FINAL REPORT

MODELING OF DEVONIAN SHALE GAS RESERVOIRS

CONTRACT #DE-AT21-78MC08216

TASK 16
(Mathematical Modeling of Shale Gas Production)
(2D MODEL)

UGR FILE # 365

JULY 31, 1980

Prepared for:

United States Department of Energy
Morgantown Energy Technology Center
Morgantown, WV 26505

By:

Science Applications, Inc.
Suite 8
Chestnut Ridge Professional Building
Morgantown, WV 26505

TABLE OF CONTENTS

	Page
Introduction	1
Summary	2
Two-Dimensional Model	4
References	10
Appendix A - Flow Equations and Concepts	12
Appendix B - Program Description	19
Appendix B1 - Program Listings	41
Appendix B2 - Sample Input Data	96
Appendix B3 - Sample Output Data	98

JGR File #365
Science Applications, Inc.
July 31, 1980

FINAL REPORT

MODELING OF DEVONIAN SHALE GAS RESERVOIRS

CONTRACT #DE-AT21-78MC08216

TASK 16
(Mathematical Modeling of Shale Gas Production)

July 31, 1980

INTRODUCTION

In recent years energy shortages have prompted both government and private industry to seek and develop unconventional oil and natural gas resources. The U.S. Department of Energy's (DOE) Eastern Gas Shales Project (EGSP) has been in existence for approximately four years and has as one of its goals the commercial development of some of the more organic rich Devonian shale reservoirs underlying the Appalachian, Michigan, and Illinois Basins. A part of this effort involves the development of mathematical models with predictive capability so that various alternative production schemes can be investigated and accurate forecasts of additional recovery can be made.

Most Devonian shale reservoirs are believed to consist of very tight porous shale formations which contain a network of randomly distributed natural fractures. Under these conditions, the fractures may provide most of the permeability to gas flow, but contribute very little to the overall storage capacity. By comparison, the matrix of the shale may provide most of the storage capacity of the shale, but has very low permeability.

Gas transport in the Devonian shale matrix can be by three primary mechanisms: Fickian diffusion, Knudsen diffusion, or Darcy flow--depending upon the pore diameter, the mean free path of the free gas molecules, and the existence of a pressure gradient. Knudsen diffusion is the term applied to diffusion in pores with small diameters and/or at low pressure when collisions of gas molecules with pore walls are more frequent than intermolecular collisions. Also, it is believed that physical desorption is a contributing factor to gas production.[1,2]

Although several studies on Devonian shale modeling have been presented[3-6], a careful review of the results indicates that much work remains to be done before adequate predictive capability exists. The two-dimensional numerical model presented here is believed to be a significant step in advancing the state-of-the art in forecasting performance of Devonian shale reservoirs. It is based on the mathematical model presented by Kucuk and Sawyer[7] and is an extension of the work presented in a previous SAI report.[8]

SUMMARY

The Department of Energy (DOE), Morgantown Energy Technology Center (METC) has been supporting the development of flow models for Devonian shale gas reservoirs. The broad objectives of this modeling program are:

1. To develop and validate a mathematical model which describes gas flow through Devonian shales.
2. To determine the sensitive parameters that affect deliverability and recovery of gas from Devonian shales.
3. To recommend laboratory and field measurements for determination of those parameters critical to the productivity and timely recovery of gas from the Devonian shales.
4. To analyze pressure and rate transient data from observation and production gas wells to determine reservoir parameters and well performance.
5. To study and determine the overall performance of Devonian shale reservoirs in terms of well stimulation, well spacing, and resource recovery as a function of gross reservoir properties such as anisotropy, porosity and thickness variations, and boundary effects.

During the previous annual period, both a mathematical model describing gas flow through Devonian shales and the software for a radial one-dimensional numerical model for single well performance were completed and placed into operation.[8] The radial flow model is a useful tool for studying single well behavior and has been used to history-match Devonian shale well production data. Also, by use of simulated pressure transient tests, it has been used as an aid in the site selection process for the Offset Well Test Program.[9] In addition, a study has been conducted using the radial flow model to determine if well interference data can be used to obtain more uniqueness in history matching.[10]

Although the radial flow model is a powerful tool for studying single well behavior, it is inadequate for determining the effects of well spacing, stimulation treatments, and variation in reservoir properties. Hence, it has been necessary to extend the model to two dimensions, maintaining full capability regarding Klinkenberg effects, desorption, and shale matrix parameters. During the current annual period, the radial flow model has been successfully extended to provide the two-dimensional capability necessary for the attainment of overall program objectives, as described above.

