NUMERICAL SIMULATION OF TWO-PHASE BOILING FLOW IN A LINEAR HORIZONTAL POROUS MEDIUM

Paul G. Atkinson
Research Assistant
Stanford University
Stanford, California

December, 1975

Grant No. AER72-03490-A03
(Previous Grant No. GI-34925)

A Report to:

Division of Advanced Energy Research and Technology
National. Science Foundation
Washington, D.C.

ACKNOWLEDGEMENT

This work was supported by the National Science Foundation. In keeping with the objectives of the RANN Program, communications were maintained with many academic and industrial agencies. Acknowledgement is made for the advice and cooperation given by these various agencies.

TABLE OF CONTENTS

PROGRAM PARAMETERS

LIST OF FIGURES

FIGURE

INTRODUCTION

This report describes a computer program for predicting transient behavior of two–phase boiling flow in a porous medium. Derivation of the necessary flow equations and a brief discussion of the finite-difference numerical method used has been previously described in report SGP–TR–1 (Kruger and Ramey, 1974). This report describes the program in greater detail, and in addition includes a source listing of the code, information on input and output parameters, and a sample run. Although an understanding of SGP-TR-1 is not essential to understanding the discussion here, it will be helpful to have it for reference purposes.

The two–phase flow finite difference simulator was developed at Stanford University during the period June 1973 to September 1974 for the purpose of investigating two–phase boiling flow experimental data such as that of Arihara (1974, pages 196-206), and Cady, Bilhartz, and Ramey (1974). Attempts to match the two–phase transient flow data of Arihara were partly successful, and were reported in SGP-TR-1. Attempts to approximately simulate the development of a superheated steam region in the experiments of Cady ~~et al~~. were unsuccessful. Apparently this was because the experimental data were influenced appreciably by capillary forces of the porous medium, which were not accounted for in the numerical model.

The equations describing two–phase boiling flow in a linear porous medium were derived in SGP–TR–1, and were presented as equations 51 and 52 in that report. For horizontal flow these equations reduce to:

$$\frac{\partial}{\partial x}\left[\gamma_1(p,S_L)\,\frac{\partial p}{\partial x}\right] = \frac{\partial \gamma_2}{\partial t}(p,S_L) \qquad (1)$$

and

$$\frac{\partial}{\partial x}\left[\gamma_3(p,S_L)\frac{\partial p}{\partial x}\right] = \frac{\partial \gamma_4}{\partial t}(p,S_L) + \dddot{q}_{loss} \ ; \qquad (2)$$

where pressure, p, and volumetric liquid saturation, S_L are the dependent variables, distance, x, and time, t, are the independent variables, and the functions γ_j, and $\dddot{q}_{loss}$ are defined in Appendix A. These functions are also presented in Appendix C of SGP-TR-1, but with different subscripts. The subscripts have been changed in this report so as to make them consistent with usage in the program listing. Other workers (Donaldson, 1968; Mercer et al., 1974; Toronyi, 1974; Brownell et al., 1975; and Lasseter et al., 1975) have reported essentially these same equations, although not always in terms of the same dependent variables.

This computer model can only describe the flow of saturated fluids. This is a consequence of the particular formulation in terms of pressure and volumetric liquid saturation as dependent variables. The use of other formulations (e.g., as by Garg et al., 1975; Mercer and Faust, 1975) may facilitate the simultaneous description of both single- and two-phase flow. Thus, in this model, if $S_L = 1.0$, then the liquid must be at the bubble point; and if $S_L = 0.0$, then the gas must be at the dew point.

The program will accept arbitrary initial conditions within the equation of state constraints, but it is limited to the specific boundary conditions of no flow at the left-hand end, and specified pressure at the right-hand end. Thus:

$$\text{at } x = 0, \ \frac{\partial p}{\partial x} = 0 \ , \qquad (3)$$

and

$$\text{at } x = L, \ p(t) = f(t), \qquad (4)$$

where f(t) is known.

One potential application of a program such as that described in this report would be the development of pressure drawdown and build-up behavior for the radial two-phase boiling flow to a single well in a thin geothermal aquifer. Converting the program from linear to radial space coordinates would require only a change in various differencing coefficients, and no change in the program structure, A specified flowrate boundary condition at the well is of more practical interest than the specified pressure condition used here. Even though such a specified flux type of boundary condition would be nonlinear for two-phase flow, this program could be modified to account for such a boundary condition by simply changing the structure of the difference equations in an appropriate manner.

The remainder of this report discusses details of both the finite difference method and the logical structure of the program. Appendix B contains a source listing of the Fortran code. Appendix C describes the subroutines used in the code, and in addition includes a flow diagram depicting the logical structure of the program. Input and output corresponding to a particular numerical simulation of the depletion experiments of Arihara are presented in Appendix D.

NOMENCLATURE AND NOTATION

1) Variable names used in the computer program are written in this report with capital letters, e.g., DELTXH. Array variables are similarly written in capital letters followed by the array dimensions in parentheses, e.g., POLD(20) .

2) Subscripts on pressure, p, and volumetric liquid saturation, S_L, indicate discretized values at the given subscripted node. For notational brevity the subscript "L" of "S" has been dropped. Thus, pressure and liquid saturation at the ith node are indicated by p_i, and S_i.

3) Line numbers in the source code are referred to with a "#"; e.g., "line #23" or "lines #240/242."

4) Superscripts indicate the time level of discretization. Thus, p_i^n is the pressure at the ith node at the nth (old and known time level, and p_i^{n+1} is at the (n+1)th (new and unknown) time level.

IMPLICIT FINITE DIFFERENCE PROCEDURE

A two time-level implicit differencing scheme with (m+1) equally spaced grid nodes was used. The Crank-Nicholson implicit scheme was chosen because it is unconditionally stable for linear systems, and of order ((At) 2 + (Ax)2) accurate. However, the backwards difference fully implicit differencing scheme probably would have been better since it appears to have better stability properties for multi-phase flow problems (Al-Hussainy, *1974)*. Such a backwards differencing scheme was used by Toronyi (1974).

The difference equations are of the form:

$$\frac{\Delta\left[\gamma_1{}^n \Delta p_i{}^n\right] + \Delta\left[\gamma_1{}^{n+1} \Delta p_i{}^{n+1}\right]}{2} = \delta\gamma_{2,i}{}^n, \tag{5}$$

$$\frac{\Delta\left[\gamma_3{}^n \Delta p_i{}^n\right] + \Delta\left[\gamma_3{}^{n+1} \Delta p_i{}^{n+1}\right]}{2} = \delta\gamma_{4,i}{}^n + \dot{q}'''_{loss}{}^{n+1/2}, \tag{6}$$

for $i = 1, 2, 3, \ldots, m$, and the expressions are in terms of the finite difference operators:

A, central difference operator in space, $\Delta u_i - \frac{\triangleq\ u_{i+1/2} - u_{i-1/2}}{(\Delta x)}$.,

the expression $\Delta\left[a\Delta u_i\right]$ is

$$\Delta\left[a\Delta u_i\right] \triangleq \frac{a_{i+1/2}\left[u_{i+1}-u_i\right] - a_{i-1/2}\left[u_i-u_{i-1}\right]}{(\Delta x)^2}$$

and δ is the forward difference operator in time, $\delta u^n = \frac{u^{n+1}-u^n}{(\Delta t)}$

Note that (At) is the time step size, and (Ax) is the uniform grid spacing. Midstream weighting of pressure and saturation was used to evaluate the $q_{i+1/2}$ and $a_{i-1/2}$ functions in the self-adjoint expression. Thus, $a_{i+1/2}$ was evaluated at the average value of pressure and saturation between the ith and (i+1)th nodes:

$$a_{i+1/2} = a(p_{i+1/2}, S_{i+1/2}) \; .$$

Apart from the observation that the evaluation of $a_{i+1/2}$ as $(a_i + a_{i+1})/2$ would have a smaller truncation error than the method used, it should be noted that some sort of upstream weighting would probably have given better results (Aziz, 1971; Lasseter and Witherspoon, 1974, pp. 86–87; Blair <u>et al</u>., 1974). While full weighting at a single upstream node appears to be generally used (Todd et al., 1972), other weighting schemes have been proposed (e.g., by Toronyi and Farouq Ali, 1974). It is common to weight saturation dependent functions at an upstream level, while at the same time evaluating the pressure dependent physical properties at midstream conditions (e.g., see Weinstein <u>et al</u>., 1974, Coats <u>et al</u>., 1974; and Toronyi, 1974).

The difference equations for the left and right-hand nodes must be modified to account for the boundary conditions. At the first node we approximate $\frac{\partial p}{\partial x} = 0$ with the central difference approximation:

$$\frac{p_2 - p_0}{(\Delta x)} = 0 \; ,$$

where p_0 is a "fictitious node."

Using a symmetry argument, we deduce that $S_0 = S_2$, where the subscript "0" again refers to the "fictitious node." The right-hand boundary condition specifies the pressure at the (m+1)th node. Thus, when the equations are differenced about the mth node, p_{m+1}^{n} becomes PB^{n}, and p_{m+1}^{n+1} becomes PB^{n+1}, where the PBs are specified. Unfortunately, there is an ambiguity with respect to the value of saturation at the right-hand boundary if anything other than full upstream weighting is used. This occurs because the value of saturation at the (m+1)th node is not specified. The computer program being described here estimated S_{m+1} by using quadratic extrapolation from S_{m-2}, S_{m-1}, and S_m (lines 8108, 132). This ambiguity does not occur if full upstream weighting is used, because in this case a value of S_{m+1} is not required.

The fully discretized equations describing the behavior of the m+1 nodes form a nonlinear nonalgebraic system of 2m equations:

$$\underset{\sim}{F}(\underset{\sim}{x}) = \underset{\sim}{0} \, , \tag{7}$$

where $\underset{\sim}{F}$ is a vector of 2m functions, and $\underset{\sim}{x}$ is the combined unknown pressure and saturation vectors. $\underset{\sim}{F}$ is given by $(F_1, F_2, \ldots, F_{2m})^t$, and $\underset{\sim}{x}$ by $(x_1, x_2, \ldots, x_{2m})^t$. The system of equations (7) are explicitly presented in Appendix E.

The nonlinear mass balance, F_{2i-1}, and energy balance equations, F_{2i}, $(i = 1, \ldots, m)$, were solved simultaneously using Newton-Raphson iteration. This solution technique is discussed in the next section.

SOLUTION OF THE DISCRETIZED NONLINEAR EQUATIONS USING NEWTON-RAPHSON ITERATION

Newton-Raphson iteration is a well known method for solving systems of equations of the form $\underset{\sim}{F}(\underset{\sim}{x}) = \underset{\sim}{0}$ (Isaacson and Keller, 1966; Carnahan, Luther, and Wilkes, 1969). In this method one successively solves the system:

$$\left[\underset{\approx}{\Phi}(\underset{\sim}{x}^{(\nu)}) \right] \underset{\sim}{\xi}^{(\nu+1)} = -\underset{\sim}{F}(\underset{\sim}{x}^{(\nu)}) \, , \tag{8}$$

$$\underset{\sim}{x}^{(\nu+1)} = \underset{\sim}{x}^{(\nu)} + \underset{\sim}{\xi}^{(\nu+1)} , \qquad (9)$$

$$\underset{\sim}{x}^{(o)} = \text{initial trial solution}, \qquad (10)$$

where: $[\underset{\sim}{\Phi}]$ is the Jacobian matrix $\left[f_{ij}\right]$, $f_{ij} \triangleq \frac{\partial F_i}{\partial x_j}$,

the superscript in parentheses indicates the iteration number; and $\underset{\sim}{\xi}$ is the correction vector after each iteration.

The matrix Φ corresponding to the discretized equations is bitridiagonal, and is presented in Fig. 1 as part of the system $\underset{\approx}{\Phi}\underset{\sim}{\xi} = -\underset{\sim}{F}$. The elements f_{ij} of $\underset{\sim}{\xi}$ are presented explicitly in Appendix E.

For $\underset{\sim}{x}^{(\nu)}$ known, the correction equation (9) is linear and algebraic. Hence it can be solved directly using Gaussian elimination. The bitridiagonal structure of $\underset{\approx}{\Phi}$ means that it has a narrow bandwidth, and hence can be solved rapidly and efficiently. Appendix F presents an algorithm which accomplishes this.

Subroutine ITSOLV (lines #81/272) executes the Newton-Raphson algorithm. Subroutine SOLVBT (lines #453/512) executes the Gaussian elimination algorithm, and is called by ITSOLV. ITSOLV continues iteration loops until either: (1) the convergence criterion is reached, i.e., until the residuals $F_i, i=1, \ldots, 2m$, at every node are less than the convergence criterion, DELTA; or (2) the number of iterations exceeds MAXNUM, in which case execution stops.

It can be shown that Newton-Raphson iteration will converge if the initial trial solution (10) is "close" enough to the correct answer (e.g., Isaacson and Keller, 1966; Carnahan <u>et al.</u>, 1969). For the discretized equations $\underset{\sim}{F}(\underset{\sim}{x}) = \underset{\sim}{0}$, it was usually adequate to use the pressures and saturations at the old time level as the initial trial solution. In order to speed up convergence, however, a weighted linear extrapolation forward in time was employed, using values at the two

$$\begin{bmatrix}
f_{11} & f_{12} & f_{13} & f_{14} & & & & & & \\
f_{21} & f_{22} & f_{23} & f_{24} & & & & & & \\
f_{31} & f_{32} & f_{33} & f_{34} & f_{35} & f_{36} & & & & \\
f_{41} & f_{42} & f_{43} & f_{44} & f_{45} & f_{46} & & & & \\
 & & f_{53} & f_{54} & f_{55} & f_{56} & f_{57} & f_{58} & & \\
 & & f_{63} & f_{64} & f_{65} & f_{66} & f_{67} & f_{68} & & \\
 & & & \ddots & \ddots & \ddots & \ddots & \ddots & & \\
 & & & & f_{2m-1,2m-3} & f_{2m-1,2m-2} & f_{2m-1,2m-1} & f_{2m-1,2m} & & \\
 & & & & f_{2m,2m-3} & f_{2m,2m-2} & f_{2m,2m-1} & 2m-2 & &
\end{bmatrix}
\begin{bmatrix}
\xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \vdots \\ \xi_{2m-1} \\ \xi_{2m-2}
\end{bmatrix}
=
\begin{bmatrix}
-F_1(\underset{\sim}{x}^{(\nu)}) \\ -F_2(\underset{\sim}{x}^{(\nu)}) \\ -F_3(\underset{\sim}{x}^{(\nu)}) \\ -F_4(\underset{\sim}{x}^{(\nu)}) \\ -F_5(\underset{\sim}{x}^{(\nu)}) \\ -F_6(\underset{\sim}{x}^{(\nu)}) \\ \vdots \\ -F_{2m-1}(\underset{\sim}{x}^{(\nu)}) \\ -F_{2m}(\underset{\sim}{x}^{(\nu)})
\end{bmatrix}$$

FIGURE 1 BITRIDIAGONAL STRUCTURE OF THE LINEAR ALGEBRAIC SYSTEM OF EQUATIONS RESULTING IN NEWTON-RAPHSON ITERATION ALGORITHM.

most recent time levels. Thus:

$$x_i^{n+1(o)} = x_i^n + (WF)\ (\Delta t_{new})\ \frac{x_i - x_i^{n-1}}{(\Delta t_{new})},$$

where (Δt_{old}) is the time step size between x_i^{n-1} and x_i^n, (At_{new}) is the time step size between x_i^n and x_i^{n+1}, and (WF) is a weight factor parameter. This is accomplished in lines #63/64 of the code.

VERIFICATION OF THE NUMERICAL SOLUTION USING A MATERIAL AND ENERGY BALANCE CHECK

Numerical solutions to nonlinear partial differential equations cannot be accepted as being valid without careful scrutiny and examination. This is particularly true when there is an absence of experimental data for the physical system being described by the equations.

One indication of numerical accuracy is the value of the residuals, $F_i(x)$, at each point, $i-1, \ldots, m$. As the numerical solution approaches the true solution to the discretized equations, the residuals should become small, and hence, if sufficient iterations are used, they are an indication of only the effects of round-off error in the machine. Small residuals at each node, however, are not necessarily an indication of physical validity of a numerical solution, since there is still a truncation error associated with the discretization. In theory this should decrease as (Ax) and (At) are decreased.

An important indication of the physical validity of the numerical solution is an overall material and energy balance. It is not obvious that this check should be a sufficient condition for physical validity, but it is clearly a necessary one. Furthermore, it is independent of the particular numerical scheme being used. The procedure simply checks to see if the numerical solution is satisfying overall mass and energy balances. It uses the computed solution to evaluate mass and energy effluxes over time, and in addition computes the mass and energy remaining in the system at various times. If overall mass and energy balances are to be satisfied, then the sum of cumulative mass or energy efflux up to a given time, and the mass or energy remaining in the system at that time, should be constant and equal to the initial mass or energy. Thus, at any time:

$$\frac{\text{Total Mass or Energy Efflux} + \text{Total Remaining Mass or Energy}}{\text{Initial Mass or Energy in the System}} = 1.$$

In the finite difference program being described here, material and energy balances were evaluated after every time step by Subroutine MEBAL (lines #361/429). Simpson's integration rule was used to evaluate the spatial integrals for mass and energy remaining in the system, and hence the total number of grid nodes had to be even. Trapezoidal integration was used to evaluate the time integrals for total mass or energy efflux.

During evaluation, the material balance check, MBAL, was evaluated by summing the total mass efflux and mass remaining at any given time, and then dividing by the initial mass in the system. The energy balance check, EBAL, was evaluated in a similar fashion. During most runs MBAL fell fairly rapidly to about 0.98, and then remained essentially constant at that value. However, EBAL rose slowly with time, reaching values of as high as 1.5 during the longest runs. This result is unsatisfactory, and it is not clear what its cause is. One possibility is that a value of saturation is required at the (m+1)th node in order to evaluate the mass efflux rate, and this value was obtained by extrapolating from internal nodes. The satisfactory behavior of the mass balance check suggests, however, that this explanation is probably incorrect.

Lasseter et al. (1974, p. 107; 1975, p. 24) have observed that an error in the overall heat balance will occur if the same grid system is used for both the mass and energy balance differential equations. However, this observation is in conflict with the experience of Toronyi (1974, p. 108), who evaluated very small mass and energy balance errors (less than 10^{-9}) using a mathematical and numerical formulation similar to that being described in this report. It is thus not apparent that the use of the same grid system for the mass and energy balance differential equations will necessarily lead to significant mass and energy balance errors. The fact that the energy balance check increases slowly while the mass balance check remains essentially constant suggests that there may be an error in the calculations which effectively acts as a heat source.

PROGRAM PARAMETERS

INPUT

With one exception, all input parameters are read in by calling Subroutine IO (lines #273/360) at line #19 in the main program. The parameters and their formatting requirements are presented in Appendix G. The only set of parameters that are not read in are the discretized values of the pressure history at the right-hand boundary condition. These values are specified in the DATA statement (lines #438/439) in Subroutine NUSTEP. PB(i) are the discretized boundary pressures in psia corresponding to the times PT(i), in seconds. The total length of the system being simulated is not read in. It is fixed by the specified grid spacing (DELTXH) and the total number of nodes (NODES), which is equal to (m+1) in the notation of the previous sections.

OUTPUT

The complete solution is printed at every nth time step, where n is the ratio of the input parameter PRNTDL to the initial time step size DELTTK. Output consists of the complete pressure and saturation vectors, PNUK1(NODES) and SNUK1(NODES). Material and energy balance information is also given. In addition, after each iteration the maximum mass and energy residuals, RMMAX and REMAX, and maximum relative changes in the solution vectors, DELPMAX and DELSMAX, are also printed. Appendix D presents the input and output corresponding to a particular simulation of the results of Arihara (1974).

If execution of the program stops normally the p_i^n, S_i^n, $p_i^{n+1(\nu)}$, and $S_i^{n+1(\nu)}$ vectors are written to UNIT 8 (see lines #76/78). These are the program variables POLD(20), SOLD(20), PNUK(20), and SNUK(20). The purpose of doing this is to facilitate restarting of the program at the time of stopping, TIME. Output to UNIT 8 occurs if

1) TIME exceeds the input parameter TMAX;

2) the material or energy balance errors become too large, i.e.,

$$|(\text{MBAL or EBAL}) - 1| \geq \text{BALDEL}; \text{ or}$$

3) there are too many Newton–Rahpson iterations in Subroutine ITSOLV (input parameter MAXNUM is the maximum number of allowed iterations).

Sample output to UNIT 8 is also presented in Appendix D.

SAMPLE VALUES OF PARAMETERS

This subsection describes the numerical values of program parameters used in the simulation runs reported in SGP-TR-1. The definitions of the parameters are described in Appendix G. A total of 21 grid nodes (NODES=21) were used in most of these runs, with an initial time step size of DELTTK = 0.5 sec. The value of the time step control criterion, DTMSCR, was initially set at 0.008 or 0.01, but had to be decreased down to 0.005 or 0.002 as the time step size increased at longer times. Thus, e.g., in the run presented in Appendix D, a value of DTMSCR = 0.01 and initial DELTTK = 0.5 sec allowed the program to reach a simulated time of 30 sec before a doubling of the time step size to 4.0 sec caused the Newton–Raphson iterations to fail to converge to the desired Criterion. 'We program was then restarted with a DELTTK = 2.0 sec, and DTMSCR = 0.002, and it then ran until a simulated time of 600 sec, with a final time step size of 16 sec.

The iteration convergence criterion, DELTA, was commonly set to 10^{-3}. Solutions obtained using this value agreed to within at least four significant figures with those obtained using a value of DELTA = 10^{-10}. The Newton-Raphson algorithm sometimes required as many as 15 iterations to converge to the DELTA criterion of 10^{-3}, and hence a value of MAXNUM = 19 was commonly used. Normally only four or five iterations were. required to reach this convergence criterion. Extra iterations were only necessary after the time step size had been doubled.

The weighting factor, WF, for extrapolating pressures and saturations to the initial guess at a new time level was commonly set at 0.9. The material and energy balance criterion, BALDEL, was usually set to the excessively large value of 0.9, since the energy balance was usually somewhat in error, and it was inconvenient for this to cause program execution to stop.

FUNCTIONAL EVALUATION

The necessary physical properties of saturated water and steam were represented in terms of cubic splines (lines #1000/1106). The splines were generated using 1967 ASME Steam Table data (Meyer et al., 1968). The use of splines has the advantage that they give a very accurate representation of physical data. In addition, they can also be used to obtain smooth first derivatives, a characteristic which is essential to the successful execution of the Newton–Raphson algorithm. However, evaluating splines can require a relatively large amount of computing time. Since most of the computing time in a program such as this is spent evaluating the nonlinear functions, this can clearly be a disadvantage.

There are a number of ways in which this execution time problem can be reduced. Weinstein (1974) has suggested that the use of equally spaced base-point splines with explicit Fortran FUNCTION statements can increase the execution speed without sacrificing the inherent accuracy of the splines. By doing this, no time needs to be spent during execution time in a logical search for the appropriate base points, and in addition the extra CPU time spent in calling a SUBPROGRAM is avoided. Execution time can be reduced by using a simpler representation of physical properties, and also by evaluating the γ_1^{n+1} and γ_3^{n+1} terms explicitly at the values of the nth time level rather implicitly at the (n+1)th time level. Coats ~~et al~~. (1974) used both linear interpolation of physical properties and explicit evaluation of the γ_1n+1 and γ_3^{n+1} terms. This is one reason why their program requires only 0.04 equivalent IBM 360/67 sec per time step-node execution time as compared to the 0.1 sec/time step-node required for this program.

CONCL COMMENTS

This report has discussed in some detail a finite-difference program for describing the boiling two-phase flow of water in a linear horizontal porous medium. Although the basic approach used appears to be sound, the program did not perform in an entirely satisfactory manner. Reasons for this are discussed throughout the report.

The formulation of the numerical solution described in this report is limited to the two-phase flow regime. Generalization of this particular formulation to the solution of the problem which includes single phase flow of compressed liquid or dry steam is not simple. It requires the incorporation of both a moving boundary and a completely different second flow regime.

