,13,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX, 3, 3,22,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 3, 3,17,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 3, 3,22,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 3, 3,21,28,NX,NY,NZ)
C
C *** ROW 4
      CALL CR(PC,0.0,1,NX, 4, 4, 1, 3,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 4, 4, 1, 3,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 4, 4, 1, 3,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 4, 4, 1, 3,NX,NY,NZ)
C
      CALL CR(PC,.15,1,NX, 4, 4, 4, 4,NX,NY,NZ)
      CALL CR(PN,0.5,1,NX, 4, 4, 4, 4,NX,NY,NZ)
      CALL CR(PE,.15,1,NX, 4, 4, 4, 4,NX,NY,NZ)
      CALL CR(PH,.75,1,NX, 4, 4, 4, 4,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX, 4, 4,13,13,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 4, 4,13,13,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 4, 4,13,13,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 4, 4,12,13,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX, 4, 4,17,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 4, 4,17,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 4, 4,17,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 4, 4,16,28,NX,NY,NZ)
C
C *** ROW 5
      CALL CR(PC,0.0,1,NX, 5, 5, 1, 3,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 5, 5, 1, 3,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 5, 5, 1, 3,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 5, 5, 1, 3,NX,NY,NZ)
C
      CALL CR(PC,.75,1,NX, 5, 5, 4, 4,NX,NY,NZ)
      CALL CR(PN,1.0,1,NX, 5, 5, 4, 4,NX,NY,NZ)
      CALL CR(PE,.75,1,NX, 5, 5, 4, 4,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX, 5, 5, 4, 4,NX,NY,NZ)
C
      CALL CR(PC,.30,1,NX, 5, 5,13,13,NX,NY,NZ)
      CALL CR(PN,1.0,1,NX, 5, 5,13,13,NX,NY,NZ)
      CALL CR(PE,.30,1,NX, 5, 5,13,13,NX,NY,NZ)
      CALL CR(PH,.30,1,NX, 5, 5,12,13,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX, 5, 5,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 5, 5,21,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 5, 5,21,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 5, 5,20,28,NX,NY,NZ)
C
C *** ROW 6
      CALL CR(PC,0.0,1,NX, 6, 6, 1, 2,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 6, 6, 1, 2,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 6, 6, 1, 2,NX,NY,NZ)
```

```
      CALL CR(PH,0.0,1,NX, 6, 6, 1, 2,NX,NY,NZ)
C
      CALL CR(PC,.25,1,NX, 6, 6, 3, 3,NX,NY,NZ)
      CALL CR(PN,0.5,1,NX, 6, 6, 3, 3,NX,NY,NZ)
      CALL CR(PE,.25,1,NX, 6, 6, 3, 3,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX, 6, 6, 3, 3,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX, 6, 6,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 6, 6,21,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 6, 6,21,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 6, 6,20,28,NX,NY,NZ)
C
C *** ROW 7
      CALL CR(PC,0.0,1,NX, 7, 7, 1, 2,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 7, 7, 1, 2,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 7, 7, 1, 2,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 7, 7, 1, 2,NX,NY,NZ)
C
      CALL CR(PC,.85,1,NX, 7, 7, 3, 3,NX,NY,NZ)
      CALL CR(PN,1.0,1,NX, 7, 7, 3, 3,NX,NY,NZ)
      CALL CR(PE,.85,1,NX, 7, 7, 3, 3,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX, 7, 7, 3, 3,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX, 7, 7,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 7, 7,21,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 7, 7,21,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 7, 7,20,28,NX,NY,NZ)
C
C *** ROW 8
      CALL CR(PC,0.0,1,NX, 8, 8, 1, 1,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 8, 8, 1, 1,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 8, 8, 1, 1,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 8, 8, 1, 1,NX,NY,NZ)
C
      CALL CR(PC,0.5,1,NX, 8, 8, 2, 2,NX,NY,NZ)
      CALL CR(PN,.75,1,NX, 8, 8, 2, 2,NX,NY,NZ)
      CALL CR(PE,0.5,1,NX, 8, 8, 2, 2,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX, 8, 8, 2, 2,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX, 8, 8,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 8, 8,21,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 8, 8,21,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 8, 8,20,28,NX,NY,NZ)
C
C *** ROW 9
      CALL CR(PC,0.0,1,NX, 9, 9, 1, 1,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 9, 9, 1, 1,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 9, 9, 1, 1,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 9, 9, 1, 1,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX, 9, 9,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX, 9, 9,21,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX, 9, 9,21,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX, 9, 9,20,28,NX,NY,NZ)
C
      CALL CR(PC,.90,1,NX, 9, 9,2,2,NX,NY,NZ)
      CALL CR(PN,1.0,1,NX, 9, 9,2,2,NX,NY,NZ)
      CALL CR(PE,.90,1,NX, 9, 9,2,2,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX, 9, 9,2,2,NX,NY,NZ)
```

```
C *** ROW 10
      CALL CR(PC,.40,1,NX,10,10, 2, 2,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,10,10, 1, 2,NX,NY,NZ)
      CALL CR(PE,.40,1,NX,10,10, 2, 2,NX,NY,NZ)
      CALL CR(PH,.20,1,NX,10,10, 2, 2,NX,NY,NZ)
C
      CALL CR(PC,.20,1,NX,10,10, 3, 3,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,10,10, 3, 3,NX,NY,NZ)
      CALL CR(PE,.20,1,NX,10,10, 3, 3,NX,NY,NZ)
      CALL CR(PH,.30,1,NX,10,10, 3, 3,NX,NY,NZ)
C
      CALL CR(PC,.80,1,NX,10,10, 4, 4,NX,NY,NZ)
      CALL CR(PN,.35,1,NX,10,10, 4, 4,NX,NY,NZ)
      CALL CR(PE,.80,1,NX,10,10, 4, 4,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,10,10, 4, 4,NX,NY,NZ)
C
      CALL CR(PC,1.0,1,NX,10,10,20,20,NX,NY,NZ)
      CALL CR(PN,1.0,1,NX,10,10,20,20,NX,NY,NZ)
      CALL CR(PE,1.0,1,NX,10,10,20,20,NX,NY,NZ)
      CALL CR(PH,.20,1,NX,10,10,20,20,NX,NY,NZ)
C
      CALL CR(PC,.20,1,NX,10,10,21,21,NX,NY,NZ)
      CALL CR(PN,1.0,1,NX,10,10,21,21,NX,NY,NZ)
      CALL CR(PE,.20,1,NX,10,10,21,21,NX,NY,NZ)
      CALL CR(PH,.15,1,NX,10,10,21,21,NX,NY,NZ)
C
      CALL CR(PC,.10,1,NX,10,10,22,22,NX,NY,NZ)
      CALL CR(PN,1.0,1,NX,10,10,22,22,NX,NY,NZ)
      CALL CR(PE,.10,1,NX,10,10,22,22,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,10,10,22,22,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,10,10,23,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,10,10,23,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,10,10,23,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,10,10,23,28,NX,NY,NZ)
C
C *** ROW 11
      CALL CR(PC,0.0,1,NX,11,11, 1, 2,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,11,11, 1, 2,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,11,11, 1, 2,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,11,11, 1, 2,NX,NY,NZ)
C
      CALL CR(PC,0.5,1,NX,11,11, 3, 3,NX,NY,NZ)
      CALL CR(PN,.90,1,NX,11,11, 3, 3,NX,NY,NZ)
      CALL CR(PE,0.5,1,NX,11,11, 3, 3,NX,NY,NZ)
      CALL CR(PH,0.6,1,NX,11,11, 3, 3,NX,NY,NZ)
C
      CALL CR(PC,.75,1,NX,11,11, 4, 4,NX,NY,NZ)
      CALL CR(PN,1.0,1,NX,11,11, 4, 4,NX,NY,NZ)
      CALL CR(PE,.75,1,NX,11,11, 4, 4,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,11,11, 4, 4,NX,NY,NZ)
C
      CALL CR(PC,.95,1,NX,11,11,23,23,NX,NY,NZ)
      CALL CR(PN,1.0,1,NX,11,11,23,23,NX,NY,NZ)
      CALL CR(PE,.95,1,NX,11,11,23,23,NX,NY,NZ)
      CALL CR(PH,.80,1,NX,11,11,23,23,NX,NY,NZ)
C
      CALL CR(PC,0.5,1,NX,11,11,24,24,NX,NY,NZ)
      CALL CR(PN,1.0,1,NX,11,11,24,24,NX,NY,NZ)
      CALL CR(PE,0.5,1,NX,11,11,24,24,NX,NY,NZ)
```

```
      CALL CR(PH,0.0,1,NX,11,11,24,24,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,11,11,25,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,11,11,25,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,11,11,25,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,11,11,25,28,NX,NY,NZ)
C
C *** ROW 12
      CALL CR(PC,0.0,1,NX,12,12, 1, 2,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,12,12, 1, 2,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,12,12, 1, 2,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,12,12, 1, 2,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,12,12,25,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,12,12,25,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,12,12,25,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,12,12,24,28,NX,NY,NZ)
C
C *** ROW 13
      CALL CR(PC,0.0,1,NX,13,13, 1, 2,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,13,13, 1, 2,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,13,13, 1, 2,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,13,13, 1, 2,NX,NY,NZ)
C
      CALL CR(PC,.90,1,NX,13,13,23,23,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,13,13,23,23,NX,NY,NZ)
      CALL CR(PE,.90,1,NX,13,13,23,23,NX,NY,NZ)
      CALL CR(PH,.80,1,NX,13,13,23,23,NX,NY,NZ)
C
      CALL CR(PC,0.5,1,NX,13,13,24,24,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,13,13,24,24,NX,NY,NZ)
      CALL CR(PE,0.5,1,NX,13,13,24,24,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,13,13,24,24,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,13,13,25,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,13,13,25,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,13,13,25,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,13,13,25,28,NX,NY,NZ)
C
C *** ROW 14
      CALL CR(PC,0.0,1,NX,14,14, 1, 2,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,14,14, 1, 2,NX,NY,NZ)
      CALL CR(PE,0.0!1,NX,14,14, 1, 2,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,14,14, 1, 2,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,14,14,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,14,14,21,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,14,14,21,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,14,14,20,28,NX,NY,NZ)
C
C *** ROW 15
      CALL CR(PC,0.0,1,NX,15,15, 1, 2,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,15,15, 1, 2,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,15,15, 1, 2,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,15,15, 1, 2,NX,NY,NZ)
C
      CALL CR(PC,.75,1,NX,15,15,23,23,NX,NY,NZ)
      CALL CR(PN,1.0,1,NX,15,15,23,23,NX,NY,NZ)
      CALL CR(PE,.75,1,NX,15,15,23,23,NX,NY,NZ)
      CALL CR(PH,0.5,1,NX,15,15,23,23,NX,NY,NZ)
```

```
C
      CALL CR(PC,.15,1,NX,15,15,24,24,NX,NY,NZ)
      CALL CR(PN,0.5,1,NX,15,15,24,24,NX,NY,NZ)
      CALL CR(PE,.15,1,NX,15,15,24,24,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,15,15,24,24,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,15,15,25,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,15,15,25,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,15,15,25,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,15,15,25,28,NX,NY,NZ)
C
C *** ROW 16
      CALL CR(PC,0.0,1,NX,16,16, 1, 2,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,16,16, 1, 2,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,16,16, 1, 2,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,16,16, 1, 2,NX,NY,NZ)
C
      CALL CR(PC,.15,1,NX,16,16,25,25,NX,NY,NZ)
      CALL CR(PN,.30,1,NX,16,16,25,25,NX,NY,NZ)
      CALL CR(PE,.15,1,NX,16,16,25,25,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,16,16,25,25,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,16,16,26,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,16,16,26,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,16,16,26,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,16,16,26,28,NX,NY,NZ)
C
C *** ROW 17
      CALL CR(PC,0.0,1,NX,17,17, 1, 2,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,17,17, 1, 3,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,17,17, 1, 2,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,17,17, 1, 2,NX,NY,NZ)
C
      CALL CR(PC,1.0,1,NX,17,17, 4, 4,NX,NY,NZ)
      CALL CR(PN,0.4,1,NX,17,17, 4, 4,NX,NY,NZ)
      CALL CR(PE,1.0,1,NX,17,17, 4, 4,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,17,17, 4, 4,NX,NY,NZ)
C
      CALL CR(PC,0.5,1,NX,17,17,25,25,NX,NY,NZ)
      CALL CR(PN,.70,1,NX,17,17,25,25,NX,NY,NZ)
      CALL CR(PE,0.5,1,NX,17,17,25,25,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,17,17,25,25,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,17,17,26,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,17,17,26,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,17,17,26,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,17,17,26,28,NX,NY,NZ)
C
C *** ROW 18
      CALL CR(PC,0.0,1,NX,18,18, 1, 3,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,18,18, 1, 3,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,18,18, 1, 3,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,18,18, 1, 3,NX,NY,NZ)
C
      CALL CR(PC,.40,1,NX,18,18, 4, 4,NX,NY,NZ)
      CALL CR(PN,0.4,1,NX,18,18, 4, 4,NX,NY,NZ)
      CALL CR(PE,.40,1,NX,18,18, 4, 4,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,18,18, 4, 4,NX,NY,NZ)
C
      CALL CR(PC,.75,1,NX,18,18,25,25,NX,NY,NZ)
```

```
      CALL CR(PN,.75,1,NX,18,18,25,25,NX,NY,NZ)
      CALL CR(PE,.75,1,NX,18,18,25,25,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,18,18,25,25,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,18,18,26,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,18,18,26,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,18,18,26,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,18,18,26,28,NX,NY,NZ)
C
C *** ROW 19
      CALL CR(PC,0.0,1,NX,19,19, 1, 3,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,19,19, 1, 3,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,19,19, 1, 3,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,19,19, 1, 3,NX,NY,NZ)
C
      CALL CR(PC,.75,1,NX,19,19, 4, 4,NX,NY,NZ)
      CALL CR(PN,1.0,1,NX,19,19, 4, 4,NX,NY,NZ)
      CALL CR(PE,.75,1,NX,19,19, 4, 4,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,19,19, 4, 4,NX,NY,NZ)
C
      CALL CR(PC,.75,1,NX,19,19,25,25,NX,NY,NZ)
      CALL CR(PN,.75,1,NX,19,19,25,25,NX,NY,NZ)
      CALL CR(PE,.75,1,NX,19,19,25,25,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,19,19,25,25,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,19,19,26,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,19,19,26,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,19,19,26,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,19,19,26,28,NX,NY,NZ)
C
C *** ROW 20
      CALL CR(PC,0.0,1,NX,20,20, 1, 3,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,20,20, 1, 3,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,20,20, 1, 3,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,20,20, 1, 3,NX,NY,NZ)
C
      CALL CR(PC,0.9,1,NX,20,20, 4, 4,NX,NY,NZ)
      CALL CR(PN,0.5,1,NX,20,20, 4, 4,NX,NY,NZ)
      CALL CR(PE,0.9,1,NX,20,20, 4, 4,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,20,20, 4, 4,NX,NY,NZ)
C
      CALL CR(PC,0.5,1,NX,20,20,25,25,NX,NY,NZ)
      CALL CR(PN,0.4,1,NX,20,20,25,25,NX,NY,NZ)
      CALL CR(PE,0.5,1,NX,20,20,25,25,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,20,20,25,25,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,20,20,26,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,20,20,26,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,20,20,26,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,20,20,26,28,NX,NY,NZ)
C
C ***  ROW  21
      CALL CR(PC,0.0,1,NX,21,21, 1, 3,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,21,21, 1, 3,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,21,21, 1, 3,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,21,21, 1, 3,NX,NY,NZ)
C
      CALL CR(PC,.05,1,NX,21,21, 4, 4,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,21,21, 4, 4,NX,NY,NZ)
      CALL CR(PE,.05,1,NX,21,21, 4, 4,NX,NY,NZ)
```

```
      CALL CR(PH,0.2,1,NX,21,21, 4, 4,NX,NY,NZ)
C
      CALL CR(PC,.45,1,NX,21,21, 5, 5,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,21,21, 5, 5,NX,NY,NZ)
      CALL CR(PE,.45,1,NX,21,21, 5, 5,NX,NY,NZ)
      CALL CR(PH,0.7,1,NX,21,21, 5, 5,NX,NY,NZ)
C
      CALL CR(PC,0.9,1,NX,21,21, 6, 6,NX,NY,NZ)
      CALL CR(PN,0.4,1,NX,21,21, 6, 6,NX,NY,NZ)
      CALL CR(PE,0.9,1,NX,21,21, 6, 6,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,21,21, 6, 6,NX,NY,NZ)
C
      CALL CR(PC,0.8,1,NX,21,21,23,23,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,21,21,23,23,NX,NY,NZ)
      CALL CR(PE,0.8,1,NX,21,21,23,23,NX,NY,NZ)
      CALL CR(PH,0.7,1,NX,21,21,23,23,NX,NY,NZ)
C
      CALL CR(PC,0.6,1,NX,21,21,24,24,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,21,21,24,24,NX,NY,NZ)
      CALL CR(PE,0.6,1,NX,21,21,24,24,NX,NY,NZ)
      CALL CR(PH,0.4,1,NX,21,21,24,24,NX,NY,NZ)
C
      CALL CR(PC,.05,1,NX,21,21,25,25,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,21,21,25,25,NX,NY,NZ)
      CALL CR(PE,.05,1,NX,21,21,25,25,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,21,21,25,25,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,21,21,26,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,21,21,26,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,21,21,26,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,21,21,26,28,NX,NY,NZ)
C
C *** ROW 22
      CALL CR(PC,0.0,1,NX,22,22, 1, 5,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,22,22, 1, 5,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,22,22, 1, 5,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,22,22, 1, 5,NX,NY,NZ)
C
      CALL CR(PC,.05,1,NX,22,22, 6, 6,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,22,22, 6, 6,NX,NY,NZ)
      CALL CR(PE,.05,1,NX,22,22, 6, 6,NX,NY,NZ)
      CALL CR(PH,.20,1,NX,22,22, 6, 6,NX,NY,NZ)
C
      CALL CR(PC,0.4,1,NX,22,22, 7, 7,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,22,22, 7, 7,NX,NY,NZ)
      CALL CR(PE,0.4,1,NX,22,22, 7, 7,NX,NY,NZ)
      CALL CR(PH,0.6,1,NX,22,22, 7, 7,NX,NY,NZ)
C
      CALL CR(PC,0.7,1,NX,22,22, 8, 8,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,22,22, 8, 8,NX,NY,NZ)
      CALL CR(PE,0.7,1,NX,22,22, 8, 8,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,22,22, 8, 8,NX,NY,NZ)
C
      CALL CR(PC,1.0,1,NX,22,22,20,20,NX,NY,NZ)
      CALL CR(PN,1.0,1,NX,22,22,20,20,NX,NY,NZ)
      CALL CR(PE,1.0,1,NX,22,22,20,20,NX,NY,NZ)
      CALL CR(PH,0.3,1,NX,22,22,20,20,NX,NY,NZ)
C
      CALL CR(PC,0.2,1,NX,22,22,21,21,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,22,22,21,21,NX,NY,NZ)
```

```
      CALL CR(PE,0.2,1,NX,22,22,21,21,NX,NY,NZ)
      CALL CR(PH,0.1,1,NX,22,22,21,21,NX,NY,NZ)
C
      CALL CR(PC,0.1,1,NX,22,22,22,22,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,22,22,22,22,NX,NY,NZ)
      CALL CR(PE,0.1,1,NX,22,22,22,22,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,22,22,22,22,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,22,22,23,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,22,22,23,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,22,22,23,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,22,22,23,28,NX,NY,NZ)
C
C *** ROW 23
      CALL CR(PC,0.0,1,NX,23,23, 1, 8,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,23,23, 1, 8,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,23,23, 1, 8,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,23,23, 1, 8,NX,NY,NZ)
C
      CALL CR(PC,0.5,1,NX,23,23, 9, 9,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,23,23, 9, 9,NX,NY,NZ)
      CALL CR(PE,0.5,1,NX,23,23, 9, 9,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,23,23, 9, 9,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,23,23,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,23,23,21,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,23,23,21,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,23,23,20,28,NX,NY,NZ)
C
C
C *** ROW 24
      CALL CR(PC,0.0,1,NX,24,24, 1, 9,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,24,24, 1, 9,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,24,24, 1, 9,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,24,24, 1, 9,NX,NY,NZ)
C
      CALL CR(PC,0.5,1,NX,24,24,10,10,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,24,24,10,10,NX,NY,NZ)
      CALL CR(PE,0.5,1,NX,24,24,10,10,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,24,24,10,10,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,24,24,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,24,24,21,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,24,24,21,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,24,24,20,28,NX,NY,NZ)
C
C
C *** ROW 25
      CALL CR(PC,0.0,1,NX,25,25, 1,10,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,25,25, 1,10,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,25,25, 1,10,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,25,25, 1,10,NX,NY,NZ)
C
      CALL CR(PC,0.5,1,NX,25,25,11,11,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,25,25,11,11,NX,NY,NZ)
      CALL CR(PE,0.5,1,NX,25,25,11,11,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,25,25,11,11,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,25,25,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,25,25,21,28,NX,NY,NZ)
```

```
      CALL CR(PE,0.0,1,NX,25,25,21,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,25,25,20,28,NX,NY,NZ)
C
C
C *** ROW 26
      CALL CR(PC,0.0,1,NX,26,26, 1,11,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,26,26, 1,11,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,26,26, 1,11,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,26,26, 1,11,NX,NY,NZ)
C
      CALL CR(PC,0.5,1,NX,26,26,12,12,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,26,26,12,12,NX,NY,NZ)
      CALL CR(PE,0.5,1,NX,26,26,12,12,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,26,26,12,12,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,26,26,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,26,26,21,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,26,26,21,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,26,26,20,28,NX,NY,NZ)
C
C *** ROW 27
      CALL CR(PC,0.0,1,NX,27,27, 1,12,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,27,27, 1,12,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,27,27, 1,12,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,27,27, 1,12,NX,NY,NZ)
C
      CALL CR(PC,0.5,1,NX,27,27,13,13,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,27,27,13,13,NX,NY,NZ)
      CALL CR(PE,0.5,1,NX,27,27,13,13,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,27,27,13,13,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,27,27,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,27,27,21,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,27,27,21,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,27,27,20,28,NX,NY,NZ)
C
C *** ROW 28
      CALL CR(PC,0.0,1,NX,28,28, 1,13,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,28,28, 1,13,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,28,28, 1,13,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,28,28, 1,13,NX,NY,NZ)
C
      CALL CR(PC,0.5,1,NX,28,28,14,14,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,28,28,14,14,NX,NY,NZ)
      CALL CR(PE,0.5,1,NX,28,28,14,14,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,28,28,14,14,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,28,28,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,28,28,21,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,28,28,21,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,28,28,20,28,NX,NY,NZ)
C
C *** ROW 29
      CALL CR(PC,0.0,1,NX,29,29, 1,14,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,29,29, 1,14,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,29,29, 1,14,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,29,29, 1,14,NX,NY,NZ)
C
      CALL CR(PC,.25,1,NX,29,29,15,15,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,29,29,15,15,NX,NY,NZ)
```

```
      CALL CR(PE,.25,1,NX,29,29,15,15,NX,NY,NZ)
      CALL CR(PH,.75,1,NX,29,29,15,15,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,29,29,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,29,29,21,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,29,29,21,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,29,29,20,28,NX,NY,NZ)
C
C *** ROW 30
      CALL CR(PC,0.0,1,NX,30,30, 1,15,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,30,30, 1,15,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,30,30, 1,15,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,30,30, 1,15,NX,NY,NZ)
C
      CALL CR(PC,0.6,1,NX,30,30,16,16,NX,NY,NZ)
      CALL CR(PN,.25,1,NX,30,30,16,16,NX,NY,NZ)
      CALL CR(PE,0.6,1,NX,30,30,16,16,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,30,30,16,16,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,30,30,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,30,30,21,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,30,30,21,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,30,30,20,28,NX,NY,NZ)
C
C *** ROW 31
      CALL CR(PC,0.0,1,NX,31,31, 1,15,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,31,31, 1,15,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,31,31, 1,15,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,31,31, 1,15,NX,NY,NZ)
C
      CALL CR(PC,0.1,1,NX,31,31,16,16,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,31,31,16,16,NX,NY,NZ)
      CALL CR(PE,0.1,1,NX,31,31,16,16,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,31,31,16,16,NX,NY,NZ)
C
      CALL CR(PC,0.0,1,NX,31,31,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,31,31,21,28,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,31,31,21,28,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,31,31,20,28,NX,NY,NZ)
C
C *** ROW 32
      CALL CR(PC,0.0,1,NX,32,32, 1,16,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,32,32, 1,16,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,32,32, 1,16,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,32,32, 1,16,NX,NY,NZ)
C
      CALL CR(PC,0.8,1,NX,32,32,17,17,NX,NY,NZ)
      CALL CR(PN,.75,1,NX,32,32,17,17,NX,NY,NZ)
      CALL CR(PE,0.8,1,NX,32,32,17,17,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,32,32,17,17,NX,NY,NZ)
C
C *** NOTE: ROWS 32-40 CONTAIN THE HOT GAS PASSAGES THROUGH
C     THE TURBINE BLADE SHANKS. THE POROSITIES FOR THESE CELLS
C     (ROWS 32-40, COLUMNS 21-25) DEPEND ON THE RATIO OF THE HOT
C     GAS INLET AREA TO THE CORRESPONDING GRID AREA (RAT)
      RAT= AINH1/ARGRD1
C
      CALL CR(PC,RAT,1,NX,32,32,21,28,NX,NY,NZ)
      CALL CR(PN,RAT,1,NX,32,32,21,28,NX,NY,NZ)
      CALL CR(PE,1.0,1,NX,32,32,21,28,NX,NY,NZ)
```

```
      CALL CR(PH,RAT,1,NX,32,32,20,28,NX,NY,NZ)
C
C *** ROW 33
      CALL CR(PC,0.0,1,NX,33,33, 1,16,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,33,33, 1,16,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,33,33, 1,16,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,33,33, 1,16,NX,NY,NZ)
C
      CALL CR(PC,.75,1,NX,33,33,17,17,NX,NY,NZ)
      CALL CR(PN,0.6,1,NX,33,33,17,17,NX,NY,NZ)
      CALL CR(PE,.75,1,NX,33,33,17,17,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,33,33,17,17,NX,NY,NZ)
C
      CALL CR(PC,RAT,1,NX,33,33,21,28,NX,NY,NZ)
      CALL CR(PN,RAT,1,NX,33,33,21,28,NX,NY,NZ)
      CALL CR(PE,1.0,1,NX,33,33,21,28,NX,NY,NZ)
      CALL CR(PH,RAT,1,NX,33,33,20,28,NX,NY,NZ)
C
C *** ROW 34
      CALL CR(PC,0.0,1,NX,34,34, 1,16,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,34,34, 1,16,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,34,34, 1,16,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,34,34, 1,16,NX,NY,NZ)
C
      CALL CR(PC,0.6,1,NX,34,34,17,17,NX,NY,NZ)
      CALL CR(PN,0.5,1,NX,34,34,17,17,NX,NY,NZ)
      CALL CR(PE,0.6,1,NX,34,34,17,17,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,34,34,17,17,NX,NY,NZ)
C
      CALL CR(PC,RAT,1,NX,34,34,21,28,NX,NY,NZ)
      CALL CR(PN,RAT,1,NX,34,34,21,28,NX,NY,NZ)
      CALL CR(PE,1.0,1,NX,34,34,21,28,NX,NY,NZ)
      CALL CR(PH,RAT,1,NX,34,34,20,28,NX,NY,NZ)
C
C *** ROW 35
      CALL CR(PC,0.0,1,NX,35,35, 1,16,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,35,35, 1,16,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,35,35, 1,16,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,35,35, 1,16,NX,NY,NZ)
C
      CALL CR(PC,0.5,1,NX,35,35,17,17,NX,NY,NZ)
      CALL CR(PN,0.4,1,NX,35,35,17,17,NX,NY,NZ)
      CALL CR(PE,0.5,1,NX,35,35,17,17,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,35,35,17,17,NX,NY,NZ)
C
      CALL CR(PC,RAT,1,NX,35,35,21,28,NX,NY,NZ)
      CALL CR(PN,RAT,1,NX,35,35,21,28,NX,NY,NZ)
      CALL CR(PE,1.0,1,NX,35,35,21,28,NX,NY,NZ)
      CALL CR(PH,RAT,1,NX,35,35,20,28,NX,NY,NZ)
C
C *** ROW 36
      CALL CR(PC,0.0,1,NX,36,36, 1,16,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,36,36, 1,16,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,36,36, 1,16,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,36,36, 1,16,NX,NY,NZ)
C
      CALL CR(PC,0.3,1,NX,36,36,17,17,NX,NY,NZ)
      CALL CR(PN,0.3,1,NX,36,36,17,17,NX,NY,NZ)
      CALL CR(PE,0.3,1,NX,36,36,17,17,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,36,36,17,17,NX,NY,NZ)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
      CALL CR(PC,RAT,1,NX,36,36,21,28,NX,NY,NZ)
      CALL CR(PN,RAT,1,NX,36,36,21,28,NX,NY,NZ)
      CALL CR(PE,1.0,1,NX,36,36,21,28,NX,NY,NZ)
      CALL CR(PH,RAT,1,NX,36,36,20,28,NX,NY,NZ)
C
C *** ROW 37
      CALL CR(PC,0.0,1,NX,37,37, 1,16,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,37,37, 1,16,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,37,37, 1,16,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,37,37, 1,16,NX,NY,NZ)
C
      CALL CR(PC,0.2,1,NX,37,37,17,17,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,37,37,17,17,NX,NY,NZ)
      CALL CR(PE,0.2,1,NX,37,37,17,17,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,37,37,17,17,NX,NY,NZ)
C
      CALL CR(PC,RAT,1,NX,37,37,21,28,NX,NY,NZ)
      CALL CR(PN,RAT,1,NX,37,37,21,28,NX,NY,NZ)
      CALL CR(PE,1.0,1,NX,37,37,21,28,NX,NY,NZ)
      CALL CR(PH,RAT,1,NX,37,37,20,28,NX,NY,NZ)
C
C *** ROW 38
      CALL CR(PC,0.0,1,NX,38,38, 1,17,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,38,38, 1,17,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,38,38, 1,17,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,38,38, 1,17,NX,NY,NZ)
C
      CALL CR(PC,RAT,1,NX,38,38,21,28,NX,NY,NZ)
      CALL CR(PN,RAT,1,NX,38,38,21,28,NX,NY,NZ)
      CALL CR(PE,1.0,1,NX,38,38,21,28,NX,NY,NZ)
      CALL CR(PH,RAT,1,NX,38,38,20,28,NX,NY,NZ)
C
C *** ROW 39
      CALL CR(PC,0.0,1,NX,39,39, 1,17,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,39,39, 1,17,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,39,39, 1,17,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,39,39, 1,17,NX,NY,NZ)
C
      CALL CR(PC,RAT,1,NX,39,39,21,28,NX,NY,NZ)
      CALL CR(PN,RAT,1,NX,39,39,21,28,NX,NY,NZ)
      CALL CR(PE,1.0,1,NX,39,39,21,28,NX,NY,NZ)
      CALL CR(PH,RAT,1,NX,39,39,20,28,NX,NY,NZ)
C
C *** ROW 40
      CALL CR(PC,0.0,1,NX,40,40, 1,16,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,40,40, 1,16,NX,NY,NZ)
      CALL CR(PE,0.0,1,NX,40,40, 1,16,NX,NY,NZ)
      CALL CR(PH,0.0,1,NX,40,40, 1,16,NX,NY,NZ)
C
C
      CALL CR(PC,1.0,1,NX,40,40,17,17,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,40,40,17,20,NX,NY,NZ)
      CALL CR(PE,1.0,1,NX,40,40,17,17,NX,NY,NZ)
      CALL CR(PH,1.0,1,NX,40,40,17,17,NX,NY,NZ)
C
      CALL CR(PC,RAT,1,NX,40,40,21,28,NX,NY,NZ)
      CALL CR(PN,0.0,1,NX,40,40,21,28,NX,NY,NZ)
      CALL CR(PE,1.0,1,NX,40,40,21,28,NX,NY,NZ)
      CALL CR(PH,RAT,1,NX,40,40,20,28,NX,NY,NZ)
```