To date, the two-dimensional model has been extensively tested for a wide variety of conditions including multiple wells with constant rate and/or constant-pressure production, variable formation thickness, and reservoir anisotropy. It is believed that this model will prove to be an important step in developing accurate predictive capability for the Devonian shales.

TWO-DIMENSIONAL MODEL

Introduction

Considerable progress has been made in formulation and numerical treatment of gas flow through Devonian shales during the last few years. The most recent contributions provide an analysis capability for dual-porosity systems with desorption and Klinkenberg effects. Progress in simulating the performance of Devonian shale gas reservoirs is, however, not so advanced. Initial simulation attempts were concerned primarily with the transport of the gas through Devonian shales. This report presents the formulation and associated numerical procedures for two-dimensional flow through Devonian shales.

The simulator was formulated to establish a model that would represent a broad range of well and boundary conditions. It is believed that the formulation accounts for the relevant mechanism of gas flow in Devonian shales. In addition to formulation and numerical procedure details, some illustrative computational results are presented.

Mathematical Formulation

The flow equations that are included in the model are presented here. It is assumed that gas transport in Devonian shale reservoirs occurs only in a porous fracture medium into which matrix blocks of contrasting physical properties deliver their gas contents. That is, the matrix acts as a uniformly distributed gas source in a fracture medium. Gas desorption from pore walls will be treated as a uniformly distributed source within the matrix blocks.

General assumptions made for the development of the mathematical model are as follows:

1. The reservoir has single-phase gas flow.
2. The reservoir is at an isothermal condition.
3. Gravity effects are neglected.
4. Surface diffusion of adsorbed molecules on the walls of the pores will be neglected in the mathematical model. However, those gases which are adsorbed on pore walls pass through more efficiently than those which are not. For example, helium is not easily adsorbed

which are adsorbed on pore walls pass through more efficiently than those which are not. For example, helium is not easily adsorbed at ordinary and higher temperatures and, therefore, does not flow readily through a porous medium. The adsorbed phase must contribute to the overall flow. Thus, neglecting surface diffusion in the model will reduce the flow rate from the matrix.

Gas flow through the fracture system can be described by the following equation:

$$\nabla \cdot \left[\rho \frac{k_f}{\mu} \nabla p_f \right] + w_m(p_f, t) = \frac{\partial}{\partial t}(\phi_f \rho) \tag{1}$$

where:

ρ = density, g/cm^3
k_f = fracture permeability, darcy
μ = viscosity, cp
ϕ_f = fracture porosity
p_f = fracture pressure, atm
t = time, sec
w_m = mass flow rate per volume of shale matrix element, g/sec/cm^3

The equation of state for real gas is given by:

$$\rho = \frac{M}{RT} \frac{p_f}{z} \tag{2}$$

where

M = molecular weight, g mole
R = universal gas constant, cm^3-atm/g mole K
T = reservoir temperature, K
z = real gas deviation factor

Substitution of eq. 2 into Eq. 1 yields:

$$\nabla \cdot \left[\frac{k_f}{\mu} \frac{p_f}{z} \nabla p_f \right] + \frac{p_f}{z} q_m = \frac{\partial}{\partial t} \left(\phi_f \frac{p_f}{z} \right) \tag{3}$$

where

q_m = volumetric flow rate per volume of shale matrix element, $cm^3/sec/cm^3$

$$= - \frac{A_m}{V_m} \frac{k_m}{\mu} \left(\frac{\partial p_m}{\partial n} \right)_{surface}$$

A_m = surface area of matrix element, cm^2
V_m = volume of element, cm^3

Equation 3 describes gas flow through the fractured shale reservoir with a source term which is the contribution from the shale matrix. Gas transport through the matrix is also described by the diffusivity equation with a source term due to desorption of gas from the pore walls of the matrix. The following equation describes the motion of gas through the matrix.

$$\nabla \cdot \left[\rho_m \frac{k_{gm}}{\mu} \nabla p_m \right] + w_d = \frac{\partial}{\partial t} (\phi_m \rho_m) \tag{4}$$

where

k_{gm} = permeability to gas, darcy
w_d = desorption rate, $g/sec/cm^3$ shale
p_m = matrix pressure, atm
ϕ_m = matrix porosity