While the formulation of the numerical solution in terms of the two-phase flow regime only may be adequate for some situations, this will not always be true. There may be important cases where the initial effect of the transition from single-phase to two-phase flow will be important. This appears to be the case, for example, in one of the two-phase depletion experiments of Arihara (1974, pp. 196-206). In these experiments, the liquid in the core was initially compressed and at uniform pressure. While in most of the experiments (Figs. E-5, E-7, and E-9) the initial temperature in the system was uniform, the experiment presented in Fig. E-3 began with a temperature at the closed end of the core that was more than 15 Fahrenheit degrees higher than at the producing end of the core. Chen (1975) has detected initial boiling at the closed end of the core during similar experiments using a capacitance liquid saturation detector (Ramey and London, 1975). Although this determination was initially unexpected and unexplained, it has since become clear that in the presence of an initial temperature gradient (closed end hotter than open end), initial boiling might be expected at the closed end of the core as well as at the producing end. This is a direct consequence of the fact that the closed end is at a higher temperature than the open end, and if pressure gradients in the core are not severe, then the closed end will reach the vapor pressure curve before the open end. While the numerical simulator that has been described in this report cannot describe such behavior, alternate formulations of the problem can. In fact, the model of Garg et al. (1975) predicted initial boiling at the closed end of the core in the experiment corresponding to Fig. E-3 of Arihara. However, the mechanism producing this unexpected behavior was unexplained, and these results were initially considered questionable.

It is reasonable to hypothesize that a single regime model for two-phase flow will be adequate to describe the behavior of compressed liquid systems that are initially at a uniform temperature. The basis for such an hypothesis is the observation that pressure transients move through the isothermal compressed liquid regime at a much faster rate than through the two-phase flow regime. Hence one would expect the overall system behavior to be dominated by flow characteristics in the two-phase regime. This would not necessarily be the case if the initial condition of the system were a compressed liquid of varying temperature. The apparent advantage of a single regime model describing two-phase boiling flow alone is that it could be expected to be less complicated and more efficient than models which describe multiple flow regimes involving both single- and two-phase flow. This is the basis for an incentive to justify the application of two-phase boiling flow models to physical systems which begin in the single-phase flow regime. This incentive will become much less compelling if fast, efficient, and accurate simulators that describe multi-region behavior can be developed.

As a final comment, it is worth remarking that extreme care is required in both the formulation and implementation of a numerical scheme such as that described in this report. Efforts should be made to explore the experiences that other workers have had with the various alternate solution techniques available. While there is often little basis for making a choice:, the experiences of other workers can often facilitate the choice between alternate methods. While such experience is often contained within the published literature, personal communication with knowledgeable workers in the field is invaluable.

REFERENCES

Al-Hussainy, R., personal communication, Stanford University, May 1974.

Arihara, N., A Study of Nonisothermal Single and Two-Phase Flow through Consolidated Sandstones, SGP-TR-2, NSF Grant No. GI-34925, Stanford 'University, November 1974.

Aziz, K., Course Notes from Seminar on Reservoir Simulation, Course No. 273, Department of Petroleum Engineering, Stanford University, Spring 197.1.

Blair, P.M., Peaceman, D.W., and Stone, H.W., Discussion of "Use of Irregular Grid in Cylindrical Coordinates," by A. Settari, K. Aziz, Soc. Petr. Engr. J., August 1974, pp. 396-412.

Brownell, D.H., Jr., Garg, S.K., and Pritchett, J.W., "Computer Simulation of Geothermal Reservoirs," SPE 5381, presented at the 45th Annual California Regional Meeting of the Society of Petroleum Engineers,,Ventura, California, April 2-4, 1975.

Cady, G.V., Bilhartz, H.L., and Ramey, H.J., Jr., "Model Studies of Geothermal Steam Production," AIChE Symposium Series "Water," 1972.

Carnahan, B., Luther, H.A., and Wilkes, J.O., Applied Numerical Methods, John Wiley & Sons, Inc., 1969.

Chen, H.K., personal communication, Stanford University, November 20, 1975.

Coats, K.H., George, W.D., Chu, C., and Marcum, B.E., "Three-Dimensional Simulation of Steamflooding," Soc. Petr. Engr. J., December 1974, pp. 573-592.

Donaldson, I.G., "The Flow of Steam/Water Mixtures through Permeable Beds: A Simple Simulation of a Natural Undisturbed Hydrothermal Region," New Zealand J. Science, 11, No. 1 (1968), pp. 3-23.

Faust, C.R., and Mercer, J.W., "Mathematical Modeling of Geothermal Systems," presented at the Second United Nations Symposium on the Development and Use of Geothermal Resources, May 19-29, 1975, San Francisco, California.

Garg, S.K., Pritchett, J.W., and Brownell, D.H., Jr., "Transport of Mass and Energy in Porous Media," presented at the Second United Nations Symposium on the Development and Use of Geothermal Resources, May 19-29, 1975, San Francisco, California.

Isaacson, E., and Keller, H.B., Analysis of Numerical Methods, John Wiley & Sons, New York, 1966.

Kruger, P., and Ramey, H.J., Jr., Stimulation and Reservoir Engineering of Geothermal Resources, Progress Report No. 3, SGP-TR-1, NSF Grant No. GI-34925, Stanford University, June 1974.

Lasseter, T.J., and Witherspoon, P.A., "Underground Storage of Liquefied Natural Gas in Cavities Created by Nuclear Explosives," Dept. of Civil Engineering, Institute of Transportation and Traffic Engineering Publication No. 74-1 (1974), University of California, Berkeley.

Lasseter, T.J., Witherspoon, P.A., and Lippmann, M.J., "The Numerical Simulation of Heat and Mass Transfer in Multi-Dimensional Two-Phase Geothermal Reservoirs," presented at the Second United Nations Symposium on the Development and Use of Geothermal Resources, May 19-29, 1975, San Francisco, California.

Mercer, J.W., Faust, C., and P'inder, G.F., "Geothermal Reservoir Simulation," Proceedings of National Science Foundation Conference on Research for the Development of Geothermal Energy Resources, Pasadena, California, 1974, pp. 256-267.

Meyer, C.A., McClintock, R.B., Silvestri, G.J., and Spencer, R.C., Jr., 1967 ASME Steam Tables, Am. Soc. Mech. Engrs., 2nd ed., New York, N.Y., 1968.

Ramey, H.J., Jr., and London, A.L., Stimulation and Reservoir Engineering of Geothermal Resources, Progress Report No. 4, SGP-TR-8, NSF Grant No. AER72-03490-A03 (Previous Grant No. GI-34925), Stanford University, August 1975.

von Rosenberg, D.A., Methods for the Numerical Solution of Partial Differential Equations., Elsevier, 1969.

Todd, M.R., O'Dell, P.M., and Hirasaki, G.J., "Methods for Increased Accuracy in Numerical Reservoir Simulators," Soc. Petr. Engr J., April 1972, pp. 515-530.

Toronyi, R.M., "Two-Phase, Two-Dimensional Simulation of a Geothermal Reservoir and the Wellbore System," Ph.D. Dissertation, Pennsylvania State University, November 1974.

Toronyi, R.M., and Farouq Ali, S.M., "Determining Interblock Transmissibility in Reservoir Simulators," JPT Forum, J. Petr. Techn., January 1974, p. 77.

Weinstein, H.G., personal communication, October 3, 1974.

Weinstein, H.G., Wheeler, J.A., and Woods, E.G., "Numerical Model for Steam Stimulation," SPE 4759, presented at Improved Oil Recovery Symposium of SPE, Tulsa, Oklahoma, April 22-24, 1974.

APPENDIX A

THE GOVERNING FLOW EQUATIONS FOR LINEAR TWO-PHASE BOILING FLOW IN A HORIZONTAL POROUS MEDIUM

The differential equations are:

$$\frac{\partial}{\partial x}\left[\gamma_1(p,S_L)\right]\cdot\frac{\partial p}{\partial x} = \frac{\partial\gamma_2}{\partial t}(p,S_L) \ , \tag{A-1}$$

$$\frac{\partial}{\partial x}\left[\gamma_3(p,S_L)\right]\cdot\frac{\partial p}{\partial x} = \frac{\partial\gamma_4}{\partial t}(p,S_L) + \dot{q}'''_{loss} \ . \tag{A-2}$$

The functions, γ_j, are defined:

$$\gamma_1 \triangleq K_L(S_L,p)\cdot\alpha_1(p) + K_g(S_L,p)\cdot\alpha_2(p) \ , \tag{A-3}$$

$$\gamma_2 \triangleq \phi\left[\alpha_3(p) + S_L\cdot\alpha_4(p)\right] \ , \tag{A-4}$$

$$\gamma_3 \triangleq K_L(S_L,p)\cdot\beta_1(p) + K_g(S_L,p)\cdot\beta_2(p) + \kappa\cdot\beta_5(p) \ , \tag{A-5}$$

$$\gamma_4 \triangleq \left[1-\phi\right]\cdot\left[C_{pr}\rho_r\right]\cdot\left[T-T_o\right] + \phi\cdot\left[\beta_3(p) + S_L\cdot\beta_4(p)\right] . \tag{A-6}$$

The dependent variables are:

p = pressure, psia;

S_L = volumetric liquid saturation, volume of liquid in pore space per total volume of pore space

The independent variables are:

x = distance, ft;

t = time, sec.

The specified functional relations for the two-phase Darcy rate equations are the Corey equations (Corey <u>et al</u>., "Three Phase Relative Permeability," <u>Trans. AIME 207</u> (1956), p. 349):

$$K_g = K_{abs}(T)\cdot(1-S_L^{*2})\cdot(1-S_L^{*})^2 \ , \tag{A-7}$$

$$K_L = K_{abs}(T)\cdot(S_L^{*})^4 \ ; \tag{A-8}$$

where:

$$S_L^{*} \triangleq (S_L-S_{Lr})/(1-S_{gr}-S_{Lr}), \tag{A-9}$$

S_{Lr} = residual liquid saturation, a linear function of temperature, liquid volume in pore space per total volume of pore space;

S_{gr} = critical gas saturation, a linear function of temperature, gas volume in pore space per total volume of pore space; and

$K_{abs}(T)$ = absolute permeability, linear function of temperature, Darcys.

Note that for saturation conditions temperature is a function of pressure, and hence the dependence of phase permeabilities on temperature, $K_g(S_L,T)$ and $K_L(S_L,T)$, is also a dependence on pressure: $K_g(S_L,p)$ and $K_L(S_L,p)$.

The physical properties of water are incorporated into the functions α_j and β_j. These are single valued functions of pressure, and are defined:

$$\alpha_1 \triangleq \rho_\ell/\mu_\ell, \ \frac{lb_m}{ft^3 cp} \ , \tag{A-10}$$

$$\alpha_2 \triangleq \rho_g/\mu_g, \ \frac{lb_m}{ft^3 cp} \tag{A-11}$$

$$a_3 \triangleq \%0' \; \frac{lb_m}{ft^3} \; . \tag{A-12}$$

$$\alpha_4 \triangleq \rho_{\ell}-\rho_g, \; \frac{lb_m}{ft^3} \; , \tag{A-13}$$

$$\beta_1 \triangleq \frac{\rho_{\ell} h_{\ell}}{\mu_{\ell}}, \; \frac{Btu}{ft^3 cp} \; , \tag{A-14}$$

$$\beta_2 \triangleq \frac{\rho_g h_g}{\mu_g}, \; \frac{Btu}{ft^3 cp} \; , \tag{A-15}$$

$$\beta_3 \triangleq \rho_g h_g, \; \frac{Btu}{ft^3} \; , \tag{A-16}$$

$$\beta_4 \triangleq \rho_{\ell} h_{\ell} - \rho_{\overset{o}{g}} h_{R'} \; \frac{\cancel{Btu}}{ft^3} \; . \tag{A-17}$$

$$\beta_5 \triangleq \frac{T_{abs} v_{fg}}{h_{fg}} \; , \; \frac{{}^oR \; ft^3}{Btu} \; ; \tag{A-18}$$

where: ρ = density, lb_m/ft^3;

μ = viscosity, c.p.;

h = specific enthalpy, BTU/lb_m oF;

T_{abs} = absolute temperature, oF;

v = specific volume, ft3/lb_m

T = temperature, oF; and

Subscripts: ℓ = liquid phase

g = gas phase

fg = change in going from liquid to gas

o = base value.

The constants in the flow equations are:

ϕ = fractional porosity, $\frac{\text{volume pore space}}{\text{bulk volume of medium}}$;

κ = effective thermal conductivity, $\frac{\text{Btu}}{\text{hr ft }^{\circ}\text{F}}$;

$(C_{pr}\rho_r)$ = specific heat content of matrix rock on a volume basis, $\frac{\text{Btu}}{\text{ft}^3\ {}^{\circ}\text{F}}$ -

The function accounting for heat losses to the environment is:

$\dot{q}'''_{loss}(x,t)$ = local heat **loss** rate from sides of the core per unit length of core and exposed surface area of core

$$= h\ (P/A) \left[T(x,t) - T_\infty\right] . \tag{A-19}$$

where h = steady state convective heat loss coefficient, $\frac{\text{BTU}}{\text{hr ft}^2\ {}^{\circ}\text{F}}$,

(P/A) = ratio of perimeter exposed to heat losses to the cross-sectional area to fluid flow.

In order to make the units given above dimensionally consistent in the governing flow equations, it is necessary to make the following conversions:

a) convert permeability, k (darcies) to (cp ft^2/sec psia) by multiplying by $1/(1.3656 \times 10^4)$;

b) convert thermal conductivity, κ $\frac{\text{BTU}}{\text{hr ft }^{\circ}\text{R}}$ to $\frac{\text{BTU}}{\text{sec ft}^2\ {}^{\circ}\text{R}}$ by multiplying by $\frac{1}{3600}$.

