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

ACKNOWLEDGEMENT .......................................................... ii

INTRODUCTION .......................................................... 1

NOMENCLATURE AND NOTATION ........................................ 3

IMPLICIT FINITE DIFFERENCE PROCEDURE .................................. 4

SOLUTION OF THE DISCRETIZED NONLINEAR EQUATIONS USING NEWTON-RAPHSON ITERATION ................................................. 6

VERIFICATION OF THE NUMERICAL SOLUTION USING A MATERIAL AND ENERGY BALANCE CHECK .............................................. 9

PROGRAM PARAMETERS

INPUT .......................................................... 11

OUTPUT .......................................................... 11

SAMPLE VALUES OF PARAMETERS ........................................ 12

FUNCTIONAL EVALUATION .................................................. 13

CONCLUDING COMMENTS ................................................... 14

REFERENCES .......................................................... 17

APPENDIX A: THE GOVERNING FLOW EQUATIONS FOR LINEAR TWO-PHASE BOILING FLOW IN A HORIZONTAL POROUS MEDIUM .................................................. 19

APPENDIX B: SOURCE LISTING OF THE COMPUTER PROGRAM ......................... 23

APPENDIX C: DESCRIPTION OF PROGRAM SUBROUTINES AND LOGICAL STRUCTURE .................................................. 43

APPENDIX D: SAMPLE INPUT AND OUTPUT ................................... 49

APPENDIX E: THE SYSTEM OF NONLINEAR DISCRETIZED EQUATIONS

\[ \mathbf{F}(\mathbf{x}) = 0 \] AND THE CORRESPONDING JACOBIAN MATRIX

\[ \mathbf{J}(\mathbf{x}) \] .................................................. 63

-iii-
LIST OF FIGURES

FIGURE

1  BITRIDIAGONAL STRUCTURE OF THE LINEAR ALGEBRAIC
    SYSTEM OF EQUATIONS RESULTING IN THE NEWTON–RAPHSON
    ITERATION ALGORITHM.  ....................................  8

C-1  FLOW DIAGRAM OF THE COMPUTER PROGRAM ........  44

F-1  BITRIDIAGONAL SYSTEM OF LINEAR ALGEBRAIC EQUATIONS ...  72
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:

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

and
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 \( \gamma_j \), and \( q_{\text{loss}} \) 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:

\[
\text{at } x = 0, \quad \frac{\partial p}{\partial x} = 0, \quad (3)
\]

and

\[
\text{at } x = L, \quad p(t) = f(t), \quad (4)
\]

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 flow rate 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., \( \text{DELT}X \). Array variables are similarly written in capital letters followed by the array dimensions in parentheses, e.g., \( \text{POLD}(20) \).

2) Subscripts on pressure, \( p \), and volumetric liquid saturation, \( S \), 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 \( i \)th 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 $i$th node at the $n$th (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:

$$\Delta \left[ \gamma_1^n \Delta p_i^n \right] + \Delta \left[ \gamma_1^{n+1} \Delta p_i^{n+1} \right] = \delta y_{2,1} u$$

(5)

$$\Delta \left[ \gamma_3^n \Delta p_i^n \right] + \Delta \left[ \gamma_3^{n+1} \Delta p_i^{n+1} \right] = \delta y_{4,1} n + \ldots \ldots$$

(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, $\Delta u_i = \frac{u_i+1/2 - u_i-1/2}{(\Delta x)}$.

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

$$\Delta \left[ a\Delta u_i \right] \Delta a_i+1/2 \left[ u_i+1 - u_i \right] - a_i-1/2 \left[ u_i-1 - u_i \right]$$

and $a$ is the forward difference operator in time, $\delta u_i^n = \frac{u_i^{n+1} - u_i^n}{(\Delta t)}$. 
Note that $\Delta t$ is the time step size, and $\Delta x$ is the uniform grid spacing. Midstream weighting of pressure and saturation was used to evaluate the $a_{i+1/2}$ 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 $i$th 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 et al., 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 et al., 1974, Coats et al., 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 $m$th node, $p_{m+1}^n$ becomes $p_{B}^n$, and $p_{m+1}^{n+1}$ becomes $p_{B}^{n+1}$, where the $p_{B}$ 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:

\[
F(x) = 0 ,
\]

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

The nonlinear mass balance, \(F_{21-1}\), 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 DISCRETE NONLINEAR EQUATIONS USING NEWTON-RAPHSON ITERATION**

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

\[
\left[ \frac{\partial F(x^{(v)})}{\partial x} \right] \xi^{(v+1)} = -F(x^{(v)}) ,
\]

\((8)\)
where: \( [\Phi] \) is the Jacobian matrix \([f_{ij}]\), \( f_{ij} \triangleq \frac{\partial F}{\partial x_j} \), the superscript in parentheses indicates the iteration number; and \( \xi \) is the correction vector after each iteration.

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

For \( \xi^{(0)} \) known, the correction equation (9) is linear and algebraic. Hence it can be solved directly using Gaussian elimination. The bitridiagonal structure of \( \Phi \) 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 \( \text{MAXNUM} \), 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 et al., 1969). For the discretized equations \( F(\tilde{x}) = 0 \), it was usually adequate to use the pressures and saturations \( \tilde{\tilde{x}} \) 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
FIGURE 1 BITRIDIAGONAL STRUCTURE OF THE LINEAR ALGEBRAIC SYSTEM OF EQUATIONS RESULTING IN NEWTON-RAPHSON ITERATION ALGORITHM.
most recent time levels. Thus:

\[ x_i^{n+1} = x_i^n + (WF) \frac{\Delta t_{new}}{\Delta t_{old}} \left( x_i^n - x_i^{n-1} \right), \]

where \((\Delta t_{old})\) is the time step size between \(x_i^{n-1}\) and \(x_i^n\), \((\Delta t_{new})\) 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:

\[
\text{Total Mass or Energy Efflux} + \text{Total Remaining Mass or Energy} = \text{Initial Mass or Energy in the System}
\]

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 10 (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 (DETXH) and the total number of nodes (NODES), which is equal to (MTL) in the notation of the previous sections.

**OUTPUT**

The complete solution is printed at every $n$ time step, where $n$ is the ratio of the input parameter PRNTDL to the initial time step size DELTK. 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, RMAX and REM, 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 $p_i^n$, $s_i^n$, $n+1(v)$, and $s_i^{n+1}(v)$ 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.,

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

3) there are too many Newton–Raphson 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 DTMSCE = 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 MAXMI = 19 was commonly used. Normally only
four or five iterations were required to reach this convergence cri-
terion. 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 $\gamma_{n+1}$ and $\gamma_{n+1}$ terms explicitly at the values of the $n$th 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 $\gamma_{n+1}$ and $\gamma_{n+1}$ 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.

CONCLUDING 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.


Aziz, K., Course Notes from Seminar on Reservoir Simulation, Course No. 273, Department of Petroleum Engineering, Stanford University, Spring 1971.


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


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.


APPENDIX A

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

The differential equations are:

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

\[ \frac{\partial}{\partial x} \left[ \gamma_3(p, S_L) \right] \cdot \frac{\partial p}{\partial x} = \frac{\partial \gamma_4}{\partial t} (p, S_L) + q_{\text{loss}}. \quad (A-2) \]

The functions, \( \gamma_j \), are defined:

\[ \gamma_1 \triangleq K_L(S_L, p) \cdot \alpha_1(p) + K_g(S_L, p) \cdot \alpha_2(p), \quad (A-3) \]

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

\[ \gamma_3 \triangleq K_L(S_L, p) \cdot \beta_1(p) + K_g(S_L, p) \cdot \beta_2(p) + \kappa \cdot \beta_5(p), \quad (A-5) \]

\[ \gamma_4 \triangleq \left[ 1 - \phi \right] \cdot \left[ \frac{c_{\text{pr}} p_r}{T_0} \right] \cdot \left[ T - T_0 \right] + \phi \cdot \left[ \beta_3(p) + S_L \cdot \beta_4(p) \right]. \quad (A-6) \]

The dependent variables are:

\[ p = \text{pressure, psia}; \]
\[ S_L = \text{volumetric liquid saturation, volume of liquid in pore space per total volume of pore space} \]

The independent variables are:

\[ x = \text{distance, ft}; \]
\[ t = \text{time, sec}. \]
The specified functional relations for the two-phase Darcy rate equations are the Corey equations (Corey et al., "Three Phase Relative Permeability," Trans. AIME 207 (1956), p. 349):

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

\[ K_L = K_{abs}(T) \cdot (S_L^*)^4; \quad (A-8) \]

where:

\[ S_L^* {\triangleq} (S_L - S_{LR})/(1 - S_{gcr} - S_{LR}), \quad (A-9) \]

- \( S_{LR} \) = residual liquid saturation, a linear function of temperature, liquid volume in pore space per total volume of pore space;
- \( S_{gcr} \) = 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, \( K(S, T) \) and \( K_g(S, T) \), is also a dependence on pressure: \( K(S, p) \) and \( K_g(S, p) \).

The physical properties of water are incorporated into the functions \( a_j \) and \( c_j \). These are single valued functions of pressure, and are defined:

\[ \alpha_1 \triangleq \frac{\rho_g}{\mu_g}, \quad \frac{1b}{ft^3 \cdot cp}, \quad (A-10) \]

\[ \alpha_2 \triangleq \frac{\rho_g}{\mu_g}, \quad \frac{1b}{ft^3 \cdot en}, \quad (A-11) \]
\[
\alpha_4 \triangleq \rho_L - \rho_g \frac{1 \text{ lb}}{\text{m}^3} \frac{\text{m}}{\text{ft}^3}, \\
\beta_1 \triangleq \frac{\rho_L h_L}{\mu_L} \frac{\text{Btu}}{\text{ft}^3 \text{cp}}, \\
\beta_2 \triangleq \frac{\rho_g h_g}{\mu_g} \frac{\text{Btu}}{\text{ft}^3 \text{cp}}, \\
\beta_3 \triangleq \rho_g h_g \frac{\text{Btu}}{\text{ft}^3}, \\
\beta_4 \triangleq \rho_L h_L - \rho_g h_g \frac{\text{Btu}}{\text{ft}^3}, \\
\beta_5 \triangleq \frac{T_{abs} v_{fg} \rho}{h_{fg}} \frac{\text{Btu}}{\text{ft}^3}.
\]

where: 
\( \rho \) = density, \( \text{lb}_m/\text{ft}^3 \);
\( \mu \) = viscosity, c.p.;
\( h \) = specific enthalpy, \( \text{BTU}/\text{lb}_m \text{ ofF} \);
\( T_{abs} \) = absolute temperature, \( \text{ofF} \);
\( v \) = specific volume, \( \text{ft}^3/\text{lb} \);
\( T \) = temperature, \( \text{ofF} \); and \( \text{m} \)

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:

\[ \phi = \text{fractional porosity}, \quad \frac{\text{volume pore space}}{\text{bulk volume of medium}}, \]

\[ \kappa = \text{effective thermal conductivity}, \quad \frac{\text{Btu}}{\text{hr ft }^{\circ}F}, \]

\[ (C_p \rho F) = \text{specific heat content of matrix rock on a volume basis}, \quad \frac{\text{Btu}}{\text{ft}^3 \cdot {^\circ}F}. \]

The function accounting for heat losses to the environment is:

\[ q_{\text{loss}}(x, t) = \text{local heat loss rate from sides of the core per unit length of core and exposed surface area of core} \]

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

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

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

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 \( (\phi \text{ ft}^2 / \text{sec psia}) \) by multiplying by \( 1/(1.3656 \times 104) \);

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

SOURCE LISTING OF THE COMPUTER PROGRAM
1. IMPLICIT REAL*8(A-H,O-Y)

2. DECLARE CPPCA BLOCKS

3. C CCMCK/FS/ FCLD, SCLD, PNUK, SNUK, PNUK1, SNUK1
4. C CCMCH/BTRSL/ CELP, DELS
5. C CCMCH/BTRIN/ A, B0, C0, D
6. C CCMCH/SD/TS, KASIF, SHRF, SURF
7. C CCMCH/EC/FECLN, PINIT, SINIT, TMAX, PNTL, SNEWBN, SCLDBK,
8. C Z PBSTCR, PCDX, BALMAX, BALMIN, NQUES, NLESS1, NLESS2, NPLUSL, MAXNUM
9. C CCMCH/CCAS1/ FCT, DELTK, HK, KAPPA, WL, DELTA, DELT, ELT, U, PA, TEXTER
10. C Z, RHCRCK, CFCC, FP, HS, OTMS, OTMSCK, MFILE
11. C CCMCH/REAL/SAVEP, CUMP, SAVEE, CUME, CHECKM, CHECKE, STARTM, STARTE, QM, QE