```
C
C-------------------    --------------    ------------------
C---  GROUP 8.DEPENDENT VARIABLES TO BE SOLVED FOR OR STORED :
C     SOLVAR(1-25)<25*.F.>,STOVAR(1-25)<25*.F.>,CONC1(1-4)<4*.T.>
C     USE FOLLOWING NAMED INTEGERS FOR ARRAY ELEMENTS 1-20:
C     P1,PP,U1,U2,V1,V2,W1,W2,M1,M2,RS,KE,EP,H1,H2,H3,C1,C2,C3,C4.
      SOLVAR(P1)= .TRUE.
      SOLVAR(PP)= .TRUE.
      SOLVAR(U1)= .TRUE.
      SOLVAR(V1)= .TRUE.
      SOLVAR(W1)= .TRUE.
      SOLVAR(KE)= .TRUE.
      SOLVAR(EP)= .TRUE.
      SOLVAR(H1)= .TRUE.
      SOLVAR(C1)= .TRUE.
CC
      STOVAR(18)= .TRUE.
      STOVAR(19)= .TRUE.
      STOVAR(21)= .TRUE.
      STOVAR(22)= .TRUE.
      STOVAR(23)= .TRUE.
C-------------------    --------------    ------------------
C---  GROUP 9. VARIABLE LABELS :
C     TITLE(1-25)<2HP1,2HPP,2HU1,2HU2,2HV1,2HV2,2HW1,2HW2,2HR1,
C     2HR2,2HRS,2HKE,2HEP,2HH1,2HH2,2HH3,2HC1,2HC2,
C     2HC3,2HC4,2HRX,2HRY,2HRZ, 2*4H****>
CC *** ENTHALPY OF THE MIXTURE
      TITLE(H1)= 4HHMIX
CC *** MASS FRACTION OF THE WATER
      TITLE(C1)= 4HMH2O
CC *** TEMPERATURE OF THE MIXTURE
      TITLE(18)= 4HTMIX
CC *** TOTAL PRESSURE
      TITLE(19)= 4HPTOT
CC *** DENSITY OF THE MIXTURE
      TITLE(2)= 4HRMIX
CC *** DENSITY OF THE WATER
      TITLE(15)= 4HRH2O
CC *** DENSITY OF THE 'HYDROGEN
      TITLE(16)= 4HRH2
CC *** EFFECTIVE VISCOSITY
      TITLE(21)= 4HEMU
CC *** PRESSURE CORRECTION
      TITLE(22)= 4HPP
CC *** CONTINUITY ERROR
      TITLE(23)= 4HCONT
C-------------------    --------------    ------------------
C---  GROUP 10 PROPERTIES:
C     IRHO1<1>,IRHO2<1>,RHO1<1.O>,RHO2<1.O>,
C     ARHO1<1.O>,BRHO1<1.O>,CRHO1<1.O>
C     IEMU1<1>,EMU1<1.O>,EMULAM<1.E-10>
C     IHSAT,H1SAT,H2SAT,PSATEX<1.O>
C     SIGMA(1-25)<1.O,2.O,1.,1.E10,1.,1.E10,1.,1.E10,
C     4*1.O,1.314,1.O,1.E10,10*1.O>
CC
C      *** UNITS ARE IN LBF, SLUGS, FEET, AND DEGREES RANKINE
CC
CC *** THE DENSITY IS CALCULATED IN GROUND CH. 10
      IRHO1=-1
CC *** SETTING IEMU1 = 2 IMPLIES THE K-EPSILON MODEL IS ACTIVE
```

```
      IEMU1= 2
CC *** SIGMA = TURB. PRANDTL OR SCHMIDT NO.
C       FOR H1 AND C1 THEY ARE .9 BASED ON CHAM TR/75, PAGE 3.2-26
      SIGMA(H1)= 0.9
      SIGMA(C1)= 0.9
CC *** LAM VISCOSITY FOR WALL FRICTION IS CALCULATED IN GROUND CH. 10
      EMULAM= -1.
C-------------------    ---------------    ------------------
C---  GROUP 11 INTER-PHASE TRANSFER PROCESSES :
C     ICFIP,CFIPS,IMDOT,CMDOT,CA1I<1.E6>,CA2I<1.E6>
C-------------------    ---------------    ------------------
C---  GROUP 12 SPECIAL SOURCES :
C     ISPCSO(1-25),AGRAVX,AGRAVY,AGRAVZ,ABUOY,HREF
C-------------------    ---------------    ------------------
C---  GROUP 13 INITIAL FIELDS :
C     FIINIT(1-25)<25*1.E-10>
C
      OMEGA = RPM*2.*PI/60.
C
      FIINIT(U1)=0.4*OMEGA
      FIINIT(V1)=0.0
      FIINIT(W1)=0.0
      FIINIT(C1)=0.1
CC *** SET TEMP AT INTERMEDIATE VALUE (LMSC-HREC TR D697954)
      FIINIT(18) = 400.
CC *** FIINIT (P1), (H1), (KE) & (EP) ARE SET BELOW IN GROUP 15
C-------------------    ---------------    ------------------
C---  GROUP 14 BOUNDARY/INTERNAL CONDITIONS :
C     ILOOP1,ILOOPN,XCYCLE<.F.>,PBAR,REGION(1-10)<10*.T.>
C    *N.B. ALL 10 REGIONS ARE DEFAULTED .TRUE..  THE USER SHOULD
C     SET REGION(I)=.FALSE. FOR UNUSED REGIONS 'I'.
      DO 140 I=1,10
 140    REGION(I)= .FALSE.
      XCYCLE = .TRUE.
C-------------------    ---------------    ------------------
C---  GROUP 15 TO 24                   REGIONS 1 TO 10
C---  ONLY THOSE REGIONS ARE ACTIVE WHICH ARE SPECIFIED BY THE
C     USER, PREFERABLY BY WAY OF:-
C     CALL PLACE(IREGN,TYPE,IXF,IXL,IYF,IYL,IZF,IZL) &
C     CALL COVAL(IREGN,VARBLE,COEFF,VALUE)
C
CC ***  'COLD'   H2 , INLET     ***
C
CC *** FEEDC1 IS SET TO THE TOTAL MASS FLOWRATE (LBM/S)
CC    AT THE COLD INLET, (SEE LMSC-HREC TR D697954),
CC    THEN CONVERTED TO SLUGS/SEC OVER SOLUTION SEGMENT
      FEEDC = FEEDC1/(G*SLICES)
CCC   AS OF 3/85 THE FLOW THROUGH THE LABY SEAL IS VARIED AS
CCC   AS A FUNCTION OF THE ECCENTRICITY OF THE ROTOR:
C
C    WEIGHTED FLOWRATE (PER SEGMENT) =(TOTAL FLOWRATE/NX)*
C    ((SM GAP HT)-COS(ANGLE)*(ECCENTRICITY))/(SM GAP HT)
C
C    FOR EIGHT CELLS IN THE X DIRECTION:
      FEDC1 = FEEDC*(GINC1S-COS(0.)*ECCENT)/GINC1S
      FEDC2 = FEEDC*(GINC1S-COS(2.*PI/8.)*ECCENT)/GINC1S
      FEDC3 = FEEDC*(GINC1S-COS(2.*PI/4.)*ECCENT)/GINC1S
      FEDC4 = FEEDC*(GINC1S-COS(2.*PI*3./8.)*ECCENT)/GINC1S
      FEDC5 = FEEDC*(GINC1S-COS(PI)*ECCENT)/GINC1S
      FEDC6 = FEEDC*(GINC1S-COS(2.*PI*5./8.)*ECCENT)/GINC1S
```

```
      FEDC7 = FEEDC*(GINC1S-COS(2.*PI*3./4.)*ECCENT)/GINC1S
      FEDC8 = FEEDC*(GINC1S-COS(2.*PI*7./8.)*ECCENT)/GINC1S
C
CC *** H1INC1 IS SET TO THE ENTHALPY (BTU/LBM) AT THE COLD
CC      INLET (TR D697954) THEN CONVERTED TO FT-LBF/SLUGS
      H1INC = H1INC1*778.16*G
CC *** ROINC IS THE DENSITY (SLUG/CU FT) AT THE COLD INLET
      ROINC = ROINC1/G
CC *** RADINC IS THE AVERAGE RADIUS (FT) OF THE COLD INLET
      RADINC = RINNER +0.05/12.
CC *** GINC1 IS SET TO THE (LARGE) GAP HT (INCHES) AT COLD INLET
CC *** CALCULATE THE INLET AREA (SQ IN) & (SQ FT/SEGMENT)
      AINC1 = GINC1 * RADINC *12.*2.*PI
      AINC = AINC1/(144.0*SLICES)
CC *** CALCULATE THE AVERAGE FEED VELOCITY AT THE COLD INLET
      W1INC = FEEDC/(ROINC*AINC)
      VELSQ=W1INC**2 + (0.5*OMEGA*RADINC)**2
C
      CALL PLACE(1,CELL,1,1,1,1,13,13)
      CALL COVAL(1,M1,FIXFLU,FEDC1/FLOAT(NX))
      CALL COVAL(1,U1,ONLYMS,0.5*OMEGA*(RADINC**2))
      CALL COVAL(1,W1,ONLYMS,W1INC)
      CALL COVAL(1,KE,ONLYMS,.01*VELSQ)
      CALL COVAL(1,EP,ONLYMS,0.16433*(.01*VELSQ)**1.5/(.1*GINC1/12.))
      CALL COVAL(1,H1,ONLYMS,H1INC)
      CALL COVAL(1,C1,ONLYMS,0.0)
C
      CALL PLACE(2,CELL,2, 2,1,1,13,13)
      CALL COVAL(2,M1,FIXFLU,FEDC2/FLOAT(NX))
      CALL COVAL(2,U1,ONLYMS,0.5*OMEGA*(RADINC**2))
      CALL COVAL(2,W1,ONLYMS,W1INC)
      CALL COVAL(2,KE,ONLYMS,.01*VELSQ)
      CALL COVAL(2,EP,ONLYMS,0.16433*(.01*VELSQ)**1.5/(.1*GINC1/12.))
      CALL COVAL(2,H1,ONLYMS,H1INC)
      CALL COVAL(2,C1,ONLYMS,0.0)
C
      CALL PLACE(3,CELL,3, 3,1,1,13,13)
      CALL COVAL(3,M1,FIXFLU,FEDC3/FLOAT(NX))
      CALL COVAL(3,U1,ONLYMS,0.5*OMEGA*(RADINC**2))
      CALL COVAL(3,W1,ONLYMS,W1INC)
      CALL COVAL(3,KE,ONLYMS,.01*VELSQ)
      CALL COVAL(3,EP,ONLYMS,0.16433*(.01*VELSQ)**1.5/(.1*GINC1/12.))
      CALL COVAL(3,H1,ONLYMS,H1INC)
      CALL COVAL(3,C1,ONLYMS,0.0)
C
      CALL PLACE(4,CELL,4, 4,1,1,13,13)
      CALL COVAL(4,M1,FIXFLU,FEDC4/FLOAT(NX))
      CALL COVAL(4,U1,ONLYMS,0.5*OMEGA*(RADINC**2))
      CALL COVAL(4,W1,ONLYMS,W1INC)
      CALL COVAL(4,KE,ONLYMS,.01*VELSQ)
      CALL COVAL(4,EP,ONLYMS,0.16433*(.01*VELSQ)**1.5/(.1*GINC1/12.))
      CALL COVAL(4,H1,ONLYMS,H1INC)
      CALL COVAL(4,C1,ONLYMS,0.0)
C
      CALL PLACE(5,CELL,5, 5,1,1,13,13)
      CALL COVAL(5,M1,FIXFLU,FEDC5/FLOAT(NX))
      CALL COVAL(5,U1,ONLYMS,0.5*OMEGA*(RADINC**2))
      CALL COVAL(5,W1,ONLYMS,W1INC)
      CALL COVAL(5,KE,ONLYMS,.01*VELSQ)
      CALL COVAL(5,EP,ONLYMS,0.16433*(.01*VELSQ)**1.5/(.1*GINC1/12.))
```

```
      CALL COVAL(5,H1,ONLYMS,H1INC)
      CALL COVAL(5,C1,ONLYMS,0.0)
C
      CALL PLACE(6,CELL,6, 6,1,1,13,13)
      CALL COVAL(6,M1,FIXFLU,FEDC6/FLOAT(NX))
      CALL COVAL(6,U1,ONLYMS,0.5*OMEGA*(RADINC**2))
      CALL COVAL(6,W1,ONLYMS,W1INC)
      CALL COVAL(6,KE,ONLYMS,.01*VELSQ)
      CALL COVAL(6,EP,ONLYMS,0.16433*(.01*VELSQ)**1.5/(.1*GINC1/12.))
      CALL COVAL(6,H1,ONLYMS,H1INC)
      CALL COVAL(6,C1,ONLYMS,0.0)
C
      CALL PLACE(7,CELL,7, 7,1,1,13,13)
      CALL COVAL(7,M1,FIXFLU,FEDC7/FLOAT(NX))
      CALL COVAL(7,U1,ONLYMS,0.5*OMEGA*(RADINC**2))
      CALL COVAL(7,W1,ONLYMS,W1INC)
      CALL COVAL(7,KE,ONLYMS,.01*VELSQ)
      CALL COVAL(7,EP,ONLYMS,0.16433*(.01*VELSQ)**1.5/(.1*GINC1/12.))
      CALL COVAL(7,H1,ONLYMS,H1INC)
      CALL COVAL(7,C1,ONLYMS,0.0)
C
      CALL PLACE(8,CELL,8, 8,1,1,13,13)
      CALL COVAL(8,M1,FIXFLU,FEDC8/FLOAT(NX))
      CALL COVAL(8,U1,ONLYMS,0.5*OMEGA*(RADINC**2))
      CALL COVAL(8,W1,ONLYMS,W1INC)
      CALL COVAL(8,KE,ONLYMS,.01*VELSQ)
      CALL COVAL(8,EP,ONLYMS,0.16433*(.01*VELSQ)**1.5/(.1*GINC1/12.))
      CALL COVAL(8,H1,ONLYMS,H1INC)
      CALL COVAL(8,C1,ONLYMS,0.0)
C
CC *** 'HOT' H2 & H2O INLET  ****
CC
CC *** FEEDH1 IS SET TO THE TOTAL MASS FLOWRATE (LBM/S)
CC    AT THE HOT  INLET, (SEE LMSC-HREC TR D697954),
CC    THEN CONVERTED TO SLUGS/SEC OVER SOLUTION SEGMENT
      FEEDH = FEEDH1/(G*SLICES)
CC *** H1INH1 IS SET TO THE ENTHALPY (BTU/LBM) AT THE HOT
CC     INLET (TR D697954) THEN CONVERTED TO FT-LBF/SLUGS
      H1INH = H1INH1*778.16*G
CC *** ROINH IS THE DENSITY (SLUG/CU FT) AT THE HOT INLET
      ROINH = ROINH1/G
CC *** RADINH IS THE AVERAGE RADIUS (FT) OF THE  HOT INLET
      RADINH = RINNER + 2.45/12.
CC *** AINH1 IS SET TO THE AREA (SQ IN) AT THE HOT INLET
CC     THEN CONVERTED TO THE INLET AREA (SQ FT) PER SEGMENT
      AINH = AINH1/(144.0*SLICES)
CC *** CALCULATE THE FEED VELOCITY AT THE  HOT INLET
      W1INH = -FEEDH/(ROINH*AINH)
CC *** TOTAL NOMINAL GRID AREA PER SEGMENT AT HOT INLET
      ARGRID=ARGRD1/(144.*SLICES)
C
      CALL PLACE(9,HIGH,1,NX,32,40,28,28)
      CALL COVAL(9,M1,FIXFLU,FEEDH/ARGRID)
      CALL COVAL(9,W1,ONLYMS,W1INH)
C
CC *** INITIALIZE ENTHALPY,TURBULENCE, AND DISSIPATION
      FIINIT(H1)=3.E7
      FIINIT(KE)=.01*((OMEGA*RADINH)**2+W1INH**2)
      FIINIT(EP)=.16433*(FIINIT(KE))**1.5/(.1*AINH/(RADINH*XULAST))
C
```

```
      VELSQ=W1INH**2
      CALL COVAL(9,KE,ONLYMS,.01*VELSQ)
      CALL COVAL(9,EP,ONLYMS,FIINIT(EP)*(.01*VELSQ/FIINIT(KE))**1.5)
      CALL COVAL(9,H1,ONLYMS,H1INH)
      CALL COVAL(9,C1,ONLYMS,H2OINH)
C
CC ***    OUTLETS   ***
CC
CC *** THE EXIT PRESSURES AROUND THE PERIPHERY OF THE
CC      AFT-PLATFORM SEAL ARE SPECIFIED IN SATELLITE, BUT
CC      ARE APPLIED AS A BOUNDARY CONDITION IN GROUND.
CC *** BELOW THE VALUES OF U1,KE,...AT THE EXIT ARE CALCULATED
CC      BEFORE THEY ARE TRANSFERED TO GROUND. THESE VALUES MUST
CC      BE SPECIFIED IN CASE THERE IS IN-FLOW AT THE PERIPHERY
CC      OF THE SEAL (EITHER TEMPORARY OR STEADY), OR NEAR BOLTS.
CC *** CALCULATE THE RADIUS AND AREA AT THE PRIMARY EXIT
      RADXIT = RINNER + YVLAST
      AEXIT=(GEXIT1/12.)*RADXIT*XULAST
CC *** ESTIMATE THE VELOCITIES AT PRIMARY EXIT
      FEEDT=FEEDC + FEEDH
      W1EXIT=-FEEDT/(ROINH*AEXIT)
      U1EXIT=OMEGA*RADXIT**2
C
      VELSQ=W1EXIT**2+(U1EXIT/RADXIT)**2
      VALKE=.01*VELSQ
      VALEP=.16433*VALKE**1.5/(.1*GEXIT1/12.)
      HEXIT=HEXIT1*778.16*G
CC *** INITIALIZE PRESSURE (PEXIT+HALF EXPECTED LOSS AT PRIMARY EXIT)
      FIINIT(P1)= PEXITA+0.5*GLOSK1*ROINH*W1EXIT**2/2.
C
C***
CC ***  ROTATING WALL AND WALL FRICTION  ***
CC
CC  ALL ROTATION AND WALL FRICTION EFFECTS SET UP IN GROUND CH. 5
C
C-------------------    --------------    ------------------
C---  GROUP 25 GROUND STATION :
C     GROSTA<.F.>,NAMLST<.F.>
C    *NAMLST ACTIVATES NAMELIST IN GROUND.
      GROSTA=.TRUE.
C-------------------    --------------    ------------------
C---  GROUP 26 SOLUTION TYPE AND RELATED PARAMETERS :
C     WHOLEP<.F.>,SUBPST<.F.>,DONACC<.F.>
      WHOLEP= .TRUE.
C-------------------    --------------    ------------------
C---  GROUP 27 SWEEP'AND ITERATION NUMBERS :
C     FSWEEP<1>,LSWEEP<1>,LITHYD<1>,LITC<1>,LITKE<1>,LITH<1>,
C     LITER(1-25)<9*1,-1,15*1>
C     IVELF<1>,NVEL<1>,IVELL<10000>,
C     IKEF<1>,NKE<1>,IKEL<10000>,
C     IENTF<1>,NENT<1>,IENTL<10000>,
C     ICNCF<1>,NCNC<1>,ICNCL<10000>,
C     IRHO1F<1>,NRHO1<1>,IRHO1L<10000>,
C     IRHO2F<1>,NRHO2<1>,IRHO2L<10000>
      LSWEEP= 200
      LITER(PP)= 15
C-------------------    --------------    ------------------
C---  GROUP 28 TERMINATION CRITERIA :
C     ENDIT(1-25)<9*1.E-10,0.5,15*1.E-10>
C-------------------    --------------    ------------------
```

```
C---  GROUP 29 RELAXATION
C     RLXP<1.>,RLXPXY<1.>,RLXPZ<1.>,RLXRHO<1.>,RLXMDT<1.>,
C     DTFALS(3-25)<23*1.E10>
      U1MAX=ABS(U1EXIT)
      W1MAX=ABS(W1INH)
      V1MAX=W1MAX
      DTFALS(V1)=YVLAST/(FLOAT(NY)*V1MAX+TINY)
      DTFALS(W1)=ZWLAST/(FLOAT(NZ)*W1MAX+TINY)
      DTFALS(KE)=1.E5*AMAX1(DTFALS(V1),DTFALS(W1))
      DTFALS(EP)=DTFALS(KE)
C------------------   --------------   ------------------
C---  GROUP 30 LIMITS :
C     VELMAX<1.E10>,VELMIN<-1.E10>,RHOMAX<1.E10>,RHOMIN<1.E-10>,
C     TKEMAX<1.E10>,TKEMIN<1.E-10>,EMUMAX<1.E10>,EMUMIN<1.E-10>,
C     EPSMAX<1.E10>,EPSMIN<1.E-10>,AMDTMX<1.E10>,AMDTMN<-1.E10>
      EMUMIN=100.*0.5E-5/G
      EMUMAX=1.E4*1.2E-3/G
      EPSMAX=1.E20
C------------------   --------------   ------------------
C---  GROUP 31 SLOWING DEVICES : SLORHO<1.>,SLOEMU<1.>
C------------------   --------------   ------------------
C---  GROUP 32 PRINT-OUT OF VARIABLES :
C     PRINT(1-25)<.T.,.F.,23*.T.>,SUBWGR<.F.>
      PRINT(2)=.TRUE.
C------------------   --------------   ------------------
C---  GROUP 33 MONITOR PRINT-OUT :
C     IXMON<1>,IYMON<1>,IZMON<1>,NPRMON<1>,NPRMNT<1>
      IZMON=9
      IYMON=13
      IXMON=1
C------------------   --------------   ------------------
C---  GROUP 34 FIELD PRINT-OUT CONTROL :
C     NPRINT<100>,NTPRIN<100>,NXPRIN<1>,NYPRIN<1>,NZPRIN<1>,
C     IZPRF<1>,ISTPRF<1>,IZPRL<10000>,ISTPRL<10000>
C     NUMCLS<10>,KOUTPT
      NPRINT = LSWEEP
C------------------   --------------   ------------------
C---  GROUP 35 TABLE CONTROL :
C     TABLES<.F.>,NTABLE,NTABVR,LINTAB,NPRTAB,NMON,
C     ITAB(1-8),MTABVR(1-8)
C------------------   --------------   ------------------
C     GROUP 36-38 ARE NOT DOCUMENTED IN THE INSTRUCTION
C     MANUAL AND ARE INTENDED FOR MAINTENANCE PURPOSES ONLY
C---  GROUP 36 DEBUG PRINT-OUT SLAB AND TIME-STEP :
C     IZPR1<1>,IZPR2<1>,ISTPR1<1>,ISTPR2<1>
C------------------   --------------   ------------------
C---  GROUP 37 DEBUG SWEEP AND SUBROUTINES :
C     KEMU,KMAIN,KINDEX,KGEOM,KINPUT,KSODAT,KCOMPF,KSORCE,
C     KSOLV1,KSOLV2,KSOLV3,KCOMPP,KADJST,KFLUX,KSHIFT,KDIF,
C     KCOMPU,KCOMPV,KCOMPW,KCOMPR,KWALL,KDBRHO<-1>,KDBEXP,KDBMDT
C     KDBGEN
C------------------   --------------   ------------------
C---  GROUP 38 MONITOR,TEST,AND FLAG :
C     MONITR<.F.>,FLAG<.F.>,TEST<.T.>,KFLAG<1>
C     END OF MAINTENANCE-ONLY SECTION
C------------------   --------------   ------------------
C---  GROUP 39 ERROR AND RESIDUAL PRINT-OUT :
C     IERRP<1000>,RESREF(1,3-24)<25*1.>,RESMAP<.F.>,
C     RESID(1-25)<2*.F.,23*.T.>,KOUTPT
      IERRP=25
```

```
      RESREF(P1)=FEEDT/ROINH
      RESREF(U1)=FEEDT*U1MAX
      RESREF(V1)=FEEDT*V1MAX
      RESREF(W1)=FEEDT*W1MAX
      RESREF(KE)=FEEDT*FIINIT(KE)
      RESREF(EP)=FEEDT*FIINIT(EP)
      RESREF(H1)=FEEDT*FIINIT(H1)
      RESREF(C1)=FEEDT*H2OINH
C------------------    --------------    ------------------
C---  GROUP 40 SPECIAL DATA : LOGIC(1..10),INTGR(1..10),RE(21..30),
C     NLSP<1>,NISP<1>,NRSP<1>,SPDATA<.F.>,LSPDA(1),ISPDA(1),RSPDA(1)
C     USE FIRST 10 ELEMENTS OF ARRAYS LOGIC & INTGR AND 21ST
C     TO 30TH OF ARRAY RE FOR TRANSFERRING SPECIAL DATA FROM
C     SATELLITE TO GROUND. BUT IF REQUIREMENTS EXCEED THIS
C     PROVISION SET SPDATA = .T., AND DIMENSION ARRAYS LSPDA,
C     ISPDA, RSPDA ABOVE AND IN GROUND AS NEEDED, AND SET HERE
C     NLSP, NISP, NRSP TO DIMENSION VALUES.
C
CC ** PASS THE FOLLOWING  INPUT GEOMETRIES, PROPERTIES, AND BOUNDARY
CC    CONDITIONS TO GROUND VIA RSPDA (FOR PRINTING ETC.)
      SPDATA=.TRUE.
      RSPDA(1)=GINC1
      RSPDA(2)=GEXIT1
      RSPDA(3)=AINH1
      RSPDA(4)=RINNER
      RSPDA(5)=H1INC1
      RSPDA(6)=H1INH1
      RSPDA(7)=ROINC1
      RSPDA(8)=ROINH1
      RSPDA(9)=ECCENT
      RSPDA(10)=RPM
      RSPDA(11)=FEEDC1
      RSPDA(12)=FEEDH1
      RSPDA(13)=H2OINH
      RSPDA(14)=W1INC
      RSPDA(15)=W1INH
C *   VARIABLES FOR EXIT AT O.D. OF AFT-PLATFORM SEAL
      RSPDA(16)=GLOSK1
C NB. PEXIT(8) ARRAY IS EQUIVALENCED TO RSPDA(17)
      RSPDA(25)=VALKE
      RSPDA(26)=VALEP
      RSPDA(27)=HEXIT
      RSPDA(28)=SLICES
      RSPDA(29)=H2OXIT
C NB. GEXIT(8) ARRAY IS EQUIVALENCED TO RSPDA(30)
C------------------    --------------    ------------------
C---  GROUP 42 RESTARTS AND DUMPS : SAVEM<.F.>,RESTRT<.F.>,KINPUT
      SAVEM = .TRUE.
      RESTRT= .TRUE.
C------------------    --------------    ------------------
C---  GROUP 43 GRAFFIC :
C     GRAPHS<.F.>,ORTHOG<.T.>,ANTSYM,NPRT<1>,ITITL<5*4H****>
C---  FOR A GRAFFIC RUN, DIMENSION PHI1 & PHI2 AS FOLLOWS:
C     PHI1(NX*NY*NZ*NM)
C     PHI2((NX+2)*(NY+2)*(NZ+2)*(NM+IBLK)) , WHERE
C     NM=NO. OF VARIABLES STORED + DENSITY(-IES)
C     IBLK=0 IF BLOCK=.FALSE.,=4 IF A 3D RUN,
C     =3 IF A 2D.YZ RUN.
      GRAPHS = .TRUE.
C------------------    --------------    ------------------
```