APPENDIX B

SOURCE LISTING OF THE COMPUTER PROGRAM

```
      IMPLICIT REAL*8(A-H,O-Y)
C
C       DECLARE COMMON BLOCKS
C
      COMMON/FS/ POLD,SOLD,PNUK,SNUK,PNUK1,SNUK1
      COMMON/BTRSOL/ DELP,DELS
      COMMON/BITRIN/ A,B,C,D
      COMMON/KSVST/ TF, KABSF, SWRF, SURF
      COMMON/BC/PECLNC, PINIT, SINIT, TMAX,PRNTOL,SNEWBN,SOLDBN,
     Z PBSTOR,PDOWN,BALMAX,BALMIN,NODES,NLESS1,NLESS2,NPLUS1,MAXNUM
      COMMON/CONST/  POR,DELTTK,HR,KAPPA,WF,DELTA, DELTXH,U,PA,TEXTER
     Z ,RHOROK,CPROCK,HP,HS,DTMS,DTMSCR,WFSTOR
      COMMON/BAL/SAVEM,CUMM,SAVEE,CUME,CHECKM,CHECKE,STARTM,STARTE,QM,QE
C
C     DECLARE VARIABLES AND VARIABLE ARRAYS
C
      REAL*8 TF(2), KABSF(2), SWRF(2), SURF(2), KAPPA
      DIMENSION  POLD(20), SOLD(20), PNUK(20), SNUK(20), PNUK1(20),
     Z SNUK1(20),DELP(20),DELS(20)
      DIMENSION  A(20,4), B(20,4), C(20,4), D(20,2)
C
C     INPUT THE DATA AND WRITE OUT ALL HEADINGS
C
      TIME=0.
C
C  ENTER BLOCKS A AND B ON FLOW DIAGRAM, FIG.1, APPENDIX C
C  SEE MARKER NO. 2, APPENDIX C
C
      CALL IC(1,TIME)
C
C INITIALIZE SYSTEM
C ENTER BLOCK C ON FLOW DIAGRAM, FIG.1, APPENDIX C
C
      TPRINT=PRNTOL + TIME
      DO 5 K=1,NODES
      PNUK1(K)=POLD(K)
    5 SNUK1(K)=SOLD(K)
      DO  6 K=1,4
      A(1,K) = 0.
    6 C(NODES,K)= 0.
C
C CONVERT KABSF,KAPPA, & U TO UNITS CONSISTENT WITH REST OF SYSTEM
C
      KABSF(1)=KABSF(1)/(1.3656D4)
      KABSF(2)=KABSF(2)/(1.3656D4)
      KAPPA=KAPPA/(1.9454D4)
      U=U/3600.000
C
C INITIALIZE MASS/ENERGY BALANCE CHECKS
C
      CUMM=0.
      CUME=0.
      CALL MEBAL(2)
C
C WRITE OUT OUTPUT FOR T=0
C
      CALL IC(2,TIME)
      CALL NUSTEP(TIME)
C
```

```
C  SUBROUTINE NUSTEP(TIME) GETS READY FOR THE NEW TIME STEP
C     CALCULATIONS. IT CHECKS DTMS AGAINST DTMSCR TO SEE IF
C     IT IS TIME TO DOUBLE DELTIK. THEN IT CALCULATES THE
C     B.C. FOR THE NEW TIME LEVEL.
C
C  SEE MARKER NO. 4, APPENDIX C
C
C *** ENTER THE TIME STEP LOOP *****
C
C ENTER SOLUTION ITERATIONS
C ENTER BLOCK D ON FLOW DIAGRAM, FIG.1, APPENDIX C
C SEE MARKER NO. 1, APPENDIX C
C
   10  CALL ITSOLV(TIME,&150,&300)
C
C  EVALUATE MAX CHANGE IN SATURATION OVER TIME STEP.
C
       DTMS = DABS(SNUK1(1) - SOLD(1))
       DO 12 K=2,NODES
   12  DTMS = DMAX1( DTMS,DABS(SNUK1(K)-SOLD(K)))
C
C  MASS/ENERGY BALANCE CHECK
C
C  ENTER BLOCK E ON FLOW DIAGRAM, FIG.1, APPENDIX C
C  SEE MARKER NO. 3, APPENDIX C
C
       CALL MEBAL(1)
C
C OUTPUT THE RESULTS IF IT IS TIME TO
C OTHERWISE PRINT RESULTS AND UPDATE KOUNTERS
C  ENTER BLOCK F ON FLOW DIAGRAM, FIG. 1, APPENDIX C
C
       IF (TIME.LT.TPRINT) GO TO 20
C
C  ENTER BLOCK G ON FLOW DIAGRAM, FIG. 1, APPENDIX C
C
       CALL IO(2,TIME)
       TPRINT=TPRINT+PRNTOL
   20  CONTINUE
C
C PREPARE TO ENTER NEW TIME STEP
C  ENTER BLOCK H ON FLOW DIAGRAM, FIG. 1, APPENDIX C
C  SEE MARKER NO. 4, APPENDIX C
C
       CALL NUSTEP(TIME)
C
C          ESTIMATE NEW SOLUTION
C
       DO 30 K=1,NODES
       PNUK(K)=PNUK1(K) + (PNUK1(K) - POLD(K))*WF
       SNUK(K) = SNUK1(K) + (SNUK1(K) - SOLD(K))*WF
       POLD(K)=PNUK1(K)
   30  SOLD(K) = SNUK1(K)
C
C  ENTER BLOCK I ON FLOW DIAGRAM, FIG. 1, APPENDIX C
C     IS THE MASS OR ENERGY BALANCE TOO LARGE?
C
       IF (CHECKM.LT.BALMIN.OR.CHECKM.GT.BALMAX.OR.CHECKE.LT.BALMIN.
     2  OR.CHECKE.GT.BALMAX)  GO TO 300
C
```

```
C  ENTER BLOCK J ON FLOW DIAGRAM, FIG. 1, APPENDIX C
C      IS IT TIME TO STOP?
C
      IF (TIME .GT. TMAX) GO TO 300
      GO TO 10
C
C  REACH HERE AFTER TOO MANY ITERATIONS IN ITSOLV, BLOCK D
C
  150 CALL MEBAL(1)
      WRITE(6,151)
  151 FORMAT(1H0,'    ********** TOO MANY ITERATIONS IN ITSOLVE ******')
  300 CONTINUE
C
C  WRITE MOST RECENT RESULTS TO UNIT 8 BEFORE STOPPING
C
      CALL IC(2,TIME)
      WRITE(8,301) TIME,DELTTK,(POLD(K),K=1,20),(SOLD(K),K=1,20),
     Z (PNUK(K),K=1,20),(SNUK(K),K=1,20)
  301 FORMAT( 2F10.3 / 16(5F10.5/))
      STOP
      END
C
C ******************************************************************
C
      SUBROUTINE ITSOLV(TIME,*,*)
      IMPLICIT REAL*8(A-H,O-Y)
C
C  SUBROUTINE FOR VERSION ** IV **
C      SOLVES OVER GRID FOR GIVEN DELTA TIME USING NEWTON-RAPHSON METHOD
C
C      FIRST EXIT IS FOR TOO MANY ITERATIONS, SECOND FOR DIVIDING BY
C      ZERO IN SOLVBT
C
      COMMON/PS/ POLD,SOLD,PNUK,SNUK,PNUK1,SNUK1
      COMMON/BTRSOL/ DELP,DELS
      COMMON/BITRIN/ A,B,C,D
      COMMON/CONST/  POR,DELTTK,HR,KAPPA,WF,DELTA, DELTXH,U,PA,TEXTER
     Z ,RHOROK,CPROCK,FF,HS,DTMS,DTMSCR,WFSTOR
      COMMON/BC/PBOUND, PINIT, SINIT, TMAX,PRNTOL,SNEWBN,SOLDBN,
     Z PBSTOP,PDOWN,BALMAX,BALMIN,NODES,NLESS1,NLESS2,NPLUS1,MAXNUM
      REAL*8 KAPPA
      DIMENSION POLD(20),SOLD(20),PNUK(20),SNUK(20),PNUK1(20),SNUK1(20),
     Z A(20,4),B(20,4),C(20,4),D(20,2),PSTO(20),SSTO(20),DPOLD(20),
     Z FKM(20),FKE(20),FKMT(20),FKET(20),F20(20),F40(20),
     Z PST(20),SST(20),F1(20),DF1P(20),DF1S(20),F3(20),DF3P(20),DF3S(20)
     Z,F2(20),DF2P(20),DF2S(20),F4(20),DF4P(20),DF4S(20), DPNEW(20),
     Z DELP(20),DELS(20),DPSAVE(20),DSSAVE(20)
C
C FIRST EVALUATE THINGS THAT STAY CONSTANT OVER A TIME STEP
C
      DO 5 K=1,NLESS1
      PSTO(K) = (POLD(K)+POLD(K+1))/2.
      SSTO(K) = (SOLD(K)+SOLD(K+1))/2.
    5 DPOLD(K) = POLD(K+1)-POLD(K)
      SOLDBN = SOLD(NLESS2)-3.*SOLD(NLESS1)+3.*SOLD(NODES)
      IF (SOLDBN.GT.SOLD(NODES))  SOLDBN = SOLD(NODES)
      PSTO(NODES) = (POLD(NODES)+PBOUND)/2.
      SSTO(NODES) = (SOLD(NODES)+SOLDBN)/2.
      DPOLD(NODES)= PBOUND-POLD(NODES)
      TTT = DABS(TIME - DELTTK)
```

```
      IF (TTT.LT.1.0D-07.AND.TTT.GT.-1.0D-07)   DPOLD(NODES) = 0.
      DO 10 K=1,NODES
      FKM(K) = PHI(1,PSTO(K),SSTO(K))*DPOLD(K)
      FKE(K) = PHI(3,PSTO(K),SSTO(K))*DPOLD(K)
      F2O(K) = PHI(2,POLD(K),SOLD(K))
   10 F4O(K) = PHI(4,POLD(K),SOLD(K))
      FKMT(1) = FKM(1)
      FKET(1) = FKE(1)
      DO 12 K=2,NODES
      FKMT(K) = FKM(K)-FKM(K-1)
   12 FKET(K) = FKE(K)-FKE(K-1)
C
C  NOW ENTER ITERATION LOOP; FIRST EVALUATE VARIOUS FUNCTIONS
C
      KOUNT=0
   15 CONTINUE
      DO 20 K=1,NLESS1
      PST(K) = (PNUK(K)+PNUK(K+1))/2.
   20 SST(K) = (SNUK(K)+SNUK(K+1))/2.
      SNEWBN = SNUK(NLESS2)-3.*SNUK(NLESS1)+3.*SNUK(NODES)
      IF (SNEWBN.GT.SNUK(NODES))  SNEWBN = SNUK(NODES)
      PST(NODES) = (PNUK(NODES)+PBOUND)/2.
      SST(NODES) = (SNUK(NODES)+SNEWBN)/2.
      DO 25 K=1,NODES
      DUMP=PST(K)
      DUMS=SST(K)
      F1(K)    = PHI(1,DUMP,DUMS)
      DF1P(K) = DIFF(1,1,DUMP,DUMS,HP,HS)
      DF1S(K) = DIFF(1,2,DUMP,DUMS,HP,HS)
      F3(K)    = PHI(3,DUMP,DUMS)
      DF3P(K) = DIFF(3,1,DUMP,DUMS,HP,HS)
      DF3S(K) = DIFF(3,2,DUMP,DUMS,HP,HS)
      DUMP=PNUK(K)
      DUMS=SNUK(K)
      F2(K)    = PHI(2,DUMP,DUMS)
      CALL GRRHOL(DUMP,GRL)
      CALL GRRHOG(DUMP,GRG)
      DF2P(K) = POR*(GRG + DUMS*(GRL-GRG))
      CALL RHOLTS(DUMP,RL)
      CALL RHOGTS(DUMP,RG)
      DF2S(K) = POR*(RL-RG)
      F4(K)    = PHI(4,DUMP,DUMS)
      CALL GRRLHL(DUMP,GRLH)
      CALL GRRGHG(DUMP,GRGH)
      DF4P(K) =POR*(GRGH + DUMS*(GRLH-GRGH))
     2 + (1.-POR)*CPROCK*RHOROK*TVHP(DUMP)*0.185052
      CALL RGHGTS(DUMP,RGHG)
      CALL RLHLTS(DUMP,RLHL)
   25 DF4S(K) = POR*(RLHL-RGHG)
C
C NOW EVALUATE AND FILL IN THE A,B,C,D ARRAYS FOR SOLVBT
C
C   FIRST DOUBLE ROW FIRST ; ALL OF THIS CAN BE DONE MORE EFFICIENTLY
C
      DPNEW(1)=PNUK(2)-PNUK(1)
      B(1,1) = HR*(DF1P(1)*DPNEW(1)-2.*F1(1)) - DF2P(1)
      B(1,2) = HR*DF1S(1)*DPNEW(1) - DF2S(1)
      B(1,3) = HR*(DF3P(1)*DPNEW(1)-2.*F3(1)) - DF4P(1)
     2   - TVHP(PNUK(1))*U*PA*DELTTK*.09252596
      B(1,4) = HR*DF3S(1)*DPNEW(1) - DF4S(1)
```

```
      C(1,1) = HR*(2.*F1(1)+DF1P(1)*DPNEW(1))
      C(1,2) = HR*DF1S(1)*DPNEW(1)
      C(1,3) = HR*(2.*F3(1)+DF3P(1)*DPNEW(1))
      C(1,4) = HR*DF3S(1)*DPNEW(1)
      D(1,1) = -(2.*HR*(F1(1)*DPNEW(1)+FKMT(1)) + F20(1) - F2(1))
      D(1,2) = -(2.*HR*(F3(1)*DPNEW(1)+FKET(1)) + F40(1) - F4(1)
     2     - ROCKHT(PNUK(1),POLD(1)) )
C
C ASSIGN THE MIDDLE ROWS
C
      DO 30 K=2,NLESS1
      DPNEW(K) = PNUK(K+1) - PNUK(K)
      A(K,1) = HR*(F1(K-1)-DPNEW(K-1)*DF1P(K-1)/2.)
      A(K,2) =-HR*DF1S(K-1)*DPNEW(K-1)/2.
      A(K,3) = HR*(F3(K-1)-DPNEW(K-1)*DF3P(K-1)/2.)
      A(K,4) =-HR*DF3S(K-1)*DPNEW(K-1)/2.
      B(K,1) = HR*(-F1(K)-F1(K-1)+(DPNEW(K)*DF1P(K)-DPNEW(K-1)*
     2     DF1P(K-1))/2. ) -        DF2P(K)
      B(K,2) = HR*(DF1S(K)*DPNEW(K)-DF1S(K-1)*DPNEW(K-1))/2.-DF2S(K)
      B(K,3) = HR*(-F3(K)-F3(K-1)+(DPNEW(K)*DF3P(K)-DPNEW(K-1)*
     2     DF3P(K-1))/2. ) -        DF4P(K)
     2   - TVHP(PNUK(K))*U*PA*DELTTK*.09252596
      B(K,4) = HR*(DF3S(K)*DPNEW(K)-DF3S(K-1)*DPNEW(K-1))/2.-DF4S(K)
      C(K,1) = HR*(F1(K)+DF1P(K)*DPNEW(K)/2.)
      C(K,2) = HR*DF1S(K)*DPNEW(K)/2.
      C(K,3) = HR*(F3(K)+DF3P(K)*DPNEW(K)/2.)
      C(K,4) = HR*DF3S(K)*DPNEW(K)/2.
      D(K,1) =-(HR*(F1(K)*DPNEW(K) - F1(K-1)*DPNEW(K-1) + FKMT(K))
     2          + F20(K) - F2(K))
   30 D(K,2) =-(HR*(F3(K)*DPNEW(K)-F3(K-1)*DPNEW(K-1)+FKET(K)) +F40(K)
     2 -F4(K) - ROCKHT(PNUK(K),POLD(K)) )
C
C  NOW FILL IN THE LAST DOUBLE ROW
C
      DPNEW(NODES) = PBOUND - PNUK(NODES)
      A(NODES,1) = HR*(F1(NLESS1)-DPNEW(NLESS1)*DF1P(NLESS1)/2.)
      A(NODES,2) =-HR*DF1S(NLESS1)*DPNEW(NLESS1)/2.
      A(NODES,3) = HR*(F3(NLESS1)-DPNEW(NLESS1)*DF3P(NLESS1)/2.)
      A(NODES,4) =-HR*DF3S(NLESS1)*DPNEW(NLESS1)/2.
      B(NODES,1) =HR*(-F1(NODES)-F1(NLESS1)+(DPNEW(NODES)*DF1P(NODES)
     2   - DPNEW(NLESS1)*
     2     DF1P(NLESS1))/2. ) - DF2P(NODES)
      B(NODES,2) = HR*(DF1S(NODES)*DPNEW(NODES)-DF1S(NLESS1)
     2  *DPNEW(NLESS1))/2.-DF2S(NODES)
      B(NODES,3) =HR*(-F3(NODES)-F3(NLESS1)+(DPNEW(NODES)
     2  *DF3P(NODES)-DPNEW(NLESS1)*
     2     DF3P(NLESS1))/2. ) - DF4P(NODES)
     2   - TVHP(PNUK(NODES))*U*PA*DELTTK*.09252596
      B(NODES,4) = HR*(DF3S(NODES)*DPNEW(NODES)-DF3S(NLESS1)
     2  *DPNEW(NLESS1))/2.-DF4S(NODES)
      D(NODES,1) =-(HR*(F1(NODES)*DPNEW(NODES) - F1(NLESS1)
     2  *DPNEW(NLESS1) + FKMT(NODES))
     2            + F20(NODES) - F2(NODES) )
      D(NODES,2) =-(HR*(F3(NODES)*DPNEW(NODES) - F3(NLESS1)
     2  *DPNEW(NLESS1) + FKET(NODES))
     2  + F40(NODES) - F4(NODES) - ROCKHT(PNUK(NODES),POLD(NODES)) )
C
C  MATRIX IS NOW FULL. SOLVE MATRIX SYSTEM
C
       CALL SOLVET(NODES,&230)
```

```
C
C SOLUTION VECTOR IS IN DELP AND DELS. EVALUATE NEW SOLUTION
C
      KOUNT = KOUNT + 1
      IF (KOUNT.LT.3.OR.MOD(KOUNT,2).NE.1) GO TO 91
      DO 90 K=1,NODES
      IF (CABS(DPSAVE(K)).LT.1.0D-30) GO TO 80
      AP = DELP(K)/DPSAVE(K)
      IF (CABS(AP).GT.0.95) GO TO 80
      PNUK1(K) = PNUK(K) + DELP(K)/(1.0 - AP)
      GO TO 81
   80 PNUK1(K) = PNUK(K) + DELP(K)
   81 IF (CABS(DSSAVE(K)).LT.1.0D-30)  GO TO 85
      AS = DELS(K)/DSSAVE(K)
      IF (CABS(AS).GT.0.95) GO TO 85
      SNUK1(K) = SNUK(K) + DELS(K)/(1.0 - AS)
      GO TO 90
   85 SNUK1(K) = SNUK(K) + DELS(K)
   90 CONTINUE
      GO TO 92
   91 DO 94 K=1,NODES
      PNUK1(K) = PNUK(K) + DELP(K)
   94 SNUK1(K) = SNUK(K) + DELS(K)
   92 DO 93 K=1,NODES
      DPSAVE(K) = DELP(K)
   93 DSSAVE(K) = DELS(K)
C
C  TEST FOR CONVERGENCE. CONDITION IS THAT THE MAX OF RMMAX AND REMAX
C   MUST BE LESS THAN DELTA. THESE ARE THE MASS AND ENERGY RESIDUALS
C   AT EACH NODE.
C
      DPMAX = CABS(DELP(1)/POLD(1))
      DSMAX = CABS(DELS(1))
      RMMAX = CABS(D(1,1))
      REMAX = CABS(D(1,2))
      DO 40 K=2,NODES
      RMMAX = DMAX1(RMMAX,CABS(D(K,1)))
      REMAX = DMAX1(REMAX,CABS(D(K,2)))
      DPMAX = DMAX1(DPMAX,CABS(DELP(K)/POLD(K)))
   40 DSMAX = DMAX1(DSMAX,CABS(DELS(K)))
      WRITE(6,201) KOUNT,DPMAX,DSMAX,RMMAX,REMAX
  201 FORMAT(' ','ITERATION NUMBER:',I2,5X,'DELPMAX=',1PD10.2,5X,
     2  'DELSMAX=',1PD10.2,5X,'RMMAX=',1PD10.2,5X,'REMAX=',1PD10.2)
      DO 45 I=1,NODES
      PNUK(I) = PNUK1(I)
   45 SNUK(I) = SNUK1(I)
      IF (KOUNT.GT.MAXNUM) RETURN 1
      IF (RMMAX.LT.DELTA.AND.REMAX.LT.DELTA) RETURN
      GO TO 15
  230 WRITE(6,232)
  232 FORMAT(1H0,'****  DIVIDED BY ZERO IN SOLVBT    ****')
      RETURN 2
      END
C
C ******************************************************************
C
      SUBROUTINE IC(L,TIME)
C
C  SEE MARKER NO. 2, APPENDIX C
C
```

```
      IMPLICIT REAL*8(A-H,O-Y)
      COMMON/KSVST/ TF, KABSF, SWRF, SCRF
      COMMON/BC/PBOUND, PINIT, SINIT, TMAX,PRNTOL,SNEWBN,SOLDBN,
     Z PBSTOR,PDOWN,BALMAX,BALMIN,NODES,NLESS1,NLESS2,NPLUS1,MAXNUM
      COMMON/PS/ POLD,SOLD,PNUK,SNUK,PNUK1,SNUK1
      COMMON/CONST/  POR,DELTTK,HR,KAPPA,WF,DELTA, DELTXH,U,PA,TEXTER
     Z ,RHOROK,CPROCK,HP,HS,DTMS,DTMSCR,WFSTOR
      COMMON/BAL/SAVEM,CUMM,SAVEE,CUME,CHECKM,CHECKE,STARTM,STARTE,QM,QE
      REAL*8 KAPPA,PNUK1(20),SNUK1(20),TF(2),KABSF(2),SWRF(2),SCRF(2)
     Z , POLD(20),SOLD(20),PNUK(20),SNUK(20)
      GO TO (10,50),L
C
C    IF L=2 WRITE OUT THE ANSWERS FOR THE MOST RECENT TIME STEP
C       IF L=1 DO THE FOLLOWING
C
C  READ THE INPUT
C
   10 READ(5,101) POR,KAPPA,U,PA,CPROCK,RHOROK
      READ(5,102) (TF(K),K=1,2), (KABSF(K),K=1,2),(SWRF(K),K=1,2),
     Z  (SCRF(K),K=1,2)
      READ(5,103)DELTTK,DELTXH,DELTA,WF,NODES,MAXNUM,BALDEL,PRNTOL,HP,HS
      READ(5,104) PINIT,SINIT,TEXTER,PBOUND,TIME,TMAX
      READ(5,105) PDOWN,DTMSCR
C
C  NOTE THAT THERE MUST BE 20 VALUES  OF EACH VECTOR READ IN, EVEN
C       THOUGH THEY ARE NOT ALL USED.
C
      READ(5,105) (POLD(K),K=1,20), (SOLD(K),K=1,20)
      READ(5,105) (PNUK(K),K=1,20),(SNUK(K),K=1,20)
  101 FORMAT(2F5.2, 4F10.3)
  102 FORMAT(2F5.1,2F10.4,4F5.2)
  103 FORMAT(F8.3,F7.3,D10.3,F5.2,2I4,F5.2,3F6.4)
  104 FORMAT(6F10.2)
  105 FORMAT( 5F10.5)
      HR = (DELTTK)/(2.*DELTXH**2)
      NLESS1 = NODES - 1
      NLESS2 = NODES - 2
      NPLUS1 = NODES + 1
      BALMAX = 1.0 + BALDEL
      BALMIN = 1.0 - BALDEL
      PBSTOR = PBOUND
      PBOUND = PINIT - PDOWN
      WFSTOR = WF
      DTMS = DTMSCR + 0.1
C
C  WRITE OUT INTRODUCTORY HEADINGS
C
      WRITE(6,201)
      WRITE(6,202)
      WRITE(6,203)
      WRITE(6,204)
      WRITE(6,205)
      WRITE(6,206) POR,KAPPA,U,PA,CPROCK,RHOROK
      WRITE(6,207) (TF(K),K=1,2),(KABSF(K),K=1,2), (SWRF(K),K=1,2)
     Z , (SCRF(K),K=1,2)
      WRITE(6,208) DELTTK,DELTXH,DELTA, WF, TMAX,PRNTOL
      WRITE(6,209) PINIT,SINIT,TEXTER,PBSTOR
      WRITE(6,210) HR,HP,HS,NODES,MAXNUM,BALDEL,PDOWN,DTMSCR
  201 FORMAT(' ',' *** VERSION IX ***',T25,'HORIZONTAL LINEAR SINGLE-CO
     ZMPONENT  TWO-PHASE  FLOW OF STEAM AND WATER THROUGH A POROUS MEDIU
```

```
     ZM')
 202  FORMAT(1H0,T5, 'HEAT LOSSES TO EXTERIOR ARE ACCOUNTED FOR',
     Z 5X,'NEWTON-RAPHSON METHOD USED TO SOLVE EQUATIONS')
 203  FORMAT(1H0, T50, ' PAUL G. ATKINSON, FALL 1973')
 204  FORMAT(1H0)
 205  FORMAT(1H0, 'SYSTEM INPUTS ARE AS FOLLOWS: ')
 206  FORMAT(1H0,'POROSITY=',F4.2,4X,'THERMAL CONDUCTIVITY=',F5.3,
     Z 4X,'U TO EXTERIOR=',F6.3,4X,'PERIMETER/AREA=',F6.3,4X,
     Z 'CPROCK=',F6.3,4X,'RHOROK=',F7.2)
 207  FORMAT(1H0,'TEMPERATURES:',2F6.1,5X,'ABSOLUTE PERMS:',2F7.4,
     Z 5X,'SWRS:',2F5.2,5X,'SORS',2F5.2)
 208  FORMAT(1H0,'DELTTK=',F7.2,5X,'DELTXH=',F7.3,5X,'DELTA=',
     Z  1PD9.1,5X,'WF=',0PF4.2,5X,'TMAX=',F10.3,5X,'PRNTDL=',F10.5)
 209  FORMAT(1H0,'INITIAL CONDITIONS ARE :::  P =',F7.2,5X,'SW = ',
     Z F7.3,5X,'TEXTER = ',F7.2,5X,'PBOUND=',F7.2,5X,
     Z'TEMPS ARE IN DEGREES F')
 210  FORMAT(1H0,'      HR IS = (1/2)*(DELTTK/DELTXH**2) =',
     Z    F10.5, '   AND IS DIMENSIONAL',15X,'HP=',F10.7,5X,
     Z 'HS=',F10.7//' NUMBER OF UNPRESCRIBED NODES (MUST BE EVEN AND LT
     Z 20)=',I3,5X,'MAXNUMBER INTERNAL ITERATIONS ALLOWED IN ITSOLV=',
     Z I3// 9X,'MAXIMUM MASS OR ENERGY BALANCE ERROR IS PLUS OR MINUS:',
     Z  F5.3//  9X,'PDOWN=',F5.1, 9X,'DTMS CRITERION=',F7.5)
      WRITE(6,220)
 220  FORMAT(1H1)
      RETURN
C
C  EXECUTE THIS SECOND PORTION IF  L=2
C
  50  WRITE(6,222) (PNUK1(K),K=1,NODES), PBOUND
      WRITE(6,222) (SNUK1(K),K=1,NODES),SNEWBN
      WRITE(6,223) TIME,QM,CUMM,CHECKM,QE,CUME,CHECKE,DELTTK,PRNTDL
 222  FORMAT(1H0,/ 2(11F11.5/))
 223  FORMAT(1H0,'TIME=',F9.3,' ** MASS RATE=',1PD9.2,'  CUMMASS OUT=',
     Z 1PD9.2,'  MBAL=',0PF7.4,' ** EN RATE=',1PD9.2,'  CUMEN OUT=',
     Z 1PD9.2,'  EBAL=',0PF7.4/'      DELTTK=', F7.3, 5X,'PRNTDL=',F7.3/'
     Z***********************************************************
     Z*******************************************************'/)
      RETURN
      END
C
C **********************************************************
C
      SUBROUTINE MEBAL(L)
C
C  SEE MARKER NO. 3, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Y)
      COMMON/BC/PBOUND, PINIT, SINIT, TMAX,PRNTDL,SNEWBN,SOLDBN,
     Z PBSTOR,PDOWN,BALMAX,BALMIN,NODES,NLESS1,NLESS2,NPLUS1,MAXNUM
      COMMON/FS/ POLD,SOLD,PNUK,SNUK,PNUK1,SNUK1
      COMMON/CONST/  POR,DELTTK,HR,KAPPA,WF,DELTA, DELTXH,U,PA,TEXTER
     Z ,RHOROK,CPROCK,HP,HS,DTMS,DTMSCK,WFSTOR
      COMMON/BAL/SAVEM,CUMM,SAVEE,CUME,CHECKM,CHECKE,STARTM,STARTE,QM,QE
      DIMENSION POLD(20),SOLD(20),PNUK(20),SNUK(20),PNUK1(20),SNUK1(20)
      REAL*8  KAPPA,UNIT(21)
C
C       MASS BALANCE FIRST  : CUMULATIVE OUTFLOW OF MASS FIRST
C                 NEED SNUK1 AT BOUNDARY:QUADRATIC EXTRAPOLATION
C
      SNEWBN = SNUK1(NLESS2) - 3.0*SNUK1(NLESS1) + 3.0*SNUK1(NODES)
```

```
      IF (SNEWBN.GT.SNUK1(NODES))  SNEWBN = SNUK1(NODES)
      QRTOT = - PHI(1,PBOUND,SNEWBN)*(3.0*PBOUND - 4.0*PNUK1(NODES)
     2   + PNUK1(NLESS1)) / (2.0*DELTXH)
      IF (L.EQ.2)  SAVEM=QRTOT
      QM = (QRTOT+SAVEM)/2.0
      CUMM = CUMM + QM * DELTTK
      SAVEM = QRTOT
C
C           TOTAL MASS REMAINING IN THE SYSTEM
C                 EVALUATE THE F(X) EXCLUDING POR & DELTXH
C
      DO 5 K=1,NODES
      CALL RHOLTS(PNUK1(K),RHOL)
      CALL RHOGTS(PNUK1(K),RHOG)
    5 UNIT(K) = SNUK1(K)*RHOL + (1.0 - SNUK1(K))*RHOG
      CALL RHOLTS(PBOUND,RHOL)
      CALL RHOGTS(PBOUND,RHOG)
      UNIT(NPLUS1) = SNEWBN* RHOL + (1.0 - SNEWBN) *RHOG
C
C                 EVALUATE THE INTEGRAL EXCLUDING THE POR & DELTXH TERMS
C
      TOTALM = UNIT(1) + UNIT(NPLUS1)
      DO 10 K=2,NODES,2
   10 TOTALM=TOTALM + 4.0*UNIT(K)
      DO 11  K=3,NLESS1,2
   11 TOTALM=TOTALM + 2.0*UNIT(K)
      REMM = TOTALM*POR*DELTXH/3.0
      IF (L.EQ.2) STARTM=(CUMM+REMM)
      CHECKM = (CUMM + REMM) / STARTM
C
C     NOW THE ENERGY BALANCE : CUMULATIVE OUTFLOW OF ENERGY FIRST
C
      TSUM = 0.0
      DO 15 K=2,NODES
   15 TSUM = TSUM + TEMPP(PNUK1(K))
      TSUM = TSUM + TEMPP(PNUK1(1)) + TEMPP(PBOUND) - 40.*TEXTER
      QRHTOT = - PHI(3,PBOUND,SNEWBN)*(3.0*PBOUND - 4.0*PNUK1(NODES)
     2  + PNUK1(NLESS1))/(2.0*DELTXH)   + PA*DELTXH*L*TSUM/2.0
      IF  (L.EQ.2)  SAVEE=QRHTOT
      QE = ( QRHTOT + SAVEE) / 2.0
      CUME = CUME + QE*DELTTK
      SAVEE=QRHTOT
C
C     TOTAL ENERGY REMAINING IN THE SYSTEM
C           CALCULATE THE F(X)'S EXCLUDING POR & DELTXH TERMS
C
      DO 20 K=1,NODES
   20 UNIT(K) = PHI(4,PNUK1(K),SNUK1(K))
      UNIT(NPLUS1) = PHI(4,PBOUND,SNEWBN)
C
C                 EVALUATE THE INTEGRAL EXCLUDING THE POR & DELTXH TERMS