11.1 C DECLARE VARIABLES AND VARIABLE ARRAYS

12. C REAL*8 TF(2), KASIF(2), SHRF(2), SURF, KAPPA
13. C DIMENSION FCLD(20), SCLD(20), PNUK(20), SNUK(20), PNUK1(20),
14. C Z SNUK1(20), CELP(20), DELS(20)
15. C DIMENSION F(20,4), I(20,4), C(20,4), D(20,2)
16. C
17. C INPUT THE DATA AND WRITE OUT ALL HEADINGS
18. C TIME=0.
18.1 C ENTER BLOCKS A AND E CA FLUX DIAGRAM, FIG.1, APPENDIX C
18.2 C SEE MARKER NO. 2, APPENDIX C
18.3 C CALL IC(1, TIME)
19. C
19.1 C INITIALIZE SYSTEM
20. C ENTER BLOCK C CA FLUX DIAGRAM, FIG.1, APPENDIX C
20.1 C
21. C TPRINT=FRACL + TIME
22. C CC 5 K=1, NCCFS
23. C FNUK(1)=PCLC(K)
24. C 5 SNUK1(K)=SCLD(K)
25. C CC 6 K=1, 4
26. C A1(K) = 0.
27. C 6 C(NCCFS,K) = C.
27.1 C CONVERT KASIF, KAPPA, & U TO UNITS CONSISTENT WITH REST OF SYSTEM
28. C
29. C KASIF(1)=KASIF(1)/(1.36504)
30. C KASIF(2)=KASIF(2)/(1.36504)
31. C KAPPA=KAPPA/(1.54545)
32. C L=UL/36CC0CC
32.1 C
33. C INITIALIZE MASS/ENERGY BALANCE CHECKS
33.1 C
34. C CLMP=0.
35. C CLME=0.
36. C CALL MPEAL(2)
36.1 C WRITE CLT CLTFL1 FCF T=0
37. C CALL IC(2, TIME)
38. C CALL MELSEF(TIME)
39. C
C SUBROUTINE ALSTEP(TIME) GETS READY FOR THE NEW TIME STEP
C IT CHECKS UMS AGAINST UMCG TO SEE IF IT IS TIME TO
C EXECUTE CELT.ps THEN IT CALCULATES THE R.C. FOR THE NEW LEVEL.
C
C SEE MARKER NO. 4, APPENDIX C
C
C *** ENTER THE TIME STEP LOOP *****
C
C ENTER SCROLLING ITERATIONS
C
C ENTER BLOCK C (CP FLOW DIAGRAM, FIG. 1, APPENDIX C
C SEE MARKER NO. 1, APPENDIX C
C
10 CALL ITSOLV(TIME, E150, E300).
C
C EVALUATE MAX CHANGE IN SATURATION OVER TIME STEP.
C
C UMS = CAUS(SNU1(K) - SUL1(K))
C CC 12 P = 2, ACCES
C 12 UMS = CMAS(DIMS, CAUS(SNU1(K) - SUL1(K)))
C
C PASS/ENERGY BALANCE CHECK
C
C ENTER BLOCK F (CN FLOW DIAGRAM, FIG. 1, APPENDIX C
C SEE MARKER NO. 3, APPENDIX C
C
C CALL MEFAL(1)
C
C CLTFLT THE RESULTS IF IT IS TIME TO
C OTHERWISE PRINT RESULTS AND UPDATE COUNTERS
C ENTER BLOCK F (CN FLOW DIAGRAM, FIG. 1, APPENDIX C
C
C IF (TIME.LT.TPRINT) GC TO 20
C
C ENTER BLOCK G (CP FLOW DIAGRAM, FIG. 1, APPENDIX C
C
C CALL IC(2, TIME)
C PRINT = TPRINT + PRATCL
C
20 CONTINUE
C
C PREPARE TO ENTER NEW TIME STEP
C ENTER BLOCK H (CN FLOW DIAGRAM, FIG. 1, APPENDIX C
C SEE MARKER NO. 4, APPENDIX C
C
C CALL ALSTEP(TIME)
C
C ESTIMATE NEW SOLUTION
C
DO 30 P = 1, ACCES
PNU1(K) = PNU1(K) + (PNU1(K) - PULD(K)) * NFW
SNL1(K) = SNL1(K) + (SNU1(K) - SUL1(K)) * NFW
PCL1(K) = PNU1(K)
30 SCL1(K) = SNL1(K)
C
C ENTER BLOCK I (CN FLOW DIAGRAM, FIG. 1, APPENDIX C
C
C IS THE PASS OF ENERGY BALANCE TOO LARGE?
C
C IF (CHECKK.LT.FALMAK) OR CHECKK.GT.FALMAK OR CHECKLK.LT.BALMIN,
C 2 CP.CHECKK.GT.FALMAX) 30 TJ 300
C
C ENTER BLCK J ON FLOW DIAGRAM, FIG. 1, APPENDIX C

IF TIME GT TMAX G0 TO 300
CC TC 1C

C REACH HERE AFTER TCC MANY ITERATIONS IN ITSOLVE, BLCK D

15C CALL MEBAL(11)
WRITE(6,151)

151 FORMAT(1HO,*
********** TOU MANY ITERATIONS IN ITSOLVE **********)

30C CONTINUE

C WRITE MOST RECENT RESULTS TO UNIT 8 BEFORE STOPPING

C CALL IC(2,TIME)
WRITE(8,31) TIME,DELTIK,(PULO(K),K=1,20),(SOLD(K),K=1,20),
2 (PNUK(K),K=1,2C),(SNUK(K),K=1,20),
301 FORMAT(2F14.3 / 16(5F10.5/))
STOP
EAC

C SLBCLTIME ITSCLV(TIME,**)
IMPLICIT REAL*8(I-H,C-Y)

C SLBCLTIME FCR VERSION ** IV **
SOLVES OVER GRIC FOR GIVEN DELTA TIME USING NEWTON-RAPHSON METHOD

C FIRST EXIT IS FCR TCC MANY ITERATIONS, SECOND FCR DIVIDING BY
C ZERO IN SCLVB

C COMCN/FS/ FCLD,SLC,PNUK,SNUK,PNUK1,SNUK1
C COMCN/BIASCL/ CELP,DELS
C COMCN/EITRI/ A8,C,D
C COMCN/CCNST/ FCF,DELTIK,HH,KAPPF,F,DELTA,DELTAX,H,PA,TEXTER
C Z,HRHCRK,CFCF,FF,HS,UTMS,UTMSK,MRSTOR
C COMCN/EC/PECUNC, PINIT, SINIT, TMAX, PANTDL, SNEWB, SLODBB
C Z PSTD,PCOE,EPMAX,BALM0,NUDES, NLESS1, NLESS2, NPLUS1, MAXNUM
C REAL*8 KAPPF
C DIMENSION FCLD(2C),SOLD(2C),PNUK(2C),SNUK(2C),PNUK1(2C),SNUK1(2C),
C A20(20,4),B20(20,4),C20(20,4),U20(20),V20(20),SSTO(2C),DPOLD(2C),
C FKM20(2C),FKET20(2C),F20(2C),F4C20(2C),
C PST20(2C),STT20(2C),F20(2C),UF120(2C),UF120(2C),F320(2C),CF320(2C),DF3S20(2C)
C FZ20(2C),DFZ20(2C),CFZ20(2C),F420(2C),UF420(2C),DF4S20(2C),DP4E20(2C),
C DLP20(2C),DLE320(2C),CPSAVE20(2C),DLSAVE20(2C)
C
C FIRST EVALUATE THINGS THAT STAY CONSTANT OVER A TIME STEP

C CC 5 K=1,NLESS1
C PSTC1K) = (FCLD(H)+PCLD(K+1))/2.
C SSTC1K) = (SCLD(K)+SCLUD(K+1))/2.
C CC 5 FCLD(K) = FCLD(K+1)-PULO(K)
C SCLUDN = SCLD(NLESS2)-3.*SCLUD(NLESS1)+3.*SCLUD(NODES)
C IF (SCLUDN.GT.0.9999) SOLDBN = SOLDN(NODES)
C PSTCLN(N)ES) = (FCLUD(NODES)+PCLUD(NODES))/2.
C SSTCLN(N)ES) = (CLUD(NODES)+SCLUD(NODES))/2.
C DPCLUD(N)ES) = PPCLUD-PCLUD(NODES)
C TTT = ABS(TIME - DELTIK)

-26-
114. IF (TTL.LT.1*CO-(7.AND.111.TT.GT.-1.0D+07) UPCLD(NODES) = 0.
115. CC 10 K=1,ACCES
116. FKN(K) = PHI(1,FSTO(K),SSTU(K)) * UPCLD(K)
117. FKE(K) = PHI(3,FSTO(K),SSTU(K)) * UPULG(K)
118. FZO(K) = PHI(2,FCLD(K),SULD(K))
119. 10 F40(K) = PHI(4,FCLD(K),SULD(K))
120. FKM(1) = FK5(1)
121. FKET(1) = FKE(1)
122. CC 12 K=2,ACCES
123. FKM(K) = FKM(K)-FKM(K-1)
124. 12 FKET(K) = FKE(K)-FKE(K-1)
125. C ASW ENTER ITERATION LOOP; FIRST EVALUATE VARIOUS FUNCTIONS
126. 1 C
127. KCOUNT=C
128. 15 CONTINUE
129. CC 20 K=1,6LESS1
130. PST(K) = (FALK(K)+PAUK(K+1))/2.
131. 20 SSK(K) = (SLNK(K)+SNUK(K+1))/2.
132. SNEWBN = SLNK(6LESS2)-3 * SNUK(6LESS1)+3 * SNUK(NODES)
133. IF (SNEWBN,CT,SLNK(NODES)) SNEWBN = SNUK(NODES)
134. PST(NODES) = (PAUK(NODES)+PBOUND)/2.
135. SSK(NODES) = (SLNK(NODES)+SNEWBN)/2.
136. CC 25 K=1,ACCES
137. CLMP=PST(K)
138. 1 CLMS=SSK(K)
139. F1(K) = PHI(1,CLMP,DUMS)
140. CF1F(K) = CFF(1,CLMP,TUMS,HP,HS)
141. DF1S(K) = CFF(1,2,CLMP,TUMS,HP,HS)
142. F3(K) = PHI(3,CLMP,DUMS)
143. CF3P(K) = CFF(2,1,CLMP,TUMS,HP,HS)
144. DF3S(K) = CFF(2,2,CLMP,TUMS,HP,HS)
145. CLMF=PALK(K)
146. CLMS=SNLK(K)
147. F2(K) = PHI(2,CLMP,DUMS)
148. CALL CFFCICL(CLMF,GRL)
149. CALL CFFCC(CLMF,GRC)
150. CF2F(K) = FCP*(CFG + LUMS*(GRL-GRC))
151. CALL P+CLTS(CLMF,GRL)
152. CALL P+CGTS(CLMF,GRG)
153. DF2S(K) = FCP*(FL-GR)
154. F4(K) = PHI(4,CLMF,LUMS)
155. CALL GRHLH(CLMF,GRLH)
156. CALL GFPGB(C,CLMF,GRGB)
157. CF4P(K) = FCP*(GFCB + DUMS*(GRLH-GRGB))
158. Z = 1 - (1,DPOR)*FCPC*CR*UR*OK*TVHP*(LUMP)*0.15052
159. CALL RGBTS(CLMF,GRGB)
160. CALL RLHTS(CLMF,RLHL)
161. 25 CF4S(K) = FCR*(FLHL-GRHG)

161. 1 C
162. Awei EVALUATE AND FILL IN THE A,B,C,U ARRAYS FOR SCLVBFT
163. 1 C
164. FIRST UCLABLE FCF FIRST; ALL OF THIS CAN BE DONE MORE EFFICIENTLY
165. 1 C
166. CCN5(1)*FALK(2)-PAUK(1)
167. E(1,1) = H*(CF1F(1)*UPNEW(1)-2*F1(1)) - DF2P(1)
168. E(1,2) = H*(CF1S(1)*CPNEW(1) - DF2S(1)
169. E(1,3) = H*(CF3F(1)*UPNEW(1)-2*F3(1)) - DF4P(1)
170. Z = TV+P+PAU(1)*U*PA*DELT*OY*52590
171. E(1,4) = H*CF3S(1)*UPNEW(1) - LF4S(1)
C ASSIGN THE MCLE ROWS

CC 30 *=2,9LESS1
GPNF[1](K) = FALK(K+1) - PNUK(K)
A(K,1) = FR*(F1(K-1)-UPNEW(K-1)+DF3P(K-1)/2).
A(K,2) = +FR*LFIS(K-1)*UPNEW(K-1)/2.
A(K,3) = +FR*F3(K-1)*UPNEW(K-1)*DF3P(K-1)/2.
A(K,4) = -FR*LFIS(K-1)*UPNEW(K-1)/2.
B(K,1) = FR*(F1(K)-F1(K-1)+DF1P(K)-DF3P(K)-UPNEW(K-1))
2 - DF1P(K-1)/2. - DF2P(K)
B(K,2) = FR*(CF3S(K)*UPNEW(K)-UPNEW(K-1))//2.-DF2S(K)
B(K,3) = FR*(-F3(K)-F3(K-1)+DF3P(K)-UPNEW(K-1))
2 - DF3P(K-1)/2. - DF3P(K)
B(K,4) = FR*(CF3S(K)*UPNEW(K)-UPNEW(K-1))//2.-DF2S(K)
C(K,1) = FR*(F1(K)+LFIS(K)-FL1(K)-UPNEW(K-1)+FKMT(K))
2 + F2(K) - F2(K)
3C C(K,2) = -FR*(F3(K)+UPNEW(K)-F3(K-1)+FKET(K)) + F4(K)
2 - F4(K) - RCCKHT(PNUK(K),PULD(K))

C FILL IN THE LAST MCLE ROW

CFAEW(NNODES) = FEGUNC - PNUK(NODES)
A(NNODES,1) = FR*(F1(NLESS1)-UPNEW(NLESS1)+DF1P(NLESS1)/2).
A(NNODES,2) = -FR*LFIS(NLESS1)*UPNEW(NLESS1)/2.
A(NNODES,3) = +FR*F3(NLESS1)*UPNEW(NLESS1)*DF3P(NLESS1).
A(NNODES,4) = -FR*LFIS(NLESS1)*UPNEW(NLESS1)/2.
E(NNODES,1) = FR*(F1(NLESS1)+DF1P(NLESS1))
2 - DF1P(NLESS1)/2. - DF2P(NLESS1)
E(NNODES,2) = FR*(CF3S(NLESS1)*UPNEW(NNODES)-DF3S(NLESS1)
2 - DF2S(NNODES).
E(NNODES,3) = -FR*(F3(NLESS1)+F3(NNODES)-F3(NLESS1))
2 + CF2P(NNODES)-CFAEW(NLESS1)
E(NNODES,4) = FR*(CF3S(NNODES)*UPNEW(NNODES)-DF3S(NLESS1)
2 - DF2S(NNODES)
E(NNODES,5) = -FR*(F1(NNODES)-F1(NLESS1))
2 + CF2P(NNODES)+FKMT(NNODES)
2 + F2C(NNODES) - F2(NNODES)
E(NNODES,6) = FR*(F3(NNODES)*UPNEW(NNODES)-F3(NLESS1)
2 + CF2P(NLESS1)+FKET(NNODES)
2 + F4C(NNODES) - F4(NNODES) - RCCKHT(PNUK(NODES),PULD(NODES))