```
            IF(IRUN.EQ.1) GO TO 900
C---  RUN2
            IF(IRUN.EQ.2) GO TO 900
C---  RUN3
            IF(IRUN.EQ.3) GO TO 900
C---  RUN4
            IF(IRUN.EQ.4) GO TO 900
C---  RUN5
            IF(IRUN.EQ.5) GO TO 900
C---  RUN6
            IF(IRUN.EQ.6) GO TO 900
C---  RUN7
            IF(IRUN.EQ.7) GO TO 900
C---  RUN8
            IF(IRUN.EQ.8) GO TO 900
C---  RUN9
            IF(IRUN.EQ.9) GO TO 900
C---  RUN10
            IF(IRUN.EQ.10) GO TO 900
C---  RUN11
            IF(IRUN.EQ.11) GO TO 900
C---  RUN12
            IF(IRUN.EQ.12) GO TO 900
C---  RUN13
            IF(IRUN.EQ.13) GO TO 900
C---  RUN14
            IF(IRUN.EQ.14) GO TO 900
C---  RUN15
            IF(IRUN.EQ.15) GO TO 900
C---  RUN16
            IF(IRUN.EQ.16) GO TO 900
C---  RUN17
            IF(IRUN.EQ.17) GO TO 900
C---  RUN18
            IF(IRUN.EQ.18) GO TO 900
C---  RUN19
            IF(IRUN.EQ.19) GO TO 900
C---  RUN20
            IF(IRUN.EQ.20) GO TO 900
C---  RUN21
            IF(IRUN.EQ.21) GO TO 900
C---  RUN22
            IF(IRUN.EQ.22) GO TO 900
C---  RUN23
            IF(IRUN.EQ.23) GO TO 900
C---  RUN24
            IF(IRUN.EQ.24) GO TO 900
C---  RUN25
            IF(IRUN.EQ.25) GO TO 900
C---  RUN26
            IF(IRUN.EQ.26) GO TO 900
C---  RUN27
            IF(IRUN.EQ.27) GO TO 900
C---  RUN28
            IF(IRUN.EQ.28) GO TO 900
C---  RUN29
            IF(IRUN.EQ.29) GO TO 900
C---  RUN30
            IF(IRUN.EQ.30) GO TO 900
  900 CONTINUE
```

```
C---  ALL RUNS
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 3 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 4 STARTS:
C------------------------------------------------------------
C  WRITE GENERAL DATA ON TO THE GUSIE1.DTA TAPE, ETC...
          IF(SPDATA) CALL WRTSPC(LSPDA,NLSP,ISPDA,NISP,RSPDA,NRSP)
          IF(BLOCK) CALL WRTPOR(PE,PN,PH,PC,NX,NY,NZ,IPLANE)
C     OLD PRACTICES RETAINED FOR REFERENCE:
C         IF(SPDATA) CALL SPCDAT(IRUN)
C         IF(BLOCK) CALL PORDAT(IRUN)
          IF(GRAPHS) CALL SORT(IRUN)
          IF(RESTRT) GO TO 902
        DO 901 INDVAR=1,25
          IF(IFIX(FIINIT(INDVAR)+0.1).NE.10101) GO TO 901
      CALL FLDDAT(IRUN)
      GO TO 902
  901 CONTINUE
  902 CALL DATAIO(WRT,10)
          IF(MONITR) CALL DATAIO(WRT,-6)
 999  CONTINUE
      STOP
      END
P!

@BRKPT PRINT$
```

ORIGINAL PAGE IS
OF POOR QUALITY

RUNKLEBIN197*TPF$(O).SLPRT

```
$BATCH
C$DIRECTIVE**MAIN
C ***
C    *FILE NAME: DSK32GRD.FTN
C ***
C    *ABSTRACT: GROUND STATION FOR SSME HPFTP APS 3-D MODEL (2 EXITS)
C ***
C    *INCLUDED SUBROUTINES: THE MODELS OF MAIN, GROUND .
C    *DOCUMENTATION: PHOENICS INSTRUCTION MANUAL (SPRING 1983).
C    *SATELLITE FILE NAME: DSKSAT.FTN
      COMMON/ISHIFT/III(57),NFMAX
C SET F-ARRAY DIMENSION AS NEEDED, & SET NFMAX ACCORDINGLY.
      COMMON F(324000)
      NFMAX=324000
      CALL MAIN1
      STOP
      END
C$DIRECTIVE**GROUND
      SUBROUTINE GROUND(IRN,ICHAP,ISTP,ISWP,IZED,INDVAR)
$INCLUDE 9,CMNGUSSI.FTN/G
$INCLUDE 9,GUSSEQUI.FTN/G
C$INCLUDE 9,NMLIST.FTN/G
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 STARTS:
C-----------------------------------------------------------------
C+++++MEANING OF SUBROUTINE ARGUMENTS:
C     IRN=RUN NUMBER                  ICHAP=CHAPTER CALLED              ISTP=TIME STEP
C     ISWP=SOLUTION SWEEP             IZED=Z-SLAB                       INDVAR: SEE CHAPTERS BELOW.
C+++++USER-INTRODUCED VARIABLES & ARRAYS:
C     TO AVOID CONFLICT WITH VARIABLE NAMES USED IN COMMON, ALL
C     VARIABLES INTRODUCED BY THE USER SHOULD HAVE NAMES STARTING
C     WITH 'G' IF REAL, 'J' IF INTEGER, AND 'G' OR 'J' IF LOGICAL.
C     THUS GDZ(IZ) MIGHT BE A Z-INTERVAL ARRAY
C     GW1(IY,IX) A 2-D ARRAY FOR AXIAL VELOCITY                         ETC.
C     USER-GENERATED SUBROUTINES SHOULD BE NAMED CORRESPONDINGLY,EG
C     SUBROUTINE GVISC(GTEMP,GCNC,GVSC), FOR COMPUTING VISCOSITY
C     FROM CONCENTRATION & TEMPERATURE.
C+++++GROUND-TO-EARTH CONNECTING SUBROUTINES:
C    *USE GET(NAME,GARRAY,NY,NX) TO PUT VALUES OF VARIABLE NAMED
C     'NAME' INTO ARRAY 'GARRAY' DIMENSIONED GARRAY(NY,NX).
C    *USE SET(NAME,IXF,IXL,IYF,IYL,GARRAY,NY,NX) TO SET VARIABLE
C     'NAME' TO GARRAY(IY,IX) OVER THE REGION: IXF-IXL & IYF-IYL.
C    *USE PRNSLB(NAME) TO PRINT VARIABLE 'NAME' OVER X-Y PLANE.
C    *USE ADD(NAME,IXF,IXL,IYF,IYL,TYPE,CM,VM,CVAR,VVAR,NY,NX)
C     TO ADD SOURCE TO VARIABLE NAMED 'NAME' (SEE CHAPTER 5).
C    *USE READIZ(IZED) IN CHAPTERS 1, 2, 8, & 9 TO ACCESS P1,...DM
C     & VOL,...AHDZ. (SEE FOOTNOTE TO LEGALITY TABLE)
C    *USE GET1D(NAME,GARRAY,NDIM) TO PUT VARIABLE NAMED 'NAME' IN
C     ONE-D ARRAY 'GARRAY' DIMENSIONED NDIM, THUS:
C     CALL GET1D(NAME,GNX,NX) FOR XG,...DXG & DIMENSION GNX(NX)
C     CALL GET1D(NAME,GNY,NY) FOR YG,...RV & DIMENSION GNY(NY)
C     CALL GET1D(NAME,GNZ,NZ) FOR ZG,...WGRID & DIMENSION GNZ(NZ).
C+++++LEGALITY TABLE FOR USE OF EARTH-CONNECTING SUBROUTINES:
C     ENTRIES IN TABLE GIVE CHAPTERS IN WHICH SUBROUTINES CAN BE
C     USED FOR VARIABLES IN LEFT-HAND COLUMN. (SUBROUTINE
C     STRIDE IS REGARDED AS BEING IN CHAPTER 3)
C     ---------------------------------------------------------
C     : VARIABLE::  GET &   :  SET  : ADD    : READIZ : GET1D :
C     :         ::   PRNSLB :       :        :        :       :
C     ---------------------------------------------------------
```

```
C      :P1  - RZ   ::     ALL    :  6 & 7 :    5     : 1,2,8,9:  NONE :
C      :P10 - RZH::3-7, 10-16:     3    :  NONE    :  NONE   :  NONE :
C      :VOL -AHDZ::     ALL    :     3    :  NONE    : 1,2,8,9:  NONE :
C      :D1DP      ::     NONE   :    10    :  NONE    :  NONE   :  NONE :
C      :D2DP      ::     NONE   :    11    :  NONE    :  NONE   :  NONE :
C      :MU1,MU1H ::  5,13-16  :    12    :  NONE    :  NONE   :  NONE :
C      :EXCO(L,H)::     NONE   :    13    :  NONE    :  NONE   :  NONE :
C      :CFP       ::      5     :    14    :  NONE    :  NONE   :  NONE :
C      :MDT       ::      5     :    15    :  NONE    :  NONE   :  NONE :
C      :HST1,HST2::   5 & 15   :    16    :  NONE    :  NONE   :  NONE :
C      :XG -WGRID::     NONE   :  NONE    :  NONE    :  NONE   :  ALL  :
C      ---------------------------------------------------------------
C      NOTES ON ABOVE TABLE:
C     *IN CHAPTERS 1, 2, 8, & 9 VARIABLES P1...DM & GEOMETRY
C      VOL...AHDZ CAN BE ACCESSED BUT ONLY IN CONJUNCTION WITH
C      USE OF READIZ, THUS:
C      DO 1 IZED=1,NZ
C      CALL READIZ(IZED)
C    1 CALL GET(... AS REQUIRED..)
C     *GEOMETRY ACCESSED BY READIZ IS THAT AT INITIAL TIME
C     *D1DP & D2DP ONLY ACCESSIBLE IN UNSTEADY FLOWS.
C+++++GROUND SERVICE SUBROUTINES:
C     *USE CONTUR(NAME,IPLANE,ILOC,NINT,I1,I2,J1,J2,GARRAY,NDIM) FOR
C      LINE-PRINTER PLOTS OF CONTOURS. 'NAME' = U1,...C4
C      'IPLANE'= XPLANE, YPLANE, OR ZPLANE                          ILOC SETS IX, IY, OR
C      IZ LOCATION OF IPLANE              I1, I2, J1, & J2 SET FIRST & LAST
C      CELLS IN HORIZ. & VERT. ON PLOT  GARRAY IS 1-D WORKING ARRAY
C      OF DIMENSION NX*NY, NX*NZ, OR NY*NZ DICTATED BY IPLANE            &
C      NDIM SETS VALUE OF DIMENSION OF GARRAY.
C     *USE FLD2DA(TITLE,GARRAY,NY,NX) TO PRINT ANY ARRAY DIMENSIONED
C      GARRAY(NY,NX)  SET 'TITLE' TO REQUIRED NAME ( 4 HOLLERITH
C      CHARACTERS ONLY).
C     *USE FLD3DA(TITLE,GARRAY,NX,NY,NZ,IPLANE,ILOC) TO PRINT ANY
C      ARRAY DIMENSIONED GARRAY(NX,NY,NZ) IN PLANE SPECIFIED BY
C      'IPLANE' & 'ILOC' AS FOR CONTUR ABOVE                          SET 'TITLE' AS FOR
C      FLD2DA.
C      VARIABLE NAMES FOR USE IN GROUND:
       COMMON/TYPE/CELL,EAST,WEST,NORTH,SOUTH,HIGH,LOW,VOLUME,WALL
       COMMON/VAR/P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,
      &KE,EP,H1,H2,H3,C1,C2,C3,C4,RX,RY,RZ,S1,S2
       COMMON/VAROLD/P10,PPO,U10,U20,V10,V20,W10,W20,R10,R20,RSO,
      &KEO,EPO,H10,H20,H30,C10,C20,C30,C40,RXO,RYO,RZO,S10,S20
       COMMON/VARLOW/P1L,PPL,U1L,U2L,V1L,V2L,W1L,W2L,R1L,R2L,RSL,
      &KEL,EPL,H1L,H2L,H3L,C1L,C2L,C3L,C4L,RXL,RYL,RZL,S1L,S2L
       COMMON/VARHI/P1H,PPH,U1H,U2H,V1H,V2H,W1H,W2H,R1H,R2H,RSH,
      &KEH,EPH,H1H,H2H,H3H,C1H,C2H,C3H,C4H,RXH,RYH,RZH,S1H,S2H
       COMMON/GMTRY/VOL,VOLO,AEAST,ANORTH,AHIGH,AEDX,ANDY,AHDZ
       COMMON/PROP/D1,D2,D1DP,D2DP,MU1,MU1LAM,EXCO,CFP,MDT,HST1,HST2
       COMMON/PRPOLD/D10,D20
       COMMON/PRPLOW/D1L,D2L,EXCOL
       COMMON/PRPHI/D1H,D2H,MU1H,EXCOH
       COMMON/VARNX/XG,XU,DXU,DXG
       COMMON/VARNY/YG,YV,DYV,DYG,R,RV
       COMMON/VARNZ/ZG,ZW1,DZW,DZG,WGRID
       COMMON/GDMSCI/XPLANE,YPLANE,ZPLANE,ITNO
       COMMON/GDMSCL/LSLAB,MSLAB,HSLAB,LAMMU
       REAL NORTH,LOW
       INTEGER    P1,PP,U1,U2,V1,V2,W1,W2,R1,R2,RS,
      &EP,H1,H2,H3,C1,C2,C3,C4,RX,RY,RZ,S1,S2
       INTEGER    P10,PPO,U10,U20,V10,V20,W10,W20,R10,R20,RSO,
```

```
     &EPO,H1O,H2O,H3O,C1O,C2O,C3O,C4O,RXO,RYO,RZO,S1O,S2O
      INTEGER    P1L,PPL,U1L,U2L,V1L,V2L,W1L,W2L,R1L,R2L,RSL,
     &EPL,H1L,H2L,H3L,C1L,C2L,C3L,C4L,RXL,RYL,RZL,S1L,S2L
      INTEGER    P1H,PPH,U1H,U2H,V1H,V2H,W1H,W2H,R1H,R2H,RSH,
     &EPH,H1H,H2H,H3H,C1H,C2H,C3H,C4H,RXH,RYH,RZH,S1H,S2H
      INTEGER VOL,VOLO,AEAST,ANORTH,AHIGH,AEDX,ANDY,AHDZ
      INTEGER D1,D1DP,D2,D2DP,EXCO,CFP,HST1,HST2
      INTEGER D1O,D2O,D1L,D2L,EXCOL,D1H,D2H,EXCOH
      INTEGER XG,XU,DXU,DXG,YG,YV,DYV,DYG,R,RV,ZG,ZW1,DZW,
     &DZG,WGRID
      INTEGER XPLANE,YPLANE,ZPLANE
      LOGICAL LSLAB,MSLAB,HSLAB,LAMMU,LSPDA
      EQUIVALENCE (M1,R1),(M2,R2)
C     SATLIT-EQUIVALENT IRUN:
      EQUIVALENCE (IRUN,INTGR(11))
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 1 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 STARTS:
C     ARRAYS ( DIMENSIONED NY,NX ) FOR USE WITH 'ADD':
C ***
      DIMENSION CVAR(40,8),VVAR(40,8),CM(40,8),VM(40,8),ZERO(40,8)
C ***
C     SPECIAL-DATA ARRAYS DIMENSIONED & DIMENSION VALUES SET HERE:
C ***
      DIMENSION LSPDA(1),ISPDA(1),RSPDA(37)
C ***
C     USER PLACES HIS VARIABLES, ARRAYS, EQUIVALENCES ETC. HERE.
C ***
      COMMON/WALLG/ GU1(40,8),GV1(40,8),GW1(40,8),GD1(40,8),GMU1L(40,8),
     . GCVAR(40,8),GVVAR(40,8),GDXU(8),GDYV(40),GR(40),GDZW(28)
C
C
      DIMENSION GH1(40,8),GC1(40,8),GPT(40,8),GW1L(40,8),GRH2O(40,8),
     . GRH2(40,8),GT1(40,8),GMU1(40,8),GT1M(40,8),GAHIGH(40,8),
     . CM1S(8),CM2S(8)
C
C     SET UP EXIT PRESSURE AND GAP DATA (TRANSFERRED FROM SATELLITE)
      DIMENSION GPEXIT(8),GGEXIT(8)
      EQUIVALENCE(RSPDA(17),GPEXIT(1)),(RSPDA(30),GGEXIT(1))
C
      INTEGER T1,T1H
      EQUIVALENCE (T1,C2),(T1H,C2H)
C
      EQUIVALENCE (GCVAR(1,1),CVAR(1,1)),(GVVAR(1,1),VVAR(1,1)),
     . (GAHIGH(1,1),ZERO(1,1))
C
      LOGICAL GVELUW,GVELVW,GVELUV,GKEEP
C ***
      DATA NLSP,NISP,NRSP/1,1,37/
      DATA CVAR,VVAR,CM,VM,ZERO/1600*0.0/
C ***
C     USER PLACES HIS DATA STATEMENTS HERE.
C ***
      DATA GH1,GC1,GPT,GW1L,GRH2O,GRH2,GT1,GT1M,GMU1/2880*0.0/
      DATA GU1,GV1,GW1,GD1,GMU1L/1600*0.0/
      DATA CM1S,CM2S,EMOUT1,EMOUT2/16*0.0,  2*0.0/
C
      DATA JMU1,JRH2O,JRH2,JT1M,JPT/21,15,16,18,19/
      DATA GPI,G,FIXVAL,GBLADE/3.1416, 32.174,  1.E10,  58.0/
      DATA CMRLX1,CMRLX2,GTINY/0.25,0.1,  1.E-10/
C ***
```

```
C
C **  DATA FOR SECOND EXIT
C     LOSS COEFFICIENT              TOTAL EXIT AREA (SQ IN)          FIRST,LAST IX-LOCATION
C     NB. MAKE SURE EXIT AREA IS PERTINENT TO CHOSEN CALCULATION DOMAIN
      DATA GLOSK2,GAXIT2,JIXE2F,JIXE2L/1.5,0.0,  0.0/
C
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 1 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 STARTS:
C     PLEASE DO NOT ALTER, OR RE-SET, ANY OF THE REMAINING
C     STATEMENTS OF THIS SECTION.
          IF(SPDATA)
     &CALL RDSPC(IRN,INTGR(12),LSPDA,NLSP,ISPDA,NISP,RSPDA,NRSP)
      CALL GRDUTY(IRN,ICHAP,IZED,INDVAR)
          IF(ICHAP.EQ.-5) GO TO 10
          IF(ICHAP.LE.0.OR.ICHAP.GT.16) RETURN
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,1200,
     &1300,1400,1500,1600),ICHAP
      RETURN
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX STANDARD SECTION 2 ENDS.
CXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX USER SECTION 2 STARTS:
C-----------------------------------------------------------------
C     CHAPTER 0: MODIFY SATLIT DATA, AT START OF EACH IRN.
C-----------------------------------------------------------------
   10 CONTINUE
C         IF(.NOT.NAMLST) RETURN
C         IF(IRN.EQ.NRUN) DATFIL=.FALSE.
C---  READ SATLIT DATA NAMELIST HERE
C     CALL WRIT40(40HENTER NAMELIST DATA FOR GROUPS 1 TO 24   )
C     READ(20,G1G24)
C     CALL WRIT40(40HENTER NAMELIST DATA FOR GROUPS 25 TO 42 )
C     READ(20,G25G42)
C
CC ** SUMMARY PRINTOUT OF INPUT DATA
      WRITE(6,21)
      WRITE(6,22) RSPDA(10),RSPDA(1),RSPDA(9),RSPDA(3),RSPDA(2),
     $RSPDA(16)
      WRITE(6,23) RSPDA(5),RSPDA(6),RSPDA(27)/(778.16*G),RSPDA(11),
     $RSPDA(12)
      WRITE(6,24) RSPDA(7),RSPDA(8),RSPDA(14),RSPDA(15),
     $RSPDA(13),RSPDA(29)
      WRITE(6,28) RSPDA(37),RSPDA(36),RSPDA(35),RSPDA(34),RSPDA(33),
     $ RSPDA(32),RSPDA(31),RSPDA(30)
      WRITE(6,25) RSPDA(24),RSPDA(23),RSPDA(22),RSPDA(21)
      WRITE(6,26) RSPDA(20),RSPDA(19),RSPDA(18),RSPDA(17)
      WRITE(6,27) GLOSK2,GAXIT2,JIXE2F,JIXE2L
C
 21   FORMAT(/////25X,21HSUMMARY OF INPUT DATA,/25X,21(1H-))
 22   FORMAT(
     $/1X,1PE12.3,2X,36HROTATIONAL SPEED OF THE DISC, (RPM).,
     $/1X,1PE12.3,2X,41HGAP SIZE AT THE LABYRINTH SEAL, (INCHES).,
     $/1X,1PE12.3,2X,46HECCENTRICITY IN THE 11:30 DIRECTION, (INCHES).,
     $/1X,1PE12.3,2X,70HTOTAL FLOW AREA (OVER 360 DEG) BETWEEN TURBINE B
     $LADE SHANKS, (SQ INS).,
     $/1X,1PE12.3,2X,85H(AVERAGE) CLEARANCE BETWEEN AFT-PLATFORM SEAL OD
     $ AND THE TURBINE BLADE LIP, (INCHES).,
     $/1X,1PE12.3,2X,64HLOSS COEFFICIENT FOR ADDITIONAL LOSSES AT EXIT N
     $EAR BLADE ROOTS.)
 23   FORMAT(
     $/1X,1PE12.3,2X,53HENTHALPY OF H2 ENTERING AT LABYRINTH SEAL, (BTU/
     $LBM).,
```

```
     $/1X,E12.3,2X,78HENTHALPY OF H2 + H2O MIXTURE ENTERING BETWEEN TURB
     $INE BLADE SHANKS, (BTU/LBM).,
     $/1X,E12.3,2X,40HENTHALPY OF TURBINE DISCHARGE (BTU/LBM).,
     $/1X,1PE12.3,2X,74HTOTAL MASS FLOWRATE OF THE H2 ENTERING THROUGH L
     $ABYRINTH SEAL (LBM/CU FT).,
     $/1X,1PE12.3,2X,83HTOTAL MASS FLOWRATE OF H2 + H2O MIXTURE ENTERING
     $ BETWEEN BLADE SHANKS, (LBM/CU FT).)
 24   FORMAT(
     $/1X,1PE12.3,2X,63HDENSITY OF THE H2 ENTERING THROUGH LABYRINTH SEA
     $L, (LBM/CU FT).,
     $/1X,1PE12.3,2X,75HDENSITY OF THE H2 + H2O MIXTURE ENTERING BETWEEN
     $ BLADE SHANKS, (LBM/CU FT).,
     $/1X,1PE12.3,2X,68HCALCULATED INLET VELOCITY OF THE H2 AT THE LABYR
     $INTH SEAL, (FT/SEC).,
     $/1X,1PE12.3,2X,90HCALCULATED INLET VELOCITY OF THE H2 + H2O MIXTUR
     $E ENTERING BETWEEN BLADE SHANKS, (FT/SEC).,
     $/1X,1PE12.3,2X,51HMASS FRACTION OF H2O ENTERING BETWEEN BLADE SHAN
     $KS.,
     $/1X,1PE12.3,2X,36HMASS FRACTON OF H2O EXITING TURBINE.)
C
 28   FORMAT(
     $/1X,1PE12.3,2X,33HEXIT GAP CLEARANCE (FEET) AT 1:00,
     $/1X,1PE12.3,2X,33HEXIT GAP CLEARANCE (FEET) AT 2:30,
     $/1X,1PE12.3,2X,33HEXIT GAP CLEARANCE (FEET) AT 4:00,
     $/1X,1PE12.3,2X,33HEXIT GAP CLEARANCE (FEET) AT 5:30,
     $/1X,1PE12.3,2X,33HEXIT GAP CLEARANCE (FEET) AT 7:00,
     $/1X,1PE12.3,2X,33HEXIT GAP CLEARANCE (FEET) AT 8:30,
     $/1X,1PE12.3,2X,34HEXIT GAP CLEARANCE (FEET) AT 10:00,
     $/1X,1PE12.3,2X,34HEXIT GAP CLEARANCE (FEET) AT 11:30 )
  25  FORMAT(
     $/1X,1PE12.3,2X,27HEXIT PRESSURE (PSF) AT 1:00,
     $/1X,1PE12.3,2X,27HEXIT PRESSURE (PSF) AT 2:30,
     $/1X,1PE12.3,2X,27HEXIT PRESSURE (PSF) AT 4:00,
     $/1X,1PE12.3,2X,27HEXIT PRESSURE (PSF) AT 5:30)
C
  26  FORMAT(
     $ 1X,1PE12.3,2X,27HEXIT PRESSURE (PSF) AT 7:00,
     $/1X,1PE12.3,2X,27HEXIT PRESSURE (PSF) AT 8:30,
     $/1X,1PE12.3,2X,28HEXIT PRESSURE (PSF) AT 10:00,
     $/1X,1PE12.3,2X,28HEXIT PRESSURE (PSF) AT 11:30 )
C
   27 FORMAT(
     $/1X,1PE12.3,2X,32HLOSS COEFFICIENT AT SECOND EXIT.,
     $/1X,1PE12.3,2X,41HTOTAL FLOW AREA AT SECOND EXIT, (SQ INS).,
     $/5X,I2,4H TO ,I2,2X,40HIX-CELLS OVER WHICH SECOND EXIT LOCATED.,
     $/////)
C
C ***
C
C CALCULATE ANY REQUIRED QUANTITIES FOR USE IN CH.5
      GRPM=RSPDA(10)
      GOMEGA=GRPM*2.*GPI/60.
      GLOSK1=RSPDA(16)
      SLICES=RSPDA(28)
      GBCELL=GBLADE/FLOAT(NX)/SLICES
      GONEBL=GPI/GBLADE
C **  GET THE PROPERTIES OF THE TURBINE EXHAUST GASES FROM SATELLITE
C     THESE VALUES ARE ASSIGNED TO ANY INFLOW AT THE 'EXIT'
C     BETWEEN THE O.D. OF THE AFT-PLATFORM SEAL AND THE BLADE LIPS
      GEXIT1=RSPDA(2)
```