C
      TOTALE = UNIT(1) + UNIT(NPLUS1)
      DO 25  K=2,NODES,2
   25 TOTALE=TOTALE + 4.0*UNIT(K)
      DO 26  K=3,NLESS1,2
   26 TOTALE=TOTALE + 2.0*UNIT(K)
      REME= TOTALE*POR*DELTXH/3.0
      IF (L.EQ.2) STARTE=(CUME+REME)
      CHECKE = (CUME + REME) / STARTE
```

```
      IF (L.EQ.1)  RETURN
C
C   OTHERWISE L=2 AND THIS IS THE FIRST TIME THROUGH
C
      STARTM=REMM
      STARTE=REME
      RETURN
      END
C
C *****************************************************************
C
      SUBROUTINE NUSTEP(TIME)
C
C  SEE MARKER NO. 4, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 KAPPA
      COMMON/BC/RECLNO, PINIT, SINIT, TMAX,PRNTOL,SNEWBN,SOLDBN,
     2 PBSTOR,PDOWN,BALMAX,BALMIN,NODES,NLESS1,NLESS2,NPLUS1,MAXNUM
      COMMON/CONST/  ROR,DELTTK,HR,KAPPA,WF,DELTA, DELTXH,U,PA,TEXTER
     2 ,RHOROK,CPROCK,HP,HS,DTMS,DTMSCR,WFSTOR
      DIMENSION PB(7), PT(7)
      DATA PB/174.,69.,54.,46.,44.,42.,39./,PT/0.,180.,300.,
     2  720.,1020.,1320.,1620./
      IF (DTMS .LE. DTMSCR) GO TO 12
      WF = WFSTOR
      GO TO 10
   12 DELTTK = 2.0 * DELTTK
      PRNTOL = 2.0 * PRNTOL
      WF = 2.0 * WFSTOR
      HR = (DELTTK)/(2.0*DELTXH**2)
   10 TIME = TIME + DELTTK
      DO 2 J=2,7
    2 IF (TIME.LT.PT(J)) GO TO 3
    3 PBOUND=PB(J-1)+(PB(J)-PB(J-1))*(TIME-PT(J-1))/(PT(J)-PT(J-1))
      RETURN
      END
C
C *****************************************************************
C
      SUBROUTINE SOLVET(N,*)
C
C  SEE MARKER NO. 5, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Z)
C
C SR SOLVES BITRIDIAGONAL SYSTEM ABC*UV=D, REF:VON ROSENBERG
C
      COMMON/BITRIN/ A,B,C,D
      COMMON/BTRSOL/ U,V
C
C BITRIN IS INPUT, BTRSOL IS OUTPUT
C
      REAL*8 A(20,4), B(20,4), C(20,4), D(20,2), U(20), V(20),
     2BETA(20,4), LAMBDA(20,4), DELTA(20,2), MU(20), GAMMA(20,2)
      REAL*8  EPS/1.0D-05/, EPS1/1.0D-50/
C
C INITIALIZE FOR DOWNSWEEP
C
      DO 5 J=1,4
```

```
      BETA(1,J) = B(1,J)
      A(1,J) = 0.
    5 C(N,J) = 0.
      DELTA(1,1) = D(1,1)
      DELTA(1,2) = D(1,2)
      MU(1) = BETA(1,1) * BETA(1,4) - BETA(1,2) * BETA(1,3)
C
C  CHECK TO SEE THAT MU .NE. ZERO, ALSO CHECK DURING DOWNSWEEP CALCLNS
C
      I=1
      IF (MU(1).LT.EPS.AND.MU(1) .GT.-EPS)     GO TO 105
      LAMBDA(1,1) = (BETA(1,4) * C(1,1) - BETA(1,2) * C(1,3)) / MU(1)
      LAMBDA(1,2) = (BETA(1,4) * C(1,2) - BETA(1,2) * C(1,4)) / MU(1)
      LAMBDA(1,3) = (BETA(1,1) * C(1,3) - BETA(1,3) * C(1,1)) / MU(1)
      LAMBDA(1,4) = (BETA(1,1) * C(1,4) - BETA(1,3) * C(1,2)) / MU(1)
      GAMMA(1,1) = (BETA(1,4)*DELTA(1,1) - BETA(1,2)*DELTA(1,2))/MU(1)
      GAMMA(1,2) = (BETA(1,1)*DELTA(1,2) - BETA(1,3)*DELTA(1,1))/MU(1)
C
C CALCULATE DOWNSWEEP COEFFICIENTS
C
      DO 10 I=2,N
      BETA(I,1)=B(I,1) - A(I,1)*LAMBDA(I-1,1) - A(I,2)*LAMBDA(I-1,3)
      BETA(I,2)=B(I,2) - A(I,1)*LAMBDA(I-1,2) - A(I,2)*LAMBDA(I-1,4)
      BETA(I,3)=B(I,3) - A(I,3)*LAMBDA(I-1,1) - A(I,4)*LAMBDA(I-1,3)
      BETA(I,4)=B(I,4) - A(I,3)*LAMBDA(I-1,2) - A(I,4)*LAMBDA(I-1,4)
      DELTA(I,1)= D(I,1) - A(I,1)*GAMMA(I-1,1) - A(I,2)*GAMMA(I-1,2)
      DELTA(I,2)= D(I,2) - A(I,3)*GAMMA(I-1,1) - A(I,4)*GAMMA(I-1,2)
      MU(I) = BETA(I,1)*BETA(I,4) - BETA(I,2)*BETA(I,3)
      IF (MU(I).LT.EPS.AND.MU(I) .GT.-EPS)     GO TO 105
      LAMBDA(I,1) = (BETA(I,4) * C(I,1) - BETA(I,2) * C(I,3)) / MU(I)
      LAMBDA(I,2) = (BETA(I,4) * C(I,2) - BETA(I,2) * C(I,4)) / MU(I)
      LAMBDA(I,3) = (BETA(I,1) * C(I,3) - BETA(I,3) * C(I,1)) / MU(I)
      LAMBDA(I,4) = (BETA(I,1) * C(I,4) - BETA(I,3) * C(I,2)) / MU(I)
      GAMMA(I,1) = (BETA(I,4)*DELTA(I,1) - BETA(I,2)*DELTA(I,2)) /MU(I)
   10 GAMMA(I,2) = (BETA(I,1)*DELTA(I,2) - BETA(I,3)*DELTA(I,1)) /MU(I)
      IF (GAMMA(N,1).LT.EPS1.AND.GAMMA(N,1).GT.-EPS1) GAMMA(N,1)=0.0
      IF (GAMMA(N,2).LT.EPS1.AND.GAMMA(N,2).GT.-EPS1) GAMMA(N,2)=0.0
C
C BACK SUBSTITUTION
C
      U(N) = GAMMA(N,1)
      V(N) = GAMMA(N,2)
      I=N-1
   99 U(I) = GAMMA(I,1) - LAMBDA(I,1)*U(I+1)  - LAMBDA(I,2)*V(I+1)
      V(I) = GAMMA(I,2) - LAMBDA(I,3)*U(I+1)  - LAMBDA(I,4)*V(I+1)
      IF (U(I).LT.EPS1.AND.U(I).GT.-EPS1)  U(I)=0.0
      IF (V(I).LT.EPS1.AND.V(I).GT.-EPS1)  V(I)=0.0
      IF (I.EQ.1) GO TO 100
      I=I-1
      GO TO 99
  100 RETURN
  105 WRITE(6,106) I
  106 FORMAT(1H1, 'DIVIDED BY ZERO TO GET MU SUB', I3)
      RETURN 1
      END
C
C ***************************************************************
C
      FUNCTION PHI(U,P,SW)
C
```

```
C  SEE MARKER NO. 6, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Y)
      COMMON/CONST/  POR,DELTTK,HR,KAPPA,WF,DELTA, DELTXH,U,PA,TEXTER
     2 ,RHOROK,CPROCK,HP,HS,DTMS,DTMSCR,WFSTOR
      REAL*8 KL, ML, KG, MG, KAPPA
      GO TO (1,2,3,4),L
    1 CALL RHOLTS(P,RL)
      CALL MLLTS(P,ML)
      CALL RHOGTS(P,RG)
      CALL MLGTS(P,MG)
      T = TEMPP(P)
      PHI = RL*PERM(1,T,SW)/ML   +  RG*PERM(2,T,SW)/MG
      RETURN
    2 CALL RHOGTS(P,RG)
      CALL RHOLTS(P,RL)
      PHI = POR*(RG + SW*(RL-RG))
      RETURN
    3 CALL RLHLTS(P,RLHL)
      CALL MLLTS(P,ML)
      CALL RGHGTS(P,RGHG)
      CALL MLGTS(P,MG)
      T= TEMPP(P)
      PHI = RLHL*PERM(1,T,SW)/ML+ RGHG*PERM(2,T,SW)/MG  + KAPPA*TVHP(P)
      RETURN
    4 CALL RLHLTS(P,RLHL)
      CALL RGHGTS(P,RGHG)
      T = TEMPP(P)
      PHI = POR*(RGHG + SW*(RLHL-RGHG))+(1.-POR)*RHOROK*CPROCK*(T-32.)
      RETURN
      END
C
C ************************************************************************
C
      FUNCTION ROCKHT(PNEW,POLD)
C
C  SEE MARKER NO. 7, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Y)
      COMMON/CONST/  POR,DELTTK,HR,KAPPA,WF,DELTA, DELTXH,U,PA,TEXTER
     2 ,RHOROK,CPROCK,HP,HS,DTMS,DTMSCR,WFSTOR
      REAL*8 KAPPA
      TOLD = TEMPP(POLD)
      TNEW = TEMPP(PNEW)
       ROCKHT =  DELTTK * PA * U * ((TNEW+TOLD)/2. - TEXTER)
      RETURN
      END
C
C ************************************************************************
C
      FUNCTION DIFF(N,M,P,S,HP,HS)
C
C  SEE MARKER NO. 8, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Y)
C
C  SUBPROGRAM TO TAKE THE DERIVATIVE OF PHI(N) WRT P (WHEN M=1) OR WRT
C    S (WHEN M=2); AT CONDITIONS P AND S.
C
C  A SECOND ORDER DIFFERENCING SCHEME (ABOUT P OR S) IS USED .
```

```
C       TRUNCATION ERROR IS OF ORDER H**2
C
        IF (M.EQ.2) GO TO 5
        DIFF = (-PHI(N,P-HP,S)+PHI(N,P+HP,S))/(2.*HP)
        RETURN
      5 DIFF = (-PHI(N,P,S-HS)+PHI(N,P,S+HS))/(2.*HS)
        RETURN
        END
C
C ******************************************************************
C
        FUNCTION PERM(J,T,SW)
C
C  SEE MARKER NO. 9, APPENDIX C
C
        IMPLICIT REAL*8(A-H,O-Y)
C
C  ****  IF J=1 LIQUID (WETTING ) CASE..... IF J=2  STEAM (NONWET) CASE
C
        COMMON/KSVS1/ TF, KABSF, SWRF, SORF
        REAL*8 TF(2), KABSF(2), SWRF(2), SORF(2), KABS
C
C KABSF, SWRF, SORF ARE GIVEN AT TWO TEMPS. USE LINEAR INTERPOLATION
C       BETWEEN THEM TO DETERMINE THE VALUE AT THE REQUIRED TEMPERATURE
C
        DUM=TF(2) - TF(1)
        IF (DUM.GT.1.0E-3.OR.DUM.LT.-1.0E-3) GO TO 5
        SWR=SWRF(1)
        SOR=SORF(1)
        KABS=KABSF(1)
        GO TO 6
      5 TDEL = (T - TF(1)) /DUM
        SWR =  SWRF(1) +  TDEL * (SWRF(2) - SWRF(1))
        SOR = SORF(1) + TDEL * (SORF(2) - SORF(1))
        KABS = KABSF(1) + TDEL * (KABSF(2) - KABSF(1))
      6 SWSTAR = (SW - SWR) / (1. - SWR - SOR)
C
C   DEFINE KABS FOR SW  .GT.(1-SOR) .OR. SW.LT.SWR
C
        IF  (SW.LT.SWR)  GO TO 20
        IF  (SW.GT.(1-SOR)) GO TO 30
C
C  IF J=1 : LIQUID CASE (WETTING PHASE); IF J=2 : STEAM CASE (NONWET)
C
        IF (J.EQ.2)  GO TO 10
        PERM = KABS * SWSTAR**4
        RETURN
     10 PERM = (1. - SWSTAR**2) *((1. - SWSTAR)**2) * KABS
        RETURN
     20 IF (J.EQ.2) GO TO 22
        PERM=0.
        RETURN
     22 PERM=KABS
        RETURN
     30 IF (J.EQ.2) GO TO 32
        PERM=KABS
        RETURN
     32 PERM=0.
        RETURN
        END
```

```
C
C ***********************************************************************
C
      BLOCK  DATA
C
C  SEE MARKER NO. 10, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Y)
      COMMON/PRESS/ XPRESS
      COMMON/LNF/ ALNFML
      COMMON/RHL/ RHOL, ARHOL,BRHOL,CRHOL
      COMMON/RHG/ RHOG, ARHOG,BRHOG,CRHOG
      COMMON/RLH/ RLHL, ARLHL,BRLHL,CRLHL
      COMMON/RGH/ RGHG, ARGHG, BRGHG, CRGHG
      COMMON/ML/ AMUL, AAMUL, BBMUL, CCMUL
      COMMON/MG/ AMUG, AAMUG, BBMUG, CCMUG
      REAL*8 XPRESS(17) /
     Z 6.70000D 01, 8.56400D 01, 1.18000D 02, 1.53000D 02,
     Z 1.95730D 02, 2.47260D 02, 3.08780D 02, 3.81540D 02, 4.66870D 02,
     Z 5.66150D 02, 6.80660D 02, 8.12560D 02, 9.62790D 02, 1.04543D 03,
     Z 1.22688D 03, 1.43150D 03, 1.66160D 03 /
      REAL*8 RHOL(17) /
     Z 5.73066D 01, 5.66251D 01, 5.59597D 01, 5.52181D 01,
     Z 5.44662D 01, 5.36481D 01, 5.27963D 01, 5.19211D 01, 5.09944D 01,
     Z 5.00000D 01, 4.89476D 01, 4.78240D 01, 4.65983D 01, 4.59559D 01,
     Z 4.46030D 01, 4.31220D 01, 4.14594D 01 /,        ARHOL(16)/
     Z-3.15923D-02,-2.71126D-02,-2.17780D-02,-1.96671D-02,
     Z-1.64492D-02,-1.50031D-02,-1.28369D-02,-1.13961D-02,-1.04213D-02,
     Z-9.60248D-03,-8.80513D-03,-8.34055D-03,-7.91905D-03,-7.64757D-03,
     Z-7.30859D-03,-7.21177D-03 / ,               BRHOL(16)/
     Z 0.00000D-01, 1.57867D-04,-9.70695D-06, 7.00784D-05,
     Z 5.22929D-06, 2.28347D-05, 1.23767D-05, 7.42511D-06, 3.99932D-06,
     Z 4.24782D-06, 2.70321D-06, 8.24319D-07, 1.98142D-06, 1.30357D-06,
     Z 5.64606D-07,-9.14366D-08 / ,               CRHOL(16)/
     Z 2.91324D-06,-2.44046D-06, 7.60432D-07,-5.05883D-07,
     Z 1.13885D-07,-5.66643D-08,-2.26846D-08,-1.33825D-08, 9.34349D-10,
     Z-4.48646D-09,-4.75546D-09, 2.56739D-09,-2.73415D-09,-1.35751D-09,
     Z-1.06872D-09, 1.32459D-10/
      REAL*8 RHOG(17) /
     Z 1.54660D-01, 2.03508D-01, 2.64005D-01, 3.38146D-01,
     Z 4.28210D-01, 5.36769D-01, 6.66300D-01, 8.21780D-01, 1.00579D 00,
     Z 1.22373D 00, 1.48166D 00, 1.78712D 00, 2.14994D 00, 2.35671D 00,
     Z 2.83118D 00, 3.40507D 00, 4.11066D 00/ ,       ARHOG(16)/
     Z 2.16272D-03, 2.14742D-03, 2.12369D-03, 2.11253D-03,
     Z 2.10550D-03, 2.10876D-03, 2.11993D-03, 2.14109D-03, 2.17312D-03,
     Z 2.21865D-03, 2.27953D-03, 2.36197D-03, 2.46958D-03, 2.53646D-03,
     Z 2.65230D-03, 2.94311D-03/ ,               BRHOG(16)/
     Z 0.00000D-01,-6.76111D-07,-1.53646D-07,-1.70917D-07,
     Z 6.55783D-09, 5.66315D-08, 1.25003D-07, 1.65735D-07, 2.09621D-07,
     Z 2.49055D-07, 2.81603D-07, 3.44410D-07, 3.71893D-07, 4.37321D-07,
     Z 4.21584D-07, 8.04114D-07/ ,               CRHOG(16)/
     Z-5.95451D-09, 6.14080D-09,-1.64469D-10, 1.38447D-09,
     Z 3.23912D-10, 3.70459D-10, 1.86601D-10, 1.71439D-10, 1.32401D-10,
     Z 9.45786D-11, 1.58966D-10, 6.09606D-11, 2.63908D-10,-2.89101D-11,
     Z 6.23155D-10,-1.16486D-09/
      REAL*8 RLHL(17) /
     Z 1.54556D 04, 1.64439D 04, 1.74203D 04, 1.83490D 04,
     Z 1.92593D 04, 2.01234D 04, 2.09556D 04, 2.17549D 04, 2.25140D 04,
     Z 2.32250D 04, 2.38815D 04, 2.44059D 04, 2.50140D 04, 2.52528D 04,
     Z 2.56735D 04, 2.59983D 04, 2.61940D 04/ ,       ARLHL(16)/
```

```
      Z 4.54552D 01, 4.00475D 01, 2.99623D 01, 2.37601D 01,
      Z 1.89758D 01, 1.49896D 01, 1.22163D 01, 9.69772D 00, 7.99970D 00,
      Z 6.40480D 00, 5.14041D 00, 4.04225D 00, 3.07891D 00, 2.70029D 00,
      Z 1.97008D 00, 1.17765D 00/ ,          BRLHL(16)/
      Z 0.00000D-01,-2.39033D-01,-1.16576D-01,-6.06285D-02,
      Z-5.12439D-02,-2.61905D-02,-1.88894D-02,-1.29760D-02,-9.26728D-03,
      Z-6.79744D-03,-4.22500D-03,-4.11334D-03,-2.29913D-03,-2.28240D-03,
      Z-1.74192D-03,-2.13078D-03/ ,          CRLHL(16)/
      Z-3.51934D-03, 1.43929D-03, 5.32656D-04, 7.32087D-05,
      Z 1.62063D-04, 3.95598D-05, 2.70966D-05, 1.44878D-05, 8.29252D-06,
      Z 7.47519D-06, 2.82627D-07, 4.02541D-06, 6.74742D-08, 9.92877D-07,
      Z-6.33462D-07, 3.08675D-06/
       REAL*8 RCHG(17) /
      Z 1.82452D 02, 2.41198D 02, 3.14193D 02, 4.03882D 02,
      Z 5.12996D 02, 6.44659D 02, 8.02227D 02, 9.89752D 02, 1.21178D 03,
      Z 1.47350D 03, 1.78125D 03, 2.14275D 03, 2.56767D 03, 2.80731D 03,
      Z 3.35098D 03, 3.99653D 03, 4.77206D 03/ ,     ARGHG(16)/
      Z 2.59924D 00, 2.58587D 00, 2.56658D 00, 2.55759D 00,
      Z 2.55268D 00, 2.55717D 00, 2.56727D 00, 2.58795D 00, 2.61681D 00,
      Z 2.65675D 00, 2.70990D 00, 2.78280D 00, 2.87325D 00, 2.92891D 00,
      Z 3.06179D 00, 3.26901D 00/ ,          BRGHG(16)/
      Z 0.00000D-01,-5.90475D-04,-8.98497D-05,-1.67026D-04,
      Z 5.22170D-05, 3.48593D-05, 1.29363D-04, 1.54860D-04, 1.83304D-04,
      Z 2.18975D-04, 2.44370D-04, 3.09170D-04, 2.92895D-04, 3.80716D-04,
      Z 3.51600D-04, 6.61113D-04/ ,          CRGHG(16)/
      Z-8.69368D-06, 5.88417D-06,-7.35068D-07, 1.71029D-06,
      Z-1.12282D-07, 5.12047D-07, 1.16811D-07, 1.11113D-07, 1.19765D-07,
      Z 7.37956D-08, 1.64010D-07,-3.61109D-08, 3.54229D-07,-5.34875D-08,
      Z 5.04207D-07,-9.57717D-07/
       REAL*8 ALNPML(8) /
      Z 2.30258D 00, 2.99573D 00, 3.91202D 00, 4.60517D 00,
      Z 5.29832D 00, 6.21461D 00, 6.90775D 00, 7.60090D 00/
       REAL*8 AMLL(8) /
      Z 3.13000D-01, 2.55000D-01, 1.97000D-01, 1.64000D-01,
      Z 1.38000D-01, 1.11000D-01, 9.40000D-02, 7.79000D-02/
      Z AAMLL(7)/
      Z-8.75223D-02,-7.59845D-02,-5.31788D-02,-4.21835D-02,
      Z-3.34444D-02,-2.60822D-02,-2.35247D-02/ ,     BBMLL(7)/
      Z 0.00000D-01, 1.88455D-02, 8.24360D-03, 7.61937D-03,
      Z 4.98850D-03, 3.04629D-03, 6.43361D-04/ ,     CCMLL(7)/
      Z 8.00481D-03,-3.05649D-03,-3.00194D-04,-1.26518D-03,
      Z-7.06549D-04,-1.15560D-03,-3.09392D-04/
       REAL*8 AMLG(8) /
      Z 1.15000D-02, 1.25000D-02, 1.34000D-02, 1.44000D-02,
      Z 1.53000D-02, 1.72000D-02, 1.91000D-02, 2.25000D-02/,
      Z AAMUG(7)/
      Z 1.58994D-03, 1.14821D-03, 1.23417D-03, 1.35300D-03,
      Z 1.57720D-03, 2.25714D-03, 3.30085D-03/ ,     BBMUG(7)/
      Z 0.00000D-01,-6.37275D-04, 7.31092D-04,-5.59668D-04,
      Z 8.83126D-04,-1.41067D-04, 2.37682D-03/ ,     CCMUG(7)/
      Z-3.06465D-04, 4.97792D-04,-6.20724D-04, 6.93836D-04,
      Z-3.72587D-04, 1.21084D-03,-1.14301D-03/
       END
C
C *************************************************************
C
       SUBROUTINE FHCLTS(X,Y)
C
C  SEE MARKER NO. 11, APPENDIX C
C
```

```
      IMPLICIT REAL*8(A-H,O-Y)
      COMMON/PRESS/ G(17)
      COMMON/FHL/ H(17), B(16), C(16), D(16)
      I=1
  100 IF (X-G(I+1))  11,11,12
   12 I=I+1
      GO TO 100
   11 T=X-G(I)
      Y=((D(I)*T + C(I))*T + B(I))*T +H(I)
      RETURN
      END
C
C ***************************************************************************
C
      SUBROUTINE FHGCTS(X,Y)
C
C  SEE MARKER NO. 12, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Y)
      COMMON/PRESS/ G(17)
      COMMON/FHG/H(17), B(16), C(16), D(16)
      I=1
  100 IF (X-G(I+1))  11,11,12
   12 I=I+1
      GO TO 100
   11 T=X-G(I)
      Y=((D(I)*T + C(I))*T + B(I))*T +H(I)
      RETURN
      END
C
C ***************************************************************************
C
      SUBROUTINE FLHCTS(X,Y)
C
C  SEE MARKER NO. 13, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Y)
      COMMON/PRESS/ G(17)
      COMMON/FLH/H(17), B(16), C(16), D(16)
      I=1
  100 IF (X-G(I+1))  11,11,12
   12 I=I+1
      GO TO 100
   11 T=X-G(I)
      Y=((D(I)*T + C(I))*T + B(I))*T +H(I)
      RETURN
      END
C
C ***************************************************************************
C
      SUBROUTINE FGHCTS(X,Y)
C
C  SEE MARKER NO. 14, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Y)
      COMMON/PRESS/ G(17)
      COMMON/FGH/H(17), B(16), C(16), D(16)
      I=1
  100 IF (X-G(I+1))  11,11,12
   12 I=I+1
```

```
      GO TO 100
   11 T=X-G(I)
      Y=((D(I)*T + C(I))*T + B(I))*T +H(I)
      RETURN
      END
C
C **********************************************************************
C
      SUBROUTINE MLLTS(P,Y)
C
C  SEE MARKER NO. 15, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Y)
      COMMON/LNP/ G(8)
      COMMON/ML/H(8), B(7), C(7), D(7)
      X = DLOG(P)
      I=1
  100 IF (X-G(I+1))  11,11,12
   12 I=I+1
      GO TO 100
   11 T=X-G(I)
      Y=((D(I)*T + C(I))*T + B(I))*T +H(I)
      RETURN
      END
C
C **********************************************************************
C
      SUBROUTINE MLGTS(P,Y)
C
C  SEE MARKER NO. 16, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Y)
      COMMON/LNP/ G(8)
      COMMON/MG/ H(8), B(7), C(7), D(7)
      X = DLOG(P)
      I=1
  100 IF (X-G(I+1))  11,11,12
   12 I=I+1
      GO TO 100
   11 T=X-G(I)
      Y=((D(I)*T + C(I))*T + B(I))*T +H(I)
      RETURN
      END
C
C **********************************************************************
C
      SUBROUTINE DPPHDL(X,Y)
C
C  SEE MARKER NO. 17, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Y)
      COMMON/PRESS/ G(17)
      COMMON/PHL/ H(17), B(16), C(16), D(16)
      I=1
  100 IF (X-G(I+1))  11,11,12
   12 I=I+1
      GO TO 100
   11 T=X-G(I)
      Y=(3.0*D(I)*T + 2.0*C(I))*T + B(I)
      RETURN
```

```
      END
C
C ***************************************************************
C
      SUBROUTINE CRFHCC(X,Y)
C
C  SEE MARKER NO. 18, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/PRESS/ G(17)
      COMMON/FHG/H(17), B(16), C(16), D(16)
      I=1
  100 IF (X-G(I+1))  11,11,12
   12 I=I+1
      GO TO 100
   11 T=X-G(I)
      Y=(3.0*D(I)*T + 2.0*C(I))*T + B(I)
      RETURN
      END
C
C ***************************************************************
C
      SUBROUTINE CFFLHL(X,Y)
C
C  SEE MARKER NO. 19, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/PRESS/ G(17)
      COMMON/FLH/H(17), B(16), C(16), D(16)
      I=1
  100 IF (X-G(I+1))  11,11,12
   12 I=I+1
      GO TO 100
   11 T=X-G(I)
      Y=(3.0*D(I)*T + 2.0*C(I))*T + B(I)
      RETURN
      END
C
C ***************************************************************
C
      SUBROUTINE CFFGHG(X,Y)
C
C  SEE MARKER NO. 20, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/PRESS/ G(17)
      COMMON/FGH/H(17), B(16), C(16), D(16)
      I=1
  100 IF (X-G(I+1))  11,11,12
   12 I=I+1
      GO TO 100
   11 T=X-G(I)
      Y=(3.0*D(I)*T + 2.0*C(I))*T + B(I)
      RETURN
      END
C
C ***************************************************************
C
      FUNCTION TEMPF(F)
C
```

```
C  SEE MARKER NO. 21, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Z)
      TEMPP= -(8400.7358) / (DLOG(P) - 15.272703) - 460.
      RETURN
      END
C
C *****************************************************************************
C
      FUNCTION TVPP(P)
C
C  SEE MARKER NO. 22, APPENDIX C
C
      IMPLICIT REAL*8(A-H,O-Z)
      T = TEMPP(P) + 460.
      TVPP = T * DEXP(( 7356.815)/(T ) - 14.65907)
      RETURN
      END
C
C
C*************     E N D    O F    P R O G R A M       ****************
C
C
```