C DETERMINE IF MAPPING IS ACCEPTABLE

CALL SCLVE(NNODES,F23C)
C SCALLTIC VECTOR IS IN VVEL AND VELS. EVALUATE NEW SOLUTION
C
KCLNT = KCLNY + 1
IF (KCLNT.LT.3L6, C.FMCR(KOUNT,2), NE.1) GO TO 91
CC 96 NE=1,ACCES
IF (CAES(CPSAVE(K)).LT.1.0D-5) GO TO 85
AP = CELP(K)/CPSAVE(K)
IF (CAES(AP).GT.C5.95) GO TO 85
FNLK1(K) = FNLK(K) + CELP(K)/(1.0 - AP)
GO TC 81
81 FNLK1(K) = FNLK(K) + CELP(K)
IF (CAES(CSSAVE(K)).LT.1.0D-5) GO TO 85
AS = CELS(K)/CSSAVE(K)
IF (CAES(AS).LT.C7.95) GO TO 85
FNLK1(K) = FNLK(K) + CELS(K)/(1.0 - AS)
GO TC 85
85 FNLK1(K) = FNLK(K) + CELS(K)
CONTINUE
GO TC 82
82 CC 54 NE=1,ACCES
FNLK1(K) = FNLK(K) + CELP(K)
SALK1(K) = SALK(K) + CELS(K)
GO TC 84
84 CC 56 NE=1,ACCES
OPSAVE(K) = CELF(K)
86 CCSAVE(K) = CCFLS(K)
871 C TEST FOR CONVERGENCE. CONDITION IS THAT THE MAX CF PMAX AND REMAX
872 C MUST BE LESS THAN CEIPE. THESE ARE THE MASS AND ENERGY RESIDUALS
873 C AT EACH NODE.
875 CPMAK = CAES(CELK(1))/PULL(1)
876 DSMAX = CAES(CELK(1))
878 RMMAK = CAES(1,1,1)
879 REMAX = CAES(1,1,2)
880 CC 48 NE=2,ACCES
PMMAK = CMAX1(PMAX,LABS(C(K,1)))
REMEX = CMAX1(REMAX,LABS(D(K,2)))
CPMAK = CMAX1(CPMAX,LABS(CELP(K))/PULL(K,1))
8854 CMAX = CMAX1(CMAX1,LABS(CELS(K)))
WRITE(6,261) KCLNT,OPMAK,USMAX,KMAX,REMAX
261 FORMT(*,*,ITRERATION NUMBER=I2,OA,DELPMAK=O,1PD10,2,5X,
262 Z,DELMAX=O,1PD10,2,5X,SMAX=O,1PD10,2,5X,REMAX=O,1PD10,2)
263 CC 45 NE=1,ACCES
FNLK1(K) = FNLK1(K)
266 SNK1(K) = SALK1(K)
267 IF (KCLNT.GT.MAXFLY) RETURN 1
268 IF (RMMAK.GT.CELF) ANL.REMAX.LT.CELT) RETURN
GO TC 15
269 WRITE(6,221)
270 221 FORMT(*,*,DIVPLOD BY ZEKL IN SOLVBT ****)
271 270 RETURN 2
272 END
C
C ***********************************************
2722 C
2723 C SUBCLTUE IC(L,1,1,F)
2731 C
2732 C SEE MARKER NO. 2, APPENDIX C
2733 C
274. IMPLICIT REAL*(8) (A-H,C-Y)
275. COMMON/KSVS1/ TF, KABSF, SWRF, SLRF
276. COMMON/EC/FECLCK, PINIT, SINIT, TMAA, PRNTOL, SNEWRN, SLOCAL,
277. Z, FSTCR, PC001, ELMAX, ELMIN, NODES, NLESS1, NLESS2, NPLUS1, MAXNUM
278. COMMON/F5/ FCCL, SCCL, PNUK, SNUK, PNUK, SNUK1
279. COMMON/CCS1/ FCTR, DELTtk, HH, KAPPA, FUL, ULETA, DELTAXH, UP, TExTER
280. Z, RFCRCK, CRFCRCK, FPRG, SUTH, DUASCK, FSTUR
281. COMMON/EA/EAVG, CUMP, SAVEE, LIMP, CHECK, CHECK, STARTM, STARTE, QM, QE
282. REAL*8 KAPPA, FNL(20), SNUK(20), TF(2), KABSF(2), SWRF(2), SCR(2)
283. Z, PCLC(20), SCCL(20), PNUK(20), SNUK(20)
284. RETURN
285. IF L=2 WRITE CLT THE ANSWERS FOR THE MOST RECENT TIME STEP
286. IF L=1 CC THE FOLLOWING
287. C READ THE INPUT
288. C 10 READ(5,101) FCR, KAPPA, U, PA, CPRCK, RHRUK
289. READ(5,102) (TF(K), K=1,2), (KABSF(K), K=1,2), (SWRF(K), K=1,2)
290. C 2 (SCR(K), K=1,2)
291. READ(5,103) CELTtK, DELTAXH, ULETA, WF, NODES, MAXNUM, BALDEL, PRNTOL, FPRG, HH
292. READ(5,104) PINIT, SINIT, TExTER, PBOUND, TIME, TMAX
293. READ(5,105) FCKn, DTMSCR
294. C ANTE THAT THERE MUST BE 20 VALUES OF EACH VECTOR READ IN, EVEN
295. C THOUGH THEY ARE NOT ALL USED.
296. C 101 FCMAT(12, F5, 2, 4F1.3)
297. 1C2 FCMAT(2F5, 1, 2FC, 4, 4F5, 2)
298. 103 FCMAT(FE, 3, F10, 5, 10, 3, FE, 2, 214, FE, 2, 3, 20)
299. 104 FCMAT(FE, 10, 2)
300. IC5 FCMAT(5, F1C, 5)
301. IF = (CELTK)/(*2, DELTAXH)(2)
302. NLESS1 = ACCES - 1
303. NLESS2 = ACCES + 1
304. NPLUS1 = ACCES - 1
305. BALMAX = 1. C + FALCEL
306. BALMIN = 1. C - FALDEL
307. PRTCR = PCRCK
308. FBOUND = PINIT - FDCnA
309. WSTCR = WF
310. GTMS = GTMSCF + C.1
311. C WRITE CLT INTRCELCTCFY HEADINGS
312. WRITE(6,201)
313. WRITE(6,202)
314. WRITE(6,203)
315. WRITE(6,204)
316. WRITE(6,205)
317. WRITE(6,206) PCR, KAPPA, U, PA, CPRCK, RHRUK
318. WRITE(6,207) (TF(K), K=1,2), (KABSF(K), K=1,2), (SWRF(K), K=1,2)
319. Z, (SCR(K), K=1,2)
320. WRITE(6,208) CELTtK, DELTAXH, ULETA, WF, TMAA, PRNTOL
321. WRITE(6,209) PINIT, SINIT, TExTER, PBBOUND, TIME, TMAX
322. WRITE(6,201) IF, WW, NODES, MAXNUM, BALDEL, PC001, DTMSCR
323. TO1 FORMAT("*", "** VERSION IX **", 125, "HORIZONTAL LINEAR SINGLE-CC
324. 2FFCEN1 TBC-PHASE FLOW OF STEAM AND WATER THROUGH A POROUS MEDIL
225. \text{**}
226. 202 \text{FORMAT(1HO,75, 'HEAT FLUXES TO EXTERIOR ARE ACCOUNTED FOR*,}
227. 5X,'NEWTON-RAPHSCH METHOD USED TO SOLVE EQUATIONS**)}
228. 203 \text{FORMAT(1HO, 150, 'PAUL C. ATKINSON, FALL 1973')}
229. 24 \text{FORMAT(1HC)}
230. 2C \text{FORMAT(1HO, 'SYSTEM INPUTS ARE AS FOLLOWS: *')}
231. 2C \text{FORMAT(1HO, 'FCFCSITY', F4.2, F4.2, 'THERMAL ConductIVITY=', F5.3,}
233. 2C \text{Z CFCACK', F6.3, F4.2, 'RHORUK=', F7.2)}
234. 2C \text{FORMAT(1HC, 'TEMPERATURES=', F2.5, 'ABSOLUTE PMX=', F2.7, F4.2,}
235. 2C \text{Z F5.5, 'SHR=', F2.5, 'SUR=', F5.5, 'SUR=', F5.5)}
236. 2C \text{FORMAT(1HO, 'CELITK=', F7.2, 'X', 'DELTA=', F7.2, 5X, 'DELTA='}}
237. 2C \text{Z IPC1, 1.5X, 'WF=', OPF4.2, 5X, 'TMA=', F10.3, 5X, 'PRNTDL=', F10.5)}
238. 2C \text{FORMAT(1HO, 'INITIAL CONDITIONS ARE : P=', F7.2, 5X, 'SH=', F7.2, 5X, 'TEMP IS IN DEGREES F')}
239. 2C \text{FORMAT(1HO, '4 = (1/2)*(DELITK/DELITX**2) = '}}
240. 2C \text{Z F10.5, ' AND IS DIMENSIONAL', F10.5, 'SH=', F10.5, 5X,}
241. 2C \text{Z I3/10, 'MAXIMUM NUMBER INTERNAL ITERATIONS ALLOWED IN ITSOLV=',}
242. 2C \text{Z I3/10, 'MAXIMUM MASS OR ENERGY BALANCE ERROR IS PLUS OR MINUS=',}
243. 2C \text{Z F5.3/10, 'FCCACK=', F5.1, 9X, 'KTH CRITERION=', F7.2)}
244. 2C \text{WRITE(6,22C)}
245. 240 \text{FORMAT(1I,1)}
246. 248 \text{RETURN}
247. 249.1 \text{C EXECUTE THIS SECTION FORTION IF L=2}
249.3 \text{C}
250. 250 \text{WRITE(6,223) (FLATL1(K),K=1, NJDES), PB huyện}
251. 251 \text{WRITE(6,224) (SNATL1(K),K=1, NODES), SNından}
252. 252 \text{WRITE(6,223) TFE, QM, CUMM, CHECKPN, CHEM, CUMH, CHEC, CUMT, CUMC,FILES, CUMR, CUMD)
253. 253 \text{WRITE(6,223) TIE, QM, CUMM, CHECKPN, CHEM, CUMH, CHEC, CUMT, CUMC,FILES, CUMR, CUMD)
254. 254 \text{WRITE(6,223) TIE, QM, CUMM, CHECKPN, CHEM, CUMH, CHEC, CUMT, CUMC,FILES, CUMR, CUMD)
255. 255 \text{WRITE(6,223) TIE, QM, CUMM, CHECKPN, CHEM, CUMH, CHEC, CUMT, CUMC,FILES, CUMR, CUMD)
256. 256 \text{WRITE(6,223) TIE, QM, CUMM, CHECKPN, CHEM, CUMH, CHEC, CUMT, CUMC,FILES, CUMR, CUMD)
257. 257 \text{RETURN}
258. 258 \text{END}
259. 260.1 \text{C}
260.2 \text{C********************RETURN********************}
260.3 \text{C}
261. \text{C SLIDECLINE RECAL()}!
261.1 \text{C}
261.2 \text{C SEE MARKER NO. 2, EFFENCEC C}
261.3 \text{C}
262. \text{IMPLICIT REAL*8(A-H,O-Y)}
263. \text{COMPS=EC/ECNLE, PINIT, SINIT, IMA, PKNTDL, SNENBN, SODBN,}
264. \text{Z PHSCR, PDWN, PALED, BALMIN, NLE1, NLE2, NPLUS, MAXNUM}
265. \text{Z COMPC/A, FCLC, KLUM, PKUK, SNUK, PKUK, SNUK}
266. \text{Z COMPC, CCNK1, FCP, CELITK, PW, KAPPA, WF, KELTA, CELITX, U, PA, TEXTER}
267. \text{Z RPMCE, CFCACK, FPC, SHS, TUM, OTASK, WSTUK}
268. \text{Z COMPC, REAL, SAVEP, SUMP, SAVEE, CUMH, CHEC, CUMR, CHEM, CESECH, STARTM, STARTK, CM, QE}
269. \text{Z IMENSIONICS, PCLC, SGLD, COL, PKUK, PKUK, SNUK, PKUK, SNUK}
270. \text{REAL*8 KAPPA, UNIT21}
271. \text{C MASS BALANCE FIRST : CUMULATIVE OUTFLOW OF MASS FIRST}
272. \text{AEEC SLK1 AT BOUNDARY=QUADRATIC EXTRAPOLATION}
273. \text{SNA=SAK1(NLESS2) - 3.0*SNUK(LNESS1) + 3.0*SNUK(NODES)
-32-

IF (SNEXTB.LT.CT.SPLKL(NCODES)) SNEXTB = SNUKL(NCODES)

CRTOT = - FI1(1,FEOUNO,SNEXTB)*(3.0*PBOND - 4.0*FNUKL(NCODES))