```
      GVALKE = RSPDA(25)
      GVALEP = RSPDA(26)
      GHEXIT = RSPDA(27)
      H2OXIT = RSPDA(29)
CC ***  NEED TO CHECK ON VALUE OF H AND H2O
C GVALKE=0.01*W1EXIT**2
      W1XITM=SQRT(100.*GVALKE)/10.
      RETURN
C------------------------------------------------------------------
C     CHAPTER 1: CALLED AT THE START OF EACH TIME STEP.
C     SET 'DT' HERE WHEN TLAST SET NEGATIVE IN BLOCK DATA.
C     'ATIME + DT' GIVES THE END TIME OF THE CURRENT TIME STEP.
C     NOT ACCESSED IF STEADY,OR PARABOLIC.
C------------------------------------------------------------------
  100 CONTINUE
      RETURN
C------------------------------------------------------------------
C     CHAPTER 2: CALLED AT THE START OF EACH SWEEP.
C------------------------------------------------------------------
  200 CONTINUE
      RETURN
C------------------------------------------------------------------
C     CHAPTER 3: CALLED AT THE START OF EACH SLAB
C     NOT ACCESSED IF PARABOLIC, BUT 'STRIDE' IS.
C------------------------------------------------------------------
  300 CONTINUE
C ***
      IF(.NOT.(RESTRT.AND.ISWP.EQ.FSWEEP)) RETURN
      CALL GET(C1,GC1,NY,NX)
      CALL GET(T1,GT1,NY,NX)
      CALL GVISC(GT1,GC1,GMU1L,NY,NX)
C ***
      RETURN
C------------------------------------------------------------------
C     CHAPTER 4: CALLED AT THE START OF EACH RE-CALCULATION OF
C     VARIABLES P1,...C4 AT CURRENT SLAB. ITNO= ITERATION NUMBER.
C------------------------------------------------------------------
  400 CONTINUE
      RETURN
C------------------------------------------------------------------
C     CHAPTER 5: GROUND CALLED WHEN SOURCE TERM IS COMPUTED.
C     INDVAR GIVES DEPENDENT VARIABLE IN QUESTION IE. U1,...C4.
C     TO ADD SOURCE TO DEPENDENT VARIABLE C1(SAY) FOR IX=IXF,IXL
C     AND IY=IYF,IYL INSERT STATEMENT:
C     IF(INDVAR.EQ.C1)
C    &CALL ADD(INDVAR,IXF,IXL,IYF,IYL,TYPE,CM,VM,CVAR,VVAR,NY,NX)
C     NOTES ON 'ADD':
C    *SOURCE= (CVAR(IY,IX)+AMAX1(0.0,MASFLO))*(VVAR(IY,IX)-PHI),
C     WHERE 'PHI' IS IN-CELL VALUE OF VARIABLE IN QUESTION.
C    *'MASFLO'= CM(IY,IX)*(VM(IY,IX)-P),
C     WHERE 'P' IS THE IN-CELL PRESSURE.
C    *FOR INDVAR= M1, OR =M2, SOURCE ADDED IS 'MASFLO' ONLY,
C     EXCEPT FOR ONEPHS=.F. & MASFLO < 0.0 (IE. OUTFLOW) WHEN
C     CM(IY,IX) IS MULTIPLIED BY R1*D1 (FOR M1) & R2*D2 (FOR M2).
C    *BOTH 'CVAR' & 'CM' ARE MUTLIPLIED BY CELL-GEOMETRY QUANTITY
C     DICTATED BY SETTING OF 'TYPE' (=CELL, EAST AREA,..VOLUME).
C    *TYPE-SPECIFIED AREAS ARE CALCULATED AS IF BLOCKAGE ABSENT,
C     BUT 'VOLUME' WITH ACCOUNT FOR ITS PRESENCE.
C    *FOR ALL SOLVED VARIABLES, INCLUDING M1 ( & M2 WHEN ONEPHS=F),
C     IF 'CM'> 0.0 CALL 'ADD'          FOR M1 & M2 ALTHOUGH 'CVAR' & 'VVAR'
```

```
C     HAVE NO SIGNIFICANCE THEY MUST BE ENTERED AS ARGUMENTS.
C    *'CVAR', 'VVAR', 'CM' & 'VM' MUST BE DIMENSIONED NY,NX.
C-----------------------------------------------------------------
  500 CONTINUE
C ***
      IF(INDVAR.NE.U1) GO TO 502
C FIX ANGULAR MOMENTUM IN CELL(S) COMPLETELY ENFRAMED BY BLADES
      IF(IZED.LT.21) GO TO 502
      CALL GET1D(R,GR,NY)
      JIYF = 32
      JIYL = 40
      DO 501 JIX=1,NX
      DO 501 JIY=JIYF,JIYL
      CVAR(JIY,JIX)=FIXVAL
  501 VVAR(JIY,JIX)=GOMEGA*GR(JIY)**2
      CALL ADD(U1,1,NX,JIYF,JIYL,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
  502 IF(INDVAR.NE.C1) GO TO 503
C
C GET ADDITIONAL VARIABLES REQUIRED FOR TOTAL PRESSURE CALCULATIONS
      CALL GET(P1,GPT,NY,NX)
      CALL GET(W1L,GW1L,NY,NX)
C
C SAVE CALCULATED EFFECTIVE VISCOSITIES FOR PRINTOUT
      CALL GET(MU1,GMU1,NY,NX)
C
C GET VARIABLES REQUIRED FOR SUBROUTINE GWALL (IN COMMON/WALLG/)
      CALL GET(U1,GU1,NY,NX)
      CALL GET(V1,GV1,NY,NX)
      CALL GET(W1,GW1,NY,NX)
      CALL GET(D1,GD1,NY,NX)
C!! NB. THE "GET" COMMAND CANNOT BE USED FOR MU1LAM IN CH. 5 AND
C!! SO THE LOCAL LAMINAR VISCOSITY ARRAY (GMU1L) MUST BE SET-UP
C!! ELSEWHERE IN GROUND. FOR THE CURRENT PROBLEM IT IS CALCULATED (FOR
C!! CONVENIENCE) IN CH. 10, AND THEN "SET" IN CH. 12 (FOR PASSING
C!! BACK TO EARTH).
C
      CALL GET1D(DXU,GDXU,NX)
      CALL GET1D(DYV,GDYV,NY)
      CALL GET1D(DZW,GDZW,NZ)
      CALL GET1D(R,GR,NY)
C
C *** CALCULATE WALL FRICTION EFFECTS ***
C
  503 GVELUW=INDVAR.EQ.U1.OR.INDVAR.EQ.W1
      GVELVW=INDVAR.EQ.V1.OR.INDVAR.EQ.W1
      GVELUV=INDVAR.EQ.U1.OR.INDVAR.EQ.V1
      GKEEP=INDVAR.EQ.KE.OR.INDVAR.EQ.EP
C
C *** ROTATING WALL(S) ***
C
C *** ROW 1
      IF(.NOT.(IZED.GE.13.AND.IZED.LE.19)) GO TO 504
      IF(.NOT.GVELUW) GO TO 5040
      CALL GWALL(INDVAR,1,NX,1,1,IZED,SOUTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,1,1,SOUTH,CM,VM,CVAR,VVAR,NY,NX)
 5040 IF(.NOT.GKEEP) GO TO 504
      CALL GWALL(INDVAR,1,NX,1,1,IZED,SOUTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,1,1,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
```

```
 504  IF(.NOT.(IZED.EQ.19)) GO TO 505
      IF(.NOT.GVELUV) GO TO 5050
      CALL GWALL(INDVAR,1,NX,1,1,IZED,HIGH,GOMEGA,O.,O.,O.,-1.)
      CALL ADD(INDVAR,1,NX,1,1,HIGH,CM,VM,CVAR,VVAR,NY,NX)
 5050 IF(.NOT.GKEEP) GO TO 505
      CALL GWALL(INDVAR,1,NX,1,1,IZED,HIGH,GOMEGA,O.,O.,O.,-1.)
      CALL ADD(INDVAR,1,NX,1,1,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROWS 2 TO 3
 505  IF(.NOT.(IZED.EQ.21)) GO TO 506
      IF(.NOT.GVELUV) GO TO 5060
      CALL GWALL(INDVAR,1,NX,2,3,IZED,HIGH,GOMEGA,O.,O.,O.,-1.)
      CALL ADD(INDVAR,1,NX,2,3,HIGH,CM,VM,CVAR,VVAR,NY,NX)
 5060 IF(.NOT.GKEEP) GO TO 506
      CALL GWALL(INDVAR,1,NX,2,3,IZED,HIGH,GOMEGA,O.,O.,O.,-1.)
      CALL ADD(INDVAR,1,NX,2,3,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 506  IF(.NOT.(IZED.GE.20.AND.IZED.LE.21)) GO 10 507
      IF(.NOT.GVELUW) GO TO 5070
      CALL GWALL(INDVAR,1,NX,2,2,IZED,SOUTH,GOMEGA,O.,O.,O.,-1.)
      CALL ADD(INDVAR,1,NX,2,2,SOUTH,CM,VM,CVAR,VVAR,NY,NX)
 5070 IF(.NOT.GKEEP) GO TO 507
      CALL GWALL(INDVAR,1,NX,2,2,IZED,SOUTH,GOMEGA,O.,O.,O.,-1.)
      CALL ADD(INDVAR,1,NX,2,2,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 507  IF(.NOT.(IZED.GE.17.AND.IZED.LE.21)) GO TO 508
      IF(.NOT.GVELUW) GO TO 5080
      CALL GWALL(INDVAR,1,NX,3,3,IZED,NORTH,GOMEGA,O.,O.,O.,-1.)
      CALL ADD(INDVAR,1,NX,3,3,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5080 IF(.NOT.GKEEP) GO TO 508
      CALL GWALL(INDVAR,1,NX,3,3,IZED,NORTH,GOMEGA,O.,O.,O.,-1.)
      CALL ADD(INDVAR,1,NX,3,3,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROW 4
 508  IF(.NOT.(IZED.EQ.16)) GO TO 509
      IF(.NOT.GVELUV) GO TO 5090
      CALL GWALL(INDVAR,1,NX,4,4,IZED,HIGH,GOMEGA,O.,O.,O.,-1.)
      CALL ADD(INDVAR,1,NX,4,4,HIGH,CM,VM,CVAR,VVAR,NY,NX)
 5090 IF(.NOT.GKEEP) GO TO 509
      CALL GWALL(INDVAR,1,NX,4,4,IZED,HIGH,GOMEGA,O.,O.,O.,-1.)
      CALL ADD(INDVAR,1,NX,4,4,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROWS 5 TO 10
 509  IF(.NOT.(IZED.GE.17.AND.IZED.LE.20)) GO 10 510
      IF(.NOT.GVELUW) GO TO 5100
      CALL GWALL(INDVAR,1,NX,5,5,IZED,SOUTH,GOMEGA,O.,O.,O.,-1.)
      CALL ADD(INDVAR,1,NX,5,5,SOUTH,CM,VM,CVAR,VVAR,NY,NX)
 5100 IF(.NOT.GKEEP) GO TO 510
      CALL GWALL(INDVAR,1,NX,5,5,IZED,SOUTH,GOMEGA,O.,O.,O.,-1.)
      CALL ADD(INDVAR,1,NX,5,5,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 510  IF(.NOT.(IZED.EQ.20)) GO TO 511
      IF(.NOT.GVELUV) GO TO 5110
      CALL GWALL(INDVAR,1,NX,5,10,IZED,HIGH,GOMEGA,O.,O.,O.,-1.)
      CALL ADD(INDVAR,1,NX,5,10,HIGH,CM,VM,CVAR,VVAR,NY,NX)
 5110 IF(.NOT.GKEEP) GO TO 511
      CALL GWALL(INDVAR,1,NX,5,10,IZED,HIGH,GOMEGA,O.,O.,O.,-1.)
      CALL ADD(INDVAR,1,NX,5,10,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 511  IF(.NOT.(IZED.EQ.21)) GO TO 512
```

```
      IF(.NOT.GVELUW) GO TO 5120
      CALL GWALL(INDVAR,1,NX,10,10,IZED,SOUTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,10,10,SOUTH,CM,VM,CVAR,VVAR,NY,NX)
 5120 IF(.NOT.GKEEP) GO TO 512
      CALL GWALL(INDVAR,1,NX,10,10,IZED,SOUTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,10,10,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 512  IF(.NOT.(IZED.EQ.22)) GO TO 513
      IF(.NOT.GVELUV) GO TO 5130
      CALL GWALL(INDVAR,1,NX,10,10,IZED,HIGH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,10,10,HIGH,CM,VM,CVAR,VVAR,NY,NX)
 5130 IF(.NOT.GKEEP) GO TO 513
      CALL GWALL(INDVAR,1,NX,10,10,IZED,HIGH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,10,10,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROW 11
 513  IF(.NOT.(IZED.EQ.23)) GO TO 514
      IF(.NOT.GVELUW) GO TO 5140
      CALL GWALL(INDVAR,1,NX,11,11,IZED,SOUTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,11,11,SOUTH,CM,VM,CVAR,VVAR,NY,NX)
 5140 IF(.NOT.GKEEP) GO TO 514
      CALL GWALL(INDVAR,1,NX,11,11,IZED,SOUTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,11,11,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 514  IF(.NOT.(IZED.EQ.24)) GO TO 515
      IF(.NOT.GVELUV) GO TO 5150
      CALL GWALL(INDVAR,1,NX,11,11,IZED,HIGH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,11,11,HIGH,CM,VM,CVAR,VVAR,NY,NX)
 5150 IF(.NOT.GKEEP) GO TO 515
      CALL GWALL(INDVAR,1,NX,11,11,IZED,HIGH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,11,11,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROWS 12 TO 13
 515  IF(.NOT.(IZED.EQ.24)) GO TO 516
      IF(.NOT.GVELUV) GO TO 5160
      CALL GWALL(INDVAR,1,NX,12,13,IZED,HIGH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,12,13,HIGH,CM,VM,CVAR,VVAR,NY,NX)
 5160 IF(.NOT.GKEEP) GO TO 516
      CALL GWALL(INDVAR,1,NX,12,13,IZED,HIGH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,12,13,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 516  IF(.NOT.(IZED.GE.21.AND.IZED.LE.24)) GO TO 517
      IF(.NOT.GVELUW) GO TO 5170
      CALL GWALL(INDVAR,1,NX,13,13,IZED,NORTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,13,13,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5170 IF(.NOT.GKEEP) GO TO 517
      CALL GWALL(INDVAR,1,NX,13,13,IZED,NORTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,13,13,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROW 14
 517  IF(.NOT.(IZED.EQ.20)) GO TO 518
      IF(.NOT.GVELUV) GO TO 5180
      CALL GWALL(INDVAR,1,NX,14,14,IZED,HIGH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,14,14,HIGH,CM,VM,CVAR,VVAR,NY,NX)
 5180 IF(.NOT.GKEEP) GO TO 518
      CALL GWALL(INDVAR,1,NX,14,14,IZED,HIGH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,14,14,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROW 15
 518  IF(.NOT.(IZED.GE.21.AND.IZED.LE.23)) GO TO 519
```

```
      IF(.NOT.GVELUW) GO TO 5190
      CALL GWALL(INDVAR,1,NX,15,15,IZED,SOUTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,15,15,SOUTH,CM,VM,CVAR,VVAR,NY,NX)
 5190 IF(.NOT.GKEEP) GO TO 519
      CALL GWALL(INDVAR,1,NX,15,15,IZED,SOUTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,15,15,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 519  IF(.NOT.(IZED.EQ.24)) GO TO 520
      IF(.NOT.GVELUV) GO TO 5200
      CALL GWALL(INDVAR,1,NX,15,15,IZED,HIGH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,15,15,HIGH,CM,VM,CVAR,VVAR,NY,NX)
 5200 IF(.NOT.GKEEP) GO TO 520
      CALL GWALL(INDVAR,1,NX,15,15,IZED,HIGH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,15,15,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROWS 16 TO 21
 520  IF(.NOT.(IZED.EQ.25)) GO TO 521
      IF(.NOT.GVELUV) GO TO 5210
      CALL GWALL(INDVAR,1,NX,16,21,IZED,HIGH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,16,21,HIGH,CM,VM,CVAR,VVAR,NY,NX)
 5210 IF(.NOT.GKEEP) GO TO 521
      CALL GWALL(INDVAR,1,NX,16,21,IZED,HIGH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,16,21,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 521  IF(.NOT.(IZED.GE.23.AND.IZED.LE.24)) GO TO 522
      IF(.NOT.GVELUW) GO TO 5220
      CALL GWALL(INDVAR,1,NX,21,21,IZED,NORTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,21,21,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5220 IF(.NOT.GKEEP) GO TO 522
      CALL GWALL(INDVAR,1,NX,21,21,IZED,NORTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,21,21,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROWS 22 TO 31
 522  IF(.NOT.(IZED.GE.21.AND.IZED.LE.22)) GO TO 523
      IF(.NOT.GVELUW) GO TO 5230
      CALL GWALL(INDVAR,1,NX,22,22,IZED,NORTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,22,22,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5230 IF(.NOT.GKEEP) GO TO 523
      CALL GWALL(INDVAR,1,NX,22,22,IZED,NORTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,22,22,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 523  IF(.NOT.(IZED.EQ.20)) GO TO 524
      IF(.NOT.GVELUW) GO TO 5240
      CALL GWALL(INDVAR,1,NX,22,31,IZED,HIGH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,22,31,HIGH,CM,VM,CVAR,VVAR,NY,NX)
 5240 IF(.NOT.GKEEP) GO TO 524
      CALL GWALL(INDVAR,1,NX,22,31,IZED,HIGH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,22,31,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C
C *** ROWS 32 TO 40
 524  IF(.NOT.(IZED.GE.21.AND.IZED.LE.28)) GO TO 5251
C
C NOTE THAT GDELTA IS SPECIFIED AS 0.5*(ANGLE BETWEEN EACH BLADE) SO
C THAT FRICTION AND KE AND EP VALUES CALCULATED AS FOR BETWEEN BLADES
      IF(.NOT.GVELVW) GO TO 5250
C
      CALL GWALL(INDVAR,1,NX,32,40,IZED,EAST,GOMEGA,0.,0.,0.,GONEBL)
C ACCOUNT FOR THERE BEING MORE THAN ONE BLADE IN EACH CELL BY
C INCREASING FRICTION LOSS(ES) PROPORTIONATELY (IE. *NO. BLADES/NX)
```

```
      DO 5241 JIX=1,NX
      DO 5241 JIY=32,40
 5241 CVAR(JIY,JIX)=CVAR(JIY,JIX)*GBCELL
      CALL ADD(INDVAR,1,NX,32,40,EAST,CM,VM,CVAR,VVAR,NY,NX)
C
      CALL GWALL(INDVAR,1,NX,32,40,IZED,WEST,GOMEGA,0.,0.,0.,GONEBL)
      DO 5242 JIX=1,NX
      DO 5242 JIY=32,40
 5242 CVAR(JIY,JIX)=CVAR(JIY,JIX)*GBCELL
      CALL ADD(INDVAR,1,NX,32,40,WEST,CM,VM,CVAR,VVAR,NY,NX)
C
 5250 IF(.NOT.GKEEP) GO TO 5251
      CALL GWALL(INDVAR,1,NX,32,40,IZED,EAST,GOMEGA,0.,0.,0.,GONEBL)
      CALL ADD(INDVAR,1,NX,32,40,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C
 5251 IF(.NOT.(IZED.GE.21.AND.IZED.LE.28)) GO TO 525
      IF(.NOT.GVELUW) GO TO 5252
      CALL GWALL(INDVAR,1,NX,32,32,IZED,SOUTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,32,32,SOUTH,CM,VM,CVAR,VVAR,NY,NX)
 5252 IF(.NOT.GKEEP) GO TO 525
      CALL GWALL(INDVAR,1,NX,32,32,IZED,SOUTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,32,32,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 525  IF(.NOT.(IZED.GE.17.AND.IZED.LE.28)) GO TO 526
      IF(.NOT.GVELUW) GO TO 5260
      CALL GWALL(INDVAR,1,NX,40,40,IZED,NORTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,40,40,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5260 IF(.NOT.GKEEP) GO TO 526
      CALL GWALL(INDVAR,1,NX,40,40,IZED,NORTH,GOMEGA,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,40,40,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** NON-ROTATING WALLS ***
C
C *** ROW 1
 526  IF(.NOT.(IZED.EQ.13)) GO TO 530
      IF(.NOT.GVELUW) GO TO 5300
      CALL GWALL(INDVAR,1,NX,1,1,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,1,1,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5300 IF(.NOT.GKEEP) GO TO 530
      CALL GWALL(INDVAR,1,NX,1,1,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,1,1,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROWS 2 TO 5
 530  IF(.NOT.(IZED.EQ.14)) GO TO 531
      IF(.NOT.GVELUV) GO TO 5310
      CALL GWALL(INDVAR,1,NX,2,5,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,2,5,LOW,CM,VM,CVAR,VVAR,NY,NX)
 5310 IF(.NOT.GKEEP) GO TO 531
      CALL GWALL(INDVAR,1,NX,2,5,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,2,5,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 531  IF(.NOT.(IZED.EQ.13)) GO TO 532
      IF(.NOT.GVELUW) GO TO 5320
      CALL GWALL(INDVAR,1,NX,5,5,IZED,SOUTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,5,5,SOUTH,CM,VM,CVAR,VVAR,NY,NX)
 5320 IF(.NOT.GKEEP) GO TO 532
      CALL GWALL(INDVAR,1,NX,5,5,IZED,SOUTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,5,5,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
```

```
  532 IF(.NOT.(IZED.EQ.12)) GO TO 533
      IF(.NOT.GVELUV) GO TO 5330
      CALL GWALL(INDVAR,1,NX,3,5,IZED,HIGH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,3,5,HIGH,CM,VM,CVAR,VVAR,NY,NX)
 5330 IF(.NOT.GKEEP) GO TO 533
      CALL GWALL(INDVAR,1,NX,3,5,IZED,HIGH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,3,5,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
  533 IF(.NOT.(IZED.GE.5.AND.IZED.LE.12)) GO TO 534
      IF(.NOT.GVELUW) GO TO 5340
      CALL GWALL(INDVAR,1,NX,3,3,IZED,SOUTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,3,3,SOUTH,CM,VM,CVAR,VVAR,NY,NX)
 5340 IF(.NOT.GKEEP) GO TO 534
      CALL GWALL(INDVAR,1,NX,3,3,IZED,SOUTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,3,3,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
  534 IF(.NOT.(IZED.EQ.5)) GO TO 535
      IF(.NOT.GVELUV) GO TO 5350
      CALL GWALL(INDVAR,1,NX,3,3,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,3,3,LOW,CM,VM,CVAR,VVAR,NY,NX)
 5350 IF(.NOT.GKEEP) GO TO 535
      CALL GWALL(INDVAR,1,NX,3,3,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,3,3,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
  535 IF(.NOT.(IZED.EQ.4)) GO TO 537
      IF(.NOT.GVELUW) GO TO 5360
      CALL GWALL(INDVAR,1,NX,4,5,IZED,SOUTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,4,5,SOUTH,CM,VM,CVAR,VVAR,NY,NX)
 5360 IF(.NOT.GKEEP) GO TO 536
      CALL GWALL(INDVAR,1,NX,4,5,IZED,SOUTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,4,5,CELL,CM,VM,CVAR,VVAR,NY,NX)
  536 IF(.NOT.GVELUV) GO TO 5370
      CALL GWALL(INDVAR,1,NX,4,4,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,4,4,LOW,CM,VM,CVAR,VVAR,NY,NX)
 5370 IF(.NOT.GKEEP) GO TO 537
      CALL GWALL(INDVAR,1,NX,4,4,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,4,4,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROWS 6 TO 9
  537 IF(.NOT.(IZED.EQ.3)) GO TO 539
      IF(.NOT.GVELUW) GO TO 5380
      CALL GWALL(INDVAR,1,NX,6,7,IZED,SOUTH,0.,0.,0.,0., 1.)
      CALL ADD(INDVAR,1,NX,6,7,SOUTH,CM,VM,CVAR,VVAR,NY,NX)
 5380 IF(.NOT.GKEEP) GO TO 538
      CALL GWALL(INDVAR,1,NX,6,7,IZED,SOUTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,6,7,CELL,CM,VM,CVAR,VVAR,NY,NX)
  538 IF(.NOT.GVELUV) GO TO 5390
      CALL GWALL(INDVAR,1,NX,6,7,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,6,7,LOW,CM,VM,CVAR,VVAR,NY,NX)
 5390 IF(.NOT.GKEEP) GO TO 539
      CALL GWALL(INDVAR,1,NX,6,7,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,6,7,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
  539 IF(.NOT.(IZED.EQ.2)) GO TO 5411
      IF(.NOT.GVELUW) GO TO 5400
      CALL GWALL(INDVAR,1,NX,8,8,IZED,SOUTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,8,8,SOUTH,CM,VM,CVAR,VVAR,NY,NX)
 5400 IF(.NOT.GKEEP) GO TO 540
      CALL GWALL(INDVAR,1,NX,8,8,IZED,SOUTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,8,8,CELL,CM,VM,CVAR,VVAR,NY,NX)
```

```
540   IF(.NOT.GVELUV) GO TO 5410
      CALL GWALL(INDVAR,1,NX,8,8,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,8,8,LOW,CM,VM,CVAR,VVAR,NY,NX)
5410  IF(.NOT.GKEEP) GO TO 5411
      CALL GWALL(INDVAR,1,NX,8,8,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,8,8,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
5411  IF(.NOT.(IZED.EQ.2)) GO TO 541
      IF(.NOT.GVELUV) GO TO 5414
      CALL GWALL(INDVAR,1,NX,9,9,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,9,9,LOW,CM,VM,CVAR,VVAR,NY,NX)
5414  IF(.NOT.GKEEP) GO TO 541
      CALL GWALL(INDVAR,1,NX,9,9,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,9,9,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROW 10
541   IF(.NOT.(IZED.EQ.1)) GO TO 545
      IF(.NOT.GVELUW) GO TO 5450
      CALL GWALL(INDVAR,1,NX,10,10,IZED,SOUTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,10,10,SOUTH,CM,VM,CVAR,VVAR,NY,NX)
      CALL GWALL(INDVAR,1,NX,10,10,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,10,10,NORTH,CM,VM,CVAR,VVAR,NY,NX)
5450  IF(.NOT.GKEEP) GO TO 545
      CALL GWALL(INDVAR,1,NX,10,10,IZED,SOUTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,10,10,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
545   IF(.NOT.(IZED.GE.2.AND.IZED.LE.4)) GO TO 546
      IF(.NOT.GVELUW) GO TO 5460
      CALL GWALL(INDVAR,1,NX,10,10,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,10,10,NORTH,CM,VM,CVAR,VVAR,NY,NX)
5460  IF(.NOT.GKEEP) GO TO 546
      CALL GWALL(INDVAR,1,NX,10,10,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,10,10,CELL,CM,VM,CVAR,VVAR,NY,NX)
546   IF(.NOT.GVELUV) GO TO 5470
      CALL GWALL(INDVAR,1,NX,10,10,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,10,10,LOW,CM,VM,CVAR,VVAR,NY,NX)
5470  IF(.NOT.GKEEP) GO TO 547
      CALL GWALL(INDVAR,1,NX,10,10,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,10,10,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROW 11
547   IF(.NOT.(IZED.GE.3.AND.IZED.LE.4)) GO TO 549
      IF(.NOT.GVELUV) GO TO 5480
      CALL GWALL(INDVAR,1,NX,11,11,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,11,11,LOW,CM,VM,CVAR,VVAR,NY,NX)
5480  IF(.NOT.GKEEP) GO TO 548
      CALL GWALL(INDVAR,1,NX,11,11,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,11,11,CELL,CM,VM,CVAR,VVAR,NY,NX)
548   IF(.NOT.GVELUW) GO TO 5490
      CALL GWALL(INDVAR,1,NX,11,11,IZED,SOUTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,11,11,SOUTH,CM,VM,CVAR,VVAR,NY,NX)
5490  IF(.NOT.GKEEP) GO TO 549
      CALL GWALL(INDVAR,1,NX,11,11,IZED,SOUTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,11,11,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROWS 12 TO 17
549   IF(.NOT.(IZED.EQ.3)) GO TO 550
      IF(.NOT.GVELUV) GO TO 5500
      CALL GWALL(INDVAR,1,NX,12,17,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,12,17,LOW,CM,VM,CVAR,VVAR,NY,NX)
```

C-2

```
 5500 IF(.NOT.GKEEP) GO TO 550
      CALL GWALL(INDVAR,1,NX,12,17,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,12,17,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 550  IF(.NOT.(IZED.GE.3.AND.IZED.LE.4)) GO TO 551
      IF(.NOT.GVELUW) GO TO 5510
      CALL GWALL(INDVAR,1,NX,17,17,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,17,17,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5510 IF(.NOT.GKEEP) GO TO 551
      CALL GWALL(INDVAR,1,NX,17,17,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,17,17,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROWS 18 TO 21
 551  IF(.NOT.(IZED.EQ.4)) GO TO 552
      IF(.NOT.GVELUV) GO TO 5520
      CALL GWALL(INDVAR,1,NX,18,20,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,18,20,LOW,CM,VM,CVAR,VVAR,NY,NX)
 5520 IF(.NOT.GKEEP) GO TO 552
      CALL GWALL(INDVAR,1,NX,18,20,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,18,20,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 552  IF(.NOT.(IZED.EQ.4)) GO TO 553
      IF(.NOT.GVELUW) GO TO 5530
      CALL GWALL(INDVAR,1,NX,21,21,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,21,21,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5530 IF(.NOT.GKEEP) GO TO 553
      CALL GWALL(INDVAR,1,NX,21,21,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,21,21,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 553  IF(.NOT.(IZED.GE.5.AND.IZED.LE.6)) GO TO 555
      IF(.NOT.GVELUW) GO TO 5540
      CALL GWALL(INDVAR,1,NX,21,21,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,21,21,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5540 IF(.NOT.GKEEP) GO TO 555
      CALL GWALL(INDVAR,1,NX,21,21,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,21,21,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROWS 22 TO 29
 555  IF(.NOT.(IZED.GE.6.AND.IZED.LE.8)) GO TO 556
      IF(.NOT.GVELUW) GO TO 5560
      CALL GWALL(INDVAR,1,NX,22,22,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,22,22,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5560 IF(.NOT.GKEEP) GO TO 556
      CALL GWALL(INDVAR,1,NX,22,22,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,22,22,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 556  IF(.NOT.(IZED.EQ.9)) GO TO 557
      IF(.NOT.GVELUW) GO TO 5570
      CALL GWALL(INDVAR,1,NX,23,23,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,23,23,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5570 IF(.NOT.GKEEP) GO TO 557
      CALL GWALL(INDVAR,1,NX,23,23,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,23,23,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 557  IF(.NOT.(IZED.EQ.10)) GO TO 558
      IF(.NOT.GVELUW) GO TO 5580
      CALL GWALL(INDVAR,1,NX,24,24,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,24,24,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5580 IF(.NOT.GKEEP) GO TO 558
      CALL GWALL(INDVAR,1,NX,24,24,IZED,NORTH,0.,0.,0.,0.,-1.)
```