The rate of desorption can be expressed as:

$$w_d = - M \frac{dc_d}{dp_m} \frac{\partial p_m}{\partial t} \tag{5}$$

where

c_d = concentration of gas at the surface of pore walls, $mole/cm^3$ shale

$\frac{dc_d}{dp_m}$ = slope of gas desorption isotherm curve, $mole/cm^3$ shale/atm

The gas permeability of a porous medium usually exceeds the liquid permeability of the same medium. The difference in these permeabilities is due to the phenomenon known as gas slippage, which is related to the mean free path of the gas molecules relative to pore diameter. Consequently, the gas permeability of a porous medium should be a function of the temperature, pressure, and the nature of the gas. Klinkenberg[11] developed the relationship between gas permeability of a porous medium to a nonreactive liquid; viz:

$$k_g = k_\ell \left(1 + \frac{b}{\bar{p}_m}\right) \tag{6}$$

where b is the Klinkenberg factor which is constant for a given gas and a given porous medium at a constant temperature. A graph of k_g versus $1/\bar{p}_m$ should result in a straight line with an intercept of k_ℓ and a slope of bk_ℓ. Thus, gas permeability is greater at low pressures. Klinkenberg factor, b, was recently given by Jones and Owens[12] as:

$$b = 12.64\ k_\ell^{-.33} \tag{7}$$

for gas sand in the permeability range of 0.1 to .0001 md.

As can be seen from Eq. 6, k_g is a function of mean pore pressure and pore diameter only because all other parameters are constant for a given gas and a given temperature. Since pore diameters are small for Devonian shales, b is expected to be large. Therefore, the Klinkenberg effect or slippage factor cannot be ignored in the Devonian shale model.

Substitution of Equations 2, 5, and 6 into Equation 4 yields:

$$\nabla\cdot\left[\frac{k_\ell}{\mu}\frac{p_m}{z}\left(1 + \frac{b}{\bar{p}_m}\right)\nabla p_m\right] = \phi_d \frac{\partial p_m}{\partial t} + \phi_m \frac{\partial}{\partial t}\left(\frac{p_m}{z}\right) \tag{8}$$

where

$$\phi_d = \frac{dc_d}{dp} RT, \text{ dimensionless}$$

The source term in Eq. 3 can be determined from the solution of Eq. 8. An iterative finite-difference solution technique is used to solve Eq. 3 and Eq. 8 simultaneously. Detailed development of the numerical procedure is presented in Appendix A.

The principal limitation at this time stems from lack of information about the desorption, and the size of the shale representative matrix element which is approximated as a cylinder. Additional indepth studies are needed to help define both the conceptual and mathematical reservoir models. Moreover, profiles of rock matrix and fluid properties are needed. This information should add considerably to the degree of definition of the model. In addition, drawdown, buildup, and interference tests would be important to calculate in situ properties of the rock by using the model. It seems likely that several wells would be needed to validate a flow model which would be relevant to conditions of the reservoir exploitation.

Wellbore options are presented below.

Wellbore Options

1. Skin Factor

For radial flow simulations, skin damage may be handled via a damaged zone permeability obtained from the equation presented by Craft and Hawkins.[13]

$$S = (k - k_d)/k_d * \ln(r_{skin}/r_w) \qquad (9)$$

where k is the undamaged permeability and k_d is the permeability of a damaged zone between r_w and r_{skin}. It is convenient to take node 1 to be within the wellbore and node 2 to be the damaged zone. Then, for a given skin factor, S, one can solve Eq. 9 for the value of k_d, using r_{skin} as the external radius of node 2 in the simulator. The value thus obtained, when input into the simulator as the permeability of node 2, will result in well performance corresponding to a dimensionless skin factor of S.

2. Wellbore Storage

Although no explicit provision for wellbore storage exists in SUGAR-MD, the effects of afterflow can be readily observed by appropriately modifying porosity values in the input data for subroutine PARM. For example, when a small grid-block is used (e.g., .25 ft x .25 ft in cartesian coordinates),

it is only necessary to assign a porosity value so that the array VP in the simulator contains the proper (desired) wellbore volume, That is, if V_W is the wellbore volume, then we must input the value of porosity (phi) which satisfies

$$(phi) * dx * dy * dz = V_W \tag{10}$$

This may be done for any number of nodes by utilizing the "special porosity values" option in the routine PARM. It should be noted that the pressure thus calculated by the simulator will be the average wellbore pressure, assuming that the grid-block dimensions are close to actual wellbore dimensions.