Z = FALNL(ALESSI) / (2.0*DELTAK)

IF (L.EQ.2) SAVEK = CRTOT

CP = (CPTCT+SAVEK)/2.0

CM = CM + CP + DELTOK

SAVEK = QRTCT

TOTAL PASS REMAINING IN THE SYSTEM

EVALUATE THE FIX) EXCLUDING PUR & DELTXK

DO 5 K=1,ACCES

CALL RCLTS(FNULK(1),RMUL)

CALL RCGTS(FALNL(1),RHO)

UNIT(K) = SNKL(1)*RMUL + (1.0 - SNKL(1))*RHO

CALL RCLTS(FECNLK,RHLK)

CALL RCGLS(FECNL,RHOG)

UNIT(NPLUS1) = SNWBN*RMUL + (1.0 - SNWBN)*RHO

EVALUATE THE INTEGRAL EXCLUDING THE PCR & DELTXK TERMS

TCTALM = UNIT(1) + UNIT(NPLUS1)

DO 10 K=2,ACCES

10 TCTALM = TCTALM + 4.0*UNIT(K)

DC 11 K=2,ALESSI,2

11 TCTALM = TCTALM + 2.0*UNIT(K)

REMM = TCTALM*ECF*DELTAX/3.0

IF (L.EQ.2) STARTK = (CM + REMM)

CHECK = (CM + REMM) / STARTK

NOW THE ENERGY BALANCE: CUMULATIVE CUTFLOW CF ENERGY FIRST

TSUM = C.C

EC 15 K=2,ACCES

15 TSLM = TSLM + TEPPD(FNULK(1))

TSUM = TSLM + TEPPD(FNKL(1)) + TEPPD(PBOND) - 40.*TEXTER

CRHTOT = - FI1(2,FEOUNO,SNWBN)*(3.0*PBOND - 4.0*PNUKL(NCODES))

2 + FALNL(ALESSI)/(12.0*DELTAX) + PA*ULLTAX*L*TSUM/2.0

606. IF (L.EQ.2) SAVEF = QHUNIT

607. CE = (CPTCT + SAVEE)/2.0

608. CUME = CUME + CE*DELTOK

609. SAVEE = CPTCT

TOTAL ENERGY REMAINING IN THE SYSTEM

CALLUTHE THE FIX)S EXCLUDING PUR & DELTXK TERMS

EC 20 K=1,ACCES

20 UNIT(K) = FI1(4,FNULK(1),SNUL(1))

UNIT(NPLUS1) = FI1(4,PBUND,SNWBN)

EVALUATE THE INTEGRAL EXCLUDING THE PCR & DELTXK TERMS

TCTAIE = UNIT(1) + UNIT(NPLUS1)

EC 25 K=2,ACCES

25 TCTAIE = TCTAIE + 4.0*UNIT(K)

DC 26 K=2,ALESSI,2

26 TCTAIE = TCTAIE + 2.0*UNIT(K)

REMF = TCTAIE*ECF*DELTAX/3.0

IF (L.EQ.2) STARTE = (CUME + REME)

CHECKE = (CM + REME) / STARTE
464. ETA(1,J) = E(1,J)
465. A(I,J) = C.
466. C(N,J) = C.
467. CTETA(1,1) = C(1,1)
468. CTETA(1,2) = C(1,2)
469. MU(1) = ETA(1,1) * ETA(1,2) * ETA(1,3)
470. I = 1
471. IF (ML(1), L=1) EPS.ANC.MU(1) @T.EPS) GOTO 105
472. LAMBA(1) = (ETA(1,4) * C(1,1) - ETA(1,2) * C(1,3)) / MU(1)
473. LAMBA(1,2) = (ETA(1,4) * C(1,1) - ETA(1,2) * C(1,4)) / MU(1)
474. LAMBA(1,3) = (ETA(1,4) * C(1,3) - ETA(1,1) * C(1,4)) / MU(1)
475. LAMBA(1,4) = (ETA(1,4) * C(1,4) - ETA(1,1) * C(1,3)) / MU(1)
476. GAMMA(1,1) = ETA(1,4) * ETA(1,1) - ETA(1,2) * ETA(1,3) / MU(1)
477. GAMMA(1,2) = ETA(1,4) * ETA(1,1) - ETA(1,2) * ETA(1,3) / MU(1)
478. DO 1=1, 10
479. I = 2, A
480. BETA(1,1) = ETA(1,1) * LAMBA(1-L,1) - ETA(1,2) * LAMBA(1-L,3)
481. BETA(1,2) = ETA(1,1) * LAMBA(1-L,2) - ETA(1,2) * LAMBA(1-L,4)
482. BETA(1,3) = ETA(1,1) * LAMBA(1-L,1) - ETA(1,2) * LAMBA(1-L,3)
483. BETA(1,4) = ETA(1,1) * LAMBA(1-L,2) - ETA(1,2) * LAMBA(1-L,4)
484. CELTA(1,1) = G(1,1) - ETA(1,1) * GAMMA(1-1,1) - ETA(1,2) * GAMMA(1-1,2)
485. CELTA(1,2) = G(1,2) - ETA(1,1) * GAMMA(1-1,2) - ETA(1,4) * GAMMA(1-1,2)
486. MU(1) = ETA(1,1) * ETA(1,2) * ETA(1,3)
487. IF (ML(1), L=1) EPS.ANC.MU(1) @T.EPS) GOTO 105
488. LAMBA(1,1) = ETA(1,4) * C(1,1) - ETA(1,2) * C(1,3) / MU(1)
489. LAMBA(1,2) = ETA(1,4) * C(1,1) - ETA(1,2) * C(1,4) / MU(1)
490. LAMBA(1,3) = ETA(1,4) * C(1,3) - ETA(1,1) * C(1,4) / MU(1)
491. LAMBA(1,4) = ETA(1,4) * C(1,4) - ETA(1,1) * C(1,3) / MU(1)
492. GAMMA(1,1) = ETA(1,4) * ETA(1,1) - ETA(1,2) * ETA(1,3) / MU(1)
493. GAMMA(1,2) = ETA(1,4) * ETA(1,1) - ETA(1,2) * ETA(1,3) / MU(1)
494. IF (GAMMA(1,1), L=1) EPS.ANC.MU(1) @T.EPS) GAMMA(1,1) = 0.0
495. IF (GAMMA(1,2), L=1) EPS.ANC.MU(1) @T.EPS) GAMMA(1,2) = 0.0
496. I = 1
497. I = 1
498. L(N) = GAMMA(1,1)
499. V(N) = GAMMA(1,2)
500. I = 1
501. L(I) = GAMMA(I,1) - LAMBA(I,1)*U(I,2) - LAMBA(I,2)*V(I,1)
502. V(I) = GAMMA(I,2) - LAMBA(I,3)*U(I,1) - LAMBA(I,4)*V(I,1)
503. IF (L(I), L=1) EPS.ANC.MU(1) @T.EPS) J(I) = 0.0
504. IF (V(I), L=1) EPS.ANC.MU(1) @T.EPS) V(I) = 0.0
505. IF (I = 1) CC TO 100
506. I = 1
507. CC TO 55
508. RETURN
509. WRITE(1, 106) I
510. 106 FORMAT(1, *CIVICECE BY ZERU TO GET MU SUB*, 13)
511. RETURN
512. END
513. FUNCTION PHI(L, P, SW)
514. \*
515. \*
C SEE MARKER AC. 6, APPENDIX C
C
REAL*8 XM, Y, Z, PHI, F, G, T, FHI, TEXP, TL

C IMPLICIT REAL*8 (A-H, O-Y)
C CCMMCK, CCNST, FCR, CINLTI, HR, KAPPA, F, DELTA, CINLTH, U, PA, TEXT
C
Z, RHOCCK, CPFCCK, PP, PS, UMS, UMSCK, WF, S

C REAL*8 XM, Y, Z, PHI, F, G, T

C REAL*8 XM, Y, Z, PHI, F, G, T

C NO TOOLS (1, 2, 3, 4, 5) "L"
C
CALL PHCLTS(P, RL)
C CALL PLLTS(F, PL)
C CALL PHCGTS(P, RC)
C CALL PLCTS(F, RC)
T = TEXP(F)
FHI = RL*FEP*(1, T, SW)/ML + RG*PERM(2, T, SW)/MG
RETURN

CALL PHCGTS(P, RC)
C CALL PHPLTS(P, RC)
C PHI = FCR*(FC + SH*(RL-RG))
RETURN

CALL PLFLTS(F, FL)
C CALL PLLTS(F, PL)
C CALL RCGRTS(F, RC)
C CALL PLGRTS(F, RG)
T = TEXP(F)
FHI = RL*FEP*(1, T, SW)/ML + RG*PERM(2, T, SW)/MG + KAPPA*TVHP(P)
RETURN

CALL FLFLTS(F, FL)
C CALL PLLTS(F, FL)
C CALL RCGRTS(F, RC)
C CALL PLGRTS(F, RG)
T = TEXP(F)
PHI = FCR*(FC + SH*(KLH-LGKH)) + (L-PK)*HRHORAK*CROCK*IT-32.1
RETURN
END

C******************************************************************************************
C FUNCTION FCRKHT(FH, FKHT, FO)
C
C SEE MARKER AC. 7, APPENDIX C
C
C******************************************************************************************
C FUNCTION DIFF(N, P, S, HP, HS)
C
C SEE MARKER AC. 8, APPENDIX C
C
C IMPLICIT REAL*8 (A-H, O-Y)
C
C SLEFRCGA TO TAKE THE DERIVATIVE OF PHI(N) WRT P (WHEN N=1) OR WRT
C S (WHEN N=2); AT CONDITIONS P AND S
C
C SECCRE CRGGA DIFFERENCING SCHEME (ABOUT P OR S) IS USED.
TRUNCATION ERROR IS CF H**2

IF (N=EC,2) CC TC 5
CIFF = (-PHI(N,F-TP,N)+PHI(N,F+HP,S))/(2*HP)
RETURN
5 CIFF = (-F11(N,F,S-HP)+PHI(N,F,S+HS))/(2*HS)
RETURN
END

FUNCTION PERN(J,1,S)
SEE MARKER NC = 5, EFFERLEIX C
 IMPLICIT REAL*8(L-H,C-Y)
**** IF J=1 LICLIC (WETTING ) CASE..... IF J=2 STEAM (NCNWET) CASE
COMMON/KSVST/ TF, KABSF, SWRF, SRF
REAL*8 TF(2), KABSF(2), SRF(2), SRF(2), KABSF

KABSF, SWRF, SRF ARE GIVEN AT MJ TEMPS. USE LINEAR INTERPOLATION
BETWEEN THEM TO DETERMINE THE VALUE AT THE REQUIRED TEMPERATURE

CUM=TF(2) - TF(1)
IF (CUM.GT.1.0*CE-3.CPT.UM.LT.-1.0*JE-3) GJ TU 5
SFR=SFRF(1)
SCK=SCRF(1)
KABSF=KABSF(1)
CC TC 6
TDEL = (T - TF(1))/CUM
SFR = SFRF(1) + TDEL *(SFRF(2) - SFRF(1))
SOR = SRF(1) + TDEL *(SRF(2) - SRF(1))
KABSF = KABSF(1) + TDEL *(KABSF(2) - KABSF(1))
6 SWSMF = (SFR - SFRF) / (1.0 - SRF - SRF)

DEFINE KABSF FOR SFR, SRF, SRF = MTS1.SRF

IF (SFR.LT.SRF) CC TC 20
IF (SFR.GT.(1-SCR)) GC TU 30

IF J=1 : LICLIC CASE (WETTING PHASE); IF J=2 : STEAM CASE (NCNWET)