```
      CALL ADD(INDVAR,1,NX,24,24,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 558  IF(.NOT.(IZED.EQ.11)) GO TO 559
      IF(.NOT.GVELUW) GO TO 5590
      CALL GWALL(INDVAR,1,NX,25,25,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,25,25,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5590 IF(.NOT.GKEEP) GO TO 559
      CALL GWALL(INDVAR,1,NX,25,25,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,25,25,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 559  IF(.NOT.(IZED.EQ.12)) GO TO 560
      IF(.NOT.GVELUW) GO TO 5600
      CALL GWALL(INDVAR,1,NX,26,26,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,26,26,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5600 IF(.NOT.GKEEP) GO TO 560
      CALL GWALL(INDVAR,1,NX,26,26,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,26,26,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 560  IF(.NOT.(IZED.EQ.13)) GO TO 561
      IF(.NOT.GVELUW) GO TO 5610
      CALL GWALL(INDVAR,1,NX,27,27,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,27,27,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5610 IF(.NOT.GKEEP) GO TO 561
      CALL GWALL(INDVAR,1,NX,27,27,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,27,27,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 561  IF(.NOT.(IZED.EQ.14)) GO TO 562
      IF(.NOT.GVELUW) GO TO 5620
      CALL GWALL(INDVAR,1,NX,28,28,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,28,28,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5620 IF(.NOT.GKEEP) GO TO 562
      CALL GWALL(INDVAR,1,NX,28,28,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,28,28,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
 562  IF(.NOT.(IZED.EQ.15)) GO TO 563
      IF(.NOT.GVELUW) GO TO 5630
      CALL GWALL(INDVAR,1,NX,29,29,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,29,29,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5630 IF(.NOT.GKEEP) GO TO 563
      CALL GWALL(INDVAR,1,NX,29,29,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,29,29,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROWS 30 TO 31
 563  IF(.NOT.(IZED.EQ.16)) GO TO 565
      IF(.NOT.GVELUW) GO TO 5640
      CALL GWALL(INDVAR,1,NX,30,31,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,30,31,NORTH,CM,VM,CVAR,VVAR,NY,NX)
 5640 IF(.NOT.GKEEP) GO TO 564
      CALL GWALL(INDVAR,1,NX,30,31,IZED,NORTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,30,31,CELL,CM,VM,CVAR,VVAR,NY,NX)
 564  IF(.NOT.GVELUV) GO TO 5650
      CALL GWALL(INDVAR,1,NX,30,31,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,30,31,LOW,CM,VM,CVAR,VVAR,NY,NX)
 5650 IF(.NOT.GKEEP) GO TO 565
      CALL GWALL(INDVAR,1,NX,30,31,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,30,31,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROWS 32 TO 37
 565  IF(.NOT.(IZED.EQ.17)) GO TO 567
      IF(.NOT.GVELUV) GO TO 5670
```

```
      CALL GWALL(INDVAR,1,NX,32,37,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,32,37,LOW,CM,VM,CVAR,VVAR,NY,NX)
 5670 IF(.NOT.GKEEP) GO TO 567
      CALL GWALL(INDVAR,1,NX,32,37,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,32,37,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROWS 38 TO 39
 567  IF(.NOT.(IZED.EQ.18)) GO TO 568
      IF(.NOT.GVELUV) GO TO 5680
      CALL GWALL(INDVAR,1,NX,38,39,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,38,39,LOW,CM,VM,CVAR,VVAR,NY,NX)
 5680 IF(.NOT.GKEEP) GO TO 568
      CALL GWALL(INDVAR,1,NX,38,39,IZED,LOW,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,38,39,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C *** ROW 40
 568  IF(.NOT.(IZED.EQ.17)) GO TO 575
      IF(.NOT.GVELUW) GO TO 5690
      CALL GWALL(INDVAR,1,NX,40,40,IZED,SOUTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,40,40,SOUTH,CM,VM,CVAR,VVAR,NY,NX)
 5690 IF(.NOT.GKEEP) GO TO 575
      CALL GWALL(INDVAR,1,NX,40,40,IZED,SOUTH,0.,0.,0.,0.,-1.)
      CALL ADD(INDVAR,1,NX,40,40,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C
C
C
C *** ACCOUNT FOR MOMENTUM LOSSES AT EXIT(S) ***
C
C *** EXIT NEAR BLADE ROOTS (O.D. OF AFT-PLATFORM SEAL)
 575  IF(.NOT.(IZED.EQ.17)) GO TO 582
      JIY = NY
      CALL GET(AHIGH,GAHIGH,NY,NX)
      CALL GET1D(DYV,GDYV,NY)
      CALL GET(W1,GW1,NY,NX)
C
C **  USE A LOSS COEFFICIENT (GLOSK1) TO COMPUTE THE PRESSURE
C     LOSS ACROSS THE EXIT AT THE O.D. OF THE AFT-PLATFORM SEAL
C     MASSFLOW = CM (EXIT PRESS - CELL PRESS)
C     CM = 2 * (EXIT AREA)/(GLOSK1*EXIT VELOCITY)
C       NOTE: SUB. FOR VELOCITY, VELOCITY =
C         W1(FULL CELL AXIAL VELOCITY)*(CELL HEIGHT/GAP SIZE)
C               SUB. FOR EXIT AREA, EXIT AREA =
C         GAHIGH(FULL CELL AREA)*(GAP SIZE/CELL HEIGHT)
      DO 580 JIX =1,NX
C PREVENT LARGE CM BY LIMITING SMALLEST W1 (=0.1*NOMINAL EXIT W1)
      ABSGW1=AMAX1(W1XITM,ABS(GW1(JIY,JIX)))
C       (CALCULATE THE LOSS COEFFICIENT CM)
      CM(JIY,JIX)=(2.0*GAHIGH(JIY,JIX)/(GLOSK1*ABSGW1+GTINY))
     .                                 *(GGEXIT(JIX)/GDYV(JIY))**2.
C PREVENT CM FROM BEHAVING ERRATICALLY IN FIRST FEW (5) SWEEPS
C NB. THE .0025 IS A 'LARGE' VALUE SUFFICIENT TO FIX P=PEXIT
      IF(ISWP.LE.5) CM(JIY,JIX)=0.0025*GAHIGH(JIY,JIX)
C GIVE SAVED CM A FINITE VALUE AT SWEEP 1
      IF(ISWP.LE.1) CM1S(JIX)=CM(JIY,JIX)
C UNDER-RELAX CM TO PREVENT INSTABILITY
      CM(JIY,JIX)=CMRLX1*CM(JIY,JIX)+(1.-CMRLX1)*CM1S(JIX)
C SAVE CM VALUE FOR RELAXATION
      CM1S(JIX)=CM(JIY,JIX)
C        (ASSIGN VM THE CIRCUMFERENTIAL EXIT PRESSURES)
```

```
      VM(JIY,JIX)=GPEXIT(JIX)
C
C **  SET THE VALUES OF U1,V1,W1,H1,KE,AND EP AT THE O.D. OF
C     THE AFT-PLATFORM SEAL IN CASE OF INFLOW.
C **  CM & VM HAVE BEEN SET ABOVE FOR PRESCRIBED MASS FLOW
C     CVAR  IS SET TO ZERO FOR ZERO DIFFUSION FLUX
      CVAR(JIY,JIX) = 0.0
C     SET VVAR TO EXTERNAL VALUE APPROPRIATE FOR THE VARIABLE
      VVAR(JIY,JIX) = 0.0
      IF (INDVAR.EQ.H1) VVAR(JIY,JIX) = GHEXIT
      IF (INDVAR.EQ.C1) VVAR(JIY,JIX) = H2OXIT
      IF (INDVAR.EQ.KE) VVAR(JIY,JIX) = GVALKE
      IF (INDVAR.EQ.EP) VVAR(JIY,JIX) = GVALEP
C ** NOTE: THE VALUES OF CVAR AND VVAR NEED NOT BE DEFINED FOR M1
C    AS THEY DO  FOR OTHER VARIABLES (REF. CHAM TR/75 SEC.4.2-9)
 580  CONTINUE
C     ****   ADD SOURCE TERM   ****
      CALL ADD(INDVAR,1,NX,JIY,JIY,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C     SUM THE MASSFLOW OUT EXIT1
      IF(.NOT.(ISWP.EQ.LSWEEP.AND.INDVAR.EQ.W1)) GO TO 582
      EMOUT1=0.0
      DO 581 JIX=1,NX
      EMOUT1=EMOUT1-GD1(JIY,JIX)*GW1(JIY,JIX)*GAHIGH(JIY,JIX)*G
 581  CONTINUE
      IF(NX.EQ.1) EMOUT1=EMOUT1*2.*GPI/XULAST
C
C
C *** SECOND EXIT ***
 582  IF(.NOT.(JIXE2F.GE.1.AND.JIXE2F.LE.NX)) GO TO 587
      IF(IZED.NE.1) GO TO 587
      JIY=10
      CALL GET(AHIGH,GAHIGH,NY,NX)
      CALL GET(W1,GW1,NY,NX)
C
C **  SUM UP THE TOTAL HIGH FACE AREA BEING CONSIDERED
      GAHSUM =0.0
      DO 583 JIX =JIXE2F,JIXE2L
 583  GAHSUM= GAHSUM +' GAHIGH(JIY,JIX)
C
C **  USE A LOSS COEFFICIENT (GLOSK2) TO COMPUTE THE PRESSURE
C     LOSS ACROSS THE EXIT
C     MASSFLOW = CM (EXIT PRESS - CELL PRESS)
C     CM = 2 * (EXIT AREA)/(GLOSK2*EXIT VELOCITY)
C        NOTE:SUBSTITUTE VEL.= W1 (FULL CELL AXIAL VELOCITY)
C                                 * (FULL CELL AREA/EXIT AREA)
      DO 584 JIX =JIXE2F,JIXE2L
C       (CALCULATE THE LOSS COEFFICIENT CM)
C           (FIRST NEED TO CALCULATE EXIT AREA PER CELL (GAREAC))
      GAREAC= (GAXIT2/144.0)*(GAHIGH(JIY,JIX)/GAHSUM)
C PREVENT LARGE CM BY LIMITING SMALLEST W1 (=0.1*NOMINAL EXIT W1*
C                                                       AREA RATIO)
      ABSGW1=AMAX1(W1XITM*GAREAC/GAHIGH(JIY,JIX),ABS(GW1(JIY,JIX)))
      CM(JIY,JIX)=(2.0*GAREAC/(GLOSK2*ABSGW1+GTINY))
     $                      *(GAREAC/(GAHIGH(JIY,JIX)))
C PREVENT CM FROM BEHAVING ERRATICALLY IN FIRST FFW (5) SWEEPS
C NB. THE .02 IS A 'LARGE' VALUE SUFFICIENT TO FIX P=PEXIT
      IF(ISWP.LE.5) CM(JIY,JIX)=0.02*GAHIGH(JIY,JIX)
C GIVE SAVED CM A FINITE VALUE AT SWEEP 1
      IF(ISWP.LE.1) CM2S(JIX)=CM(JIY,JIX)
```

```
C UNDER-RELAX CM TO PREVENT INSTABILITY
      CM(JIY,JIX)=CMRLX2*CM(JIY,JIX)+(1.-CMRLX2)*CM2S(JIX)
C SAVE CM VALUE FOR RELAXATION
      CM2S(JIX)=CM(JIY,JIX)
C        (ASSIGN VM THE SPECIFIED EXIT PRESSURE (SAME AS EXIT 1))
      VM(JIY,JIX)=GPEXIT(JIX)
C
C **  SET THE VALUES OF U1,V1,W1,H1,KE,AND EP AT THE EXIT
C     IN CASE OF INFLOW.
C **  CM & VM HAVE BEEN SET ABOVE FOR PRESCRIBED MASS FLOW
C     CVAR  IS SET TO ZERO FOR ZERO DIFFUSION FLUX
      CVAR(JIY,JIX) = 0.0
C     SET VVAR TO EXTERNAL VALUE APPROPRIATE FOR THE VARIABLE
C     OUT OF LAZINESS AND FOR WANT OF ANYTHING BETTER, THE
C     VALUES BELOW ARE THE SAME AS FOR EXIT1
      VVAR(JIY,JIX) = 0.0
      IF (INDVAR.EQ.H1) VVAR(JIY,JIX) = GHEXIT
      IF (INDVAR.EQ.C1) VVAR(JIY,JIX) = H2OXIT
      IF (INDVAR.EQ.KE) VVAR(JIY,JIX) = GVALKE
      IF (INDVAR.EQ.EP) VVAR(JIY,JIX) = GVALEP
  584 CONTINUE
C     ****   ADD SOURCE TERM    ****
      CALL ADD(INDVAR,JIXE2F,JIXE2L,JIY,JIY,CELL,CM,VM,CVAR,VVAR,NY,NX)
C
C SUM THE MASSFLOW OUT EXIT2
      IF(.NOT.(ISWP.EQ.LSWEEP.AND.INDVAR.EQ.W1)) GO TO 587
      EMOUT2=0.0
      DO 585 JIX=JIXE2F,JIXE2L
      EMOUT2=EMOUT2-GD1(JIY,JIX)*GW1(JIY,JIX)*GAHIGH(JIY,JIX)*G
  585 CONTINUE
      IF(NX.EQ.1) EMOUT2=EMOUT2*2.*GPI/XULAST
C
C
C **** RESET CM AND VM SO THAT THEY DON'T INTERFERE WITH 'GWALL'
  587 DO 590 JIX=1,NX
      DO 590 JIY=1,NY
      CM(JIY,JIX)=0.0
      VM(JIY,JIX)=0.0
  590 CONTINUE
C
C
C
C
C *** CALCULATE TOTAL PRESSURES
      IF(INDVAR.NE.C1) GO TO 599
      DO 595 JIX=1,NX
      DO 595 JIY=1,NY
      UVEL=GU1(JIY,JIX)/GR(JIY)
      VVEL=GV1(JIY,JIX)
      IF(JIY.GT.1) VVEL=0.5*(VVEL+GV1(JIY-1,JIX))
      WVEL=GW1(JIY,JIX)
      IF((IZED.EQ.17.AND.JIY.EQ.40).OR.(IZED.EQ.13.AND.JIY.EQ.1))
                                                     GO TO 594
      IF(IZED.GT.1) WVEL=0.5*(WVEL+GW1L(JIY,JIX))
  594 VELSQ=UVEL*UVEL+VVEL*VVEL+WVEL*WVEL
  595 GPT(JIY,JIX)=GPT(JIY,JIX)+0.5*GD1(JIY,JIX)*VELSQ
C
C ***
  599 RETURN
C-------------------------------------------------------------------------
```

```
C     CHAPTER 6: CALLED AT THE END OF EACH VARIABLE-RECALCULATION
C     CYCLE COMMENCED AT CHAPTER 4. ITNO = ITERATION NUMBER.
C-----------------------------------------------------------------
  600 CONTINUE
      RETURN
C-----------------------------------------------------------------
C     CHAPTER 7: CALLED AT END OF EACH SLAB-WISE CALCULATION.
C-----------------------------------------------------------------
  700 CONTINUE
C ***
C PASS CALCULATED AUXILIARY VARIABLES BACK TO EARTH
      CALL SET(JMU1,1,NX,1,NY,GMU1,NY,NX)
      CALL SET(JPT,1,NX,1,NY,GPT,NY,NX)
      IF(.NOT.(IRHO1.EQ.-1)) RETURN
      CALL SET(JT1M,1,NX,1,NY,GT1M,NY,NX)
C     CALL SET(JRH2O,1,NX,1,NY,GRH2O,NY,NX)
CC    CALL SET(JRH2,1,NX,1,NY,GRH2,NY,NX)
C ***
C......AUTOPLOT FILE
      IF(MOD(ISWP,NPRMON).EQ.O.AND.IZED.EQ.IZMON) THEN
      JSTP=ISTP
      JSWP=ISWP
      CALL AUTMON(JSTP,JSWP)
      ENDIF
C......
      RETURN
C-----------------------------------------------------------------
C     CHAPTER 8: CALLED AT THE END OF EACH SWEEP
C     NOT ACCESSED IF PARABOLIC.
C-----------------------------------------------------------------
  800 CONTINUE
      RETURN
C-----------------------------------------------------------------
C     CHAPTER 9: CALLED AT THE END OF EACH TIME STEP
C     NOT ACCESSED IF PARABOLIC.
C-----------------------------------------------------------------
  900 CONTINUE
C ***
      WRITE(6,991)  EMOUT1,EMOUT2
  991 FORMAT(/////1X,1PE12.3,2X,72HCALCULATED (TOTAL) MASS OUTFLOW RATE
     .AT EXIT NEAR BLADE ROOTS (LBM/SEC).,/,
     ./1X,E12.3,2X,62HCALCULATED (TOTAL) MASS OUTFLOW RATE AT SECOND EXI
     .T (LBM/SEC).,/////)
C ***
      RETURN
C-----------------------------------------------------------------
C     CHAPTER 10: SET PHASE 1 DENSITY HERE WHEN IRHO1=-1 IN DATA.
C     SET CURRENT-Z 'SLAB' DENSITY, D1, IF MSLAB=.T.,
C     EG.  IF(MSLAB) CALL SET(D1,1,NX,1,NY,GD1,NY,NX).
C     SET NEXT LARGER-Z 'SLAB' DENSITY, D1H, IF HSLAB=.T. & PARAB=F
C     EG.  IF(HSLAB) CALL SET(D1H,1,NX,1,NY,GD1H,NY,NX).
C     SET D(LN(D1))/DP (IE. D1DP) FOR UNSTEADY FLOW,
C     EG.  IF(MSLAB) CALL SET(D1DP,1,NX,1,NY,GD1DP,NY,NX).
C-----------------------------------------------------------------
 1000 CONTINUE
C
C *** CALCULATE TEMP, DENSITY AND VISCOSITY OF HYDROGEN/WATER MIXTURE
      IF(MSLAB) GO TO 1001
      JH1=H1H
      JC1=C1H
```

```
      JD1=D1H
      JT1=T1H
      GO TO 1002
 1001 JH1=H1
      JC1=C1
      JD1=D1
      JT1=T1
C
 1002 CALL GET(JH1,GH1,NY,NX)
      CALL GET(JT1,GT1,NY,NX)
      CALL GET(JC1,GC1,NY,NX)
C
C DEDUCE TEMPERATURE OF MIXTURE FROM CALCULATED MIXTURE ENTHALPY
      CALL GTEMP(GH1,GC1,GT1,NY,NX,MSLAB)
C
C CALCULATE DENSITIES FROM DEDUCED MIXTURE TEMPERATURE
      CALL GRHO(GT1,GC1,GD1,GRH2O,GRH2,NY,NX,MSLAB)
C
C PASS CALCULATED MIXTURE DENSITY BACK TO EARTH
      CALL SET(JD1,1,NX,1,NY,GD1,NY,NX)
C
      IF(.NOT.MSLAB) RETURN
C
C CALCULATE THE LAMINAR VISCOSITY OF THE MIXTURE ("SET" IN CH. 12)
      CALL GVISC(GT1,GC1,GMU1L,NY,NX)
C
C SAVE MSLAB TEMPERATURES
      DO 1010 IX=1,NX
      DO 1010 IY=1,NY
 1010 GT1M(IY,IX)=GT1(IY,IX)
C ***
      RETURN
C-------------------------------------------------------------------
C     CHAPTER 11: SET PHASE 2 DENSITY HERE WHEN IRHO2=-1 IN DATA.
C     SET CURRENT-Z 'SLAB' DENSITY, D2, IF MSLAB=.T.,
C     EG.  IF(MSLAB) CALL SET(D2,1,NX,1,NY,GD2,NY,NX).
C     SET NEXT LARGER-Z 'SLAB' DENSITY, D2H, IF HSLAB=.T. & PARAB=F
C     EG.  IF(HSLAB) CALL SET(D2H,1,NX,1,NY,GD2H,NY,NX).
C     SET D(LN(D2))/DP FOR UNSTEADY FLOW,
C     EG.  IF(MSLAB) CALL SET(D2DP,1,NX,1,NY,GD2DP,NY,NX). ,
C-------------------------------------------------------------------
 1100 CONTINUE
      RETURN
C-------------------------------------------------------------------
C     CHAPTER 12: SET PHASE 1 VISCOSITY HERE WHEN IEMU1=-1 IN DATA.
C     SET CURRENT-Z 'SLAB' VISCOSITY (MU1), IF MSLAB=.T.,
C     EG.  IF(MSLAB) CALL SET(MU1,1,NX,1,NY,GVISC,NY,NX).
C     SET NEXT LARGER-Z 'SLAB' VISC. (MU1H), IF HSLAB=.T. & PARAB=F
C     EG.  IF(HSLAB) CALL SET(MU1H,1,NX,1,NY,GVSCH,NY,NX).
C
C     CHAPTER ALSO ACCESSED WHEN EMULAM=-1.0 IN DATA, SO THAT THE
C     LAMINAR VISCOSITY WHICH APPEARS IN WALL FUNCTIONS & IN THE
C     KE-EP TURBULENCE MODEL (IEMU1=2) MAY BE SET NON-CONSTANT.
C     SET CURRENT-Z 'SLAB' VALUE (MU1LAM) WHEN LAMMU=.T.,
C     EG.  IF(LAMMU) CALL SET(MU1LAM,1,NX,1,NY,GVSCL,NY,NX).
C-------------------------------------------------------------------
 1200 CONTINUE
C ***
C PASS CALCULATED MIXTURE VISCOSITY BACK TO EARTH
      IF(LAMMU) CALL SET(MU1LAM,1,NX,1,NY,GMU1L,NY,NX)
```

```
C ***
      RETURN
C---------------------------------------------------------------------
C     CHAPTER 13: SET EXCHANGE COEFFICIENT (E.C.) FOR VARIABLE
C     INDVAR WHEN SIGMA(INDVAR)=-1.0 IN DATA.
C     SET CURRENT-Z 'SLAB' E.C. (EXCO) IF MSLAB=.T.,
C     EG.  IF(MSLAB) CALL SET(EXCO,1,NX,1,NY,GEXCO,NY,NX).
C     SET NEXT SMALLER-Z 'SLAB' E.C. (EXCOL) IF LSLAB=.T.,
C     EG.  IF(LSLAB) CALL SET(EXCOL,1,NX,1,NY,GEXCOL,NY,NX).
C     SET NEXT LARGER-Z  'SLAB' E.C. (EXCOH) IF HSLAB=.T.,
C     EG.  IF(HSLAB) CALL SET(EXCOH,1,NX,1,NY,GEXCOH,NY,NX).
C     NOTE: FOR MSLAB, INDVAR=U1,..C4  FOR LSLAB, INDVAR=U1L,..C4L
C     & FOR HSLAB, INDVAR=U1H,..C4H. IF PARAB=.T. SET MSLAB ONLY.
C---------------------------------------------------------------------
 1300 CONTINUE
      RETURN
C---------------------------------------------------------------------
C     CHAPTER 14: SET INTER-PHASE FRICTION COEFFICIENT (CFP) HERE
C     WHEN ICFIP = -1 IN DATA          ITS UNITS = FORCE / (CELL * RELATIVE
C     SPEED OF PHASES).
C---------------------------------------------------------------------
 1400 CONTINUE
      RETURN
C---------------------------------------------------------------------
C     CHAPTER 15: SET INTER-PHASE MASS-TRANSFER RATE PER CELL (MDT)
C     HERE WHEN IMDOT = -1 IN DATA.
C---------------------------------------------------------------------
 1500 CONTINUE
      RETURN
C---------------------------------------------------------------------
C     CHAPTER 16: SET HERE PHASE 1 & 2 SATURATION ENTHALPIES
C     ( HST1 & HST2) WHEN IHSAT = -1 IN DATA.
C---------------------------------------------------------------------
 1600 CONTINUE
      RETURN
      END
      SUBROUTINE GTEMP(GH1,GC1,GT1,NY,NX,MSLAB)
C---------------------------------------------------------------------
C PURPOSE:- TO DETERMINE THE TEMPERATURE OF THE HYDROGEN/WATER MIXTURE
C           FROM THE CALCULATED MIXTURE ENTHALPY
C------------------------------------------------------------------------
C
C CURVE FITS OF THE ANALYTICAL FORM:-
C  HH2=CH2+BH2*T+AH2*T**2
C  HH2O=CH2O+BH2O*T+AH2O*T**2
C
C REFERENCES:-
C  H2:
C  H2O:
C
C RANGES OF VALIDITY:-
C  H2: T=170 TO 2000 DEG R
C  H2O: T=490 TO 2060 DEG R (BUT EXTRAPOLATION BELOW THIS O.K.)
C
C UNITS:-
C  H IN BTU/LBM AND T IN DEG R
C  H'S CONVERTED TO FT-LBF/SLUG BEFORE RETURN TO GROUND
C------------------------------------------------------------------------
      DIMENSION GH1(NY,NX),GC1(NY,NX),GT1(NY,NX),CH2O(6),BH2O(6),
     . AH2O(6),CH2(2),BH2(2),AH2(2)
```

```
      LOGICAL MSLAB
C----------------------------------------------------------------------
C
C HYDROGEN ENTHALPY CURVE FIT DATA
      DATA CH2/-357.6903,-45.88906/
      DATA BH2/4.468995,3.557702/
      DATA AH2/-5.92706E-4,-7.15694E-6/
C WATER ENTHALPY CURVE FIT DATA
      DATA CH2O
     ./-424.5938,2289.552,-7363.69,599.5881,-307.5449,-96.3053/
      DATA BH2O
     ./0.82414,-4.577089,6.913,-1.27177,1.190721,1.285063/
      DATA AH2O
     ./1.3067E-4,2.815249E-3,0.0,1.369267E-3,0.0,-1.75707E-4/
C
C UNIT CONVERSION FACTOR
      DATA CONVH/25036.52/
C CONVH = CONVERSION FACTOR FROM BTU/LBM TO FT.LBF/SLUG = 778.16*G
C
      DATA TINY/1.E-10/
C----------------------------------------------------------------------
C
      DO 50 IX=1,NX
      DO 50 IY=1,NY
      ENTH=GH1(IY,IX)/CONVH
      IF(ENTH.LE.TINY) GO TO 35
      TEMP=GT1(IY,IX)
      XH2O=GC1(IY,IX)
C
C DETERMINE WHICH OF THE SIX WATER ENTHALPY/TEMP CURVE FITS TO USE
      IF(TEMP.GE.10..AND.TEMP.LT.975.) GO TO 12
      IF(TEMP.GE.975..AND.TEMP.LT.1184.6) GO TO 13
      IF(TEMP.GE.1184.6.AND.TEMP.LT.1223.3) GO TO 14
      IF(TEMP.GE.1223.3.AND.TEMP.LT.1281.4) GO TO 15
      IF(TEMP.GE.1281.4.AND.TEMP.LT.1400.) GO TO 16
      IF(TEMP.GE.1400..AND.TEMP.LT.2000.) GO TO 161
      GO TO 90
 12   IHW=1
      GO TO 17
 13   IHW=2
      GO TO 17
 14   IHW=3
      GO TO 17
 15   IHW=4
      GO TO 17
 16   IHW=5
      GO TO 17
 161  IHW=6
C
C DETERMINE WHICH OF THE TWO HYDROGEN ENTHALPY/TEMP CURVE FITS TO USE
 17   IF(TEMP.GE.10..AND.TEMP.LT.508.) GO TO 18
      IF(TEMP.GE.508..AND.TEMP.LT.2000.) GO TO 19
 18   IHH=1
      GO TO 20
 19   IHH=2
C
C SOLVE QUADRATIC IN T TO DETERMINE LOCAL MIXTURE TEMPERATURE (DEG R)
 20   CC=CH2(IHH)*(1.-XH2O)+XH2O*CH2O(IHW)-ENTH
      BB=BH2(IHH)*(1.-XH2O)+XH2O*BH2O(IHW)
      AA=AH2(IHH)*(1.-XH2O)+XH2O*AH2O(IHW)
```