APPENDIX C

DESCRIPTION OF PROGRAM SUBROUTINES AND LOGICAL STRUCTURE

This appendix contains a flow diagram (Fig. **C-1)** showing the various components of the logical structure of the program, and an extended table giving information on the numerous subroutines in the program.

The extended table comprising the remainder of this appendix contains information on subroutine contents, functions, dummy arguments, and locations in the source listing. The MARKER **NUMBER** in the left-hand column is specified in order to ease reference to this appendix from COMMENT cards in the program source listing.

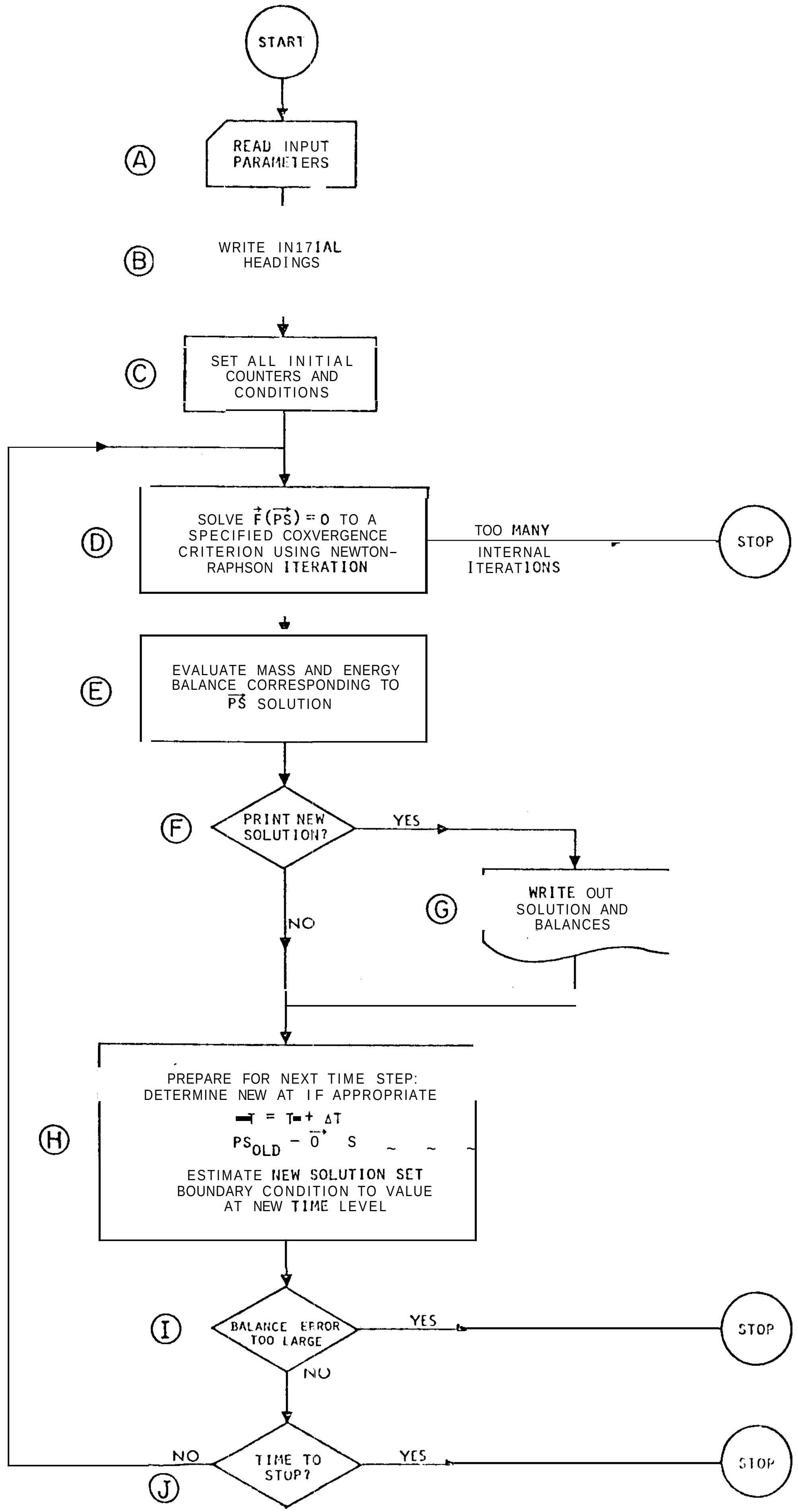

FIGURE C-1. FLOW DIAGRAM OF THE COMPUTER PROGRAM

MARKER NUMBER	SUBROUTINE NAME AND PARAMETERS	LINE NUMBERS IN SOURCE LISTING	COMMENTS AND DESCRIPTION
(1)	ITSOLV(TIME,*,*)	81/272	Executes Block D in the flow diagram, Fig C-1 Called at line #46 only. This is a Newton-Raphson iteration algorithm on the nonlinear discretized system of equation. There is a normal return (line #267) if convergence occurs successfully The first anomalous return (line #266) occurs if there are too many iterations, and the second (line #271) occurs if there is division by zero or a very small number in SUBROUTINE SOLVBT (marker no. 5).
(2)	IO(TIME)	273/360	Controls all input from the card reader (UNIT 5) and output to the line printer (UNIT 6). If L=1 the subroutine executes Block A and part of Blocks B and C of the flow diagram, Fig. C-1. If L=2 it writes the current values of the variables of interest to UNIT 6. L=1 is called only at line #19. L=2 is called at lines #38, 56.
(3)	MEBAL(L)	361/429	Executes Block E of the flow diagram, Fig. C-1. Evaluates current values of the mass and energy balance checks. If L=2 (line #36) the subroutine is being called for the first time, and some of the variables must be initialized All succeeding calls (line #71) are with L=1.
(4)	NUSTEP(TIME)	430/452	Executes Block I of the Flowchart. Prepares for the next time step after a successfully completed Newton-Raphson iteration procedure. Contains the controls for increasing the time step size. This subroutine also includes the specified boundary condition information at the right-hand end of the system (lines 438/439). This information is required in order to fix the right-hand boundary pressure for the new time level. The subroutine is called once at line #39 to initialize for the first time step calculations, and afterwards is called only at line #60.

MARKER NUMBER	SUBROUTINE NAME AND PARAMETERS	LINE NUMBERS IN SOURCE LISTING	COMMENTS AND DESCRIPTION
(5)	SOLVBT(N,*)	453/512	Executes the Gaussian elimination algorithm described in Appendix F. normal return for successful execution. The anomalous return occurs if there is division by zero or a very small number as defined in line #461. This subroutine is called only at line #223 in SUBROUTINE ITSOLV.

The rest of the subroutines listed below are evaluations of the various functions required to generate the numerical solution. The first four subroutines are SUBROUTINE FUNCTIONS of more than one variable. All of the remaining subroutines are functions containing equation of state information for saturated water and steam, and are hence single-valued functions of pressure.

(6)	PHI(L,P,SW)	513/542	This SUBROUTINE FUNCTION returns the value of γ_L as defined in Appendix A. The function is evaluated at the pressure, P, psia, and at the volumetric liquid saturation, SW.
(7)	ROCKHT(PNEW,POLD)	543/552	This SUBROUTINE FUNCTION returns the heat loss rate per unit bulk volume of core averaged over the time interval (TIME) to (TIME + DELTTK). This function is the same as $\dot{q}'''$ in Appendix A (Eq. A-19), and $\gamma_{5,i}^{n+1/2}$ in Appendix E (Eq. E-9).
(8)	DIFF(N,M,P,S,HP,HS)	553/564	A SUBROUTINE FUNCTION which returns the derivative of γ_N with respect to P (if M=1) or S (if M=2) using a centered difference approximation to the first derivative. If M=1 the spacing used is HP on either side of the center node, and if M=2 the spacing is HS. This function is called in SUBROUTINE ITSOLV to evaluate various elements of the Jacobian matrix whose analytic evaluation would be tedious.
(9)	PERM(J,T,SW)	565/602	A SUBROUTINE FUNCTION which returns the absolute permeability to water (if J=1) or steam (if J=2) corresponding to the temperature, T, and liquid saturation, SW. The permeabilities are evaluated using the Corey equations (see Appendix A, Eqs. A-7, 8,9).

MARKER NUMBER	SUBROUTINE NAME AND PARAMETERS	LINE NUMBERS IN SOURCE LISTING	COMMENTS AND DESCRIPTION
(10)	BLOCK DATA	1000/1106	A BLOCK DATA listing containing the cubic spline coefficients for evaluating the functions which contain the physical properties of saturated steam and water.

The following subroutines evaluate various physical properties of saturated steam and water as a function of pressure. The last two subroutines (Markers No. 21 & 22) are FUNCTION SUBROUTINES which evaluate analytic approximations to the physical properties. All the other subroutines are of the form SUBROUTINE NAME(X,Y), and return the physical property value Y, evaluated from the cubic spline coefficients, corresponding to the pressure X.

(11)	RHOLTS(X,Y)	1107/1118	Density of saturated water, ρ_L, lb_m/ft^3
(12)	RHOGTS(X,Y)	1119/1130	Density of saturated steam, ρ_g, lb_m/ft^3
(13)	RLHLTS(X,Y)	1131/1142	Density x specific enthalpy of saturated water, $\rho_L h_L$, BTU/ft^3
(14)	RGHGTS(X,Y)	1143/1154	Density x specific enthalpy of saturated steam, $\rho_g h_g$, BTU/ft^3

NOTE: The spline fits for viscosity are based a logarithmic pressure scale rather than a linear scale

(15)	MULTS(X,Y)	1159/1167	Viscosity of saturated water, μ_L, cp
(16)	MUGTS(X,Y)	1168/1180	Viscosity of saturated steam, μ_g, cp
(17)	GRRHOL(X,Y)	1181/1192	Differential of ρ_L with respect to pressure, $(lb_m/ft^3)/psia$.
(18)	GRRHOG(X,Y)	1193/1204	Differential of ρ_g with respect to pressure, $(lb_m/ft^3)/psia$.
(19)	RRLHL(X,Y)	1205/1216	Differential of $\rho_L h_L$ with respect to pressure, $BTU/ft^3)/psia$.
(20)	GRRGHG(X,Y)	1217/1228	Differential of $\rho_g h_g$ with respect to pressure, $(BTU/ft^3)/psia$

MARKER NUMBER	SUBROUTINE NAME AND PARAMETERS	LINE NUMBERS IN SOURCE LISTING	COMMENTS AND DESCRIPTION
(21)	TEMPP(P)	1229/1233	A FUNCTION SUBROUTINE which evaluates the saturation temperature, in ^{o}F, corresponding to a pressure, P, psia. An analytic approximation to the vapor pressure curve due to Whiting and Ramey (personal communication, H.J. Ramey, Jr., 1972) was used. (See also p. 23 of the Stanford Geothermal Program Technical Report Number 6 by Ramey et al., April 1974]
(22)	T (P]	1234/1239	A FUNCTION SUBROUTINE which evaluates the term β_5 as defined in Appendix A, Eq. A-18. An analytic approximation due to Whiting and Ramey (personal communication, H.J. Ramey, Jr., 1972) is used to represent v_{fg}/h_{fg} as a function of temperature. SUBROUTINE FUNCTION TEMPP(P) is used to determine the absolute saturation temperature, T_{abs}, ^{o}R.