IF (J.EQ.2) CC TC 10
PERN = KABSF * SWSMF**4
RETURN
1C PERN = (1.0 - SWSMF**4) * ((1.0 - SWSMF)**2) * KABSF
RETURN
20 IF (J.EQ.2) CC TC 22
PERN=0.
RETURN
2 IF PERN>KABSF
RETURN
30 IF (J.EQ.2) CC TC 32
PEPN=KABSF
PEPN=0.
RETURN
END
IMPLICIT REAL*8(A-,C-,Y)
COMMON/FRES/ G(17)
COMMON/FLH/ T(I1), B(10), C(10), U(10)
I=1
1CC IF (X-G(I+1)) 11,11,12
12 I=I+1
114.
GC TC 1CO
115.
11 T=X-G(I)
116.
Y=((C(I))T+C(I))T+B(I))T+H(I)
117.
RETURN
118.
END
118.1 C
118.2 C**********************************************************************************************
118.3 C
SLRACTIME FHECTS(X,Y)
119.1 C
119.2 C SEE MARKER NC. 12, APPENDIX C
1200 C
IMPLICIT REAL*8(A-,C-,Y)
121.
COMMON/FRES/ G(17)
122.
COMMON/FLH/ T(I1), B(16), C(10), U(10)
123.
I=1
124.
1CC IF (X-G(I+1)) 11,11,12
125.
12 I=I+1
126.
GO TC 1CO
127.
11 T=X-G(I)
128.
Y=((C(I))T+C(I))T+B(I))T+H(I)
129.
RETURN
130.
END
130.1 C
130.2 C**********************************************************************************************
130.3 C
SLRACTIME FLHLTS(X,Y)
131.
131.1 C
131.2 C SEE MARKER NC. 12, APPENDIX C
131.3 C
IMPLICIT REAL*8(A-,C-,Y)
132.
COMMON/FRES/ G(17)
133.
COMMON/FLH/ T(I1), B(16), C(10), U(16)
134.
I=1
135.
1CG IF (X-G(I+1)) 11,11,12
137.
12 I=I+1
138.
GC TO 1CC
139.
11 T=X-G(I)
140.
Y=((C(I))T+C(I))T+B(I))T+H(I)
141.
RETURN
142.
END
142.1 C
142.2 C**********************************************************************************************
142.3 C
SLRACTIME FHCCTS(X,Y)
143.
143.1 C
143.2 C SEE MARKER NC. 14, APPENDIX C
143.3 C
IMPLICIT REAL*8(A-,C-,Y)
144.
COMMON/FRES/ G(17)
145.
COMMON/FLH/ T(I1), B(16), C(10), U(10)
146.
I=1
147.
1CC IF (X-G(I+1)) 11,11,12
149.
12 I=I+1
GO TO 100  
11 T=X-G(I)  
12 Y=(D(I)*T + C(I))*T + b(I)*T + h(I)  
13 RETURN  
14 END  
15 C  
16 SLCECLINE P = T5(P,Y)  
17 C  
18 SEE MARKET NO. 15, APPENDIX C  
19 C  
20 IMPLICIT REAL*8(A-T,C-Y)  
21 CMPCN/LHF/ E(8)  
22 CMPCN/ML/H(E), E(7), C(7), D(7)  
23 X = CLCG(P)  
24 I=1  
25 IF (X-G(I+1)) 11,11,12  
26 I=I+1  
27 GO TO 100  
28  
29 T=X-G(I)  
30 Y=(D(I)*T + C(I))*T + b(I)*T + h(I)  
31 RETURN  
32 END  
33 C  
34 SLCECLINE P = T5(F,Y)  
35 C  
36 SEE MARKET NO. 16, APPENDIX C  
37 C  
38 IMPLICIT REAL*8(A-T,C-Y)  
39 CMPCN/LHF/ E(8)  
40 CMPCN/ML/H(E), E(7), C(7), D(7)  
41 X = CLCG(P)  
42 I=1  
43 IF (X-G(I+1)) 11,11,12  
44 I=I+1  
45 GO TO 100  
46  
47 T=X-G(I)  
48 Y=(C(I)*T + C(I))*T + b(I)*T + h(I)  
49 RETURN  
50 END  
51 C  
52 SLCECLINE CFF+L(X,Y)  
53 C  
54 SEE MARKET NO. 17, APPENDIX C  
55 C  
56 IMPLICIT REAL*8(A-T,C-Y)  
57 CMPCN/FF/PRESS/ G(17)  
58 CMPCN/ML/H(E), E(10), C(10), D(10)  
59 I=1  
60 IF (X-G(I+1)) 11,11,12  
61 I=I+1  
62 GO TO 100  
63  
64 T=X-G(I)  
65 Y=(3*C*E(I)*T + 2.0*C(I))*T + b(I)  
66 RETURN  
67
**FUNCTION** (FREAL*(X,Y))

**SUBROUTINE** (FREAL*(X,Y))

**SEE** MARKER AC. 18, APPENDIX C

**IMPLICIT** REAL*8(A-H,C-O)

**COMMON**/PRESS/ G(17)

**COMMON**/FGH/F(17), B(16), C(16), D(16)

**IF** (X-G(I)) 11,11,12

**RETURN**

END
C SEE MARKER NC. 21, APPENDIX C

C

1230.C
1231.
C IMPLICIT REAL*8(I-F,C-Y)
1231.TEMFP = -(84CC.735E) / (DLOG(P) - 15.272703) - 460.
1232.
C RETURN
1233.
C END
1233.

C

1233.1.C
1233.2.C
1233.C
1234.C
1234.1.C
1234.2.C
1234.3.C
1235.
1236.
1237.
1238.
1239.
1240.
1241.
1242.
1243.
1244.

C FUNCTION TVFP(P)
C
SEE MARKER NC. 22, APPENDIX C
C
C IMPLICIT REAL*8(I-H,C-Y)
C T = TEMFP(F) + 46C.
C TVFP = T * (EXP(T 7356.815)/T) - 14.65907)
C RETURN
C END

C

C****************** EN DC C F PROGRAM ******************
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.
START

A
READ INPUT PARAMETERS

B
WRITE INITIAL HEADINGS

C
SET ALL INITIAL COUNTERS AND CONDITIONS

D
SOLVE $f(p_s) = 0$ TO A SPECIFIED CONVERGENCE CRITERION USING NEWTON-RAPHSON ITERATION

- STOP

E
EVALUATE MASS AND ENERGY BALANCE CORRESPONDING TO PS SOLUTION

F
PRINT NEW SOLUTION?

G
WRITE OUT SOLUTION AND BALANCES

- STOP

H
PREPARE FOR NEXT TIME STEP:
- DETERMINE NEW AT IF APPROPRIATE
  $t = t + \Delta t$
  $p_s^{\text{old}} - \Delta t$
- ESTIMATE NEW SOLUTION SET
- BOUNDARY CONDITION TO VALUE AT NEW TIME LEVEL

I
BALANCE ERROR TOO LARGE

- STOP

J
TIME TO STOP?

- STOP

FIGURE C-1. FLOW DIAGRAM OF THE COMPUTER PROGRAM
<table>
<thead>
<tr>
<th>MARKER NUMBER</th>
<th>SUBROUTINE NAME AND PARAMETERS</th>
<th>LINE NUMBERS IN SOURCE LISTING</th>
<th>COMMENTS AND DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>ITSOLV(TIME,<em>,</em>)</td>
<td>81/272</td>
<td>Executes Block D in the flow diagram, Fig. C-1. Call 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 the program division by zero or a very small number in SUBROUTINE SOLVB (marker no. 5).</td>
</tr>
<tr>
<td>(2)</td>
<td>IO(TIME)</td>
<td>73/360</td>
<td>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.</td>
</tr>
<tr>
<td>(3)</td>
<td>MEBAL(L)</td>
<td>361/429</td>
<td>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.</td>
</tr>
<tr>
<td>(4)</td>
<td>NUSTEP(TIME)</td>
<td>430/452</td>
<td>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.</td>
</tr>
<tr>
<td>MARKER NUMBER</td>
<td>SUBROUTINE NAME AND PARAMETERS</td>
<td>LINE NUMBERS IN SOURCE LISTING</td>
<td>COMMENTS AND DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------</td>
<td>-------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>(5)</td>
<td>SOLVBT(N,* )</td>
<td>453/512</td>
<td>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.</td>
</tr>
<tr>
<td>8</td>
<td>PHI(L,P,SW)</td>
<td>513/542</td>
<td>This SUBROUTINE FUNCTION returns the value of $\gamma_i$ as defined in Appendix A. The function is evaluated at the pressure, $P$, psia, and at the volumetric liquid saturation, $SW$.</td>
</tr>
<tr>
<td>7</td>
<td>ROCKHT(PNEW, POLD)</td>
<td>543/552</td>
<td>This SUBROUTINE FUNCTION returns the heat loss rate per unit bulk volume of core averaged over the time interval (TIME) to (TIME + DELT). This function is the same as $q'''$ in Appendix A (Eq. A-19), and $\gamma_{5,1}^{n+1/2}$ in Appendix E (Eq. E-9).</td>
</tr>
<tr>
<td>8</td>
<td>DIFF(N, M, P, S, HP, S )</td>
<td>553/584</td>
<td>A SUBROUTINE FUNCTION which returns the derivative of $\gamma_N$ 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.</td>
</tr>
<tr>
<td>9</td>
<td>PERM(J, T, SW)</td>
<td>565/802</td>
<td>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).</td>
</tr>
<tr>
<td>MARKER NUMBER</td>
<td>SUBROUTINE NAME AND PARAMETERS</td>
<td>LINE NUMBERS IN SOURCE LISTING</td>
<td>COMMENTS AND DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------</td>
<td>-------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>1.0</td>
<td>BLOCK DATA</td>
<td>1005/1196</td>
<td>A BLOCK DATA listing containing the cubic spline coefficients for evaluating the functions which contain the physical properties of saturated steam and water.</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MARKER NUMBER</th>
<th>SUBROUTINE NAME AND PARAMETERS</th>
<th>LINE NUMBERS IN SOURCE LISTING</th>
<th>COMMENTS AND DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>RHOILS(X,Y)</td>
<td>1107/1118</td>
<td>Density of saturated water, $\rho_L$, lbm/ft$^3$</td>
</tr>
<tr>
<td>12</td>
<td>RHOILS(X,Y)</td>
<td>1119/1130</td>
<td>Density of saturated steam, $\rho_g$, lbm/ft$^3$</td>
</tr>
<tr>
<td>13</td>
<td>RLHLTS(X,Y)</td>
<td>1131/1142</td>
<td>Density $\times$ specific enthalpy of saturated water, $\rho_L h_L$, BTU/ft$^3$</td>
</tr>
<tr>
<td>14</td>
<td>RCHILTS(X,Y)</td>
<td>1143/1154</td>
<td>Density $\times$ specific enthalpy of saturated steam, $\rho_g h_g$, BTU/ft$^3$</td>
</tr>
</tbody>
</table>

NOTE: The spline fits for viscosity are based on a logarithmic pressure scale rather than a linear scale.
<table>
<thead>
<tr>
<th>MARKER NUMBER</th>
<th>SUBROUTINE NAME AND PARAMETERS</th>
<th>LINE NUMBERS IN SOURCE LISTING</th>
<th>COMMENTS AND DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(21)</td>
<td>TEMPP(P)</td>
<td>1229/1233</td>
<td>A FUNCTION SUBROUTINE which evaluates the saturation temperature, in °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).</td>
</tr>
<tr>
<td>(22)</td>
<td>T (P)</td>
<td>1234/1239</td>
<td>A FUNCTION SUBROUTINE which evaluates the term ( \beta_c ) 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_f L f / h_f ) as a function of temperature. SUBROUTINE FUNCTION TEMPP(P) is used to determine the absolute saturation temperature, ( T_{abs} ) °R.</td>
</tr>
</tbody>
</table>
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 | 2208 | 2209 | 2210 | 2211 | 2212 | 2213 | 2214 | 2215 | 2216 | 2217 | 2218 | 2219 | 2220 | 2221 | 2222 | 2223 | 2224 | 2225 | 2226 | 2227 |
|---|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
|   | .36  | 3.0  | 0.0  | 24   | 0.023| 1.5  | .4     | 0.05  | 1.0E+04| 0.9  | 20   | 19   | 0.9  | 0.50  | 0.0001| 0.0001| 174  | 1.0  | 270  | 174  |
|   | 1.00 | 1.00 | 0.0000 | 0.10000| .30  | .30  | .005  | .05  | 1.0E-04| 0.9  | 20   | 19   | 0.9  | 0.50  | 0.0001| 0.0001| 1.0  | 1.0  | 1.0  | 1.0  |
|   | 50   | 50   | 50   | 50   | 50   | 50   | 50    | 50    | 50    | 50    | 50   | 50   | 50    | 50    | 50    | 50    | 50    | 50    | 50    | 50    |

---

<table>
<thead>
<tr>
<th></th>
<th>33.500</th>
<th>2.000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>173.99996</td>
<td>173.99993</td>
</tr>
<tr>
<td></td>
<td>173.99833</td>
<td>173.99641</td>
</tr>
<tr>
<td></td>
<td>173.94385</td>
<td>173.89696</td>
</tr>
<tr>
<td></td>
<td>172.80593</td>
<td>171.37393</td>
</tr>
<tr>
<td></td>
<td>172.60580</td>
<td>168.95860</td>
</tr>
<tr>
<td></td>
<td>0.99999</td>
<td>0.99999</td>
</tr>
<tr>
<td></td>
<td>0.99970</td>
<td>0.99936</td>
</tr>
<tr>
<td></td>
<td>0.99004</td>
<td>0.98171</td>
</tr>
<tr>
<td></td>
<td>0.83264</td>
<td>0.77791</td>
</tr>
<tr>
<td></td>
<td>173.99992</td>
<td>173.99989</td>
</tr>
<tr>
<td></td>
<td>173.99757</td>
<td>173.99494</td>
</tr>
<tr>
<td></td>
<td>173.92968</td>
<td>173.87404</td>
</tr>
<tr>
<td></td>
<td>172.52293</td>
<td>170.87225</td>
</tr>
<tr>
<td></td>
<td>0.99999</td>
<td>0.99999</td>
</tr>
<tr>
<td></td>
<td>0.99957</td>
<td>0.99910</td>
</tr>
<tr>
<td></td>
<td>0.98752</td>
<td>0.97763</td>
</tr>
<tr>
<td></td>
<td>0.81501</td>
<td>0.76828</td>
</tr>
</tbody>
</table>
**Version IX**

**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**

**Density** = 0.36  **Thermal Conductivity** = 3.000  **U to Exterior** = 0.0  **Prism Length/Area** = 24.000  **CpRoC* = 0.230  **RhoCk** = 165.00

**Temperatures** = 100.0  **Absolute Perms** = 0.1000  **SHS** = 0.30  **SHS** = 0.05  **SHS** = 0.05

**DELTK** = 0.50  **DELTH** = 0.50  **DELTA** = 1.00-04  **WF** = 90  **TMAX** = 150.000  **PNTOL** = 0.50000

**Initial Conditions Are:**  **P** = 174.00  **SW** = 1.000  **TEXT** = 370.00  **PBOUND** = 174.00  **Temps Are in Degrees F**