```
C
      IF(ABS(AA).LE.TINY) GO TO 28
      ROOT=SQRT(BB*BB-4.*AA*CC)
      T1=(-BB+ROOT)/(2.*AA)
      T2=(-BB-ROOT)/(2.*AA)
      IF(AA.LT.O.) GO TO 27
C AA POSITIVE
      TEMP=AMAX1(T1,T2)
      GO TO 40
C AA NEGATIVE
 27   TEMP=AMIN1(T1,T2)
      GO TO 40
C AA ZERO
 28   TEMP=-CC/BB
      GO TO 40
C
C SET TEMP TO ZERO IN FULLY BLOCKED CELLS
 35   TEMP=0.0
C
 40   GT1(IY,IX)=TEMP
C
 50   CONTINUE
C
      RETURN
C-------------- DE-BUG -----------------------------------------
 90   WRITE(6,91)
      WRITE(6,92) IY,IX,TEMP,ENTH,MSLAB
 92   FORMAT(//1X,2I4,1P2E12.3,1L1)
 91   FORMAT(///1X,88H*** TEMPERATURE OUT OF RANGE OF CURVE FITS IN SUBR
      OUTINE GTEMP. EXECUTION TERMINATED *** )
      STOP
      END
      SUBROUTINE GRHO(GT1,GC1,GD1,GRH2O,GRH2,NY,NX,MSLAB)
C-------------------------------------------------------------------
C PURPOSE:- TO CALCULATE THE DENSITIES OF THE MIXTURE, HYDROGEN AND
C           WATER AT THE MIXTURE TEMPERATURE DERIVED FROM THE
C           CALCULATED MIXTURE ENTHALPY (IN SUBROUTINE GTEMP)
C-------------------------------------------------------------------
C
C CURVE FITS OF THE ANALYTICAL FORM:-
C  RH2=EXP(CH2+BH2*LN(TEMP)+AH2*LN(TEMP)**2)
C  RH2O=FH2O+EH2O*T+DH2O*T**2+CH2O*T**3+BH2O*T**4+AH2O*T**5
C
C REFERENCES:-
C  H2:
C  H2O:
C
C RANGES OF VALIDITY:-
C  H2: T=170 TO 2000 DEG R (BUT EXTRAPOLATION DOWN TO 150 O.K.)
C  H2O: T=490 TO 2060 DEG R
C  (NB. H2O AT T BELOW 490 GIVEN DENSITY OF H2O AT FREEZING)
C
C UNITS:-
C  RHO IN LBM/FT**3  AND T IN DEG R
C  RHO'S CONVERTED TO SLUG/CU FT BEFORE RETURNING TO GROUND
C-------------------------------------------------------------------
      DIMENSION GT1(NY,NX),GRH2O(NY,NX),GRH2(NY,NX),GC1(NY,NX),
     . GD1(NY,NX),FH2O(4),EH2O(4),DH2O(4),CH2O(4),BH2O(4),AH2O(4)
      LOGICAL MSLAB
C-------------------------------------------------------------------
```

```
C
C HYDROGEN DENSITY CURVE FIT DATA
      DATA CH2,BH2,AH2/4.579578,-0.5199177,-2.86885E-2/
C WATER DENSITY CURVE FITS DATA
      DATA FH2O/-82.117,-2177.783,119.1372,30.17724/
      DATA EH2O/0.62353,7.12733,-4.770357E-2,-2.573409E-2/
      DATA DH2O/-6.77963E-4,-4.54395E-3,-1.00694E-4,6.195714E-6/
      DATA CH2O/-3.41207E-7,-1.91391E-6,5.516186E-8,0.0/
      DATA BH2O/9.23406E-10,1.686E-9,0.0,0.0/
      DATA AH2O/-3.9688E-13,0.0,0.0,0.0/
C
C UNIT CONVERSION FACTOR
      DATA CONVR/32.174/
C CONVR = G = 32.174, TO CONVERT LBM/FT**3 TO SLUG/FT**3
C
      DATA RH2OF/62.578/
C RH2OF = WATER DENSITY AT FREEZING (APPROX 490 DEG R), IN LBM/FT**3
      DATA TINY/1.E-10/
C-----------------------------------------------------------------
C
      DO 20 IX=1,NX
      DO 20 IY=1,NY
      TEMP=GT1(IY,IX)
      CONC=GC1(IY,IX)
      IF(TEMP.LE.TINY) GO TO 18
      TEMPLN=ALOG(TEMP)
C
C DETERMINE WHICH OF,THE 4 WATER DENSITY/TEMPERATURE CURVE FITS TO USE
      IF(TEMP.GE.10..AND.TEMP.LT.1180.) GO TO 12
      IF(TEMP.GE.1180..AND.TEMP.LT.1250.) GO TO 13
      IF(TEMP.GE.1250..AND.TEMP.LT.1380.) GO TO 14
      IF(TEMP.GE.1380..AND.TEMP.LT.2000.) GO TO 141
      GO TO 50
 12   IT=1
      GO TO 15
 13   IT=2
      GO TO 15
 14   IT=3
      GO TO 15
 141  IT=4
C
 15   IF(TEMP.LE.490.) GO TO 16
C DENSITY OF WATER (IN SLUG/FT**3)
      RH2O=(FH2O(IT)+EH2O(IT)*TEMP+DH2O(IT)*TEMP**2
     . +CH2O(IT)*TEMP**3+BH2O(IT)*TEMP**4+AH2O(IT)*TEMP**5)/CONVR
      GO TO 17
C TRAP WATER DENSITY TO ITS VALUE AT FREEZING FOR TEMPS BELOW FREEZING
 16   RH2O=RH2OF/CONVR
C
C DENSITY OF HYDROGEN (IN SLUG/FT**3)
 17   RH2=(EXP(CH2+BH2*TEMPLN+AH2*TEMPLN**2))/CONVR
      GO TO 19
C
C SET DENSITIES TO TINY IN FULLY BLOCKED CELLS
 18   RH2O=TINY
      RH2=TINY
C
C CALCULATE THE MIXTURE DENSITY
 19   GD1(IY,IX)=1./(CONC/RH2O+(1.-CONC)/RH2)
C
```

```
      IF(.NOT.MSLAB) GO TO 20
C SAVE MSLAB DENSITIES FOR PRINTOUT FROM EARTH
      GRH2D(IY,IX)=RH2O
      GRH2(IY,IX)=RH2
C
 20   CONTINUE
C
      RETURN
C--------------- DE-BUG ---------------------------------------------
 50   WRITE(6,51)
      WRITE(6,52) IY,IX,TEMP,CONC,MSLAB
 51   FORMAT(///1X,87H*** TEMPERATURE OUT OF RANGE OF CURVE FITS IN SUBR
     .OUTINE GRHO. EXECUTION TERMINATED *** )
 52   FORMAT(//1X,2I4,1P2E12.3,1L1)
      STOP
      END
      SUBROUTINE GVISC(GT1,GC1,GMU1L,NY,NX)
C-------------------------------------------------------------------
C PURPOSE:- TO CALCULATE THE LAMINAR VISCOSITY OF THE MIXTURE
C-------------------------------------------------------------------
C
C CURVE FITS OF THE ANALYTICAL FORM:-
C  EMUH2=DH2+CH2*TEMP+BH2*TEMP**2+AH2*TEMP**3
C  EMUH2O=EXP(FH2O+EH2O*TEMP+DH2O*TEMP**2+CH2O*TEMP**3+BH2O*TEMP**4
C                                                  +AH2O*TEMP**5)
C
C REFERENCES:-
C  H2:
C  H2O:
C
C RANGES OF VALIDITY:-
C  H2: T=170 TO 2000 DEG R (BUT EXTRAPOLATION DOWN TO 150 O.K.)
C  H2O: T=490 TO 1752 DEG R
C  (NB. H2O AT T BELOW 490 GIVEN VISCOSITY OF H2O AT FREEZING)
C
C UNITS:-
C  EMU IN LBM/FT.SEC AND T IN DEG R
C  EMU'S CONVERTED TO SLUG/FT.SEC BEFORE RETURNING TO GROUND
C-------------------------------------------------------------------
      DIMENSION GT1(NY,NX),GC1(NY,NX),GMU1L(NY,NX)
C-------------------------------------------------------------------
C
C HYDROGEN VISCOSITY CURVE FIT DATA
      DATA DH2,CH2,BH2,AH2/0.4989,-5.4575E-5,5.1824E-7,-1.4948E-10/
C WATER VISCOSITY CURVE FIT DATA
      DATA FH2O,EH2O,DH2O,CH2O,BH2O,AH2O/20.5532,-6.52199E-2,
     . 3.2726E-5,6.6687E-8,-8.3627E-11,2.6237E-14/
      DATA FH2OA,EH2OA/6.5334525E-3,1.11E-5/
C
C UNIT CONVERSION FACTOR
      DATA CONVM/32.174/
C CONVM = G = 32.174, TO CONVERT LBM TO SLUG
      DATA EMULWF/1.2446E-3/
C EMULWF=LAM VISCOSITY OF WATER AT FREEZING (490 DEG R)             IN LBM/FT.SEC
C-------------------------------------------------------------------
C
      DO 20 IX=1,NX
      DO 20 IY=1,NY
      TEMP=GT1(IY,IX)
      CONC=GC1(IY,IX)
```

```
C
C VISCOSITY OF HYDROGEN (IN LBM/FT.SEC)
      EMUH2=(DH2+CH2*TEMP+BH2*TEMP**2+AH2*TEMP**3)*1.E-5
C
      IF(TEMP.LE.490.) GO TO 17
      IF(TEMP.GE.1392.) GO TO 16
C VISCOSITY OF WATER (IN LBM/FT.SEC)
      EMUH2O=EXP(FH2O+EH2O*TEMP+DH2O*TEMP**2+CH2O*TEMP**3+BH2O*TEMP**4
     +               +AH2O*TEMP**5)*1.E-3
      GO TO 18
 16   EMUH2O=FH2OA+EH2OA*TEMP
      GO TO 18
C
C TRAP WATER VISCOSITY TO ITS VALUE AT FREEZING FOR TEMPS BELOW FREEZING
 17   EMUH2O=EMULWF
C
C CALCULATE THE MIXTURE VISCOSITY (IN SLUGS/FT.SEC)
 18   GMU1L(IY,IX)=1./(CONC/EMUH2O+(1.-CONC)/EMUH2)/CONVM
C
 20   CONTINUE
C
      RETURN
      END
C--- OCTOBER.1984,  CHAM (NA) GROUND SUBPROGRAM "GWALL", TO FACILITATE
C    THE SETTING OF AN UNLIMITED NUMBER OF WALL SURFACES IN THE SPRING
C    1983 VERSION OF PHOENICS.
C
      SUBROUTINE GWALL(JVAR,JIXF,JIXL,JIYF,JIYL,JIZ,GWTYPE,
                           GUWALL,GVWALL,GWWALL,GHWALL,GDELTA)
C-----------------------------------------------------------------
$INCLUDE 9,CMNGUSSI.FTN/G (NLIST)
$INCLUDE 9,GUSSEQUI.FTN/G (NLIST)
C-------------------------------------------------------------------
C
C PURPOSE:- TO COMPUTE CVAR (=GCVAR) AND VVAR (=GVVAR) FOR TURBULENT
C           AND LAMINAR WALL FUNCTIONS
C
C-----------------------------------------------------------------
C ANY QUESTIONS (OR PROBLEMS) ON THE USE OF THIS SUBPROGRAM  SHOULD BE
C ADDRESSED TO:
C  L.W. KEETON, CHAM (NA) INC.,
C  1525-A SPARKMAN DRIVE,
C  HUNTSVILLE, AL 35802, U.S.A.
C  TEL: (205) 830-2620
C-----------------------------------------------------------------
C RESTRICTIONS:-
C  1. GWALL IS NOT VALID FOR 2-FLUID MODEL CALCULATIONS.
C  2. PROVISION FOR A MOVING GRID HAS NOT YET BEEN INCLUDED.
C-----------------------------------------------------------------
C NOTES:-
C  1. THIS GROUND SUBPROGAM IS INTENDED TO FACILITATE THE SETTING OF
C     APPOPRIATE WALL BOUNDARY CONDITIONS VIA GROUND FOR THOSE CASES
C     WHEN THE 10 REGIONS OF THE SATELLITE ARE INSUFFICIENT. INSTEAD
C     OF USING A SPECIAL REGION TO SPECIFY THE PRESENCE OF A WALL THE
C     GROUND USER SUBPROGRAM GWALL CAN NOW BE USED INSTEAD. THE MODELS
C     EMPLOYED ARE IDENTICAL TO THOSE CURRENTLY INCORPORATED IN EARTH
C     SUBPROGRAM "WALL" (SEE NOTE 3 BELOW).
C
C  2. TO FACILITATE CROSS-CHECKING, WHERE FEASABLE, ALL VARIABLE NAMES
C     AND CODING IN GWALL ARE SIMILAR TO THOSE USED IN EARTH SUBPROGRAM
```

```
C      "WALL".
C
C  3. THE WALL BOUNDARY CONDITION TREATMENT USED HEREIN IS EXACTLY AS
C     DESCRIBED IN THE SPRING 1983 PHOENICS USER'S MANUAL (CHAM TR/75),
C     ON PAGES 3.2-47 TO 49. ITS MAIN FEATURES ARE OUTLINED BELOW.
C
C  4. THE QUANTITIES GCVAR (=GCOEFF) AND GVVAR (=GVALUE) COMPUTED IN
C     GWALL ARE EQUIVALENT TO THE "COEFFICIENT" AND "VALUE" QUANTITIES
C     (CP1R1, VP1R1 ETC.) DISCUSSED IN THE USER'S MANUAL ON PAGES 3.2-41
C     TO 49. THE GWALL ARRAYS GCVAR AND GVVAR ARE IDENTICALLY EQUIVALENT
C     TO THE GROUND ARRAYS CVAR AND VVAR, RESPECTIVELY, AND MUST
C     BE EQUIVALENCED TO EACH OTHER IN GROUND (SEE NOTES 5 AND 6 BELOW).
C     SPECIFIC EXAMPLES OF THE USES OF CVAR AND VVAR IN GROUND ARE
C     GIVEN IN SECTION 4 OF THE USER'S MANUAL ON PAGE 4.3-2.
C
C  5. TO USE GWALL, THE USER MUST PROVIDE, IN GROUND CH. 5
C     --------- (FOR EACH SEPARATE REGION OF WALL, AT EACH IZ-SLAB):-
C               ---------------------------------------------------
C      A. VIA SUBROUTINE ARGUMENTS: THE VARIABLE INDEX (E.G. U1,KE,..),
C         JVAR       THE FIRST AND LAST IX AND IY-CELL COORDINATES OF THE
C         REGION OF CELLS CONTAINING (OR NEIGHBOURING) THE WALL, JIXF,
C         JIXL,JIYF,JIYL          THE CURRENT IZ-SLAB COORDINATE, JIZ  THE WALL
C         'TYPE' (E.G. NORTH,SOUTH,..), GWTYPE                         THE 3 COMPONENTS OF WALL
C         VELOCITY, GUWALL,GVWALL,GWWALL (SEE NOTE 8 BELOW)            THE WALL
C         ENTHALPY, GHWALL        AND THE PERPENDICULAR DISTANCE FROM THE
C         WALL, GDELTA (SEE NOTE 12 BELOW)
C      B. VIA "CALL GET" STATEMENTS IN GROUND CH. 5: THE 3 COMPONENTS OF
C         FLUID VELOCITIES, GU1,GV1,GW1                                AND THE DENSITIES, GD1
C      C. VIA "CALL GET1D" STATEMENTS IN GROUND CH.5: THE CELL WIDTHS IN
C         THE 3 COORDINATE DIRECTIONS, GDXU,GDYV,GDZW                  AND THE RADII AT
C         EACH CELL GRID NODE, GR       AND
C      D. VIA LOCAL CALCULATION IN GROUND CH.5 (SEE NOTES 9 AND 10): THE
C         CURRENT IZ-SLAB LAMINAR VISCOSITIES, GMU1L.
C
C     THE REQUIRED CVAR AND VVAR VALUES ARE THEN RETURNED TO GROUND FOR
C     EACH VARIABLE, THROUGH COMMON/WALLG/, VIA THE GCVAR AND GVVAR
C     ARRAYS (WHICH ARE EQUIVALENCED TO CVAR AND VVAR), RESPECTIVELY.
C
C  6. SUBROUTINE GWALL SHOULD BE CALLED, SEPARATELY, FOR EACH CONTINUOUS
C     REGION OF CELLS AT THE CURRENT SLAB CONTAINING (OR NEIGHBOURING)
C     A WALL. CONSEQUENTLY, THE QUANTITIES IN GROUP A ABOVE MUST BE
C     PROVIDED ON A REGION-BY-REGION BASIS. THE VARIABLES IN GROUPS B,
C     C AND D, HOWEVER, SHOULD BE OBTAINED ONCE ONLY FOR EACH IZ-SLAB
C     VIA "CALL GET" OR "CALL GET1D" STATEMENTS (SEE USER'S MANUAL, PAGE
C     4.2-26) OR LOCAL CALCULATION, RESPECTIVELY, AS DESCRIBED IN NOTE 5
C     ABOVE. THESE VARIABLES ARE STORED IN GROUND IN THE LOCAL ARRAY
C     NAMES GIVEN AND ARE THEN PASSED TO SUBROUTINE GWALL VIA THE
C     GROUND/WALL COMMON BLOCK "WALLG". THE NAMES AND SEQUENCE OF THE
C     ELEVEN ARRAYS IN COMMON/WALLG/ MUST NOT BE ALTERED BY THE USER.
C     FURTHERMORE, THEY MUST BE APPROPRIATELY AND CONSISTENTLY
C     DIMENSIONED IN BOTH GROUND AND GWALL, VIZ: GU1,GV1,GW1,GD1,GMU1L,
C     GCVAR,GVVAR(NY,NX)          GDXU(NX)                             GDYV,GR(NY)  AND GDZW(NZ). IN
C     ADDITION, THE GCVAR AND GVVAR ARRAYS MUST BE EQUIVALENCED IN
C     GROUND TO CVAR AND VVAR, RESPECTIVELY, SO THAT THEIR VALUES AS
C     CALCULATED IN GWALL CAN BE PASSED BACK TO GROUND (VIA
C     COMMON/WALLG/) FOR USE IN THE CORRESPONDING CALLS TO "ADD".
C
C  7. GROUND SUBPROGRAM GWALL MUST BE CALLED FOR EACH CELL OR REGION
C     OF CELLS WHERE A SPECIAL WALL BOUNDARY TREATMENT IS REQUIRED.
C     GWALL MUST BE CALLED SEPARATELY FOR EACH VARIABLE INFLUENCED BY
```

```
C     THE WALL, I.E. FOR U1,V1,W1,H1,KE AND EP AS NECESSARY, FOLLOWED BY
C     A CALL TO ADD TO INCLUDE THE APPROPRIATE CVAR AND VVAR SOURCE
C     MODIFICATION TO THE RELEVANT F.D. EQUATIONS. IT SHOULD BE STRESSED
C     THAT EVERY CALL TO GWALL IN GROUND FOR ANY VARIABLE MUST BE
C     FOLLOWED IMMEDIATELY BY A CORRESPONDING CALL TO ADD, I.E. WITH
C     SAME VARIABLE INDEX, REGION OF CELLS AND CELL TYPE (UNLESS EITHER:
C     A. AREAS CALCULATED LOCALLY - SEE NOTE 9 BELOW                        OR B. THE VARIABLE
C     INDEX (JVAR) IS EITHER KE OR EP, IN WHICH CASE THE CALL TO ADD
C     TYPE MUST ALWAYS BE "CELL" (DUE TO THEIR VALUES BEING FIXED)).
C     NOTE ALSO THAT GWALL MUST BE CALLED SEPARATELY (AND REPEATEDLY)
C     FOR EACH DIFFERENT WALL TYPE (E.G. NORTH,HIGH,..) THAT MIGHT
C     OCCUR IN ANY CELL OR CELLS.
C
C  8. FOR POLAR COORDINATES (WHEN UR IS SOLVED-FOR RATHER THAN U) THE
C     GUWALL AND GU1 QUANTITIES ARE ASSUMED (IN GWALL) TO BE
C     ANGULAR VELOCITY (OMEGA = U/R) AND UR-AT-THE-CELL, RESPECTIVELY.
C     THE GUWALL (=OMEGA) IS THEN MULTIPLIED (WITHIN GWALL, WHENEVER
C     CARTES=.F.) BY THE LOCAL RADIUS**2 TO GIVE THE REQUIRED LOCAL
C     UR-AT-THE-WALL VALUE. IF THIS IS NOT APPROPRIATE FOR ANY
C     PARTICULAR PROBLEM THEN THE MULTIPLICATION BY R**2 SHOULD BE
C     SUPPRESSED BY THE USER IN GWALL AND THE DESIRED GUWALL VALUE,
C     RATHER THAN OMEGA, SHOULD THEN BE FED VIA THE GUWALL ARGUMENT.
C
C  9. GCVAR IS NOT MULTIPLIED BY THE APPROPIATE AREAS OF CONTACT WITHIN
C     GWALL. THIS MUST BE DONE BY THE USER WITHIN GROUND CH. 5 ITSELF,
C     FOR EACH PARTICULAR VARIABLE, AS NECESSARY, AFTER CVAR
C     HAS BEEN RETURNED FROM GWALL. THIS CAN BE DONE EITHER BY
C     EXPLICITLY CALCULATING THE APPROPRIATE AREAS OR VIA THE "TYPE"
C     SPECIFICATION IN THE CALL TO ADD. IF THE AREAS ARE CALCULATED
C     LOCALLY (AND THEN MULTIPLIED TO CVAR) THEN THE CALL TO ADD "TYPE"
C     SHOULD BE PER "CELL" FOR EVERY VARIABLE (AND REGION) SO TREATED.
C
C  10.THE LAMINAR VISCOSITIES AT EACH IZ-SLAB WHERE GWALL IS TO BE CALLED
C     MUST BE SET-UP AND STORED IN THE LOCAL ARRAY GMU1L, WITHIN GROUND
C     CH.5 (SEE NOTE 5 ABOVE). THESE CAN BE SET EITHER: A. TO A CONSTANT
C     VALUE EVERYWHERE (E.G. EMULAM)   OR, B. DETERMINED LOCALLY (BASED
C     ON LOCAL CONDITIONS) AS DESCRIBED IN NOTE 11 BELOW.
C
C  11.THE LAMINAR VISCOSITIES USED IN THE WALL FUNCTIONS WITHIN THE
C     PHOENICS SUBPROGRAM WALL CAN BE SET TO A NON-CONSTANT VALUE BY
C     SETTING EMULAM= -1. IN THE SATELLITE, AND INSERTING APPOPRIATE
C     CODING IN GROUND CH.12, AS DESCRIBED ON PAGE 4.3-14 OF THE USER'S
C     MANUAL. IF GWALL IS TO BE USED, HOWEVER, THE CODING FOR VARYING
C     LAMINAR VISCOSITY (LOCAL ARRAY: GMU1L) MUST BE INCLUDED IN CH.5
C     (AND NOT CH. 12). THIS IS BECAUSE THE CALL TO CH.12 ORIGINATES
C     FROM WITHIN THE PHOENICS WALL SUBPROGRAM, WHICH WILL NOT BE
C     ACCESSED IF GWALL IS USED INSTEAD. THE CALCULATED GMU1L VALUES
C     MUST, HOWEVER, STILL BE "SET" IN CH. 12.
C
C  12.FOR THOSE CASES WHEN THE WALLS ARE ALIGNED WITH CELL FACES, THE
C     PERPENDICULAR DISTANCES FROM THE WALL (GDELTA) ARE NORMALLY
C     EXACTLY EQUAL TO ONE HALF THE APPROPRIATE CELL WIDTH (I.E. DX/2
C     ETC.). WHEN SUCH A TREATMENT IS APPROPRIATE THE USER OF GWALL
C     NEEDS SIMPLY TO SET GDELTA TO ANY NEGATIVE VALUE (E.G. -1.) AND
C     THE APPROPRIATE HALF-CELL WIDTH(S) WILL THEN BE AUTOMATICALLY
C     USED INSIDE GWALL. IF THIS TREATMENT IS NOT DESIRED, HOWEVER,
C     THE APPROPRIATE NORMAL DISTANCES MUST BE SPECIFIED AS AN
C     ARGUMENT (CELL-BY-CELL OR REGION-BY-REGION) IN THE GWALL CALL
C     STATEMENT. HOWEVER, IT SHOULD BE NOTED THAT, IN POLAR GEOMETRIES
C     WHEN THE NORMAL DISTANCE FROM AN EAST OR WEST WALL IS BEING
```

```
C     SPECIFIED EXPLICITLY ONLY THE ANGLE (I.E. DX/2) BETWEEN THE GRID
C     NODE AND WALL SURFACE NEEDS TO BE SPECIFIED AS THE CORRESPONDING
C     NORMAL DISTANCE IS DEDUCED FROM WITHIN GWALL ITSELF BY MULTIPLYING
C     THE SPECIFIED ANGLE BY THE LOCAL RADIUS.
C
C-----------------------------------------------------------------
C DESCRIPTION OF THE WALL BOUNDARY TREATMENT EMPLOYED:-
C
C  A. THE TURBULENT WALL SHEAR STESS IS CALCULATED FROM A WALL FUNCTION
C     BASED ON THE LOGARITHMIC LAW OF THE WALL (REF: LAUNDER AND
C     SPALDING (1972), "MATHEMATICAL MODELS OF TURBULENCE"). THE
C     SO-CALLED "LOG LAW OF THE WALL" IS GIVEN BY:
C
C      UPLUS=(1./AK)*LOG(EWALL*YPLUS)
C
C     WHERE
C      UPLUS=UGRID/USTAR
C      YPLUS=RHO*USTAR*DELTA/EMUL
C     AND
C      AK=VON KARMANN CONSTANT (=0.435)
C      EWALL=EMPIRICAL CONSTANT (=9.0 FOR SMOOTH WALL)
C      UGRID=VELOCITY AT THE NEAR-WALL GRID NODE
C      USTAR=WALL SHEAR VELOCITY (=SQRT(TAUW/RHO))
C      RHO=DENSITY
C      DELTA=PERPENDICULAR DISTANCE OF NEAR-WALL NODE FROM WALL
C      EMUL=LAMINAR VISCOSITY
C
C     THE TURBULENT WALL SHEAR STRESS IS THEN GIVEN BY:
C
C      TAUW=EMUL*YPLUS/UPLUS
C
C     THE QUANTITIES YPLUS AND UPLUS ARE COMMONLY REFERRED TO AS
C     THE NORMALISED DISTANCE AND VELOCITY, RESPECTIVELY. IN THE
C     CODING OF GWALL BELOW UPLUS AND YPLUS ARE NOT SEEN EXPLICITLY,
C     HOWEVER, THEY CAN BE DEDUCED AS FOLLOWS:
C
C      UPLUS=1./SQRT(GS)
C      YPLUS=REYNO*SQRT(GS)=REYNO*YPLUS
C
C     WHERE
C      SQRT(GS)=USTAR/UGRID
C     THAT IS
C      GS=TAUW/(RHO*UGRID**2)
C     AND
C      REYNO=RHO*UGRID*DELTA/EMUL
C
C  B. DUE TO THE IMPLICIT RELATIONSHIP BETWEEN UPLUS AND YPLUS THEIR
C     VALUES ARE OBTAINED ITERATIVELY. THE ITERATIVE PROCEDURE'S
C     INITIAL GUESS FOR GS (WHERE UPLUS=1./SQRT(GS)) IS TAKEN FROM
C     KUTATELADZE AND LEONTIEV "TURBULENT BOUNDARY LAYERS", VIZ:
C
C      GS=A*(REYNOLDS NO)**B
C
C     WHERE A (=8.74) AND B (=0.142857) ARE TAKEN FROM TABLE 3-1 OF
C     THE ABOVE REFERENCE.
C
C  C. THE TURBULENT KINETIC ENERGY AND DISSIPATION RATE VALUES ARE THEN
C     FIXED AT THE VALUES WHICH WOULD PREVAIL AT THE NEAR-WALL GRID
C     NODES IF THE SUPPOSED UNIVERSAL LOGARITHMIC VELOCITY PROFILE
C     PREVAILED.
```

```
C
C  D. FOR LAMINAR WALL SHEAR STRESS (REYNOLDS NO .LE. 132.25) A
C     LINEAR VELOCITY PROFILE IS ASSUMED NEAR TO THE WALL.
C
C  E. THE WALL HEAT TRANSFER RATE IS EVALUATED FROM THE CHILTON-
C     COLBURN FORM OF THE REYNOLD'S ANALOGY, AS DESCRIBED IN THE
C     PHOENICS USER'S MANUAL, PAGE 3.2-48.
C
C---------------------------------------------------------------------
      COMMON/WALLG/ GU1(40,8),GV1(40,8),GW1(40,8),GD1(40,8),GMU1L(40,8),
     . GCVAR(40,8),GVVAR(40,8),GDXU(8),GDYV(40),GR(40),GDZW(28)
C--------------------------------------------------------------------
      DATA JVISIT,JSOUTH,JU1,JV1,JW1,JKE,JEP,JH1/0,4,3,5,7,12,13,14/
      DATA GAFRIC,GBFRIC,GAK,GEWALL,GTAUDK/8.74,0.142857,0.435,9.0,0.3/
      DATA GREAT/1.E10/
C--------------------------------------------------------------------
CHAPTER 0                    PRELIMINARIES
C---------------------------------------------------------------------
      IF(JVISIT.GT.0) GO TO 10
      JVISIT=JVISIT+1
      GACON=1./GAFRIC**(2./(1.+GBFRIC))
      GBCON=(1.-GBFRIC)/(1.+GBFRIC)
      GAKRA=1./SIGMA(24)**0.666667
      GWALC=0.16433/GAK
C
 10   DO 390 JIX=JIXF,JIXL
      DO 390 JIY=JIYF,JIYL
C
      RWDRP2=1.
      GRGRID=GR(JIY)
      JWTYPE=IFIX(GWTYPE)
      GDEL=GDELTA
C
      GWALLU=GUWALL
C FOLLOWING STATEMENT SHOULD BE SUPPRESSED (IE. "COMMENTED OUT") IF
C NOT APPROPRIATE (SEE NOTE 8 ABOVE)
      IF(.NOT.CARTES) GWALLU=GUWALL*GRGRID**2
C
      GOTO (13,13,12,12,11,11), JWTYPE
C DETERMINE DISTANCE FROM WALL AND RELATIVE VELOCITY PARALLEL TO WALL
C
C HIGH OR LOW WALL
 11   IF(GDELTA.LT.Q.) GDEL=0.5*GDZW(JIZ)
      GV1WAL=GVWALL
      GV2WAL=GWALLU/GRGRID
      GV1CEL=GV1(JIY,JIX)
      GV2CEL=GU1(JIY,JIX)/GRGRID
      GO TO 17
C
C NORTH OR SOUTH WALL
 12   IF(GDELTA.LT.0.) GDEL=0.5*GDYV(JIY)
      GV1WAL=GWWALL
      GV2WAL=GWALLU/GRGRID
      GV1CEL=GW1(JIY,JIX)
      GV2CEL=GU1(JIY,JIX)/GRGRID
      IF(.NOT.(.NOT.CARTES.AND.JVAR.EQ.JU1)) GO TO 17
C WHEN SOLVING FOR UR, MODIFY U SO THAT NEAR-WALL U IS EMPLOYED
      GFAC=0.5
      IF(JWTYPE.EQ.JSOUTH) GFAC=-0.5
      RWDRP2=(1.+GFAC*GDYV(JIY)/GRGRID)**2
```