APPENDIX D
SAMPLE INPUT AND OUTPUT

The run presented in this appendix corresponds to a particular simulation of the two--phase boiling flow experiments of Arihara (1974) as described by Kruger and Ramey (1974). The first page of this appendix contains the input: data cards (for UNIT 5) at the top, and the output to UNIT 8 at the bottom. This output occurred at termination of the program due to too many iterations in SUBROUTINE ITSOLV. The remainder of the appendix consists of output to the line printer (UNIT 6). The run was later restarted using the data on UNIT 8 as initial conditions, and with a smaller time step size control parameter, DTMSCR, of 0.002.

```
2207.    .36  3.0    0.0       24.        0023      165.
2208.   1000 100.   0.100000 0.100000   .30   030   005   .05
2209.    0.50     0.05     1.00-04 0.9   20   19 0.9   0.50   0.0001   0.0001
2210.    174.         1.0        370.        174.        000     150.0
2211.      000        0.010
2212.   174.        174.       174.       174.        174.
2213.   174.        174.       174.       174.        174.
2214.   174.        174.       174.       174.        174.
2215.   174.        174.       174.       174.        174.
2216.   100         100        1.0        1.0         100
2217.   1.0         100        1.0        1.0         1.0
2218.   1.0         1.0        1.0        1.0         1.0
2219.   100         100        1.0        1.0         1.0
2220.   174.        174.       174.       174.        174.
2221.   174.        174.       174.       174.        174.
2222.   174.        174.       174.       174.        174.
2223.   174.        174.       174.       174.        174.
2224.   1.0         100        1.0        1.0         100
2225.   1.0         100        1.0        1.0         1.0
2226.   1.0         1.0        1.0        1.0         100
2227.   1.0         1.0        1.0        1.0         1.0
```

```
     33.500      2.000
  173.99996  173.99993  173.99986  173.93967  173.99925
  173.99833  173.99641  173.99251  173.98486  173.97037
  173.94385  173.89696  173.81674  173.68359  173.42781
  172.80593  171.37393  168.95860  165.51532  160.99467
    0.99999    0.99999    0.99998    0.99994    0.99987
    0.99970    0.99936    0.99867    0.99732    0.99475
    0.99004    0.98171    0.96742    0.94371    0.90077
    0.83264    0.77791    0.74334    0.71843    0.69702
  173.99992  173.99989  173.99977  173,99949  173.99887
  173,99757  173.99494  173.98978  173.97996  173.96189
  173.92968  173.87404  173.78082  173.62651  173.30791
  172.52293  170.87225  168.26007  164.62138  159.83806
    0.99999    0,99998    0.99996    0.99991    0.99980
    0.99957    0,99910    0.99819    0.99645    0.99324
    0,98752    0,97763    0.96101    0.93367    0.85353
    0,81501    0.76828    0.73646    0.71322    0.69183
```

```
*** VERSION IX 00*   HORIZONTAL LINEAR SINGLE-COMPONENT  TWO-PHASE  FLOW OF STEAM AND WATER THROUGH A POROUS MEDIUM

  HEAT LOSSES TO EXTERIOR ARE ACCOUNTED FOR      NEWTON-RAPHSON METHOD USED TO SOLVE EQUATIONS

                                             PAUL G. ATKINSON, FALL 1973


SYSTEM INPUTS ARE AS FOLLOWS

POROSITY=0.36    THERMAL CONDUCTIVITY=3.000    U TO EXTERIOR= 0 0      PERIMETER/AREA= 24.000    CPROCK= 0.230    RHOROK= 165.00

TEMPERATURES  100 0 100.0     ABSOLUTE PERMS  0.1000 0 1000     SWRS= 0 30 0 30     SORS 0.05 0.05

DELTTK=   0 50     DELTXH=  0 050    DELTA= 1.00-04     WF=0 90     TMAX=    150 000     PANTOL=   0.50000

INITIAL CONDITIONS ARE :::  P = 174.00     SW =   1.000     TEXTER = 370.00     PBOUND= 174.00     TEMPS ARE IN DEGREES F

    HR IS = (1/Z)*(DELTTK/DELTXH**2  = 100.00000   AND IS DIMENSIONAL              HP= 0.0001000     HS= 0 0001000

NUMBER OF UNPRESCRIBED NODES (MUST BE EVEN AND LT  Z0)= Z0     MAXNUMBER INTERNAL ITERATIONS ALLOWED IN ITSOLV=  9

       MAXIMUM MASS OR ENERGY BALANCE ERROR IS PLUS OR MINUS 0 900

                                         =0 01000
```

```
 174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000
 174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000

   1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000
   1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000

TIME=     0.0   ** MASS RATE= 0.0        CUMMASS OUT= 0.0        MBAL= 1.0000 ** EN RATE= 0.0        CUMEN OUT= 0.0        EBAL= 1.0000
    DELTTX=  0.500      PRNODL=  0.500
**************************************************************************************************************

ITERATION NUMBER: 1      DELPMAX=  1.17D-04      DELSMAX=  3.63D-03      RMMAX=  [illegible].20D-02      REMAX=  2.82D 01
ITERATION NUMBER: 2      DELPMAX=  7.32D-07      DELSMAX=  3.29D-06      RMMAX=  3.86D-07      REMAX=  3.21D-03
ITERATION NUMBER: 3      DELPMAX=  5.67D-00      DELSMAX=  2.46D-08      RMMAX=  8.07D-12      REMAX=  2.37D-05

 174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000
 174.00000  174.00000  174.00000  174.00000  174.00000  173.99909  173.00000  173.90853  173.97070  173.90833

   1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000
   1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    0.99993    0.00975    0.00637    0.98935

TIME=     0.500 ** MASS RATE= 1.12D-02  CUMMASS OUT= 5.59D-03  MBAL= 0.9999 ** EN RATE= 3.84D 00  CUMEN OUT= 1.92D 00  EBAL= 1.0002
    DELTTK=  0.500      PRNTDL=  0.500
**************************************************************************************************************

ITERATION NUMBER: 1      DELPMAX=  9.59D-04      DELSMAX=  3.07D-02      RMMAX=  7.73D-01      REMAX=  2.65D 02
ITERATION NUMBER: 2      DELPMAX=  5.61D-05      DELSMAX=  2.01D-03      RMMAX=  5.07D-02      REMAX=  1.74D 01
ITERATION NUMBER: 3      DELPMAX=  7.13D-06      DELSMAX=  2.16D-04      RMMAX=  5.67D-03      REMAX=  1.95D 00
ITERATION NUMBER: 4      DELPMAX=  1.26D-07      DELSMAX=  6.13D-07      RMMAX=  3.48D-05      REMAX=  1.23D-02
ITERATION NUMBER: 5      DELPMAX=  8.02D-10      DELSMAX=  1.03D-08      RMMAX=  7.77D-08      REMAX=  2.25D-05

 174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000
 174.00000  174.00000  174.00000  173.99909  173.90994  173.00040  173.09006  173.07015  173.78507  173.12500

   1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000
   1.00000    1.00000    1.00000    1.00000    0.00000    0.90001    0.99030    0.00407    0.00003    0.89323

TIME=     1.500 ** MASS RATE= 2.93D-02  CUMMASS OUT= 3.49D-02  MBAL= 0.9973 ** EN RATE= 1.00D 01  CUMEN OUT= 1.20D 01  EBAL= 1.0002
    DELTTK=  1.000      PRNODL=  1.000
**************************************************************************************************************

ITERATION NUMBER: 1      DELPMAX=  2.22D-04      DELSMAX=  6.90D-03      RMMAX=  1.61D-01      REMAX=  5.54D 01
ITERATION NUMBER: 2      DELPMAX=  9.75D-06      DELSMAX=  5.07D-04      RMMAX=  1.21D-02      REMAX=  4.12D 00
ITERATION NUMBER: 3      DELPMAX=  1.63D-06      DELSMAX=  7.20D-05      RMMAX=  1.74D-03      REMAX=  5.93D-01
ITERATION NUMBER: 4      DELPMAX=  2.91D-08      DELSMAX=  2.61D-08      RMMAX=  7.67D-06      REMAX=  2.73D-03
ITERATION NUMBER: 5      DELPMAX=  1.98D-10      DELSMAX=  2.31D-09      RMMAX=  9.64D-09      REMAX=  4.24D-06

 174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000
 174.00000  174.00000  173.00000  173.99993  173.90953  173.09704  173.03241  173.00383  173.58303  172.54167

   1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000
```

```
  1.00000    1.00000    1.00000    0.99090    0.99992    0.99948    0.99085    0.98316    0.92361    0.82422

TIME=     2.500 ** MASS RATE= 3.49D-02  CUMMASS OUT= 0.95D-02  MBAL= 0.9049 ** EN RATE= 1.20D 01  CUMEN OUT= 2.43D 01  EBAL= 1.0005
   DELTTK=  1.000       PRNTDL=  1.000
****************************************************************************************************************************

ITERATION NUMBER: 1      DELPMAX=  1.19D-04      DELSMAX=  4.96D-03      RMMAX=  1.13D-01      REMAX=  3.82D 01
ITERATION NUMBER: 2      DELPMAX=  1.06D-05      DELSMAX=  1.54D-03      RMMAX=  3.28D-02      REMAX=  1.11D 01
ITERATION NUMBER: 3      DELPMAX=  2.72D-06      DELSMAX=  3.03D-04      RMMAX=  6.52D-03      REMAX=  2.21D 00
ITERATION NUMBER: 4      DELPMAX=  1.21D-07      DELSMAX=  4.08D-07      RMMAX=  2.53D-05      REMAX=  9.08D-03
ITERATION NUMBER: 5      DELPMAX=  1.91D-09      DELSMAX=  1.13D-07      RMMAX=  2.55D-06      REMAX=  8.68D-04
ITERATION NUMBER: 6      DELPMAX=  1.26D-10      DELSMAX=  8.03D-09      RMMAX=  1.47D-07      REMAX=  4.90D-05


 174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000
 174.00000  173.90999  173.99995  173.99968  173.99522  173.99079  173.95697  173.82750  173.40917  171.95833


  1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000
  1.00000    1.00000    0.99999    0.09994    0.09909    0.90830    0.98275    0.96911    0.89745    0.77736

TIME=     3.500 ** MASS RATE= 3.30D-02  CUMMASS OUT= 1.03D-01  MBAL= 0.9931 ** EN RATE= 1.14D 01  CUMEN OUT= 3.54D 01  FBAL= 1.0011
   DELTTK=  1.000       PRNTDL=  1.000
****************************************************************************************************************************

ITERATION NUMBER: 1      DELPMAX=  2.63D-04      DELSMAX=  2.19D-03      RMMAX=  4.48D-02      REMAX=  1.54D 01
ITERATION NUMBER: 2      DELPMAX=  7.56D-06      DELSMAX=  7.72D-04      RMMAX=  1.50D-02      REMAX=  5.02D 00
ITERATION NUMBER: 3      DELPMAX=  1.18D-06      DELSMAX=  1.86D-04      RMMAX=  3.68D-03      REMAX=  1.23D 00
ITERATION NUMBER: 4      DELPMAX=  7.34D-08      DELSMAX=  2.88D-07      RMMAX=  1.27D-05      REMAX=  4.60D-03
ITERATION NUMBER: 5      DELPMAX=  4.43D-10      DELSMAX=  1.94D-08      RMMAX=  4.33D-07      REMAX=  1.48D-04
ITERATION NUMBER: 6      DELPMAX=  2.51D-11      DELSMAX=  2.94D-09      RMMAX=  5.76D-08      REMAX=  1.93D-05


 174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000
 173.99990  173.90996  173.90931  173.99902  173.99533  173.90906  173.92245  173.76606  173.20369  171.37500


  1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000
  1.00000    0.99909    0.90900    0.90083    0.90917    0.00641    0.98020    0.05462    0.87306    0.74335

TIME=     4.500 ** MASS RATE= 3.14D-02  CUMMASS OUT= 1.34D-0   MBAL= 0.9910 ** EN RATE= 1.11D 01  CUMEN OUT= 4.05D 01  EBAL= 1.0017
   DELTTK=  1.000       PRNTDL=  1.000
****************************************************************************************************************************

ITERATION NUMBER: 1      DELPMAX=  4.07D-04      DELSMAX=  3.19D-03      RMMAX=  3.65D-02      REMAX=  1.25D 01
ITERATION NUMBER: 2      DELPMAX=  2.82D-05      DELSMAX=  1.28D-03      RMMAX=  2.35D-02      REMAX=  7.80D 00
ITERATION NUMBER: 3      DELPMAX=  6.54D-06      DELSMAX=  3.55D-04      RMMAX=  6.64D-03      REMAX=  2.20D 00
ITERATION NUMBER: 4      DELPMAX=  2.12D-07      DELSMAX=  6.13D-07      RMMAX=  3.10D-05      REMAX=  1.13D-02
ITERATION NUMBER: 5      DELPMAX=  1.89D-10      DELSMAX=  7.64D-08      RMMAX=  1.55D-06      REMAX=  5.23D-04
ITERATION NUMBER: 6      DELPMAX=  3.30D-10      DELSMAX=  9.74D-09      RMMAX=  1.69D-07      REMAX=  5.54D-05


 174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000
 173.99993  173.99989  173.99943  173.00765  173.99031  173.96335  173.38106  173.05059  172.95231  170.7010[illegible]


  1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000
  1.00000    0.09993    0.00991    0.99058    0.99828    0.99359    0.97881    0.93917    0.85547    0.72471

TIME=     5.500 ** MASS RATE= 3.04D-02  CUMMASS OUT=  65D-01  MBAL= 0.9904 ** EN RATE= 1.00D 01  CUMEN OUT= 5.74D 01  EBAL= 1.0025
```

```
    DELTTK=  1.000      PRNTDL=  1.000
*************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  4.450-04     DELSMAX=  3.230-03     RMMAX=  4.660-02     REMAX=  1.600 01
ITERATION NUMBER: 2     DELPMAX=  4.470-05     DELSMAX=  1.550-03     RMMAX=  2.770-02     REMAX=  9.120 00
ITERATION NUMBER: 3     DELPMAX=  1.220-05     DELSMAX=  4.770-04     RMMAX=  8.680-03     REMAX=  2.860 00
ITERATION NUMBER: 4     DELPMAX=  3.130-07     DELSMAX=  8.290-07     RMMAX=  4.160-05     REMAX=  1.530-02
ITERATION NUMBER: 5     DELPMAX=  1.710-09     DELSMAX=  1.570-07     RMMAX=  3.030-06     REMAX=  1.010-03
ITERATION NUMBER: 6     DELPMAX=  1.050-09     DELSMAX=  1.540-08     RMMAX=  2.300-07     REMAX=  7.290-05


 174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  173.99999
 173.99594  173.99973  173.99882  173.99531  173.98284  173.94341  173.83369  173.56153  172.65528  170.20833


   1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000
   0.99999    0.09995    0.90979    0.99017    0.90000    0.03905    0.97033    0.92[illegible]    0.833      0.72001

TIME=    6.500 ** MASS RATE= 3.140-02  CUMMASS OUT= .960-01  MBAL= 0 9890 ** EN RATE= 1.140 01  CUMEN OUT= 9.880 01  EBAL= 1.003
    DELTTK=  1.000      PRNTDL=  1.000
*************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  3.910-04     DELSMAX=  3.490-03     RMMAX=  6.330-02     REMAX=  2.180 01
ITERATION NUMBER: 2     DELPMAX=  4.100-05     DELSMAX=  1.290-03     RMMAX=  2.310-02     REMAX=  7.580 00
ITERATION NUMBER: 3     DELPMAX=  1.250-05     DELSMAX=  4.330-04     RMMAX=  7.830-03     REMAX=  2.580 00
ITERATION NUMBER: 4     DELPMAX=  2.600-07     DELSMAX=  5.350-07     RMMAX=  2.900-05     REMAX=  1.080-02
ITERATION NUMBER: 5     DELPMAX=  1.790-09     DELSMAX=  1.300-07     RMMAX=  2.480-06     REMAX=  8.230-04
ITERATION NUMBER: 6     DELPMAX=  1.040-09     DELSMAX=  1.030-08     RMMAX=  1.340-07     REMAX=  4.070-05


 174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  173.99999  173.99997
 173.99986  173.99941  173.99701  173.99175  173.07230  173.91873  173.78249  173.44372  172.32591  169.62500


   1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    0.99989
   0.99987    0.99900    0.99959    0.99354    0.99517    0.03556    0.96117    0.90349    0.82541    0.72694

TIME=    7.500 ** MASS RATE= 3.490-02  CUMMASS OUT= 2 310-01  MBAL= 0.9390 ** EN RATE= 1.260 01  CUMEN OUT= 8.[illegible] 01  EBAL= 1 0047
    DELTTK=  1.000      PRNTDL=  1.000
*************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  3.270-04     DELSMAX=  2.900-03     RMMAX=  6.050-02     REMAX=  2.100 01
ITERATION NUMBER: 2     DELPMAX=  1.860-05     DELSMAX=  5.230-04     RMMAX=  9.360-03     REMAX=  3.060 00
ITERATION NUMBER: 3     DELPMAX=  5.910-06     DELSMAX=  1.870-04     RMMAX=  3.390-03     REMAX=  1.110 00
ITERATION NUMBER: 4     DELPMAX=  1.460-07     DELSMAX=  1.930-07     RMMAX=  6.710-06     REMAX=  2.830-03
ITERATION NUMBER: 5     DELPMAX=  1.290-09     DELSMAX=  6.130-09     RMMAX=  5.060-08     REMAX=  1.180-05


 174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  173.99998  173 90993
 173.99970  173.99837  173.99599  173.98081  173.96017  173.89062  173.72871  173.31983  171.97505  169.04167


   1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    0.99999
   0.99995    0.09080    0.90929    0.99706    0.99203    0.93054    0.95158    0.33332    0.31196    0.73507

TI       8.500 ** MASS RATE= 4.010-02  CUMMASS OUT= 2.710-01  MBAL= 0.9837 ** EN RATE= 1.440 01  CUMEN OUT= 9.590 01  EBAL= 1.0063
    ELTTK=  1.000      PRNTDL=  1.000
**   *************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  3 0C0-04     DELSMAX=  2 120-03     RMMAX=  5 410-02     REMAX=  1.930 01
```

```
ITERATION NUMBER: 2     DELPMAX=  8.14D-06     DELSMAX=  2.92D-04     RMMAX=  5.36D-03     REMAX=  1.76D 00
ITERATION NUMBER: 3     DELPMAX=  3.97D-C6     DELSMAX=  1.12D-04     RMMAX=  2.01D-03     REMAX=  6.58D-01
ITERATION NUMBER: 4     DELPMAX=  2.90D-06     DELSMAX=  3.06D-07     RMMAX=  3.40D=04     REMAX=  1.27D-01
ITERATION NUMBER: 5     DELPMAX=  2.22D-08     DELSMAX=  1.82D-07     RMMAX=  2.60D-06     REMAX=  8.42D-04
ITERATION NUMBER: 6     DELPMAX=  3.05D-09     DELSMAX=  1.84D-08     RMMAX=  3.03D-07     REMAX=  1.04D-04
ITERATION NUMBER: 7     DELPMAX=  3.42D-10     DELSMAX=  9.14D-09     RMMAX=  1.64D-07     REMAX=  5.35D-05


 174.C000C  174.00000  174.00000  174.0C000  174.00000  174.00000  174.00000  174.00000  173.00999  173.99996  173.09985
 173.99944  173.90304  173.00350  173.98030  173.94507  173.85032  173.07160  173.17447  171.00725  168.45833


   1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    0.99990    0.99997
   0.90990    0.99965    0.99336    0.99652    0.90025    0.07400    0.94147    0.30439    0.70704    0.74210

TIME=     9.500 ** MASS RATE= 4.55D-02  CUMMASS OUT= 3 10D-01  MBAL= 0 9884 ** EN RATE= 1.62O 01  CUMEN OUT= 1 12O O2  EBAL= 1.0081
    DELTTK=  1.000      PRNTDL=  1.000
************************************************************************************************************** ****

ITERATION NUMBER: 1     DELPMAX=  3.[illegible]D-04     DELSMAX=  1.98D-03     RMMAX=  5.14D-02     REMAX=  1.84D 01
ITERATION NUMBER: 2     DELPMAX=  2.33D-C5     DELSHAX=  6.24D-04     RHMAX=  1.13D-02     REMAX=  3.68D 00
ITERATION NUMBER: 3     DELPMAX=  1.04D-05     DELSHAX=  2.46D-04     RMMAX=  4.39D-03     REMAX=  1.43D 00
ITERATION NUMBER: 4     DELPMAX=  2.E0D-07     DELSHAX=  2.11D-07     RMMAX=  3.52D-05     REMAX=  1.28D-02
ITERATION NUMBER: 5     DELPMAX=  1.03D-09     DELSMAX=  5.07D-08     RMMAX=  1.06D-06     REMAX=  3.54D-04
ITERATION NUMBER: 6     DELPMAX=  3.74D=09     DELSMAX=  3.07D-07     RMMAX=  5.98D-06     REMAX=  1.99D-03
ITERATION NUMBER: 7     DELPMAX=  4.57D-09     DELSMAX=  1.10D-07     RMMAX=  1.97D-06     REMAX=  6.40D-04
ITERATION NUMBER: 8     DELPMAX=  1.41D-09     DELSMAX=  4.64D-09     RMMAX=  4.44D-08     REMAX=  1.49D-05


 174.CC000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  173.99998  173.99903  173 09972
 173.99904  173.00084  173.09031  173.97243  173.92753  173.82404  173.01030  173.00E70  171.21824  167.87500


   1.C0000    1.00000    1.00000    1.00000    1.C0000    1.00000    1.00000    1.00000    1.00000    0.99999    0.99995
   0.90983    0.99944    0.9032E    0.09511    0.98714    0.06832    0.93084    0.94610    0.78309    0.74363

TIME=    10.500 ** MASS RATE= 4.93D-02  CUMMASS OUT= 3.00O-01  HBAL= 0.0EE2 ** EN RATE= 1 75D 01  CUMEN OUT= 1 30O O2  EBAL= 1.0101
    DELTTK=  1.000      PRNTDL=  1.000
************************************************ ****************************************************************

ITERATION NUMBER: 1     DELPMAX=  3.45D-04     DELSMAX=  1.90D-03     RMMAX=  3.43D-02     REMAX=  1.28D 01
ITERATION NUMBER: 2     DELPMAX=  3.02D-05     DELSMAX=  6.66D-04     RMMAX=  1.18D-02     REMAX=  3.84D 00
ITERATION NUMBER: 3     DELPMAX=  1.36D-05     DELSMAX=  2.68D-04     RMMAX=  4.70D-03     REMAX=  1.52D 00
ITERATION NUMBER: 4     DELPMAX=  3.53D-07     DELSMAX=  2.96D-07     RMMAX=  3.84D-05     REMAX=  1.40D-02
ITERATION NUMBER: 5     DELPMAX=  2.85D-09     DELSMAX=  1.22D-08     RMMAX=  4.78D-08     REMAX=  7.34D-06


 174.CC0C0  174.00000  174.00000  174.00000  174.00000  174.00000  174.00CCC  173.99999  173.99996  173.99986  173.99953
 173.99844  173.99520  173.98617  173.96292  173.9C788  173.78864  173.54177  172.81727  170.80554  167.29167


   1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    0.99090    E.99998    0.99902
   0.09972    0.99915    0.00755    0.09342    0.00363    0.90233    0.01038    0.E2090    0.77040    C.74086

TIME=    11.500 ** MASS RATE= 5.10D-02  CUMMASS OUT= 4.17D-01  MBAL= 0.9880 ** EN RATE= 1 81D 01  CUMEN OUT= 1 48O O2  EBAL= 1.0121
    DELTTK=  1.000      PRNTDL=  1.000
******************************************************************************************************************

ITERATION NUMBEP: 1     DELPMAX=  3.72D-C4     DELSMAX=  2.21D-03     RMMAX=  3.91D-02     REMAX=  1.35D 01
ITERATION NUMBER: 2     DELPMAX=  2.89D-05     DELSMAX=  5.41D-04     RMMAX=  9.45D-03     REMAX=  3.04D 00
```

```
ITERATION NUMBER: 3     DELPMAX=  1.330-05     DELSMAX=  2.190-04     RMMAX=  3.780-03     REMAX=  1.210 00
ITERATION NUMBER: 4     DELPMAX=  1.090-06     DELSMAX=  3.880-07     RMMAX=  1.120-04     REMAX=  4.210-02
ITERATION NUMBER: 5     DELPMAX=  8.630-09     DELSMAX=  3.800-08     RMMAX=  5.780-08     REMAX=  3.580-05


174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  173.99408   73.09993  173.99976  173.99923
173.99761  173.99300  173.98707  173.95183  173.88020  173.75000  173.40204  172.59005   70.76725  166.70833


  1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    0.99999    0.99996    0.99986
  0.99958    0.99977    0.99064    0.99146    0.97973    0.95550    0.90086    0.81589    0.75860    0.73501

TIME=   12.500 ** MASS RATE= 5.060-02  CUMMASS OUT= 4.670-01  MBAL= 0.987 ** EN RATE= 1.800 01  CUMEN OUT= 1.660 02  EBAL= 1.0141
   DELTTK=  1.000      PRNTOL=  1.000
*****************************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  3.910-04     DELSMAX=  2.210-03     RMMAX=  3.930-02     REMAX=  1.360 01
ITERATION NUMBER: 2     DELPMAX=  2.040-05     DELSMAX=  3.420-04     RMMAX=  5.910-03     REMAX=  1.890 00
ITERATION NUMBER: 3     DELPMAX=  9.890-06     DELSMAX=  1.390-04     RMMAX=  2.370-03     REMAX=  7.500-01
ITERATION NUMBER: 4     DELPMAX=  4.010-07     DELSMAX=  4.900-07     RMMAX=  4.050-05     REMAX=  1.480-02
ITERATION NUMBER: 5     DELPMAX=  4.130-09     DELSMAX=  2.440-08     RMMAX=  2.620-07     REMAX=  6.970-05


174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  173.99999  173.99997  173.99988  173.99962  173.99881
173.99651  173.99036  173.97500  173.93936  173.86279  173.71041  173.37111  172.33855  169.90388  166.12500


  1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    0.99999    0.99998    0.99999    0.99978
  0.99938    0.99839    0.90557    0.99024    0.97560    0.94838    0.89338    0.80021    0.74895    0.72750

TIME=   13.500 ** MASS RATE= 4.870-02  CUMMASS OUT= 5.160-01  MBAL= 0.9875 ** EN RATE= 1.740 01  CUMEN OUT= 1.830 02  EBAL= 1.0161
   DELTTK=  1.000      PRNTOL=  1.000
*****************************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  3.930-04     DELSMAX=  2.130-03     RMMAX=  3.960-02     REMAX=  1.380 01
ITERATION NUMBER: 2     DELPMAX=  8.820-06     DELSMAX=  1.550-04     RMMAX=  2.680-03     REMAX=  8.580-01
ITERATION NUMBER: 3     DELPMAX=  5.140-06     DELSMAX=  6.400-05     RMMAX=  1.070-03     REMAX=  3.360-01
ITERATION NUMBER: 4     DELPMAX=  4.290-07     DELSMAX=  4.950-07     RMMAX=  2.560-05     REMAX=  9.060-03
ITERATION NUMBER: 5     DELPMAX=  6.110-09     DELSMAX=  4.120-08     RMMAX=  5.630-07     REMAX=  1.630-04
ITERATION NUMBER: 6     DELPMAX=  1.330-09     DELSMAX=  1.410-08     RMMAX=  2.290-07     REMAX=  7.110-05


174.00000  174.00000  174.00000  174.00000  174.00000  174.00000  173.99993  173.99994  173.09081  173.99941  173.99825
173.99509  173.03706  173.90097  173.92533  173.83720  173.60008  173.26430  172.05049  169.01022  165.54167


  1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    0.99999    0.99997    0.99989    0.99969
  0.99913    0.99771    0.99431    0.99675    0.97104    0.94000    0.87913    0.79470    0.74164    0.71993

TIME=   14.500 ** MASS RATE= 4.620-02  CUMMASS OUT= 5.620-01  MBAL= 0.9872 ** EN RATE= 1.670 01  CUMEN OUT= 2.000 02  EBAL= 1.0179
   DELTTK=  1.000      PRNTOL=  1.000
*****************************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  3.750-04     DELSMAX=  1.880-03     RMMAX=  3.730-02     REMAX=  1.310 01
ITERATION NUMBER: 2     DELPMAX=  5.600-06     DELSMAX=  2.570-05     RMMAX=  5.810-04     REMAX=  2.000-01
ITERATION NUMBER: 3     DELPMAX=  7.900-07     DELSMAX=  8.530-06     RMMAX=  1.400-04     REMAX=  4.350-02
ITERATION NUMBER: 4     DELPMAX=  1.390-07     DELSMAX=  5.290-07     RMMAX=  5.940-06     REMAX=  1.990-03
ITERATION NUMBER: 5     DELPMAX=  3.070-08     DELSMAX=  3.400-07     RMMAX=  5.590-06     REMAX=  1.740-03
ITERATION NUMBER: 6     DELPMAX=  2.310-09     DELSMAX=  4.520-08     RMMAX=  9.700-07     REMAX=  3.330-04
ITERATION NUMBER: 7     DELPMAX=  1.040-09     DELSMAX=  1.210-08     RMMAX=  2.000-07     REMAX=  6.240-05
```

```
 174.00000  174.00000  174.00000  174.00000  174.00000  173.99999  173.99997  173.99990  173.99970  173.99912  173.99751
 173.99333  173.98313  173.95985  173.90087  173.80930  173.61826  173.14017  171.73583  168.92039  164.95833

   1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    0.99999    0.99998    0.99905    0.99908    0.99030
   0.99832    0.99701    0.99237    0.98399    0.96609    0.93216    0.86441    0.78695    0.73635    0.71330

TIME=   15.500 ** MASS RATE= 4.37D-02  CUMMASS OUT= 6.06D-01  MBAL= 0.9069 ** EN RATE= 1.50D 01  CUMEN OUT= 2.36D 02  EBAL= 1. 197
   DELTTK = 1.000      PRNTDL=  1.000
*****************************************************************************************************************************

ITERATION NUMBER: 1      DELPMAX=  3.38D-04      DELSMAX=  1.69D-03      RMMAX=  3.18D-02      REMAX=  1.14D 01
ITERATION NUMBER: 2      DELPMAX=  1.85D-05      DELSMAX=  1.65D-04      RMMAX=  2.66D-03      REMAX=  8.10D-01
ITERATION NUMBER: 3      DELPMAX=  6.14D-06      DELSMAX=  6.44D-05      RMMAX=  1.05D-03      REMAX=  3.26D-01
ITERATION NUMBER: 4      DELPMAX=  2.03D-07      DELSMAX=  3.40D-07      RMMAX=  8.72D-06      REMAX=  2.78D-03
ITERATION NUMBER: 5      DELPMAX=  2.56D-09      DELSMAX=  4.31D-08      RMMAX=  7.53D-07      REMAX=  2.40D-04
ITERATION NUMBER: 6      DELPMAX=  2.07D-09      DELSMAX=  3.55D-08      RMMAX=  6.27D-07      REMAX=  2.00D-04
ITERATION NUMBER: 7      DELPMAX=  1.24D-09      DELSMAX=  1.31D-08      RMMAX=  2.15D-07      REMAX=  6.65D-05


 174.00000  174.00000  174.00000  174.00000  174.00000  173.99999  173.99995  173.99985  173.99956  173.99873  173.99657
 173.99118  173.97854  173.95066  173.89235  173.73013  173.56409  172.99669  171.40058  168.41676  164.37500

   1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    0.99999    0.99997    0.99992    0.99978    0.99939
   0.99334    0.99020    0.99125    0.98097    0.90084    0.92288    0.83907    0.78044    0.73329    0.70822

TIME=   16.500 ** MASS RATE= 4.17D-02  CUMMASS OUT= 6.48D-01  MBAL= 0.9860 ** EN RATE= 1.53D 01  CUMEN OUT= 2.31D 02  EBAL= 1.0213
   DELTTK=  1.000      PRNTDL=  1.000
*****************************************************************************************************************************

ITERATION NUMBER: 1      DELPMAX=  2.92D-04      DELSMAX=  1.83D-03      RMMAX=  3.36D-02      REMAX=  1.16D 01
ITERATION NUMBER: 2      DELPMAX=  2.51D-05      DELSMAX=  2.31D-04      RMMAX=  3.73D-03      REMAX=  1.14D 00
ITERATION NUMBER: 3      DELPMAX=  9.08D-06      DELSMAX=  9.16D-05      RMMAX=  1.49D-03      REMAX=  4.60D-01
ITERATION NUMBER: 4      DELPMAX=  1.70D-07      DELSMAX=  2.24D-07      RMMAX=  4.78D-06      REMAX=  2.27D-03
ITERATION NUMBER: 5      DELPMAX=  3.41D-09      DELSMAX=  4.81D-08      RMMAX=  8.21D-07      REMAX=  2.59D-04
ITERATION NUMBER: 6      DELPMAX=  1.96D-09      DELSMAX=  1.92D-08      RMMAX=  3.10D-07      REMAX=  9.49D-05


 174.00000  174.00000  174.00000  174.00000  173.99999  173.99998  173.99993  173.99978  173.99936  173.99824  173.99540
 173.98861  173.97324  173.94044  173.87446  173.74937  173.50259  172.83262  171.05297  167.91742  163.79167

   1.00000    1.00000    1.00000    1.00000    1.00000    1.00000    0.99909    0.99996    0.99989    0.99969    0.99910
   0.90763    0.00526    0.98943    0.97760    0.95536    0.91271    0.83539    0.77463    0.73124    0.70522

TIME=   17.500 ** MASS RATE= 4.06D-02  CUMMASS OUT= 6.88D-01  MBAL= 0.9804 ** EN RATE= 1.49D 01  CUMEN OUT= 2.46D 02  EBAL= 1.0230
   DELTTK=  1.000      PRNTDL=  1.000
*****************************************************************************************************************************

ITERATION NUMBER: 1      DELPMAX=  2.53D-04      DELSMAX=  1.75D-03      RMMAX=  3.20D -2      REMAX=  1.11D 01
ITERATION NUMBER: 2      DELPMAX=  2.32D-05      DELSMAX=  2.10D-04      RMMAX=  3.40D -3      REMAX=  1.04D 00
ITERATION NUMBER: 3      DELPMAX=  8.63D-06      DELSMAX=  8.49D-05      RMMAX=  1.38D -3      REMAX=  4.25D-01
ITERATION NUMBER: 4      DELPMAX=  1.41D-07      DELSMAX=  1.44D-07      RMMAX=  5.25D -6      REMAX=  2.31D-03
ITERATION NUMBER: 5      DELPMAX=  3.83D-09      DELSMAX=  4.91D-08      RMMAX=  8.30D -7      REMAX=  2.60D-04
ITERATION NUMBER: 6      DELPMAX=  1.31D-09      DELSMAX=  1.16D-08      RMMAX=  2.72D -7      REMAX=  9.61D-05

 174.00000  174.00000  174.00000  174.00000  173.99999  173.99996  173.99980  173.99967  173.99910  173.99701  173.99399
```

```
173.98560  173.96723  173.92918  173.83407  173.71055  173.43 6   172.64059  170.69823  167.42834  163.20833

   1.00000    1.00000    1.00000    1.00000    1.00000    0.99999    0.99908    0.9990 *   0.99984    0.99958    0.99893
   0.99745    0.99419    0.98743    0.97410    0.94051    0.90180    0.82205    0.769 1    0.72971    0.70384

TIME=   18.500 ** MASS RATE= 4.030-02  CUMMASS OUT= 7 280- 1  MBAL=   9802 ** EN RATE= 1 490 01  CUMEN OUT= 2 01  02    AL= 1.0246
    DELTTK=  1.000     PRNTOL=  1.000
************************************************** ******* ************************************************ ***** ** ******* ****

ITERATION NUMBER: 1     DELPMAX=  2.340-04     DELSMAX=  1.650-03     RMMAX=  3.070-02     REMAX=  1.070 01
ITERATION NUMBER: 2     DELPMAX=  1.410-05     DELSMAX=  1.180-04     RMMAX=  1.910-03     REMAX=  5.790-01
ITERATION NUMBER: 3     DELPMAX=  5.060-06     DELSMAX=  4.870-05     RMMAX=  7.960-04     REMAX=  2.440-01
ITERATION NUMBER: 4     DELPMAX=  1.300-07     DELSMAX=  1.060-07     RMMAX=  5.490-06     REMAX=  2.350-03
ITERATION NUMBER: 5     DELPMAX=  3.580-09     DELSMAX=  4.450-08     RMMAX=  7.560-07     REMAX=  2.360-04
ITERATION NUMBER: 6     DELPMAX=  1.360-09     DELSMAX=  4.950-09     RMMAX=  1.430-07     REMAX=  5.330-05


174.00000  174.00000  174.00000  173.99999  173.99998  173.99994  173.99983  173.99954  173.99876  173.99684  173.99230
173.98212  173.96048  173.91684  173.83336  173.68128  173.35466  172.44220  170.33982  166.95077  162.62500

   1.00000    1.00000    1.00000    1.00000    1.00000    0.99990    0.99097    0.99992    0. 9978    0.99944    0.99804
   0.99682    0.99295    0.98524    0.97030    0.94325    0.89033    0.81004    0.70365    0. 2808    0.70333

TIME=   19.500 ** MASS RATE= 4.060-02  CU MASS OUT= 7.090-01  MBAL= 0.0800 ** EN RATE= 1.500 01  CUMEN OUT= 2. 8D 02  EBAL= 1.0263
    DELTTK=  1.000     PRNTDL=  1.000
********** * ***** ************************************ ************ ***************************** * ********** ** ** **************

ITERATION NUMBER: 1     DELPMAX=  2.390-04     DELSMAX=  1.500-03     RMMAX=  2.870-02     REMAX=  1.000 01
ITERATION NUMBER: 2     DELPMAX=  2.250-06     DELSMAX=  3.630-06     RMMAX=  5.650-05     REMAX=  2.550-02
ITERATION NUMBER: 3     DELPMAX=  4.200-08     DELSMAX=  5.660-07     RMMAX=  9.460-06     REMAX=  2.960-03
ITERATION NUMBER: 4     DELPMAX=  2.050-08     DELSMAX=  4.740-08     RMMAX=  2.690-07     REMAX=  4.720-06


174.00000  174.00000  174.00000  173.99999  173.99997  173.99992  173.99976  173.99936  173.99834  173.99590  173.99032
173.97815  173.95297  173.90338  173.81037  173.64319  173.26574  172.21786  169.97788  166.48145  162.04167

   1.00000    1.00000    1.00000    1.00000    0.99999    0.99990    0.99990    0.99989    0.99971    0.99927    0.99828
   0.99013    0.99106    0.98284    0.90620    0.93654    0.87851    0.79960    0.75821    0.72596    0.70283

TIME=   20.500 ** MASS RATE= 4.130-02  CUMMASS OUT= 8.100-01  HBAL= 0 9958 ** EN RATE= 1.590 01  CUMEN OUT= 2 910 02  EBAL= 1 0281
    DELTTK=  1.000     PRNTOL=  1.000
********************************************* ************* *************** ******************** ***** ************************

ITERATION NUMBER: 1     DELPMAX=  2.630-04     DELSMAX=  1.310-03     RMMAX=  2.560-02     REMAX=  9.050 00
ITERATION NUMBER: 2     DELPMAX=  8.570-06     DELSMAX=  1.000-04     RMMAX=  1.680-03     REMAX=  5.280-01
ITERATION NUMBER: 3     DELPMAX=  4.570-06     DELSMAX=  4.180-05     RMMAX=  6.860-04     REMAX=  2.090-01
ITERATION NUMBER: 4     DELPMAX=  2.280-07     DELSMAX=  4.190-07     RMMAX=  1.110-05     REMAX=  4.670-03
ITERATION NUMBER: 5     DELPMAX=  1.010-08     DELSMAX=  1.070-07     RMMAX=  1.820-06     REMAX=  5.620-04
ITERATION NUMBER: 6     DELPMAX=  5.160-09     DELSMAX=  4.280-08     RMMAX=  6.790-07     REMAX=  2.040-04
ITERATION NUMBER: 7     DELPMAX=  1.820-09     DELSMAX=  1.650-08     RMMAX=  2.710-07     REMAX=  8.240-05


174.00000  174.00000  174.00000  173.99999  173.99996  173.99988  173.99967  173.99914  173.99783  173.99479  173.98802
173.97367  173.94467  173.88879  173.78604  173.60134  173.16493  171.97649  169.61094  166.01441  161.45833

   1.00000    1.00000    1.00000    1.00000    0.99999    0.99995    0.99994    0.99085    0.99962    0     08    0.99788
```

```
  0.99533   0.99013   0.98025   0.96192   0.92926   0.86656   0.79074   0.75287   0.72321   .70175

TIME=   21.500 ** MASS RATE= 4.20D-02  CUMMASS OUT= 8.52D-01  MBAL= 0.9857 00  EN RATE= 1.55D 01  CUMEN OUT= 3.07D 02  EBAL= 1.0298
    DELTTK=  1.000      PRNTDL=  1.000
**************************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  2.97D-04     DELSMAX=  1.41D-03     RMMAX=  2.60D-02     REMAX=  8.93D 00
ITERATION NUMBER: 2     DELPMAX=  1.41D-05     DELSMAX=  1.48D-04     RMMAX=  2.47D-03     REMAX=  7.57D-01
ITERATION NUMBER: 3     DELPMAX=  7.09D-06     DELSMAX=  6.22D-05     RMMAX=  1.02D-03     REMAX=  3.08D-01
ITERATION NUMBER: 4     DELPMAX=  2.57D-07     DELSMAX=  2.44D-07     RMMAX=  1.20D-05     REMAX=  5.10D-03
ITERATION NUMBER: 5     DELPMAX=  6.52D-09     DELSMAX=  7.24D-08     RMMAX=  1.25D-06     REMAX=  3.88D-04
ITERATION NUMBER: 6     DELPMAX=  2.16D-09     DELSMAX=  1.56D-08     RMMAX=  3.77D-07     REMAX=  1.34D-04
ITERATION NUMBER: 7     DELPMAX=  5.51D-10     DELSMAX=  5.03D-09     RMMAX=  8.32D-08     REMAX=  2.53D-05


 174.00000  174.00000  173.99999  173.99998  173.99994  173.99983  173.99955  173.99886  173.99721  173.99348  173.98539
 173.96865  173.93557  173.87315  173.76070  173.55470  173.05078  171.72019  169.23651  165.54332  160.87500


   1.00000   1.00000   1.00000   1.00000   0.99999   0.99997   0.99992   0.99980   0.99951   0.99884   0.99741
   0.99444   0.98857   0.97746   0.95740   0.92130   0.85472   0.78334   0.74777   0.71995   0.69987

TIME=   22.500 ** MASS RATE= 4.25D-02  CUMMASS OUT= 8 95D 01  MBAL= 0.9855 00  EN RATE=  570  1  CUMEN OUT= 3 22D 02  EBAL= 1   16
    DELTTK=  1.000      PRNTDL=  1.000
**************************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  3.29D-04     DELSMAX=  1.45D-03     RMMAX=  2.66D-02     REMAX=  9.15D 00
ITERATION NUMBER: 2     DELPMAX=  1.44D-05     DELSMAX=  1.46D-04     RMMAX=  2.43D-03     REMAX=  7.42D-01
ITERATION NUMBER: 3     DELPMAX=  7.33D-06     DELSMAX=  6.12D-05     RMMAX=  1.00D-03     REMAX=  3.01D-01
ITERATION NUMBER: 4     DELPMAX=  2.88D-07     DELSMAX=  2.46D-07     RMMAX=  1.23D-05     REMAX=  5.33D-03
ITERATION NUMBER: 5     DELPMAX=  5.66D-09     DELSMAX=  6.34D-08     RMMAX=  1.10D-06     REMAX=  3.43D-04
ITERATION NUMBER: 6     DELPMAX=  1.85D-09     DELSMAX=  1.61D-08     RMMAX=  3.75D-07     REMAX=  1.32D-04
ITERATION NUMBER: 7     DELPMAX=  6.08D-10     DELSMAX=  5.26D-09     RMMAX=  8.67D-08     REMAX=  2.62D-05


 174.00000  174.00000  173.99999  173.99997  173.99991  173.99977  173.99940  173.99851  173.99647  173.99195  173.98240
 173.96307  173.92568  173.95648  173.73403  173.50281  172.92215  171.45085  168.85240  165.06334  160.29167


   1.00000   1.00000   1.00000   0.99999   0.99998   0.99990   0.99989   0.99974   0.99937   0.99857   0.99638
   0.99345   0.98081   0.97450   0.95265   0.91268   0.84320   0.77714   0.74305   0.71642   0.69726

TIME=   23.500 ** MASS RATE= 4.25D-02  CUMMASS OUT= 9 37D-01  MBAL= 0.9854 00  EN RATE= 1.58D 01  CUMEN OUT= 3.38D 02  EBAL= 1.0334
    DELTTK=  1.000      PRNTDL=  1.000
**************************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  3.52D-04     DELSMAX=  1.45D-03     RMMAX=  2.66D-02     REMAX=  9.16D 00
ITERATION NUMBER: 2     DELPMAX=  1.08D-05     DELSMAX=  1.11D-04     RMMAX=  1.85D-03     REMAX=  5.66D-01
ITERATION NUMBER: 3     DELPMAX=  5.84D-06     DELSMAX=  4.65D-05     RMMAX=  7.58D-04     REMAX=  2.26D-01
ITERATION NUMBER: 4     DELPMAX=  3.18D-07     DELSMAX=  1.90D-07     RMMAX=  1.17D-05     REMAX=  5.24D-03
ITERATION NUMBER: 5     DELPMAX=  3.30D-09     DELSMAX=  4.33D-08     RMMAX=  7.74D-07     REMAX=  2.43D-04
ITERATION NUMBER: 6     DELPMAX=  2.17D-09     DELSMAX=  1.70D-08     RMMAX=  2.76D-07     REMAX=  8.22D-05


 174.00000  173.99999  173.99998  173.99995  173.99988  173.99968  173.99921  173.99810  173.99560  173.99020  173.97904
 173.95693  173.91499  173.83874  173.70569  173.44517  172.77835  171.17002  168.45758  164.57220  159.70833


   1.00000   1.00000   1.00000   0.99999   0.99998   0.99994   0.99986   0.99966   0.99922   0.99 26   0.99628
   0.99230   0.98491   0.97133   0.94761   0.90348   0.83222   0.77184   0.73878   0.71291   0.69 22
```

```
TIME=   24.500 ** MASS RATE= 4.21D-02  CUMMASS OUT= 0.79D-01  MBAL= 0.9852     EN RATE= 1.57D 01  CUMEN OUT= 3.54D 02  EBAL= 1.0352
    DELTTK=  1.000      PRNTOL=  1.000
*************************************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  3.61D-04     DELSMAX=  1.42D-03     RMMAX=  2.65D-02     REMAX=  9.16D 00
ITERATION NUMBER: 2     DELPMAX=  5.19D-06     DELSMAX=  6.35D-05     RMMAX=  1.07D-03     REMAX=  3.32D-01
ITERATION NUMBER: 3     DELPMAX=  3.49D-06     DELSMAX=  2.66D-05     RMMAX=  4.33D-04     REMAX=  1.28D-01
ITERATION NUMBER: 4     DELPMAX=  3.70D-07     DELSMAX=  1.10D-07     RMMAX=  1.15D-05     REMAX=  5.40D-03
ITERATION NUMBER: 5     DELPMAX=  2.18D-10     DELSMAX=  2.07D-08     RMMAX=  4.08D-07     REMAX=  1.34D-04
ITERATION NUMBER: 6     DELPMAX=  1.02D-09     DELSMAX=  7.84D-09     RMMAX=  1.28D-07     REMAX=  3.80D-05


 173.99999  173.99999  173.99998  173.99994  173.99984  173.99958  173.99898  173.99761  173.99459  173.98821  173.97529
 173.95021  173.90347  173.81980  173.67543  173.38104  172.61915  170.87886  168.05246  164.07030  159.12500


   1.00000    0.00000    1.00000    0.99990    0.99997    0.99993    0.99982    0.99958    0.99904    0.99[illegible]91    0.99502
   0.90117    0.98286    0.[illegible]6790    0.94223    0.89378    0.82196    0.76715    0.73500    0.70905    0.6[illegible]108

TIME=   25.500 ** MASS RATE= 4.16D-02  CUMMASS OUT= 1.02D 00  MBAL= 0.9851     EN RATE= 1.56D 01  CUMEN OUT= 3.69D 02  EBAL= 1.0370
    DELTTK=  1.000      PRNTOL=  1.000
*************************************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  3.58D-04     DELSMAX=  1.36D-03     RMMAX=  2.57D-02     REMAX=  8.94D 00
ITERATION NUMBER: 2     DELPMAX=  2.98D-06     DELSMAX=  1.86D-05     RMMAX=  3.40D-04     REMAX=  1.13D-01
ITERATION NUMBER: 3     DELPMAX=  1.06D-06     DELSMAX=  7.91D-06     RMMAX=  1.28D-04     REMAX=  3.79D-02
ITERATION NUMBER: 4     DELPMAX=  3.24D-07     DELSMAX=  4.77D-08     RMMAX=  9.65D-06     REMAX=  4.56D-03
ITERATION NUMBER: 5     DELPMAX=  4.03D-09     DELSMAX=  1.31D-08     RMMAX=  1.56D-07     REMAX=  3.47D-05


 173.99999  173.99999  173.99997  173.99991  173.99978  173.99946  173.99871  173.99703  173.99342  173.98596  173.97114
 173.94289  173.89109  173.79956  173.64304  173.30953  172.44475  170.57829  167.63868  163.56010  158.54167


   1.00000    1.00000    0.99999    0.99998    0.99906    0.99990    0.90977    0.99947    0.99883    0.99751    0.90458
   0.98980    0.98006    0.96435    0.93654    0.85308    0.81253    0.70282    0.73166    0.70676    0.68812

TIME=   26.500 ** MASS RATE= 4.09D-02  CUMMASS OUT= 1.08D 00  MBAL= 0.9840     EN RATE= 1.40 01  CUMEN OUT= 3.85D 02  EBAL= 1.0387
    DELTTK=  1.000      PRNTOL=  1.000
*************************************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  3.46D-04     DELSMAX=  1.26D-03     RMMAX=  2.43D-02     REMAX=  8.48D 00
ITERATION NUMBER: 2     DELPMAX=  5.15D-06     DELSMAX=  1.54D-05     RMMAX=  2.13D-04     REMAX=  5.02D-02
ITERATION NUMBER: 3     DELPMAX=  9.07D-07     DELSMAX=  6.14D-06     RMMAX=  9.89D-05     REMAX=  2.89D-02
ITERATION NUMBER: 4     DELPMAX=  1.22D-07     DELSMAX=  5.77D-08     RMMAX=  2.87D-06     REMAX=  1.39D-03
ITERATION NUMBER: 5     DELPMAX=  1.64D-10     DELSMAX=  4.91D-09     RMMAX=  1.00D-07     REMAX=  3.37D-05


 173.99999  173.99998  173.99995  173.99988  173.99971  173.99930  173.99838  173.99635  173.99208  173.98343  173.96657
 173.93495  173.87783  173.77839  173.60774  173.22966  172.25575  170.26904  167.21863  163.04512  157.95833

   1.00000    1.00000    0.99999    0.99098    0.[illegible]    0.99988    0.99971    0.9993    0.99860    0.99706    0.99[illegible]7
   0.98840    0.97830    0.96057    0.93039    0.[illegible]334    0.80398    0.75870    0.7286    0.70430    0.68553

TIME=   27.500 ** MASS RATE= 4.02D-02  CUMMASS OUT= 1.10D 00  MBAL= 0.9847     EN RATE= 1.52D 01  CUMEN OUT= 4.00D 02  EBAL= 1.0404
    DELTTK=  1.000      PRNTOL=  1.000
*************************************************************************************************************************************

ITERATION      : 1     DELPMAX=   30D 04     DELSMAX=  1.15D-03     RMMAX=  2.            REMAX=  7.89D 00
```