**R I** = (1/2)*IKDELTK/DELTH** = 1000000  **I** is dimensional

**H** = 0.0001000  **HS** = 0.0001000

**Number of Unpresribed Nodes (Must Be Even and LT 20)** = 20  **Max Number Internal Iterations Allowed in ITSOLV** = 9

**Maximum Error in Energy Balance Error Is PLUS OR MINUS 0.900**

= 0.01000
<table>
<thead>
<tr>
<th>TIME</th>
<th>0.00</th>
<th><strong>MASS RATE</strong></th>
<th>0.00</th>
<th>CUMM OUT= 0.00</th>
<th>WARN= 1.0000</th>
<th>EN RATE= 0.00</th>
<th>CUMEN OUT= 1.0000</th>
<th>EBAL= 1.0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELT = 0.500</td>
<td>DEL = 0.500</td>
<td>DEL = 0.500</td>
<td>DEL = 0.500</td>
<td>DEL = 0.500</td>
<td>DEL = 0.500</td>
<td>DEL = 0.500</td>
<td>DEL = 0.500</td>
<td>DEL = 0.500</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>1</td>
<td>DELPMAX = 9.500-04</td>
<td>DELSMAX = 3.600-03</td>
<td>RMMAX = 2.100-02</td>
<td>REMAX = 2.500-02</td>
<td>REMAX = 2.500-02</td>
<td>REMAX = 2.500-02</td>
<td>REMAX = 2.500-02</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>2</td>
<td>DELPMAX = 5.010-05</td>
<td>DELSMAX = 2.600-03</td>
<td>RMMAX = 1.400-02</td>
<td>REMAX = 1.400-02</td>
<td>REMAX = 1.400-02</td>
<td>REMAX = 1.400-02</td>
<td>REMAX = 1.400-02</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>3</td>
<td>DELPMAX = 1.700-07</td>
<td>DELSMAX = 6.100-05</td>
<td>RMMAX = 0.900-01</td>
<td>REMAX = 0.900-01</td>
<td>REMAX = 0.900-01</td>
<td>REMAX = 0.900-01</td>
<td>REMAX = 0.900-01</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>4</td>
<td>DELPMAX = 9.600-00</td>
<td>DELSMAX = 0.300-00</td>
<td>RMMAX = 1.500-00</td>
<td>REMAX = 1.500-00</td>
<td>REMAX = 1.500-00</td>
<td>REMAX = 1.500-00</td>
<td>REMAX = 1.500-00</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>5</td>
<td>DELPMAX = 9.600-00</td>
<td>DELSMAX = 0.300-00</td>
<td>RMMAX = 1.500-00</td>
<td>REMAX = 1.500-00</td>
<td>REMAX = 1.500-00</td>
<td>REMAX = 1.500-00</td>
<td>REMAX = 1.500-00</td>
</tr>
<tr>
<td>Time</td>
<td>Mass Rate</td>
<td>Cum Mass Out</td>
<td>En Rate</td>
<td>Cum En Out</td>
<td>Fl Bal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-----------</td>
<td>--------------</td>
<td>---------</td>
<td>------------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.500</td>
<td>3.490-02</td>
<td>2.430-01</td>
<td>1.200</td>
<td>1.430-00</td>
<td>1.0005</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iteration Number</th>
<th>DELPHMAX</th>
<th>DELSMAX</th>
<th>RMMAK</th>
<th>REMAX</th>
<th>3.820</th>
<th>0.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.100-04</td>
<td>2.960-03</td>
<td>1.30-01</td>
<td>2.380</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.050-05</td>
<td>3.650-04</td>
<td>1.100</td>
<td>2.200</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.300-06</td>
<td>4.850-07</td>
<td>0.500-03</td>
<td>2.550</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.150-07</td>
<td>5.850-08</td>
<td>1.470-07</td>
<td>8.450</td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Mass Rate</th>
<th>Cum Mass Out</th>
<th>En Rate</th>
<th>Cum En Out</th>
<th>Fl Bal</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.500</td>
<td>3.140-07</td>
<td>4.850-01</td>
<td>1.100-01</td>
<td>1.0007</td>
<td>1.0005</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iteration Number</th>
<th>DELPHMAX</th>
<th>DELSMAX</th>
<th>RMMAK</th>
<th>REMAX</th>
<th>1.250</th>
<th>0.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.070-04</td>
<td>3.190-03</td>
<td>3.850-02</td>
<td>2.450</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2.870-05</td>
<td>1.290-03</td>
<td>2.350-02</td>
<td>7.450</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>6.590-06</td>
<td>3.550-04</td>
<td>6.640-03</td>
<td>2.200</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3.670-07</td>
<td>6.130-05</td>
<td>1.130-02</td>
<td>1.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.890-10</td>
<td>5.640-06</td>
<td>1.550-06</td>
<td>5.230</td>
<td>0.000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Mass Rate</th>
<th>Cum Mass Out</th>
<th>En Rate</th>
<th>Cum En Out</th>
<th>Fl Bal</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.500</td>
<td>3.048-02</td>
<td>5.740-01</td>
<td>1.000</td>
<td>1.0005</td>
<td>1.007</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iteration Number</th>
<th>DELPHMAX</th>
<th>DELSMAX</th>
<th>RMMAK</th>
<th>REMAX</th>
<th>0.8247</th>
<th>0.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.500-06</td>
<td>0.9998</td>
<td>0.9998</td>
<td>0.9988</td>
<td>0.92361</td>
<td>0.8247</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>DELPMAX</td>
<td>DELSMAX</td>
<td>RMMAX</td>
<td>REMAX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>---------</td>
<td>---------</td>
<td>-------</td>
<td>-------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>4,450-04</td>
<td>3,230-03</td>
<td>4,660-02</td>
<td>1,660-01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4,70-05</td>
<td>1,550-03</td>
<td>2,770-02</td>
<td>9,120-00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1,220-05</td>
<td>4,770-04</td>
<td>8,680-03</td>
<td>2,860-00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3,130-07</td>
<td>8,290-07</td>
<td>4,160-05</td>
<td>1,530-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1,710-09</td>
<td>1,570-07</td>
<td>3,030-06</td>
<td>1,010-03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1,050-09</td>
<td>1,540-08</td>
<td>2,300-07</td>
<td>7,260-05</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIME</th>
<th>6.500 **</th>
<th>MASS RATE</th>
<th>3.140-02</th>
<th>CUMMASS OUT</th>
<th>0.960-01</th>
<th>MABL = 0.989</th>
<th>** 1.490-01</th>
<th>CUMEN OUT</th>
<th>0.880</th>
<th>EBAL = 1.003</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTKE</td>
<td>1.000</td>
<td>PRNDL = 1.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ITERATION NUMBER</th>
<th>DELPMAX</th>
<th>DELSMAX</th>
<th>RMMAX</th>
<th>REMAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3,910-04</td>
<td>1,490-03</td>
<td>6,330-02</td>
<td>2,180-01</td>
</tr>
<tr>
<td>2</td>
<td>4,100-05</td>
<td>1,290-03</td>
<td>2,310-02</td>
<td>7,580-00</td>
</tr>
<tr>
<td>3</td>
<td>1,250-05</td>
<td>4,330-04</td>
<td>7,830-03</td>
<td>2,580-00</td>
</tr>
<tr>
<td>4</td>
<td>2,600-07</td>
<td>5,350-07</td>
<td>2,900-05</td>
<td>1,080-02</td>
</tr>
<tr>
<td>5</td>
<td>1,790-09</td>
<td>1,300-07</td>
<td>2,460-06</td>
<td>8,230-04</td>
</tr>
<tr>
<td>6</td>
<td>1,040-09</td>
<td>1,030-08</td>
<td>1,340-07</td>
<td>4,070-05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIME</th>
<th>7.000 **</th>
<th>MASS RATE</th>
<th>3.490-02</th>
<th>CUMMASS OUT</th>
<th>2.310-01</th>
<th>MABL = 0.939</th>
<th>** 1.560-01</th>
<th>CUMEN OUT</th>
<th>8.160</th>
<th>EBAL = 1.007</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTKE</td>
<td>1.000</td>
<td>PRNDL = 1.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ITERATION NUMBER</th>
<th>DELPMAX</th>
<th>DELSMAX</th>
<th>RMMAX</th>
<th>REMAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3,270-04</td>
<td>2,900-03</td>
<td>6,050-02</td>
<td>2,100-01</td>
</tr>
<tr>
<td>2</td>
<td>1,860-05</td>
<td>5,230-04</td>
<td>9,360-03</td>
<td>3,060-00</td>
</tr>
<tr>
<td>3</td>
<td>5,910-06</td>
<td>1,870-04</td>
<td>3,390-03</td>
<td>1,110-00</td>
</tr>
<tr>
<td>4</td>
<td>1,460-07</td>
<td>1,930-07</td>
<td>6,710-06</td>
<td>2,830-03</td>
</tr>
<tr>
<td>5</td>
<td>1,290-09</td>
<td>6,130-08</td>
<td>5,060-08</td>
<td>1,180-05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIME</th>
<th>8.000 **</th>
<th>MASS RATE</th>
<th>4.010-02</th>
<th>CUMMASS OUT</th>
<th>2.710-01</th>
<th>MABL = 0.9887</th>
<th>** 1.440-01</th>
<th>CUMEN OUT</th>
<th>9.580</th>
<th>EBAL = 1.003</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTKE</td>
<td>1.000</td>
<td>PRNDL = 1.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ITERATION NUMBER</th>
<th>DELPMAX</th>
<th>DELSMAX</th>
<th>RMMAX</th>
<th>REMAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3,000-04</td>
<td>2,120-03</td>
<td>4,100-02</td>
<td>1,930-01</td>
</tr>
<tr>
<td>ITERATION NUMBER: 1</td>
<td>DFLMAX = 3, 98x-05</td>
<td>DELSMAX = 1, 980-03</td>
<td>RMMAX = 5, 140-02</td>
<td>REMAX = 1, 840 01</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------</td>
<td>------------------</td>
<td>------------------</td>
<td>------------------</td>
</tr>
<tr>
<td>ITERATION NUMBER: 2</td>
<td>DFLMAX = 2, 330-05</td>
<td>DELSMAX = 6, 240-04</td>
<td>RMMAX = 1, 130-02</td>
<td>REMAX = 3, 680 00</td>
</tr>
<tr>
<td>ITERATION NUMBER: 3</td>
<td>DFLMAX = 1, 540-05</td>
<td>DELSMAX = 2, 460-04</td>
<td>RMMAX = 4, 390-03</td>
<td>REMAX = 1, 430 00</td>
</tr>
<tr>
<td>ITERATION NUMBER: 4</td>
<td>DFLMAX = 2, 800-04</td>
<td>DELSMAX = 2, 110-07</td>
<td>RMMAX = 3, 520-05</td>
<td>REMAX = 1, 280-02</td>
</tr>
<tr>
<td>ITERATION NUMBER: 5</td>
<td>DFLMAX = 1, 630-09</td>
<td>DELSMAX = 5, 070-08</td>
<td>RMMAX = 1, 060-06</td>
<td>REMAX = 3, 540-04</td>
</tr>
<tr>
<td>ITERATION NUMBER: 6</td>
<td>DFLMAX = 3, 740-05</td>
<td>DELSMAX = 3, 070-07</td>
<td>RMMAX = 5, 980-06</td>
<td>REMAX = 1, 990-03</td>
</tr>
<tr>
<td>ITERATION NUMBER: 7</td>
<td>DFLMAX = 4, 570-09</td>
<td>DELSMAX = 1, 100-07</td>
<td>RMMAX = 6, 600-04</td>
<td>REMAX = 1, 940-05</td>
</tr>
</tbody>
</table>

| TIME = 10, 500 | ** MASS RATE = 4, 930-02 | CUMMASS OUT = 3, 080-01 | MEAN = 0, 9884 | EN RATE = 1, 620 01 | CUMEN OUT = 1, 120 02 | EBRL = 1, 081 01 |