```
      GO TO 17
C
C EAST OR WEST WALL
 13   IF(GDELTA.LT.O.) GDEL=0.5*GDXU(JIX)
      IF(.NOT.CARTES) GDEL=GDEL*GRGRID
      GV1WAL=GWWALL
      GV2WAL=GVWALL
      GV1CEL=GW1(JIY,JIX)
      GV2CEL=GV1(JIY,JIX)
C
C CALCULATE RELATIVE VELOCITY OF FLUID PARALLEL TO WALL
  17  CONTINUE
      GSPEED=SQRT((GV1CEL-GV1WAL)**2+(GV2CEL-GV2WAL)**2)
C
C SET APPROPRIATE WALL VALUE (I.E. ITS VELOCITY OR ENTHALPY)
      GVPHI=0.0
      IF(JVAR.EQ.JKE.OR.JVAR.EQ.JEP) GO TO 18
      IF(JVAR.EQ.JU1) GVPHI=GWALLU/RWDRP2
      IF(JVAR.EQ.JV1) GVPHI=GVWALL
      IF(JVAR.EQ.JW1) GVPHI=GWWALL
      IF(JVAR.EQ.JH1) GVPHI=GHWALL
C
 18   GRHO=GD1(JIY,JIX)
      GDELMU=GDEL/GMU1L(JIY,JIX)
      GREYNO=GRHO*GSPEED*GDELMU
      GVALUE=0.0
      GCOEFF=0.0
C
      IF(JVAR.EQ.JKE.OR.JVAR.EQ.JEP) GO TO 100
      IF(JVAR.EQ.JU1.OR.JVAR.EQ.JV1.OR.JVAR.EQ.JW1) GO TO 300
      IF(JVAR.EQ.JH1) GO TO 301
      GO TO 350
C---------------------------------------------------------------------
CHAPTER 1      FIX TURBULENT KINETIC ENERGY (KE)
C---------------------------------------------------------------------
 100  GS=GACON*AMAX1(GREYNO,1.0)**(GBCON-1.)
      DO 101 JITS=1,3
      GSHALF=SQRT(GS)
 101  GS=(GAK/ALOG(1.01+GEWALL*GREYNO*GSHALF))**2
      GTAU=GS*GREYNO*GSPEED/GDELMU
      GTKE=AMIN1(AMAX1(GTAU/(GRHO*GTAUDK),TKEMIN),TKEMAX)
C
      IF(JVAR.EQ.JEP) GO TO 200
C
      GVALUE=GTKE
      GCOEFF=GREAT
C
      GO TO 350
C---------------------------------------------------------------------
CHAPTER 2      FIX KE DISSIPATION RATE (EP)
C---------------------------------------------------------------------
 200  GVALUE=GWALC*SQRT(GTKE)*GTKE/GDEL
      GCOEFF=GREAT
C
      GO TO 350
C---------------------------------------------------------------------
CHAPTER 3     WALL FRICTION (U1,V1,W1) AND HEAT TRANSFER (H1)
C---------------------------------------------------------------------
C--- LAMINAR
C WALL FRICTION
```

```
 300  GCOEFF=RWDRP2/GDELMU
      GO TO 302
C HEAT TRANSFER
 301  GCOEFF=GAKRA*RWDRP2/GDELMU
C
 302  IF(GREYNO.LE.132.25) GO TO 310
C--- TURBULENT
      GS=GACON*GREYNO**(GBCON-1.)
      DO 303 JITS=1,3
      GSHALF=SQRT(GS)
 303  GS=(GAK/ALOG(1.01+GEWALL*GREYNO*GSHALF))**2
      GCOEFF=GCOEFF*GS*GREYNO
C
 310  GVALUE=GVPHI
C
C--- SET UP GCVAR (=CVAR) AND GVVAR (=VVAR) ARRAYS
 350  GCVAR(JIY,JIX)=GCOEFF
      GVVAR(JIY,JIX)=GVALUE
C
 390  CONTINUE
C
      RETURN
C-----------------------------------------------------------------------
      END
      SUBROUTINE AUTMON(ISTP,ISWP)
$INCLUDE CMNGUSSI.FTN/G (NLIST)
$INCLUDE GUSSEQUI.FTN/G (NLIST)
      DIMENSION ISOLV(25)
      LOGICAL FIRST
      DATA FIRST/.TRUE./
      DATA KSTP/O/
C... USER DIMENSIONED (NY X NX) ARRAY FOR GETTING VARIABLES
      DIMENSION GDUM(40,8)
C
      IF(FIRST) THEN
      OPEN(20,FILE='AUTOMON.DTA',STATUS='RENEW',RECL=20,FORM=
     +     'FORMATTED')
      NUMSOL = O
      DO 10 I = 1,25
      IF(SOLVAR(I).OR.STOVAR(I)) THEN
      ISOLV(NUMSOL+1) = I
      NUMSOL = NUMSOL+1
      ENDIF
   10 CONTINUE
      FIRST=.FALSE.
      ENDIF
C
      IF(KSTP.NE.ISTP) THEN
       IF(.NOT.STEADY) WRITE(20,'("TIME STEP NO. ",I3)') ISTP
       WRITE(20,'(I2)') NUMSOL
       DO 15 I = 1,NUMSOL
   15  WRITE(20,'(A4)') TITLE(ISOLV(I))
       KSTP = ISTP
      ENDIF
C
      WRITE(20,'(I3)') ISWP
      DO 20 II = 1,NUMSOL
      CALL GET(ISOLV(II),GDUM,NY,NX)
      WRITE(20,'(1PE10.3)') GDUM(IYMON,IXMON)
   20 CONTINUE
```

```
      C
                RETURN
                END .
      P!