.51617

```
ITERATION NUMBER: 2     DELPMAX=  7.82D-06     DELSMAX=  3.4 0-05     RMMAX=  5.19D-04     REMAX=  1.40D-01
ITERATION NUMBER: 3     DELPMAX=  2.08D-06     DELSMAX=  1.3 0-05     RMMAX=  2.24D-04     REMAX=  6.52D-02
ITERATION NUMBER: 4     DELPMAX=  1.64D-07     DELSMAX=  4.4 D-08     RMMAX=  3.59D-08     REMAX=  1.87D-03
ITERATION NUMBER: 5     DELPMAX=  1.04D-09     DELSMAX=  8.9 D-10     RMMAX=  4.08D-0      REMAX=  1.77D-05          .996

 173.99998  173.99997  173.99994  173.99985  173.99963  173.99912  173.99799  173.99557  173.99057  173.98062  173.98157
 173.92639  173.86379  173.75651  173.55883  173.14028  172.05303  169.95181  166.79473  162.52895  157.37500

   1.00000    1.00000    0.99999    0.99997    0.99993    0.99984    0.99964    0.99921    0.99833    0.99657    0.99319
   0.98654    0.97580    0.95667    0.92370    0.86288    0.79632    0.75469    0.72596    0.70220    0.68340

TIME=   28.500 ** MASS RATE= 3.97D-02  CUMMASS OUT= 1.14D 00  MBAL= 0.9845 **  EN RATE= 1.510 01  CUMEN OUT= 4.15D 02  EBAL= 1.0421
   DELTTK=  1.000     PRNTDL=  1.000
**************************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  3.14D-04     DELSMAX=  1.17D-03     RMMAX=  2.14D-02     REMAX=  7.37D 00
ITERATION NUMBER: 2     DELPMAX=  8.25D-06     DELSMAX=  3.69D-05     RMMAX=  5.66D-04     REMAX=  1.53D-01
ITERATION NUMBER: 3     DELPMAX=  2.31D-06     DELSMAX=  1.51D-05     RMMAX=  2.42D-04     REMAX=  7.02D-02
ITERATION NUMBER: 4     DELPMAX=  1.63D-07     DELSMAX=  3.60D-08     RMMAX=  3.58D-06     REMAX=   .86D-03
ITERATION NUMBER: 5     DELPMAX=  1.09D-09     DELSMAX=  5.20D-10     RMMAX=  3.38D-08     REMAX=  1.56D-05
                                                                                     .51457
 173.99998  173.99996  173.99992  173.99980  173.99952  173.99890  173.99754  173.99467  173.98887  173.97752  173.95612
 173.91721  173.84897  173.73356  173.52615  173.04037  171.83769  169.62725  166.36893  162.01438  156.79167

   1.00000    0.99999    0.99999    0.99996    0.99992    0.99980    0.99956    0.99906    0.99803    0.9960     0.90222
   0.98531    0.97316    0.95258    0.91652    0.85250    0.78948    0.75078    0.72340    0.70037    0.6816

TIME=   29.500 ** MASS RATE= 3.94D-02  CUMMASS OUT= 1.1  00  MBAL= 0.9843 **  EN RATE= 1.500 01  CUMEN OUT= 4.30D 02  EBAL= 1.0438
   DELTTK=  1.000     PRNTDL=  1.000
**************************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  3.04D-04     DELSMAX=  1.19D-03     RMMAX=  2.18D-02     REMAX=   51D 00
ITERATION NUMBER: 2     DELPMAX=  6.65D-06     DELSMAX=  2.61D-05     RMMAX=  3.03D-04     REMAX=   03D-01
ITERATION NUMBER: 3     DELPMAX=  1.68D-06     DELSMAX=  1.07D-05     RMMAX=  1.71D-04     REMAX=   93D-02
ITERATION NUMBER: 4     DELPMAX=  1.46D-07     DELSMAX=  2.97D-08     RMMAX=  3.25D-06     REMAX=   68D-03
ITERATION NUMBER: 5     DELPMAX=  9.23D-10     DELSMAX=  7.69D-10     RMMAX=  3.49D-08     REMAX=   53D-05

 173.99997  173.99995  173.99989  173.99974  173.99940  173.99864  173.99701  173.99365  173.98697  173.97410  173.95022
 173.90741  173.83331  173.70931  173.47932  172.92911  171.61092  169.29598  165.94237  161.50291  156.20833
                                                                                     .51147
   0.99999    0.99999    0.99998    0.99995    0.99980    0.99976    0.99967    0.99888    0.99769    0.99541    0.99117
   0.98357    0.97037    0.94827    0.90886    0.84230    0.78338    0.74608    0.72090    0.69868    0.68031

TIME=   30.500 ** MASS RATE= 3.92D-02  CUMMASS OUT= 1.22D 00  MBAL= 0.9841 **  EN RATE= 1.50D 01  CUMEN OUT= 4.45D 02  EBAL= 1.0455
   DELTTK=  1.000     PRNTDL=  1.000
**************************************************************************************************************************

ITERATION NUMBER: 1     DELPMAX=  3.00D-04     DELSMAX=  1.21D-03     RMMAX=  2.22D-02     REMAX=  7.67D 00
ITERATION NUMBER: 2     DELPMAX=  3.73D-06     DELSMAX=  6.95D-06     RMMAX=  8.51D-05     REMAX=  1.38D-02
ITERATION NUMBER: 3     DELPMAX=  4.66D-07     DELSMAX=  2.78D-06     RMMAX=  4.46D-05     REMAX=  1.27D-02
ITERATION NUMBER: 4     DELPMAX=  7.75D-08     DELSMAX=  3.06D-08     RMMAX=  1.71D-06     REMAX=  8.89D-04
ITERATION NUMBER: 5     DELPMAX=  1.29D-10     DELSMAX=  4.33D-09     RMMAX=  8.20D-08     REMAX=  2.64D-05
```

```
173.99996  173.99993  173.99986  173.99967  173.99925  173.99833  173.99641  173.99251  173.98486  173.97037  173.94385
173.89696  173.81674  173.68359  173.42781  172.80593  171.37393  168.95860  165.51532  160.99467  155.62500


 0.99999    0.99999    0.99998    0.99994    0.99087    0.09970    0.99936    0.09867    0.90732    0.99405    0.09004
 0.98171    0.96742    0.94371    0.90077    0.83264    0.17791    0.74334    0.11845    0.60702    0.67909

TIME=   31.500 ** MASS RATE= 3.91D-02  CUMMASS OUT= 1.26D 00  MBAL= 0.9330 ** EN RATE= 1.500 01  CUMEN OUT= 4.630 02  EBAL= 1.047Z
   DELTTK= 1.000       PRNTOL=  1.000
*****************************************44*********  ****0*4a4*********@@4*@******0***444*******  ****0*4*  *0**04  ●●●4**4●●4444*4*44

ITERATION NUMBER: 1      DELPMAX=  1.390-03      DELSMAX=  3.220-03      RMMAX=  1.100-01      REMAX=  4.290 01
ITERATION NUMBER: 2      DELPMAX=  2.200-04      DELSMAX=  1.950-03      RMMAX=  2.960-02      REMAX=  8.430 00
ITERATION NUMBER: 3      DELPMAX=  2.140-04      DELSMAX=  1.830-03      RMMAX=  2.760-02      REMAX=  7.810 00
ITERATION NUMBER: 4      DELPMAX=  1.310-04      DELSMAX=  3.750-04      RMMAX=  3.320-03      REMAX=  9.920-01
ITERATION NUMBER: 5      DELPMAX=  5.220-05      DELSMAX=  4.430-04      RMMAX=  6.660-03      REMAX=  1.880 00
ITERATION NUMBER: 6      DELPMAX=  3.040-05      DELSMAX=  3.880-04      RMMAX=  6.410-03      REMAX=  1.930 00
ITERATION NUMBER: 7      DELPMAX=  3.920-05      DELSMAX=  3.350-04      RMMAX=  5.050-03      REMAX=  1.430 00
ITERATION NUMBER: 8      DELPMAX=  2.450-05      DELSMAX=  6.970-05      RMMAX=  5.790-04      REMAX=  1.730-01
ITERATION NUMBER: 9      DELPMAX=  1.050-05      DELSMAX=  8.930-05      RMMAX=  1.340-03      REMAX=  3.800-01
ITERATION NUMBER:10      DELPMAX=  5.920-06      DELSMAX=  7.780-05      RMMAX=  1.290-03      REMAX=  3.900-01
ITERATION NUMBER:11      DELPMAX=  7.800-06      DELSMAX=  6.680-05      RMMAX=  1.010-03      REMAX=  2.850-01
ITERATION NUMBER:12      DELPMAX=  4.890-06      DELSMAX=  1.390-05      RMMAX=  1.150-04      REMAX=  3.420-02
ITERATION NUMBER:13      DELPMAX=  2.110-06      DELSMAX=  1.790-05      RMMAX=  2.690-04      REMAX=  7.620-02
ITERATION NUMBER:14      DELPMAX=  1.180-06      DELSMAX=  1.560-05      RMMAX=  2.590-04      REMAX=  7.810-02
ITERATION NUMBER:15      DELPMAX=  1.560-06      DELSMAX=  1.340-05      RMMAX=  2.020-04      REMAX=  5.710-02
ITERATION NUMBER:16      DELPMAX=  9.780-07      DELSMAX=  2.780-06      RMMAX=  2.290-05      REMAX=  6.850-03
ITERATION NUMBER:17      DELPMAX=  4.220-07      DELSMAX=  3.580-06      RMMAX=  5.390-05      REMAX=  1.520-02
ITERATION NUMBER:18      DELPMAX=  2.370-07      DELSMAX=  3.120-06      RMMAX=  5.180-05      REMAX=  1.560-02
ITERATION NUMBER:19      DELPMAX=  3.130-07      DELSMAX=  2.680-06      RMMAX=  4.040-05      REMAX=  1.140-02
ITERATION NUMBER:20      DELPMAX=  1.960-07      DELSMAX=  5.570-07      RMMAX=  4.590-06      REMAX  [illegible].370-03

   4●@@●*@@4* TOO MANY ITERATIONS IN ITSOLVE *****


173.99992  173.99989  173.99977  173.99949  173.99887  173.99757  173.99494  173.98978  173.97996  173.96189  173.92968
173.87404  173.78082  173.62651  173.30791  172.52293  170.87225  168.26007  164.62138  159.83806  154.45833


 0.99999    0.99908    0.99996    0.99991    0.99930    0.99957    0.99910    0.99919    0.99645    0.99324    0.0875Z
 0.97763    0.96101    0.93367    0.88353    0.81531    0.76828    0.73646    0.71322    0.69183    0.67229

TIME=   33.500 ** MASS RATE= 3.750-02  CUMMASS OUT= 1.33D 00  MBAL= 0.0033 *0 EN RATE= 1.450 01  CUMEN OUT= 6.890 02  EBAL= 1 0503
   DELTTK= 2.000       PRNTOL=  2.000
***************************************************4**********4*  *40******  @4*****4@a440**a*********444**4**@****************  **  4*  4******4***4●●
```

(.50824)

$$\gamma_{5,i}^{n+1/2} \triangleq \frac{UP}{A}\left[\frac{T_i^{n+1} + T_i^{n}}{2} - T_\infty\right] \; ; \qquad \text{(E-9)}$$

where the T_i are saturation temperatures corresponding to the pressures p_i,

U = overall steady state heat transfer coefficient from the core to the environment, [BTU/hr ft2 °F)],

P = the perimeter of the system corresponding to U, [ft],
A = cross-sectional area to fluid flow, [ft2],

T_∞ = temperature of the environment, [°F],
ϕ = porosity of the porous medium, [ft^3/ft3],

c_{pr} = specific heat of the rock matrix, [BTU/lb_m °F],
ρ_r = density of the rock matrix, [lb_m/ft^3].