<p>| DELTJK = 1, 000 | PREDTL = 1, 000 | PRTOL = 1, 000 |</p>
<table>
<thead>
<tr>
<th>TIME</th>
<th>D M S</th>
<th>DESCRIPTION</th>
<th>DELTMASS</th>
<th>DELTMASS</th>
<th>RMASS</th>
<th>REMASS</th>
<th>TIME</th>
<th>D M S</th>
<th>DESCRIPTION</th>
<th>DELTMASS</th>
<th>DELTMASS</th>
<th>RMASS</th>
<th>REMASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.500</td>
<td>00</td>
<td><strong>MASS RATE= 4.030-02</strong> CUMMASS OUT= 7.280 -</td>
<td><strong>1 Mn=</strong></td>
<td>98.62</td>
<td><strong>EN RATE= 1.490 01</strong> CUMEN OUT= 2.01 02</td>
<td>AL = 1.0246</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>DFTTOL = 1.000</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>PRNTDL = 1.000</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DELTMASS</td>
<td>DELTMASS</td>
<td>RMASS</td>
<td>REMASS</td>
<td>TIME</td>
<td>D M S</td>
<td>DESCRIPTION</td>
<td>DELTMASS</td>
<td>DELTMASS</td>
<td>RMASS</td>
<td>REMASS</td>
</tr>
<tr>
<td>174.000</td>
<td>00</td>
<td><strong>MASS RATE= 4.030-02</strong> CUMMASS OUT= 7.280 -</td>
<td><strong>1 Mn=</strong></td>
<td>98.62</td>
<td><strong>EN RATE= 1.490 01</strong> CUMEN OUT= 2.01 02</td>
<td>AL = 1.0246</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>DFTTOL = 1.000</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>PRNTDL = 1.000</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>178.000</td>
<td>00</td>
<td><strong>MASS RATE= 4.030-02</strong> CUMMASS OUT= 7.280 -</td>
<td><strong>1 Mn=</strong></td>
<td>98.62</td>
<td><strong>EN RATE= 1.490 01</strong> CUMEN OUT= 2.01 02</td>
<td>AL = 1.0246</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>DFTTOL = 1.000</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>PRNTDL = 1.000</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIME</td>
<td>24.500</td>
<td>MASS RATE</td>
<td>4.21D-05</td>
<td>CU MASS OUT</td>
<td>0.790D-01</td>
<td>MULAL</td>
<td>0.9552</td>
<td>REN</td>
<td>RATE</td>
<td>1.570 D-01</td>
<td>CUMEN</td>
<td>OUT</td>
<td>3.54D-02</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>-----------</td>
<td>-----------</td>
<td>-------------</td>
<td>-----------</td>
<td>--------</td>
<td>--------</td>
<td>-----</td>
<td>------</td>
<td>-----------</td>
<td>--------</td>
<td>-----</td>
<td>----------</td>
</tr>
<tr>
<td>DELTA</td>
<td>1.00D</td>
<td>PRNTLTL</td>
<td>1.00D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ITERATION NUMBER</th>
<th>1</th>
<th>DELPMAX</th>
<th>3.61D-04</th>
<th>DELSMA</th>
<th>1.42D-03</th>
<th>RMMAX</th>
<th>2.65D-02</th>
<th>wEMAX</th>
<th>9.150 D-00</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITERATION NUMBER</td>
<td>2</td>
<td>DELPMAX</td>
<td>5.19D-06</td>
<td>DELSMA</td>
<td>6.35D-05</td>
<td>RMMAX</td>
<td>1.07D-03</td>
<td>wEMAX</td>
<td>3.32D-01</td>
</tr>
<tr>
<td>------------------</td>
<td>---</td>
<td>----------</td>
<td>---------</td>
<td>--------</td>
<td>-----------</td>
<td>-------</td>
<td>----------</td>
<td>------</td>
<td>--------------</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>3</td>
<td>DELPMAX</td>
<td>3.49D-06</td>
<td>DELSMA</td>
<td>2.66D-05</td>
<td>RMMAX</td>
<td>4.33D-04</td>
<td>wEMAX</td>
<td>1.28D-01</td>
</tr>
<tr>
<td>------------------</td>
<td>---</td>
<td>----------</td>
<td>---------</td>
<td>--------</td>
<td>-----------</td>
<td>-------</td>
<td>----------</td>
<td>------</td>
<td>--------------</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>4</td>
<td>DELPMAX</td>
<td>3.70D-07</td>
<td>DELSMA</td>
<td>1.19D-07</td>
<td>RMMAX</td>
<td>5.40D-03</td>
<td>wEMAX</td>
<td>3.70D-03</td>
</tr>
<tr>
<td>------------------</td>
<td>---</td>
<td>----------</td>
<td>---------</td>
<td>--------</td>
<td>-----------</td>
<td>-------</td>
<td>----------</td>
<td>------</td>
<td>--------------</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>5</td>
<td>DELPMAX</td>
<td>2.18D-10</td>
<td>DELSMA</td>
<td>2.07D-08</td>
<td>RMMAX</td>
<td>4.06D-07</td>
<td>wEMAX</td>
<td>1.34D-04</td>
</tr>
<tr>
<td>------------------</td>
<td>---</td>
<td>----------</td>
<td>---------</td>
<td>--------</td>
<td>-----------</td>
<td>-------</td>
<td>----------</td>
<td>------</td>
<td>--------------</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>6</td>
<td>DELPMAX</td>
<td>1.02D-09</td>
<td>DELSMA</td>
<td>7.84D-09</td>
<td>RMMAX</td>
<td>1.29D-07</td>
<td>wEMAX</td>
<td>3.80D-05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIME</th>
<th>25.500</th>
<th>MASS RATE</th>
<th>4.16D-02</th>
<th>CUMMASS OUT</th>
<th>1.02D</th>
<th>CBAL</th>
<th>0.8851</th>
<th>REN</th>
<th>RATE</th>
<th>1.56D</th>
<th>CBAL</th>
<th>1.0370</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTA</td>
<td>1.00D</td>
<td>PRNTLTL</td>
<td>1.00D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ITERATION NUMBER</th>
<th>1</th>
<th>DELPMAX</th>
<th>3.58D-04</th>
<th>DELSMA</th>
<th>1.36D-03</th>
<th>RMMAX</th>
<th>2.57D-02</th>
<th>wEMAX</th>
<th>8.94D-00</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITERATION NUMBER</td>
<td>2</td>
<td>DELPMAX</td>
<td>2.98D-06</td>
<td>DELSMA</td>
<td>1.86D-05</td>
<td>RMMAX</td>
<td>8.40D-04</td>
<td>wEMAX</td>
<td>1.30D-01</td>
</tr>
<tr>
<td>------------------</td>
<td>---</td>
<td>----------</td>
<td>---------</td>
<td>--------</td>
<td>-----------</td>
<td>-------</td>
<td>----------</td>
<td>------</td>
<td>--------------</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>3</td>
<td>DELPMAX</td>
<td>1.06D-06</td>
<td>DELSMA</td>
<td>2.81D-06</td>
<td>RMMAX</td>
<td>1.28D-04</td>
<td>wEMAX</td>
<td>3.76D-02</td>
</tr>
<tr>
<td>------------------</td>
<td>---</td>
<td>----------</td>
<td>---------</td>
<td>--------</td>
<td>-----------</td>
<td>-------</td>
<td>----------</td>
<td>------</td>
<td>--------------</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>4</td>
<td>DELPMAX</td>
<td>3.24D-07</td>
<td>DELSMA</td>
<td>4.77D-08</td>
<td>RMMAX</td>
<td>9.65D-06</td>
<td>wEMAX</td>
<td>4.56D-03</td>
</tr>
<tr>
<td>------------------</td>
<td>---</td>
<td>----------</td>
<td>---------</td>
<td>--------</td>
<td>-----------</td>
<td>-------</td>
<td>----------</td>
<td>------</td>
<td>--------------</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>5</td>
<td>DELPMAX</td>
<td>4.03D-09</td>
<td>DELSMA</td>
<td>1.31D-08</td>
<td>RMMAX</td>
<td>1.56D-07</td>
<td>wEMAX</td>
<td>3.67D-05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIME</th>
<th>26.500</th>
<th>MASS RATE</th>
<th>4.09D-02</th>
<th>CUMMASS OUT</th>
<th>1.08D</th>
<th>CBAL</th>
<th>0.9840</th>
<th>REN</th>
<th>RATE</th>
<th>1.4D</th>
<th>CBAL</th>
<th>1.0367</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTA</td>
<td>1.00D</td>
<td>PRNTLTL</td>
<td>1.00D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ITERATION NUMBER</th>
<th>1</th>
<th>DELPMAX</th>
<th>3.46D-04</th>
<th>DELSMA</th>
<th>1.50D-03</th>
<th>RMMAX</th>
<th>2.43D-02</th>
<th>wEMAX</th>
<th>8.48D-00</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITERATION NUMBER</td>
<td>2</td>
<td>DELPMAX</td>
<td>5.15D-06</td>
<td>DELSMA</td>
<td>1.54D-05</td>
<td>RMMAX</td>
<td>2.13D-04</td>
<td>wEMAX</td>
<td>5.02D-02</td>
</tr>
<tr>
<td>------------------</td>
<td>---</td>
<td>----------</td>
<td>---------</td>
<td>--------</td>
<td>-----------</td>
<td>-------</td>
<td>----------</td>
<td>------</td>
<td>--------------</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>3</td>
<td>DELPMAX</td>
<td>9.70D-07</td>
<td>DELSMA</td>
<td>9.10D-06</td>
<td>RMMAX</td>
<td>9.89D-05</td>
<td>wEMAX</td>
<td>3.89D-02</td>
</tr>
<tr>
<td>------------------</td>
<td>---</td>
<td>----------</td>
<td>---------</td>
<td>--------</td>
<td>-----------</td>
<td>-------</td>
<td>----------</td>
<td>------</td>
<td>--------------</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>4</td>
<td>DELPMAX</td>
<td>1.22D-07</td>
<td>DELSMA</td>
<td>5.77D-08</td>
<td>RMMAX</td>
<td>2.87D-06</td>
<td>wEMAX</td>
<td>1.39D-03</td>
</tr>
<tr>
<td>------------------</td>
<td>---</td>
<td>----------</td>
<td>---------</td>
<td>--------</td>
<td>-----------</td>
<td>-------</td>
<td>----------</td>
<td>------</td>
<td>--------------</td>
</tr>
<tr>
<td>ITERATION NUMBER</td>
<td>5</td>
<td>DELPMAX</td>
<td>1.64D-10</td>
<td>DELSMA</td>
<td>4.91D-09</td>
<td>RMMAX</td>
<td>1.00D-07</td>
<td>wEMAX</td>
<td>3.37D-05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TIME</th>
<th>27.500</th>
<th>MASS RATE</th>
<th>4.02D-02</th>
<th>CUMMASS OUT</th>
<th>1.10D</th>
<th>CBAL</th>
<th>0.9847</th>
<th>REN</th>
<th>RATE</th>
<th>1.52D</th>
<th>CBAL</th>
<th>1.0364</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELTA</td>
<td>1.00D</td>
<td>PRNTLTL</td>
<td>1.00D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| ITERATION | 1 | DELPMAX | 3D04 | DELSMA | 1.15D-03 | RMMAX | 2. | wEMAX | 7.89D-00 |
\[ \gamma_{5,i}^{n+1/2} \Delta \frac{U}{A} \frac{T_{i}^{n+1} + T_{i}^{n}}{2} - T_{\infty} ; \]  
(E-9)

where the \( T_{i} \) are saturation temperatures corresponding to the pressures \( P_{i} \).

\[ U = \text{overall steady state heat transfer coefficient from the core to the environment, [BTU/hr ft}^{2} {^\circ F}] , \]

\[ P = \text{the perimeter of the system corresponding to } U \text{, [ft]}, \]

\[ A = \text{cross-sectional area to fluid flow, [ft}^{2}], \]

\[ T_{\infty} = \text{temperature of the environment, [}^\circ F], \]

\[ \phi = \text{porosity of the porous medium, [ft}^{3}/\text{ft}^{3}], \]

\[ c_{pr} = \text{specific heat of the rock matrix, [BTU/1bm} {^\circ F}], \]

\[ \rho_{r} = \text{density of the rock matrix, [lbm/ft}^{3}]. \]

For the internal nodes, the equations are:

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

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

\[ (E-10) \]

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

\[ + \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] \}

\[ + \gamma_{4,i}^{n} - \gamma_{4,i}^{n+1} - (\Delta t) \cdot \gamma_{5,i}^{n+1/2}. \]  
(E-11)
At the mth node next to the right-hand boundary we have:

\[
F_{2m-1} = (HR) \left\{ \gamma_{1,m-1} \left[ p_{m-1}^{n} - p_{m}^{n+1} \right] \right. + \gamma_{1,m} \left[ pB_{n+1}^{n} - p_{m}^{n+1} \right] \\
+ \left. \left[ \gamma_{1,m-1} \left[ p_{m-1}^{n} - p_{m}^{n} \right] + \gamma_{1,m} \left[ p_{B}^{n} - p_{m}^{n} \right] \right] \right\} \\
+ \gamma_{2,m}^{n} - \gamma_{2,m}^{n+1},
\]

\[\text{(E-12)}\]

\[
F_{2m} = (HR) \left\{ \gamma_{3,m-1} \left[ p_{m-1}^{n} - p_{m}^{n+1} \right] \right. + \gamma_{3,m} \left[ pB_{n+1}^{n} - p_{m}^{n+1} \right] \\
+ \left. \left[ \gamma_{3,m-1} \left[ p_{m-1}^{n} - p_{m}^{n} \right] + \gamma_{3,m} \left[ p_{B}^{n} - p_{m}^{n} \right] \right] \right\} \\
+ \gamma_{4,m}^{n} - \gamma_{4,m}^{n+1} - (\Delta t) \cdot \gamma_{5,m}^{n+1/2},
\]

\[\text{(E-13)}\]

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

When the right-hand node has a constant specified pressure, then \( pB_{n} = pB_{n+1}^{n+1} \). 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 1122 and \( pB_{n+1}^{n+1} \) 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 \( \left[ \hat{G}(x) \right] \), 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 \( p_{n}^{n} \) and \( s_{n}^{n} \) are all zero.
The $f_{ij}$ functions are presented below:

For $F_1(x)$:

$$f_{11} = (HR) \cdot \left\{ \frac{\partial}{\partial p} (\gamma_{1,1}^*) \cdot \left[ P_2 - P_1 \right] - 2\gamma_{1,1}^* \right\} - \frac{\partial}{\partial S} (\gamma_{2,1}) , \quad (E-14)$$

$$f_{12} = (HR) \cdot \left\{ \frac{\partial}{\partial S} (\gamma_{1,1}^*) \cdot \left[ P_2 - P_1 \right] \right\} - \frac{\partial}{\partial S} (\gamma_{2,1}) , \quad (E-15)$$

$$f_{13} = (HR) \cdot \left\{ \frac{\partial}{\partial p} (\gamma_{1,1}^*) \cdot \left[ P_2 - P_1 \right] + 2\gamma_{1,1}^* \right\} , \quad (E-16)$$

$$f_{14} = (HR) \cdot \frac{\partial}{\partial S} (\gamma_{1,1}^*) \cdot \left[ P_2 - P_1 \right] , \quad (E-17)$$

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

For $F_2(x)$:

$$f_{21} = (HR) \cdot \left\{ \frac{\partial}{\partial p} (\gamma_{3,1}^*) \cdot \left[ P_2 - P_1 \right] - 2\gamma_{3,1}^* \right\}$$