@BRKPT PRINT$
```

APPENDIX B: PROPERTY CURVE FITS

The individual enthalpy curves for water and hydrogen have been combined in order to calculate a mixture enthalpy, Enthalpy_{mix}, defined as:

$$\text{Enthalpy}_{mix}(T) = (\text{Mass Ratio } H_2O) * \text{Enthalpy Water } (T) + (1\text{-Mass Ratio } H_2O) * \text{Enthalpy Hydrogen } (T)$$

This combined property curve is needed to be able to calculate the temperature of any given mixture of water and hydrogen in the aft-platform seal cavity, based on the mixture ratio and enthalpy calculated by the model. From the temperature are then calculated other fluid properties, such as density and viscosity. The curve fits used to compute these properties are depicted in Figures B-1 to B-6.

ENTHALPY OF WATER[1]

CURVE FIT I $(492 \le T < 975R)$ $H\ (\mathrm{Btu/lbm}) = -424.5938 + .82414T + 1.3067\times10^{-4}T^2$

CURVE FIT II $(975R \le T < 1184.6R)$ $H = 2289.552 - 4.577089T + 2.815249\times10^{-3}T^2$

CURVE FIT III $(1184.6R \le T < 1223.3R)$ $H = -7363.69 + 6.913T$

CURVE FIT IV $(1223.3R \le T < 1281.4R)$ $H = 599.5881 - 1.27177T + 1.369267\times10^{-3}T^2$

CURVE FIT V $(1281.4R \le T \le 1400R)$ $H = -307.5449 + 1.190721T$

STANDARD ERROR = 4.08 Btu/lbm

---

[1] These curves were fit to data taken from *Thermodynamic Properties of Steam*, Joseph Keenan and Frederick Keyes, (New York: Wiley and Sons, 1936) pp. 72-75.

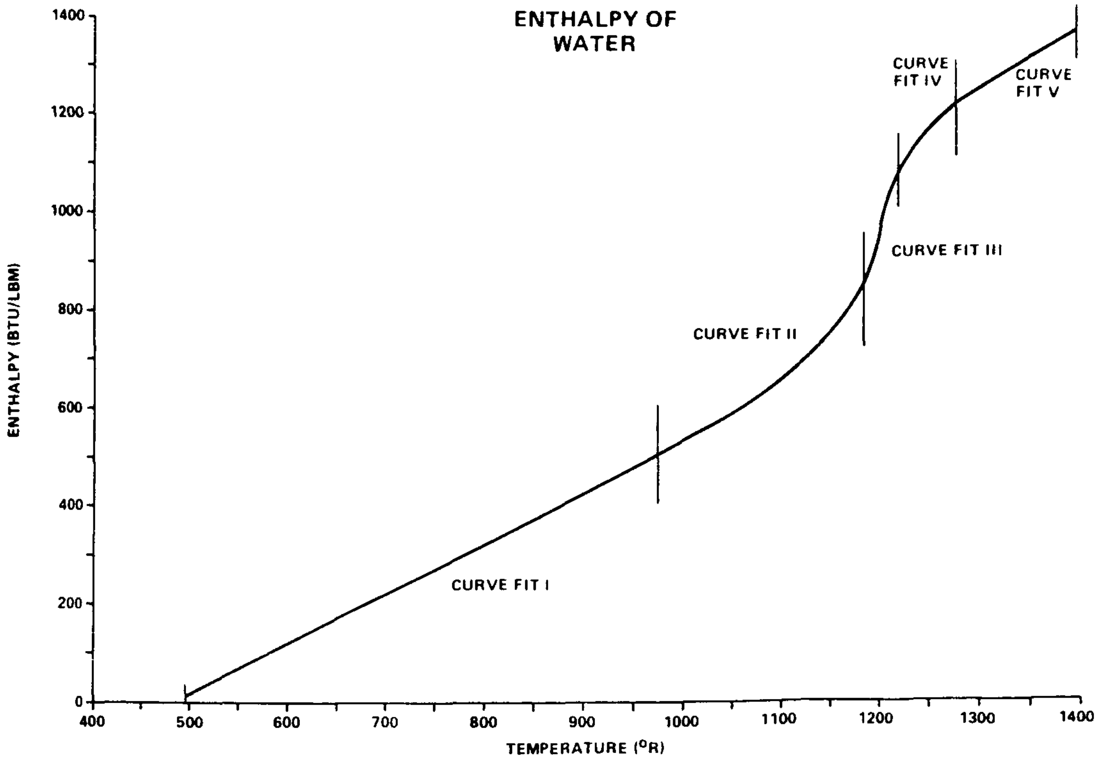


Figure B-1.

DENSITY OF WATER[2]

CURVE FIT I
($490R \leq T < 1180R$)

$$\text{density } (\text{lbm/ft}^3) = -82.117 + .62353T - 6.77693\times10^{-4}T^2 - 3.41207\times10^{-7}T^3 + 9.23406\times10^{-10}T^4 - 3.9688\times10^{-13}T^5$$

CURVE FIT II
($1180R \leq T < 1250R$)

$$\text{density} = -2177.783 + 7.12733T - 4.54395\times10^{-3}T^2 - 1.91391\times10^{-6}T^3 + 1.686\times10^{-9}T^4$$

CURVE FIT III
($1250R \leq T \leq 1400R$)

$$\text{density} = 119.1372 - 4.770357\times10^{-2}T - 1.00694\times10^{-4}T^2 + 5.516186\times10^{-8}T^3$$

STANDARD ERROR = 0.61 lbm/ft^3

---

[2] These curves are fit to data taken from Keenan, pp. 72-75.

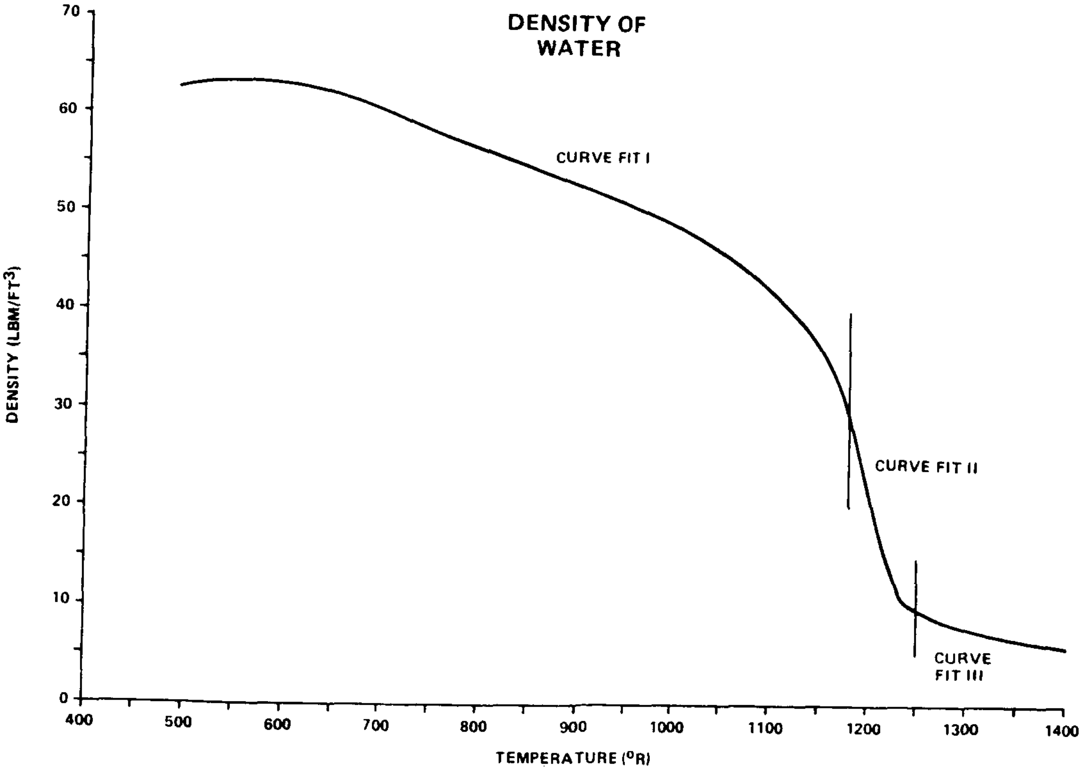


Figure B-2.

VISCOSITY OF WATER[3]

$$\text{VISC}\ (\times 10^3) = e^{\left[20.5532 - 6.52199\times 10^{-2}T + 3.2726\times 10^{-5}T^2 + 6.6687\times 10^{-8}T^3 - 8.3627\times 10^{-11}T^4 + 2.6237\times 10^{-14}T^5\right]}$$

STANDARD ERROR $(\times 10^3)$ = 0.0066 lb/ft-sec

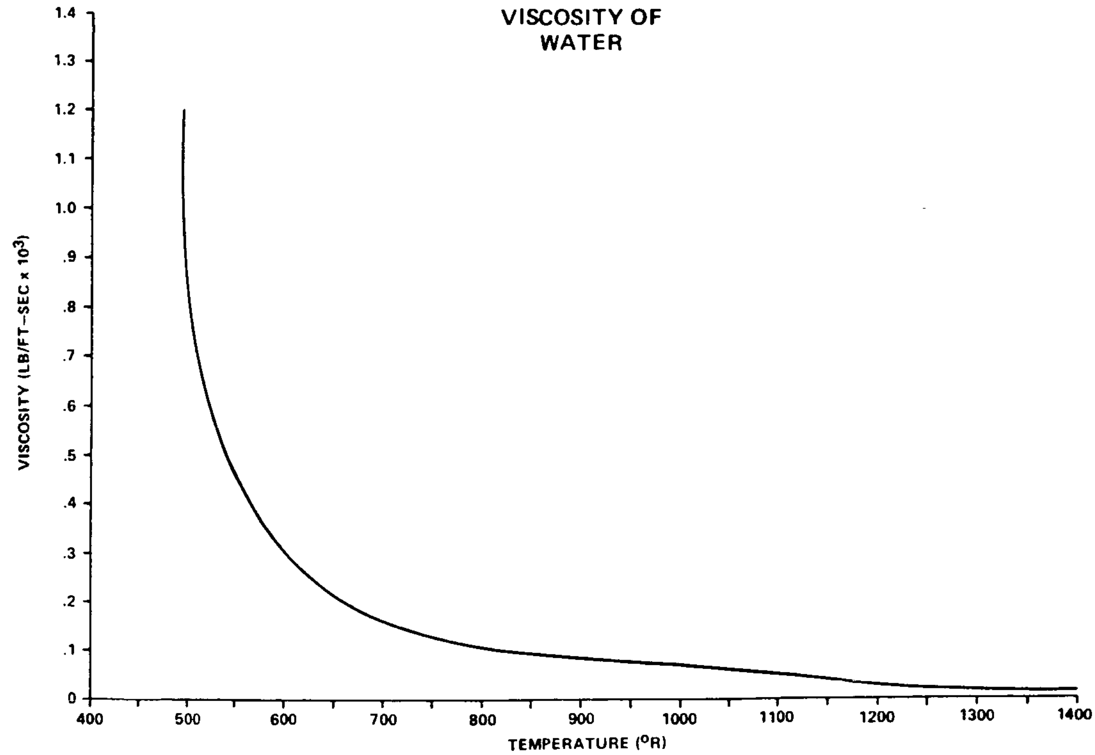


Figure B-3.

---

[3] This curve is fit to data taken from Steam Tables, Joseph Keenan, et al., (New York: Wiley and Sons, Inc., 1969) p. 113.

ENTHALPY OF HYDROGEN[4]

CURVE FIT I
($170R \leq T \leq 508R$) H (Btu/lbm) = $-5.92706\times10^{-4}T^2 + 4.468995T$
-357.6903

CURVE FIT II
($508R < T \leq 2000R$) $H = -7.15694\times10^{-6}T^2 + 3.557702T - 45.88906$

STANDARD ERROR = 4.39 Btu/lbm

---

[4] These curves are fit to data taken from the Hydrogen Technological Survey - Thermophysical Properties, Robert D. McCarty, (Washington, D.C.: NASA Scientific and Technical Information Office, 1975) p. 472.

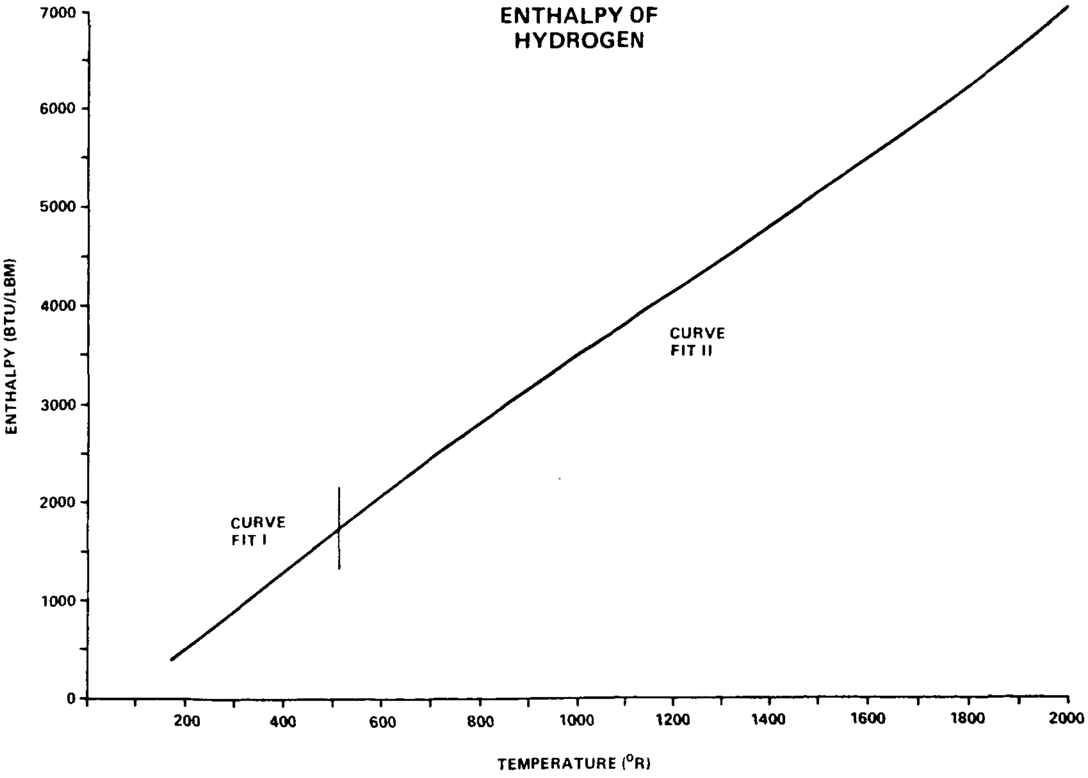


Figure B-4.

DENSITY OF HYDROGEN[5]

$$\text{density (lbm/ft}^3\text{)} = e^{\left[-5.26685 + 3.049183(\ln H) - .41497(\ln H)^2 + 1.40759\times10^{-2}(\ln H)^3\right]}$$

where H is the enthalpy of hydrogen.

STANDARD ERROR = .0189 lbm/ft^3

---

[5] This curve is fit to data taken from McCarty, p. 472.

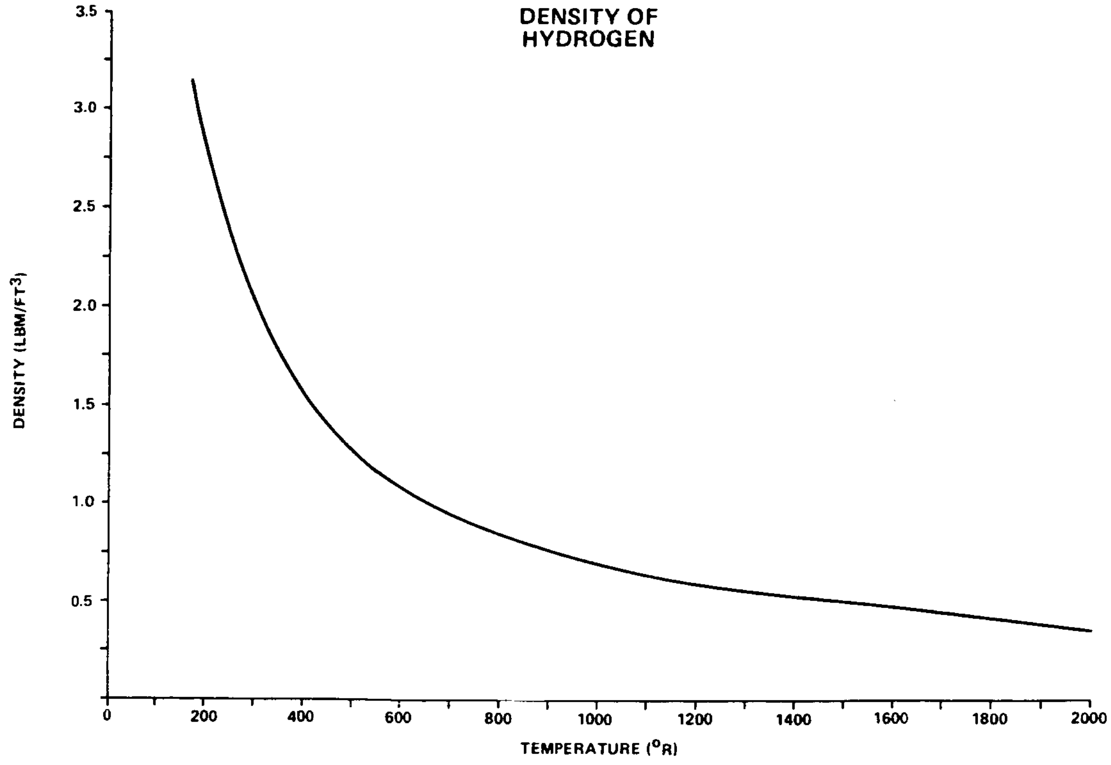


Figure B-5.

VISCOSITY OF HYDROGEN[6]

$$\text{VISC. (lbm/ft-sec} \times 10^5) = .4989 - 5.4575\times10^{-5}T + 5.1824\times10^{-7}T^2 - 1.4948\times10^{-10}T^3$$

STANDARD ERROR $(\times10^5)$ = 0.00047 lbm/ft-sec

[6] This curve is fit to data taken from the Hydrogen Technological Survey - Thermophysical Properties, Robert D. McCarty, (Washington, D.C.: Scientific and Technical Information Office, NASA, 1975) p. 473.

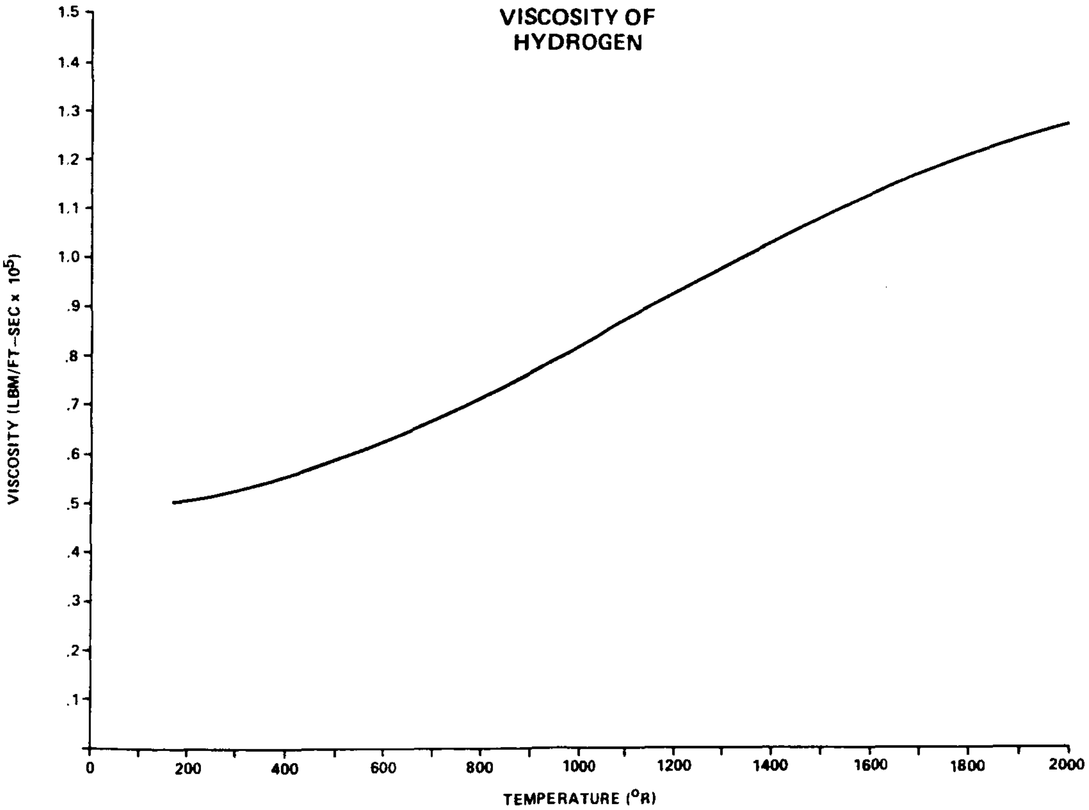


Figure B-6.

APPENDIX C: CONVERGENCE CHARACTERISTICS

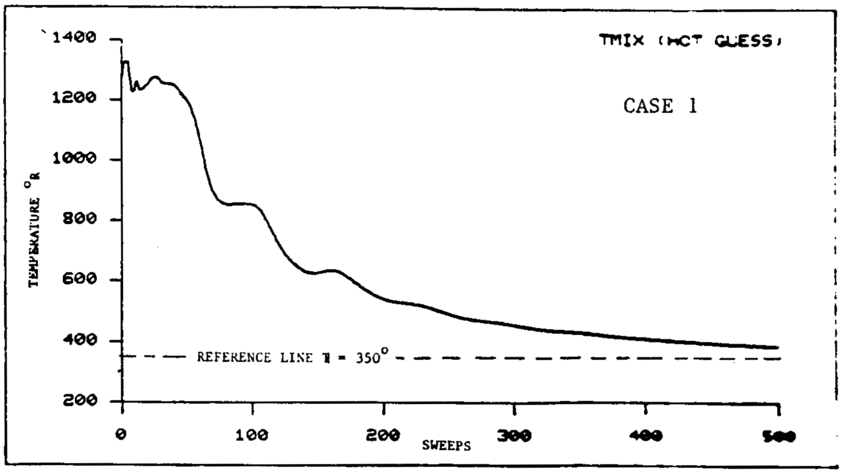
The insensitivity of the model to the initial values chosen for temperature and velocity is demonstrated by the solution sets given below. There were four different test cases run using identical boundary conditions but with the different guesses of velocity and temperature listed in the following table. Cases 1 to 3 were run to test the sensitivity of the solution to the initial temperature guess, and case 4 was run to check the sensitivity of the solution to the initial choice of the velocity field.

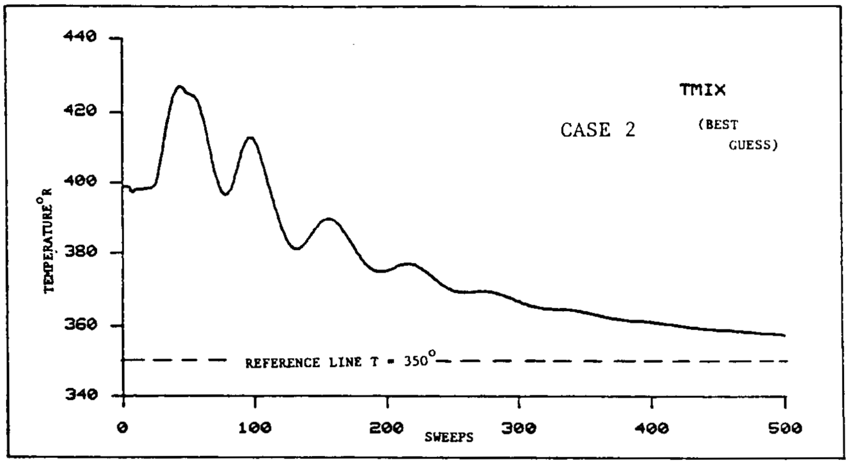
INITIAL FIELD VALUES

	Temperature	Theta Velocity
CASE 1	Hot Guess	Omega = 0.4 Disk Omega
CASE 2	Best Guess	Same as above
CASE 3	Cold Guess	Same as above
CASE 4	Best Guess	Omega = 0

(Omega in radians/sec)

After 500 sweeps, the values of velocity, temperature, and pressure, at a reference point in the middle of the cavity, have converged to the extent shown in Figures C-1 to C-8. By 500 sweeps the constant pressure lines (Fig. C-2) for all four cases are very similar, as are the streamlines (Fig. C-8), and to a lesser extent the temperature profiles (Figs. C-1 and C-7). Of the three, temperature is the slowest to recover from a poor initial guess. However, even with an initial temperature estimate 1000°R off the final values, the temperature at the monitoring point has converged to within 200° of the final value after 200 sweeps, and to within 75° of the final value after 400 sweeps. This is an acceptable convergence rate for our application, especially since the magnitude of the error is readily apparent from the slope of the temperature convergence curve (Fig. C-1). In the event that greater accuracy were required, the solution could be bracketed or else extended the necessary number of sweeps.





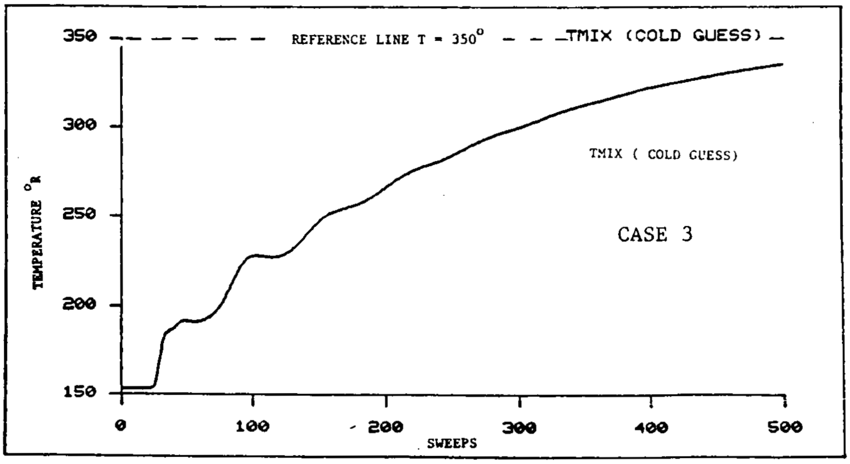


Figure C-1. Temperature convergence, cases 1 to 3.

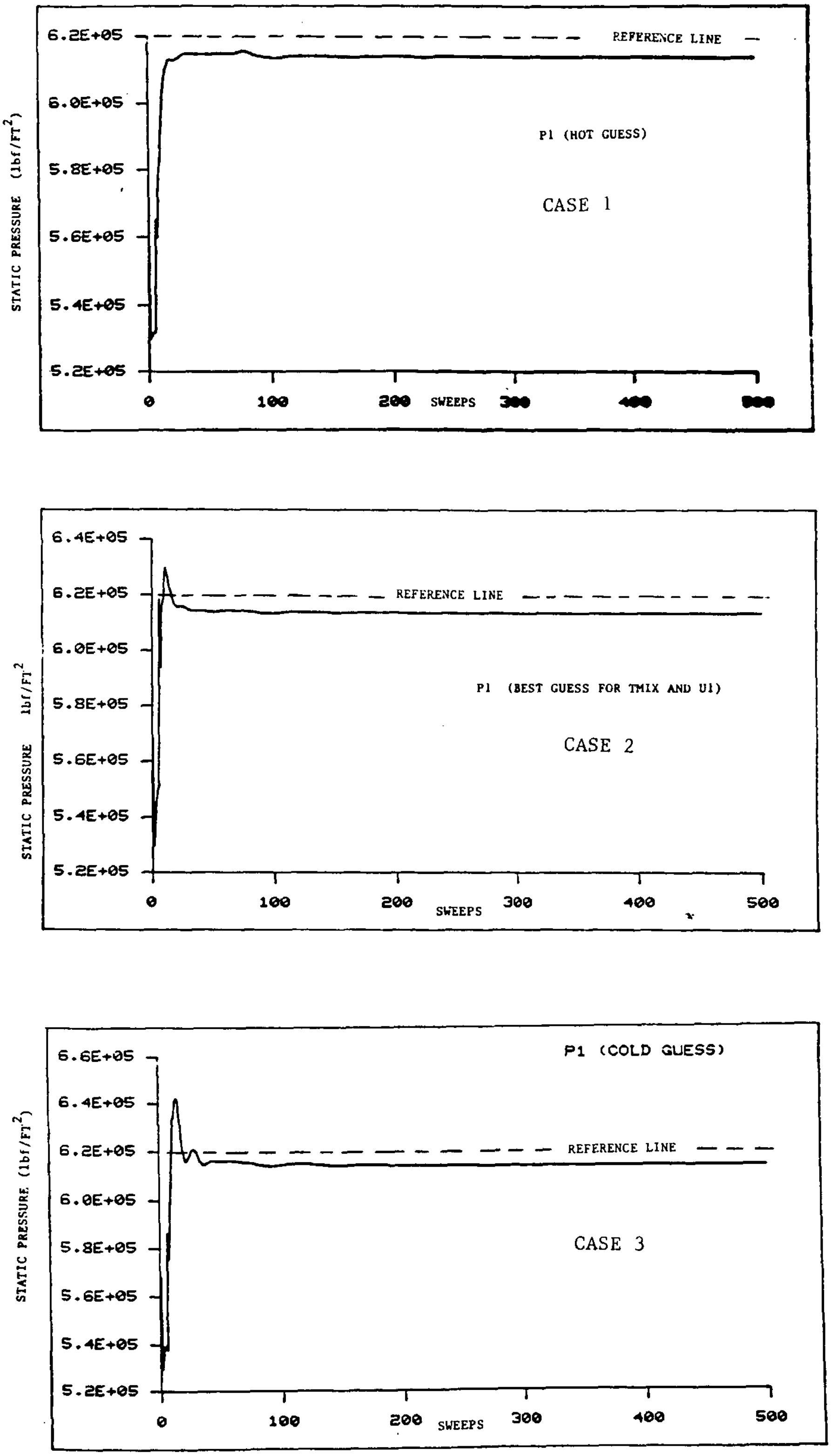
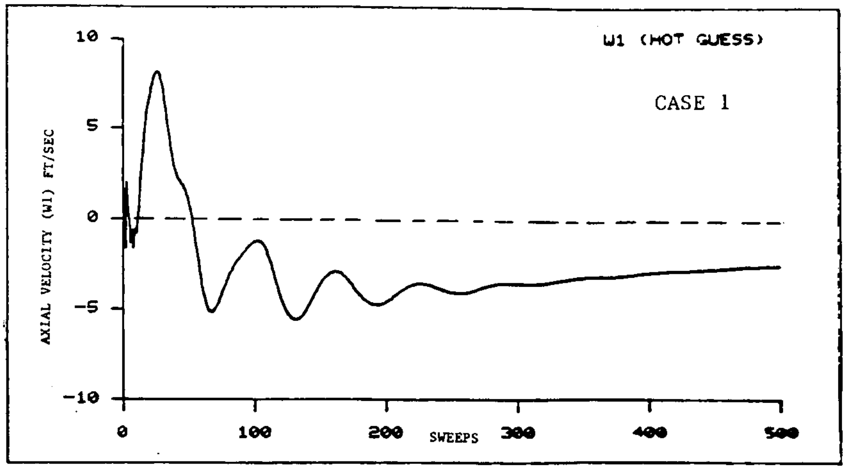
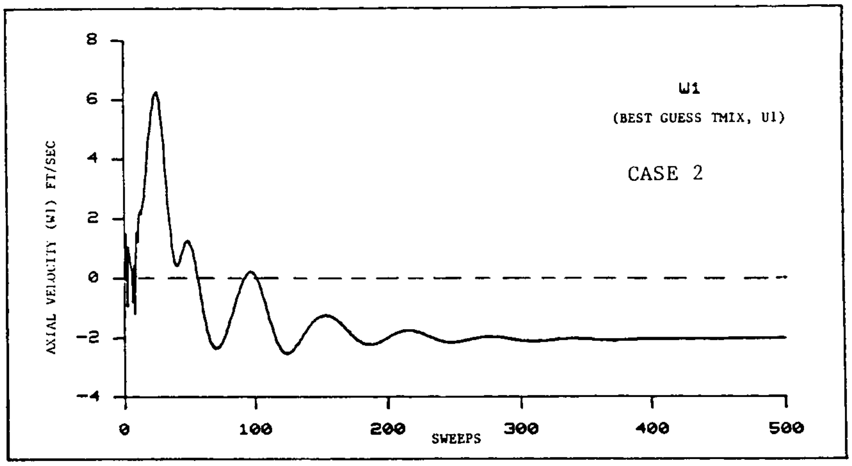


Figure C-2. Pressure convergence, cases 1 to 3.





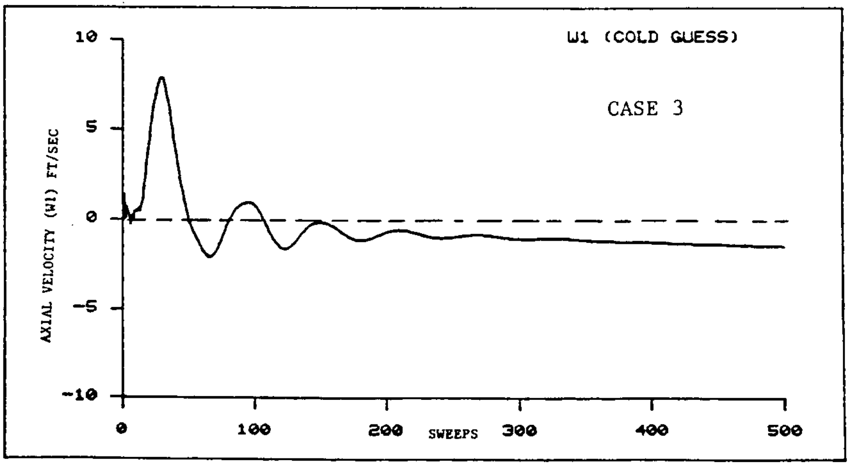


Figure C-3. Circumferential velocity convergence, cases 1 to 3.

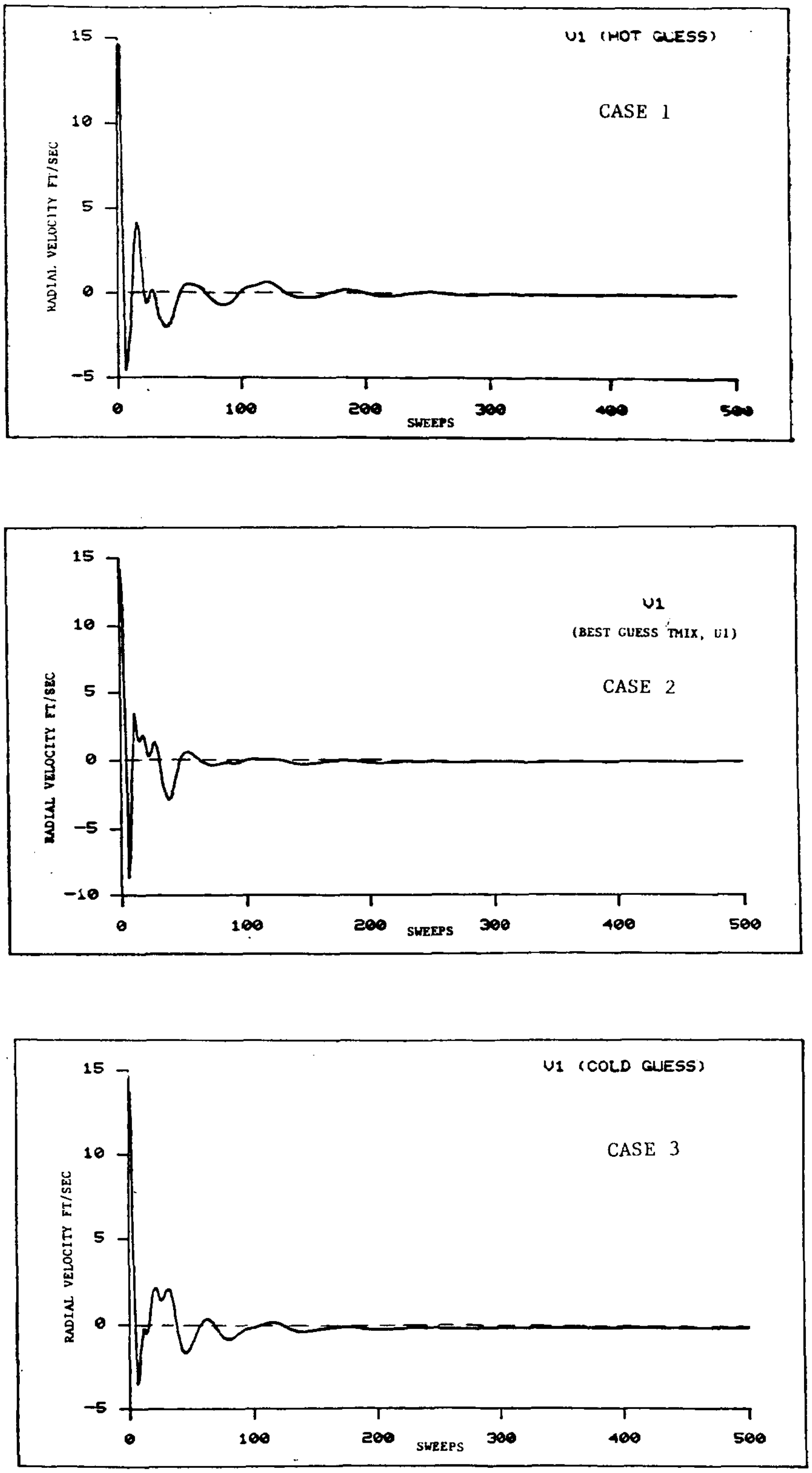


Figure C-4. Radial velocity convergence, cases 1 to 3.

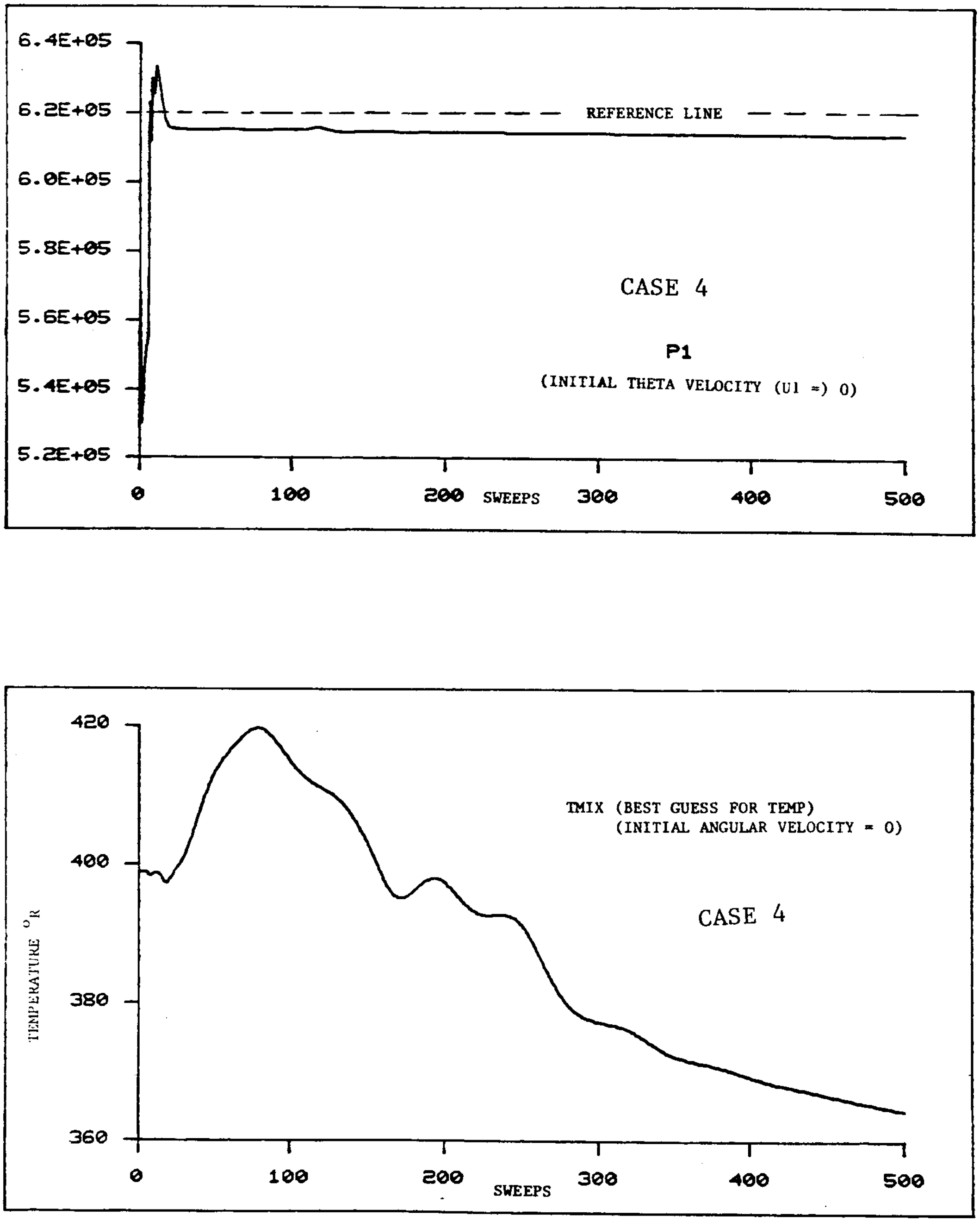


Figure C-5. Pressure and temperature convergence, case 4.

ORIGINAL PAGE IS
OF POOR QUALITY

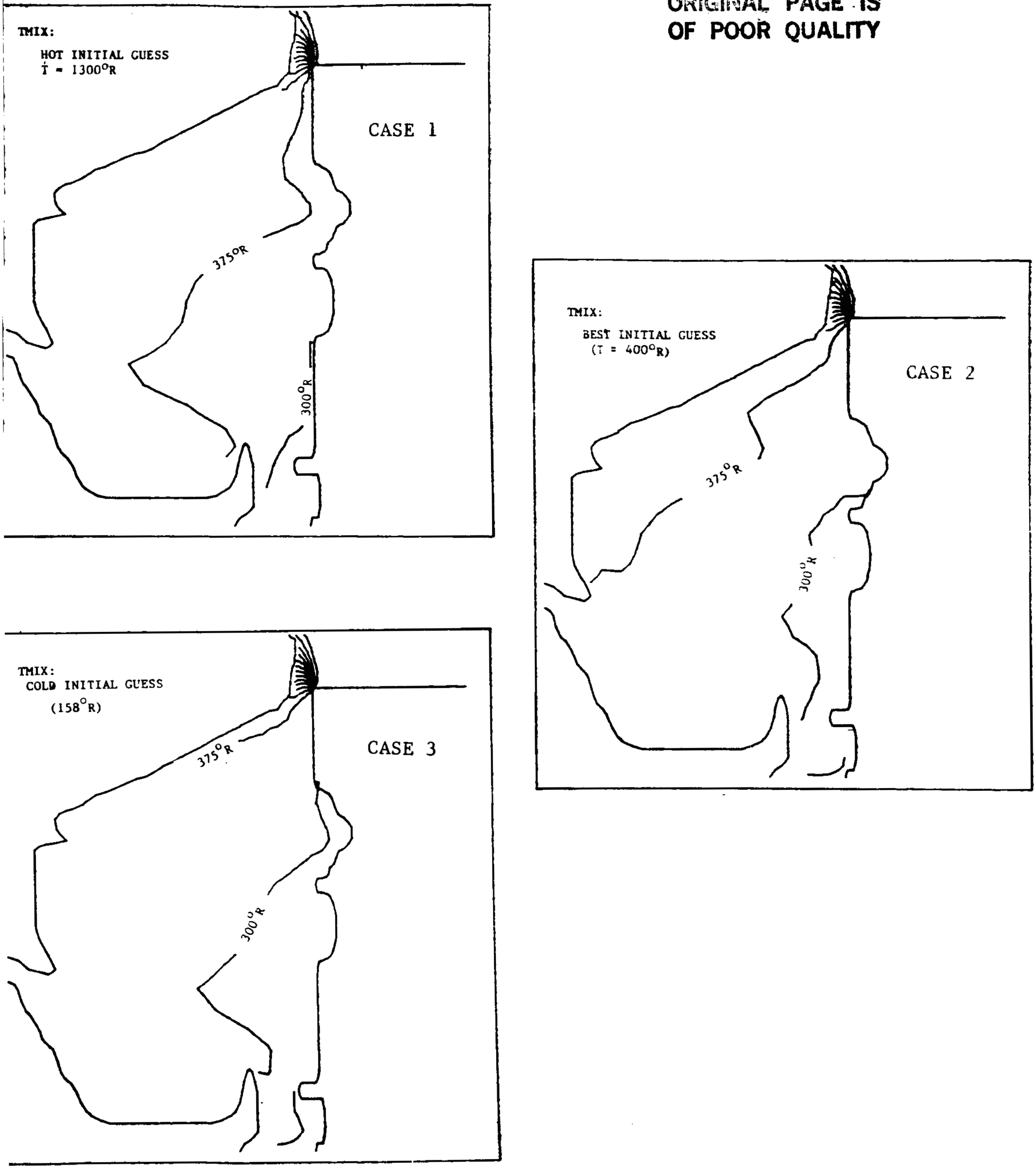


Figure C-6. Temperature fields, cases 1 to 3.

ORIGINAL PAGE IS
OF POOR QUALITY

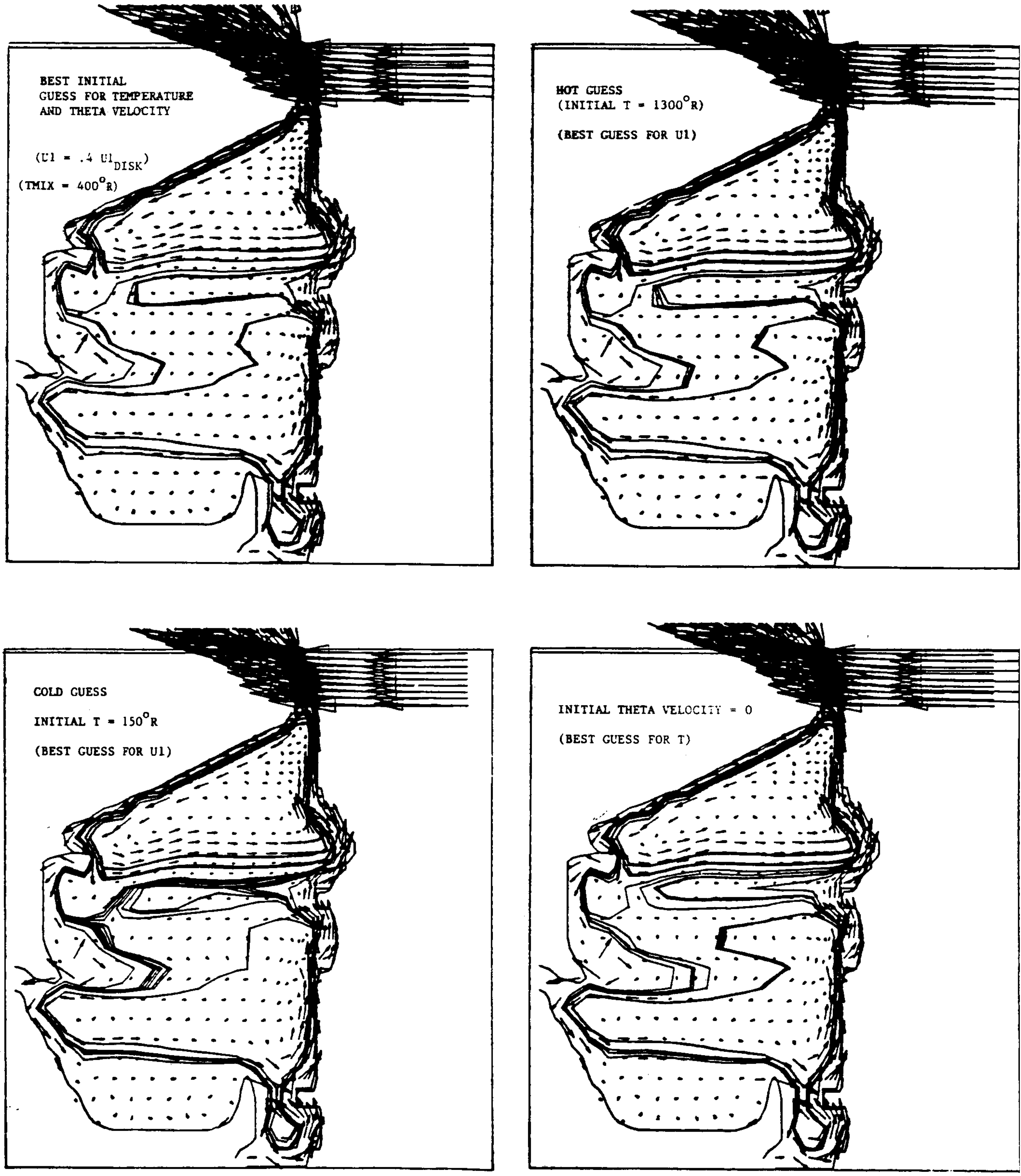


Figure C-7. Velocity field and streamlines, cases 1 to 4.

1. REPORT NO.
NASA TP-2685

2. GOVERNMENT ACCESSION NO.

3. RECIPIENT'S CATALOG NO.

4. TITLE AND SUBTITLE
Space Shuttle Main Engine High Pressure Fuel Pump
Aft Platform Seal Cavity Flow Analysis

5. REPORT DATE
January 1987

6. PERFORMING ORGANIZATION COD

7. AUTHOR(S)
S. A. Lowry and L. W. Keeton*

8. PERFORMING ORGANIZATION REPO

9. PERFORMING ORGANIZATION NAME AND ADDRESS
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

10. WORK UNIT NO.
M-544

11. CONTRACT OR GRANT NO.

13. TYPE OF REPORT & PERIOD COVE
Technical Paper

12. SPONSORING AGENCY NAME AND ADDRESS
National Aeronautics and Space Administration
Washington, D.C. 20546

14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES
Prepared by Structures and Propulsion Laboratory, Science and Engineering Directorate.

*L. W. Keeton is employed by CHAM (NA) Inc., Huntsville, Alabama

16. ABSTRACT

A general purpose, three-dimensional computational fluid dynamics code named PHOENICS, developed by CHAM Inc., is used to model the flow in the aft-platform seal cavity in the high pressure fuel pump of the space shuttle main engine. The model is used to predict the temperatures, velocities, and pressures in the cavity for six different sets of boundary conditions. The results are presented as input for further analysis of two known problems in the region, specifically: erratic pressures and temperatures in the adjacent coolant liner cavity and cracks in the blade shanks near the outer diameter of the aft-platform seal.

17. KEY WORDS
Space Shuttle Main Engine
High Pressure Fuel Turbopump
Aft Platform Seal Cavity
Computational Fluid Dynamics
Turbine Blades

18. DISTRIBUTION STATEMENT
Unclassified — Unlimited

Subject Category 34

19. SECURITY CLASSIF. (of this report)
Unclassified

20. SECURITY CLASSIF. (of this page)
Unclassified

21. NO. OF PAGES
137

22. PRICE
A07

For sale by National Technical Information Service, Springfield, Virginia 221

NASA-Langley, 19