For the internal nodes, the equations are:

$$F_{2i-1} = (HR)\cdot\Big\{\gamma_{1,i-1}^{*n+1}\left[p_{i-1}^{n+1} - p_i^{n+1}\right] + \gamma_{1,i}^{*n+1}\left[p_{i+1}^{n+1} - p_i^{n+1}\right]$$

$$+\gamma_{1,i-1}^{*n}\left[p_{i-1}^{n} - p_i^{n}\right] + \gamma_{1,i}^{*n}\left[p_{i+1}^{n} - p_i^{n}\right]\Big\} + \gamma_{2,i}^{n} - \gamma_{2,i}^{n+1}$$

(E-10)

$$F_{2i} = (HR)\cdot\Big\{\gamma_{3,i-1}^{*n+1}\left[p_{i-1}^{n+1} - p_i^{n+1}\right] + \gamma_{3,i}^{*n+1}\left[p_{i+1}^{n+1} - p_i^{n+1}\right]$$

$$+\gamma_{3,i-1}^{*n}\left[p_{i-1}^{n} - p_i^{n}\right] + \gamma_{3,i}^{*n}\left[p_{i+1}^{n} - p_i^{n}\right]\Big\}$$

$$+\gamma_{4,i}^{n} - \gamma_{4,i}^{n+1} - (\Delta t)\cdot\gamma_{5,i}^{n+1/2} \; . \qquad \text{(E-11)}$$

At the mth node next to the right-hand boundary we have:

$$F_{2m-1} = (HR)\cdot\Big\{\gamma_{1,m-1}^{*n+1}\left[p_{m-1}^{n+1}-p_m^{n+1}\right] + \gamma_{1,m}^{*n+1}\left[PB^{n+1}-p_m^{n+1}\right]$$

$$+ \gamma_{1,m-1}^{*n}\left[p_{m-1}^{n}-p_m^{n}\right] + \gamma_{1,m}^{n}\left[PB^{n}-p_m^{n}\right]\Big\}$$

$$+ \gamma_{2,m}^{n} - \gamma_{2,m}^{n+1}\ , \qquad \text{(E-12)}$$

$$F_{2m} = (HR)\cdot\Big\{\gamma_{3,m-1}^{*n+1}\left[p_{m-1}^{n+1}-p_m^{n+1}\right]$$

$$- \gamma_{3,m}^{*n+1}\left[PB^{n+1}-p_m^{n+1}\right]$$

$$+ \gamma_{3,m-1}^{*n}\left[p_{m-1}^{n}-p_m^{n}\right] + \gamma_{3,m}^{*n}\left[PB^{n}-p_m^{n}\right]\Big\}$$

$$+ \gamma_{4,m}^{n} - {}_{4,m}^{n+1} - (\Delta t)\cdot\gamma_{5,m}^{n+1/2}\ , \qquad \text{(E-13)}$$

where PB^n is the specified pressure at the right-hand node at the nth (old) time level, and PB^{n+1} is at the (n+1)th (new) time level.

When the right-hand node has a constant specified pressure, then $PB^n = PB^{n+1}$. The program was initially written for this case, and when it was converted to the varying pressure case I forgot to distinguish between PB^n in line #112 and PB^{n+1} in line 8134. It *is* not clear how much effect this mistake will have on the calculations.

In deriving the functions f_{ij} of the matrix $[\underset{\approx}{\Phi}(\underset{\sim}{x})]$, it is helpful to remember that the unknown vector x consists of p and S values at the (n+1)th time level. Hence, derivatives of the portions of F(x) which depend only on the values of p^n and S^n are all zero.

The f_{ij} functions are presented below:

For $F_1(\underset{\sim}{x})$:

$$f_{11} = (HR) \cdot \left\{ \frac{\partial}{\partial p} (\gamma_{1,1}^{*}) \cdot \left[p_2 - p_1 \right] - 2\gamma_{1,1}^{*} \right\} - \frac{\partial}{\partial p} (\gamma_{2,1}) \ , \quad \text{(E-14)}$$

$$f_{12} = (HR) \cdot \left\{ \frac{\partial}{\partial S} (\gamma_{1,1}^{*}) \cdot \left[p_2 - p_1 \right] \right\} - \frac{\partial}{\partial S} (\gamma_{2,1}) \ , \quad \text{(E-15)}$$

$$f_{13} = (HR) \cdot \left\{ \frac{\partial}{\partial p} (\gamma_{1,1}^{*}) \ . \ \left[p_2 - p_1 \right] + 2\gamma_{1,1}^{*} \right\} , \quad \text{(E-16)}$$

$$f_{14} = (HR) \cdot \frac{\partial}{\partial S} (\gamma_{1,1}^{*}) \cdot \left[p_2 - p_1 \right] \ , \quad \text{(E-17)}$$

$$f_{1,j} = 0 \text{ for } j > 4 \ . \quad \text{(E-18)}$$

For $F_2(\underset{\sim}{x})$:

$$f_{21} = (HR) \cdot \left\{ \frac{\partial}{\partial p} (\gamma_{3,1}^{*}) \cdot \left[p_2 - p_1 \right] - 2\gamma_{3,1}^{*} \right\}$$

$$- \frac{\partial}{\partial p} (\gamma_{4,1}) - (\Delta t) \cdot \frac{\partial}{\partial p} (\gamma_{5,1}^{n+1/2}) \ , \quad \text{(E-19)}$$

$$f_{22} = (HR) \cdot \left\{ \frac{\partial}{\partial S} (\gamma_{3,1}^{*}) \cdot \left[p_2 - p_1 \right] \right\} - \frac{\partial}{\partial S} (\gamma_{4,1}) \ , \quad \text{(E-20)}$$

$$f_{23} = (HR) \cdot \left\{ \frac{\partial}{\partial p} (\gamma_{3,1}^{*}) \cdot \left[p_2 - p_1 \right] + 2\gamma_{3,1}^{*} \right\} \quad \text{(E-21)}$$

$$f_{24} = (HR) \cdot \left\{ \frac{\partial}{\partial S} (\gamma_{3,1}^{*}) \cdot \left[p_2 - p_1 \right] \right\} , \qquad \text{(E-22)}$$

$$f_{2j} = 0 \text{ for } j > 4 . \qquad \text{(E-23)}$$

For $F_{2i-1}(\underset{\sim}{x})$, $i = 2, 3, \ldots, m$:

$$f_{2i-1,j} = 0 \text{ for } j \leq 21-4 \text{ and } j \geq 2i+3 .$$

$$f_{2i-1,2i-3} = (HR) \cdot \left\{ \frac{1}{2} \cdot \frac{\partial}{\partial p} (\gamma_{1,i-1}^{*}) \cdot \left[p_{i-1} - p_i \right] + \gamma_{1,i-1}^{*} \right\} , \qquad \text{(E-24)}$$

$$f_{2i-1,2i-2} = \frac{1}{2} \cdot \left\{ (HR) \cdot \frac{\partial}{\partial S} (\gamma_{1,i-1}^{*}) \cdot \left[p_{i-1} - p_i \right] \right\} , \qquad \text{(E-25)}$$

$$\begin{aligned} f_{2i-1,2i-1} = (HR) \cdot \Bigg\{ & \frac{1}{2} \cdot \frac{\partial}{\partial p} (\gamma_{1,i}^{*}) \cdot \left[p_{i+1} - p_i \right] - \gamma_{1,i}^{*} \\ & + \frac{1}{2} \cdot \frac{\partial}{\partial p} (\gamma_{1,i-1}^{*}) \cdot \left[p_{i-1} - p_i \right] - \gamma_{1,i-1}^{*} \Bigg\} \\ & - \frac{\partial}{\partial p} (\gamma_{2,i}) , \qquad \text{(E-26)} \end{aligned}$$

$$\begin{aligned} f_{2i-1,2i} = \frac{1}{2} \cdot (HR) \cdot \Bigg\{ & \frac{\partial}{\partial S} (\gamma_{1,i}^{*}) - \left[p_{i+1} - p_i \right] \\ & + \frac{\partial}{\partial S} (\gamma_{1,i-1}^{*}) \cdot \left[p_{i-1} - p_i \right] \Bigg\} - \frac{\partial}{\partial S} (\gamma_{2,i}) , \qquad \text{(E-27)} \end{aligned}$$

$$f_{2i-1,2i+1} + (HR) \cdot \left\{ \frac{1}{2} \cdot \frac{\partial}{\partial p} (\gamma_{1,i}{}^{*}) \cdot \left[p_{i+1} - p_i \right] + \gamma_{1,i}{}^{*} \right\} , \qquad \text{(E-28)}$$

$$f_{2i-1,2i+2} = \frac{1}{2} \cdot (HR) \cdot \frac{\partial}{\partial S} (\gamma_{1,i}{}^{*}) . \qquad \text{(E-29)}$$

For $F_{2i}(\underline{x})$, $i = 2, 3, \ldots, m$:

$$f_{2i,j} = 0 \text{ for } j \leq 2i-4 \text{ and } j \geq 2i+3 , \qquad \text{(E-30)}$$

$$f_{2i,2i-3} = (HR) \cdot \left\{ \frac{1}{2} \cdot \frac{\partial}{\partial p} (\gamma_{3,i-1}{}^{*}) \cdot \left[p_{i-1} - p_i \right] + \gamma_{3,i-1}{}^{*} \right\} , \qquad \text{(E-31)}$$

$$f_{2i,2i-2} = \frac{1}{2} \cdot (HR) \cdot \frac{\partial}{\partial S} (\gamma_{3,i-1}{}^{*}) \cdot \left[p_{i-1} - p_i \right] , \qquad \text{(E-32)}$$

$$\begin{aligned} f_{2i,2i-1} = (HR) \cdot \Big\{ & \frac{1}{2} \cdot \frac{\partial}{\partial p} (\gamma_{3,i}{}^{*}) \cdot \left[p_{i+1} - p_i \right] - \gamma_{3,i}{}^{*} \\ & - \frac{1}{2} \cdot \frac{\partial}{\partial p} (\gamma_{3,i-1}{}^{*}) \cdot \left[p_{i-1} - p_i \right] - \gamma_{3,i}{}^{*} \Big\} \\ & - \frac{\partial}{\partial p} (\gamma_{4,i}) - (\Delta t) \cdot \frac{\partial}{\partial p} (\gamma_{5,i}{}^{n+1/2}) , \end{aligned} \qquad \text{(E-33)}$$

$$\begin{aligned} f_{2i,2i} = \frac{1}{2} \cdot (HR) \cdot \Big\{ & \frac{\partial}{\partial S} (\gamma_{3,i}{}^{*}) \cdot \left[p_{i+1} - p_i \right] \\ & + \frac{\partial}{\partial S} (\gamma_{3,i-1}{}^{*}) \cdot \left[p_{i-1} - p_i \right] \Big\} - \frac{\partial}{\partial S} (\gamma_{4,i}) , \end{aligned} \qquad \text{(E-34)}$$

$$f_{2i,2i+1} = (HR) \cdot \left\{ \frac{1}{2} \cdot \frac{\partial}{\partial p} (\gamma_{3,i}^{*}) \cdot \left[p_{i+1} - p_i \right] + \gamma_{3,i}^{*} \right\} , \qquad \text{(E-35)}$$

$$f_{2i,2i+2} = \frac{1}{2} \cdot (HR) \cdot \frac{\partial}{\partial S} (\gamma_{3,i}^{*}) \cdot \left[p_{i+1} - p_i \right] . \qquad \text{(E-36)}$$

Note that all evaluations of the functions f_{ij} occur at values of p and S at the (n+1)th tine level, and hence for notational brevity the (n+1) superscripts have been dropped in presenting the f_{ij}'s.

APPENDIX F

GAUSSIAN ELIMINATION ALGORITHM FOR THE SOLUTION OF A BITRIDIAGONAL SYSTEM OF LINEAR ALGEBRAIC EQUATIONS

(from D. A. von Rosenburg, *Methods for the Numerical Solution of Partial Differential Equations*, Elsevier, 1969, Appendix C)

The equations are of the form:

$$a_{i,1} u_{i-1} + a_{i,2} v_{i-1} + b_{i,1} u_i + b_{i,2} v_i$$

$$+ c_{i,1} u_{i+1} + c_{i,2} v_{i+1} = d_{i,1} ,$$

$$a_{i,3} u_{i-1} + a_{i,4} v_{i-1} + b_{i,3} u_i + b_{i,4} v_i$$

$$+ c_{i,3} u_{i+1} + c_{i,4} v_{i+1} = d_{i,2} ,$$

for $i = 1, \ldots, m$;

where in the application of interest:

$$a_{1,j} = c_{m,j} = 0, \text{ for } j = 1, \ldots, 4 .$$

This system is written in matrix form in Fig. F-1.

$$
\left[\begin{array}{cccccccccc}
b_{11} & b_{12} & C_{11} & C_{12} & & & & & & \\
b_{13} & b_{14} & C_{13} & C_{14} & & & & & & \\
a_{21} & a_{22} & b_{21} & b_{22} & C_{21} & C_{22} & & & & \\
& \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & & & \\
& & a_{i,1} & a_{i,2} & b_{i,1} & b_{i,2} & C_{i,1} & C_{i,2} & & \\
& & a_{i,3} & a_{i,4} & b_{i,3} & b_{i,4} & C_{i,3} & C_{i,4} & & \\
& & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \\
& & & & & & a_{m,1} & a_{m,2} & b_{m,1} & b_{m,2} \\
& & & & & & a_{m,3} & a_{m,4} & b_{m,3} & b_{m,4}
\end{array}\right]
\left[\begin{array}{c}
u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ u_i \\ v_i \\ \vdots \\ u_m \\ v_m
\end{array}\right]
=
\left[\begin{array}{c}
d_{11} \\ d_{12} \\ d_{21} \\ d_{22} \\ \vdots \\ d_{i,1} \\ d_{i,2} \\ \vdots \\ d_{m,1} \\ d_{m,2}
\end{array}\right]
$$

FIGURE F-1. BITRIDIAGONAL SYSTEM OF LINEAR ALGEBRAIC EQUATIONS

The algorithm is:

Compute :

$$\beta_{i,1} = b_{i,1} - a_{i,1} \lambda_{i-1,1} - a_{i,2} \lambda_{i-1,3} ,$$

$$\beta_{i,2} = b_{i,2} - a_{i,1} \lambda_{i-1,2} - a_{i,2} \lambda_{i-1,4} ,$$

$$\beta_{i,3} = b_{i,3} - a_{i,3} \lambda_{i-1,1} - a_{i,4} \lambda_{i-1,3} ,$$

$$\beta_{i,4} = b_{i,4} - a_{i,3} \lambda_{i-1,2} - a_{i,4} \lambda_{i-1,4} ,$$

with $\beta_{1,j} = b_{1,j}$ for $j = 1, 2, 3, 4$;

and :

$$\delta_{i,1} = d_{i,1} - a_{i,1} \gamma_{i-1,1} - a_{i,2} \gamma_{i-1,2} ,$$

$$\delta_{i,2} = d_{i,2} - a_{i,3} \gamma_{i-1,1} - a_{i,4} \gamma_{i-1,2} ,$$

with $\delta_{1,1} = d_{1,1}$ and $\delta_{1,2} = d_{1,2}$;

and :

$$\mu_i = \beta_{i,1} \beta_{i,4} - \beta_{i,2} \beta_{i,3} .$$

The $\beta_{i,j}$, $\delta_{i,j}$, and μ_i are evaluated to aid the computation of what follows. They do not need to be stored after computation of:

$$\lambda_{i,1} = (\beta_{i,4}\, c_{i,1} - \beta_{i,2}\, c_{i,3})/\mu_i \ ,$$

$$\lambda_{i,2} = (\beta_{i,4}\, c_{i,2} - \beta_{i,2}\, c_{i,4})/\mu_i \ ,$$

$$\lambda_{i,3} = (\beta_{i,1}\, c_{i,3} - \beta_{i,3}\, c_{i,1})/\mu_i \ ,$$

$$\lambda_{i,4} = (\beta_{i,1}\, c_{i,4} - \beta_{i,3}\, c_{i,2})/\mu_i \ ,$$

and:

$$\gamma_{i,1} = (\beta_{i,4}\, \delta_{i,1} - \beta_{i,2}\, \delta_{i,2})/\mu_i \ ,$$

$$\gamma_{i,2} = (\beta_{i,1}\, \delta_{i,2} - \beta_{i,3}\, \delta_{i,1})/\mu_i \ .$$

Values of $\lambda_{i,j}$ and $\gamma_{i,j}$, $j = 1, 2, 3, 4$, must be stored, since they are used in the back substitution:

$$u_m = \gamma_{m,1} \ ,$$

$$v_m = \gamma_{m,2} \ ,$$

and:

$$u_i = \gamma_{i,1} - \lambda_{i,1}\, u_{i+1} - \lambda_{i,2}\, v_{i+1} \ ,$$

$$v_i = \gamma_{i,2} - \lambda_{i,3} u_{i+1} - \lambda_{i,4} v_{i+1} ,$$

for $i = m-1, m-2, \ldots, 1$.

This algorithm is contained in subroutine **SOLVBT** (lines #453/512).

APPENDIX G

DESCRIPTION OF INPUT PARAMETERS AND THEIR FORMATTING REQUIREMENTS

Information is given in the order: Format Requirement, Variable Name, and Description.

FORMAT	VARIABLE NAME	DESCRIPTION
CARD 1 - Physical Data for Core		
F5.2	POR	Porosity, ft3 pore volume/ft3 bulk
F5.2	KAPPA	Axial Thermal Conductivity, BTU/(hr ft °F)
F10.3	U	Steady state heat loss coefficient to environment, BTU/(hr ft2 °F)
F10.3	PA	Ratio of perimeter over which heat losses are occurring to cross sectional area to flow, ft^{-1}
F10.3	CPROCK	Specific heat of the rock matrix, BTU/(lb_m °F)
F10.3	RHOROK	Density of the rock matrix, ft3/lb_m.

CARD 2 - Relative Permeability Information

The relative permeability relationships are given by the Corey equations as defined by Eqs. A-7, 8, and 9 in Appendix A.

Two values of the absolute permeability, K_{abs}, and the residual gas and liquid saturations, S_{rg} and S_{rL} respectively, are read in. These values correspond to the two temperatures, T_{f1} and T_{f2}. Linear interpolation between the two values of K_{abs}, S_{rg}, and S_{rL} is used for temperatures different from T_{f1} and T_{f2}. Thus we have the input parameters:

F5.1	**TF(1)**	Temperature levels for interpolation,
F5.1	**TF(2)**	°F;

FORMAT	VARIABLE NAME	DESCRIPTION
F10.4	KABSF(1)	Absolute permeability at the two
F10.4	KABSF(2)	temperature levels, ^{o}F
F5.2	SWRF(1)	Residual liquid saturations at the
F5.2	SWRF(2)	two temperature levels;
F5.2	SORF(1)	Residual gas saturations at the two
F5.2	SORF(2)	temperature levels.

CARD 3 - Parameters for the Numerical Solution

FORMAT	VARIABLE NAME	DESCRIPTION
F8.3	DELTTK	Initial time step size, sec;
F7.3	DELTXH	Uniform mesh size, ft;
D10.2	DELTA	Convergence criterion applied to the mass and energy equation residuals at each node;
F5.2	**WF**	Weighting factor used at each node for linear estimation of the zeroth iteration level vector at a new time step, based on the results of the last two time steps;
I4	NODES	Total number of mesh nodes used, including the endpoints. Must be odd and less than or equal to 21;
14	**MAXNUM**	Maximum number of iterations allowed in the Newton-Raphson solution of the nonlinear system of discretized equations;
F5.2	BALDEL	Maximum allowed error in both the mass (CHECKM) and energy (CHECKE) balance check. Execution stops if $\lvert 1-x \rvert \geq \text{BALDEL}$, where x = CHECKM or CHECKE:
F8.4	PRNTDL	Parameter for controlling output, whereby the solution is printed every nth time step, where n = (PRNTDL)/(DELTTK);

FORMAT	VARIABLE NAME	DESCRIPTION
F8.4	HP	Parameters used in Subprogram FUNCTION DIFF (lines #553/564) to evaluate derivatives of γ_1 or γ_3 using the centered difference approximation to the first derivative: $\left.\frac{df}{dx}\right\|_{x_0} = \frac{f(x_0+h) - f(x_0-h)}{h}$, where h = HP for derivatives with respect to P, and h = HS for derivatives with respect to S.
F8.4	HS	

CARD 4 - Various System and Run-Time Parameters

FORMAT	VARIABLE NAME	DESCRIPTION
F10.2	PINIT	Initial pressure and saturation of system. These values appear in the main heading, but are not used anywhere in the current version of the program.
F10.2	SINIT	
F10.2	TEXTER	Temperature of the environment surrounding the core system, °F.
F10.2	PBOUND	Current value of the specified pressure at $x = L$. The input value appears in the main heading, but is not used anywhere in the program. The variable itself is used in the program, but its value changes.
F10.2	TIME	The time corresponding to the initial conditions on card 6/13 below, sec;
F10.2	TMAX	Maximum running time, sec.

CARD 5 - Step Size Control Parameter

FORMAT	VARIABLE NAME	DESCRIPTION
	PDOWN	Dummy input. Appears in the main heading, but is not used in the current version of the program;

FORMAT	VARIABLE NAME	DESCRIPTION
	DTMSCR	Criterion used to increase the time step size. For $\text{DTMS} = \underset{\text{all nodes}}{\text{max over}} \lvert \text{SNUK1(i)} - \text{SOLD(i)} \rvert$ The time step size is doubled if DIMS $\leq$ DTMSCR.

CARDS 6/9 - Initial Pressure Distribution in the System

5F10.5 on each card, POLD(1), POLD(2), ..., POLD(20), in psia;

CARDS 10/13 - Initial Volumetric Liquid Saturation in the System

5F10.5 on each card, SOLD(1), SOLD(2), ..., SOLD(20), dimensionless;

CARDS 14/17 - Estimate of the Pressure Solution at the Second Time Level

5F10.5 on each card, PNUK(1), PNUK(2), ..., PNUK)20), in psia;

CARDS 18/21 - Estimate of the Saturation Solution at the Second Time Level

5F10.5 on each card, SNUK(1), SNUK(2), ..., SNUK(20), dimensionless.

Note that even though the 21 mesh node capacity might not be used, dummy pressure and saturation information must be supplied at the nodes not being used such that the above format specifications are satisfied.