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

$$f_{22} = (HR) \cdot \left\{ \frac{\partial}{\partial S} (\gamma_{3,1}^*) \cdot \left[ P_2 - P_1 \right] \right\} - \frac{\partial}{\partial S} (\gamma_{4,1}) , \quad (E-20)$$

$$f_{23} = (HR) \cdot \left\{ \frac{\partial}{\partial p} (\gamma_{3,1}^*) \cdot \left[ P_2 - P_1 \right] + 2\gamma_{3,1}^* \right\} \quad (E-21)$$
\[ f_{24} = (HR) \cdot \left\{ \frac{\partial}{\partial S} (\gamma_{3,1}^{*}) \cdot \left[ p_{2} - p_{1} \right] \right\}, \quad (E-22) \]

\[ f_{2j} = 0 \text{ for } j > 4. \quad (E-23) \]

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

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

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

\[ f_{2i-1,2i-2} = \frac{1}{2} \cdot \left\{ (HR) \cdot \frac{\partial}{\partial p} (\gamma_{1,i}^{*}) \cdot \left[ p_{i+1} - p_{1} \right] + \gamma_{1,i}^{*} \right\}, \quad (E-25) \]

\[ f_{2i-1,2i-1} = (HR) \cdot \left\{ \frac{1}{2} \cdot \frac{\partial}{\partial p} (\gamma_{1,i}^{*}) \cdot \left[ p_{i+1} - p_{1} \right] - \gamma_{1,i}^{*} \right\} \]

\[ + \frac{1}{2} \cdot \frac{\partial}{\partial p} (\gamma_{1,i-1}^{*}) \cdot \left[ p_{i-1} - p_{1} \right] - \gamma_{1,i-1}^{*} \right\} \]

\[ - \frac{\partial}{\partial p} (\gamma_{2,i}) \quad (E-26) \]

\[ f_{2i-1,2i} = \frac{1}{2} \cdot (HR) \cdot \left\{ \frac{\partial}{\partial S} (\gamma_{1,i}^{*}) - \left[ p_{i+1} - p_{1} \right] \right\} \]

\[ + \frac{\partial}{\partial S} (\gamma_{1,i-1}^{*}) \cdot \left[ p_{i-1} - p_{1} \right] \right\} - \frac{\partial}{\partial S} (\gamma_{2,i}) \quad (E-27) \]
\[ f_{2i-1,2i+1} + (HR) \cdot \left\{ \frac{1}{2} \cdot \frac{\partial}{\partial p} (\gamma_{1,i}^*) \cdot \left[ p_{i+1} - p_1 \right] + \gamma_{1,i}^* \right\}, \quad (E-28) \]

\[ f_{2i-1,2i+2} = \frac{1}{2} \cdot (HR) \cdot \frac{\partial}{\partial s} (\gamma_{1,i}^*). \quad (E-29) \]

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

\[ f_{2i, j} = 0 \text{ for } j < 2i-4 \text{ and } j \geq 2i+3, \quad (E-30) \]

\[ f_{2i, 2i-3} = (HR) \cdot \left\{ \frac{1}{2} \cdot \frac{\partial}{\partial p} (\gamma_{3,i-1}^*) \cdot \left[ p_{i-1} - p_1 \right] + \gamma_{3,i-1}^* \right\}, \quad (E-31) \]

\[ f_{2i, 2i-2} = \frac{1}{2} \cdot (HR) \cdot \frac{\partial}{\partial s} (\gamma_{3,i-1}^*) \cdot \left[ p_{i-1} - p_1 \right], \quad (E-32) \]

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

\[ - \frac{\partial}{\partial p} (\gamma_{4,i}) \cdot (\Delta t) \cdot \frac{\partial}{\partial s} (\gamma_{5,i}^{n+1/2}), \quad (E-33) \]

\[ f_{2i, 2i} = \frac{1}{2} \cdot (HR) \cdot \left\{ \frac{\partial}{\partial s} (\gamma_{3,i}^*) \cdot \left[ p_{i+1} - p_i \right] \right\}

\[ + \frac{\partial}{\partial s} (\gamma_{3,i-1}^*) \cdot \left[ p_{i-1} - p_1 \right] \right\} - \frac{\partial}{\partial s} (\gamma_{4,i}), \quad (E-34) \]
\[ f_{2i, 2i+1} = (HR) \cdot \left\{ \frac{1}{2} \cdot \frac{3}{p} (\gamma_{3,1}^*) \cdot \left[ p_1^{i+1} - p_1^i \right] + \gamma_{3,1}^* \right\}, \quad (E-35) \]

\[ f_{2i, 2i+2} = \frac{1}{2} \cdot (HR) \cdot \frac{3}{S} (\gamma_{3,1}^*) \cdot \left[ p_1^{i+1} - p_1^i \right]. \quad (E-36) \]

Note that all evaluations of the functions \( f_{ij} \) occur at values of \( p \) and \( S \) at the \((n+1)\)th time 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

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:

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

This system is written in matrix form in Fig. F-1.
\[
\begin{bmatrix}
    b_{11} & b_{12} & c_{11} & c_{12} \\
    b_{13} & b_{14} & c_{13} & c_{14} \\
    a_{21} & a_{22} & b_{21} & b_{22} & c_{21} & c_{22} \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    a_{i1} & a_{i2} & b_{i1} & b_{i2} & c_{i1} & c_{i2} \\
    a_{i3} & a_{i4} & b_{i3} & b_{i4} & c_{i3} & c_{i4} \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    a_{m1} & a_{m2} & b_{m1} & b_{m2} & c_{m1} & c_{m2} \\
    a_{m3} & a_{m4} & b_{m3} & b_{m4} & c_{m3} & c_{m4} \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    u_1 \\
    v_1 \\
    u_2 \\
    v_2 \\
    \vdots \\
    u_i \\
    v_i \\
    \vdots \\
    u_m \\
    v_m \\
\end{bmatrix}
= \begin{bmatrix}
    d_{11} \\
    d_{12} \\
    d_{21} \\
    d_{22} \\
    \vdots \\
    d_{i1} \\
    d_{i2} \\
    \vdots \\
    d_{m1} \\
    d_{m2} \\
\end{bmatrix}
\]

**Figure F-1. Bitridiagonal System of Linear Algebraic Equations**
The algorithm is:

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

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

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

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

with \( \beta_{i,j} = b_{i,j} \) 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 \( \delta_{i,1} = d_{i,1} \) and \( \delta_{i,2} = d_{i,2} \);

and:

\[ \mu_{i} = \beta_{i,1} \beta_{i,4} - \beta_{i,2} \beta_{i,3} \]
The $\delta_{i,j}$, $\delta_{i,j}$, and $\psi_j$ are evaluated to aid the computation of what follows. They do not need to be stored after computation of:

$$
\lambda_{1,1} = (\beta_{i,4} c_{i,1} - \beta_{i,2} c_{i,3})/\psi_1 ,
$$

$$
\lambda_{1,2} = (\beta_{i,4} c_{i,2} - \beta_{i,2} c_{i,4})/\psi_1 ,
$$

$$
\lambda_{1,3} = (\beta_{i,1} c_{i,3} - \beta_{i,3} c_{i,1})/\psi_1 ,
$$

$$
\lambda_{1,4} = (\beta_{i,1} c_{i,4} - \beta_{i,3} c_{i,2})/\psi_1 ,
$$

and:

$$
\gamma_{1,1} = (\beta_{i,1} \delta_{i,1} - \delta_{i,2})/\psi_1 ,
$$

$$
\gamma_{1,2} = (\beta_{i,1} \delta_{i,2} - \delta_{i,1})/\psi_1 .
$$

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:

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

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

and:

$$
u_1 = \gamma_{1,1} - \lambda_{1,1} u_{i+1} - \lambda_{1,2} v_{i+1} ,$$
\[ v_i = \gamma_{i,2} - \lambda_{i,3} u_{i+1} - \lambda_{i,4} v_{i+1}, \]

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.

<table>
<thead>
<tr>
<th>FORMAT</th>
<th>VARIABLE NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CARD 1 - Physical Data for Core</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F5.2</td>
<td>POR</td>
<td>Porosity, ft$^3$ pore volume/ft$^3$ bulk</td>
</tr>
<tr>
<td>F5.2</td>
<td>KAPPA</td>
<td>Axial Thermal Conductivity, BTU/ (hr ft $^\circ$F)</td>
</tr>
<tr>
<td>F10.3</td>
<td>U</td>
<td>Steady state heat loss coefficient to environment, BTU/ (hr ft$^2$ $^\circ$F)</td>
</tr>
<tr>
<td>F10.3</td>
<td>PA</td>
<td>Ratio of perimeter over which heat losses are occurring to cross sectional area to flow, ft$^{-1}$</td>
</tr>
<tr>
<td>F10.3</td>
<td>CPROCK</td>
<td>Specific heat of the rock matrix, BTU/(lb $^\circ$F)</td>
</tr>
<tr>
<td>F10.3</td>
<td>RHOROK</td>
<td>Density of the rock matrix, ft$^3$/lb$_m$.</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>FORMAT</th>
<th>VARIABLE NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>F5.1</td>
<td>TF(1)</td>
<td>Temperature levels for interpolation, $^\circ$F;</td>
</tr>
<tr>
<td>F5.1</td>
<td>TF(2)</td>
<td>$^\circ$F;</td>
</tr>
<tr>
<td>FORMAT</td>
<td>VARIABLE NAME</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>--------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>F10.4</td>
<td>KABSF(1)</td>
<td>Absolute permeability at the two temperature levels, °F</td>
</tr>
<tr>
<td>F10.4</td>
<td>KABSF(2)</td>
<td></td>
</tr>
<tr>
<td>F5.2</td>
<td>SWRF(1)</td>
<td>Residual liquid saturations at the two temperature levels;</td>
</tr>
<tr>
<td>F5.2</td>
<td>SWRF(2)</td>
<td></td>
</tr>
<tr>
<td>F5.2</td>
<td>SORF(1)</td>
<td>Residual gas saturations at the two temperature levels.</td>
</tr>
<tr>
<td>F5.2</td>
<td>SORF(2)</td>
<td></td>
</tr>
</tbody>
</table>

**CARD 3 - Parameters for the Numerical Solution**

<table>
<thead>
<tr>
<th>FORMAT</th>
<th>VARIABLE NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>F8.3</td>
<td>DELTTK</td>
<td>Initial time step size, sec;</td>
</tr>
<tr>
<td>F7.3</td>
<td>DELTXH</td>
<td>Uniform mesh size, ft;</td>
</tr>
<tr>
<td>D10.2</td>
<td>DELTA</td>
<td>Convergence criterion applied to the mass and energy equation residuals at each node;</td>
</tr>
<tr>
<td>F5.2</td>
<td>WF</td>
<td>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;</td>
</tr>
<tr>
<td>I4</td>
<td>NODES</td>
<td>Total number of mesh nodes used, including the endpoints. Must be odd and less than or equal to 21;</td>
</tr>
<tr>
<td>14</td>
<td>MAXNUM</td>
<td>Maximum number of iterations allowed in the Newton-Raphson solution of the nonlinear system of discretized equations;</td>
</tr>
<tr>
<td>F5.2</td>
<td>BALDEL</td>
<td>Maximum allowed error in both the mass (CHECKM) and energy (CHECKE) balance check. Execution stops if</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>F8.4</td>
<td>PRNTDL</td>
<td>Parameter for controlling output, whereby the solution is printed every nth time step, where $n = (\text{PRNTDL})/(\text{DELTTK})$;</td>
</tr>
</tbody>
</table>
Parameters used in Subprogram FUNCTION DIFF (lines 553/564) to evaluate derivatives of \( \gamma_1 \) or \( \gamma_3 \) using the centered difference approximation to the first derivative:

\[
\frac{df}{dx}\bigg|_{x_0} = \frac{f(x_0+h) - f(x_0-h)}{h},
\]

where \( h = HP \) for derivatives with respect to \( P \), and \( h = HS \) for derivatives with respect to \( S \).

**CARD 4 - Various System and Run-Time Parameters**

- **F10.2 PINIT**  
  Initial pressure and saturation of the system. These values appear in the main heading, but are not used anywhere in the current version of the program.

- **F10.2 SINIT**  
  Initial pressure and saturation of the system. These values appear in the main heading, but are not used anywhere in the current version of the program.

- **F10.2 TEXTER**  
  Temperature of the environment surrounding the core system, \( \Delta F \).

- **F10.2 PBOLD**  
  Current value \( \Delta F \) 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, \( \text{sec} \).

- **F10.2 TMAX**  
  Maximum running time, \( \text{sec} \).

**CARD 5 - Step Size Control Parameter**

- **PDOWN**  
  Dummy input. Appears in the main heading, but is not used in the current version of the program;
FORMAT | VARIABLE NAME | DESCRIPTION
--- | --- | ---
 | DIMSCR | Criterion used to increase the time step size. For

\[
\text{DIM} = \max_{\text{all nodes}} |\text{SNUK}(i) - \text{SOLD}(i)|
\]

The time step size is doubled if \(\text{DIM} < \text{DIMSCR}\)

CARDS 6/9 - Initial Pressure Distribution in the System
5F10.5 on each card. \(\text{POLD}(1), \text{POLD}(2), \ldots, \text{POLD}(20)\), in psia;

CARDS 10/13 - Initial Volumetric Liquid Saturation in the System
5F10.5 on each card. \(\text{SOLD}(1), \text{SOLD}(2), \ldots, \text{SOLD}(20)\), dimensionless;

CARDS 14/17 - Estimate of the Pressure Solution at the Second Time Level
5F10.5 on each card. \(\text{PNUK}(1), \text{PNUK}(2), \ldots, \text{PNUK}(20)\), in psia;

CARDS 18/21 - Estimate of the Saturation Solution at the Second Time Level
5F10.5 on each card. \(\text{SNUK}(1), \text{SNUK}(2), \ldots, \text{SNUK}(20)\), 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.