In order to actually see the "afterflow" rates, each node at which wellbore storage is to occur should also be specified as a "history node" as discussed in Appendix B. This will assure that, upon shut-in, the afterflow rates are printed at each well.

REFERENCES

1. Thomas, J., Jr., and Frost, R.R., "Internal Surface Area and Porosity in Eastern Gas Shales from the Sorption of Nitrogen, Carbon Dioxide, and Methane," Proceedings, First Eastern Gas Shales Symposium, USDOE, MERC/SP-77/5, March 1978.

2. Frost, R.R., and Thomas, J., Jr., "Use of Internal Surface Area and High-Pressure Methane Sorption Data to Estimate Capacity for Gas Production from the New Albany Shale Group," Preprints, Second Eastern Gas Shales Symposium, USDOE, METC/SP-77/6, Vol. I, October 1978.

3. Holditch, S.A., and Laufer, W.L., "The Analysis of Fractured Gas Wells Using Reservoir Simulation," Paper No. SPE 7473, 53rd Annual Fall Technical Conference and Exhibition SPE-AIME, Houston, TX, October 2-4, 1978.

4. Kuuskraa, V.A., Brashear, J.P., Doscher, T.M., and Elkins, L.E., "Enhanced Recovery of Unconventional Gas," Main Report, Vol. II, (for DOE, Contract No. EF-77-C-01-2705), Lewin and Associates, Inc., February 1978.

5. Springer, T.E., and Nuckols, E.B., "Cottageville Gas Field Correlation Analyses for Reservoir Modeling," Proceedings, Third Eastern Gas Shales Symposium, USDOE, METC/SP-79/6, October 1-3, 1979.

6. Intercomp, "Analysis of Gas Production from Eastern Gas Shales," Final Report to U.S. DOE (Contract EW-78-C-21-8485), November 1979.

7. Kucuk, F., and Sawyer, W.K., "Modeling of Devonian Shale Gas Reservoir Performance," Proceedings Third Eastern Gas Shales Symposium, USDOE, METC/SP-79/6, October 1-3, 1979.

8. Science Applications, Inc. (SAI), "Modeling of Devonian Shale Gas Reservoirs, Final Report to USDOE (Task 10A, Contract #DE-AT21-78MC08216), September 1979.

9. Science Applications, Inc. (SAI), "Site Selection for the Offset Well Test Program," Final Report to USDOE (Tasks 33 and 20-4, Contract #DE-AT21-78MC08216), May 9, 1980.

10. Sawyer, W.K., Alam, J., and Rose, W.D., "Utilizing Well Interference Data in Unconventional Gas Reservoirs," Paper No. SPE 8940, First SPE/DOE Unconventional Gas Recovery Symposium, Pittsburgh, PA, May 18-20, 1980.

11. Klinkenberg, L.J., "The Permeability of Porous Media to Liquids and Gases," Drilling and Production Practice (1941), p. 200-213.

12. Jones, F.O., and Owens, W.W., "A Laboratory Study of Low Permeability Gas Sands," SPE Paper 7551, presented at SPE Symposium on Low Permeability Gas Reservoirs, SPE-AIME, Denver, CO, May 20-22, 1979.

13. Craft, B.C., and Hawkins, M.F., Applied Petroleum Reservoir Engineering, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1959.

14. Young, D.M., Iterative Solution of Large Linear Systems, Academic Press, 1971.

15. Price, H.S., and Coats, K.H., "Direct Methods in Reservoir Simulation," SPE Preprint No. 4278, January 1973.

APPENDIX A

FLUID FLOW EQUATIONS AND CONCEPTS

Reservoir Nodes

With reference to Figure 1a and the basic definitions

F = flux, rate of flow of quantity being conserved/unit cross-sectional area normal to the direction of flow

C = concentration/unit volume

q = input rate/unit volume,

we see that the fundamental conservation relationship has been changed

$$\text{Net Input} = \text{Net Change} \tag{A-1}$$

When expressed over a time interval t is

$$\begin{aligned}&\{(F_x)_{x-\Delta x} - (F_x)_{x+\Delta x}\}\,\Delta y\Delta z + \{(F_y)_{y-\Delta y} - (F_y)_{y+\Delta y}\}\,\Delta x\Delta z + \\ &\{(F_z)_{z-\Delta z} - (F_z)_{z+\Delta z}\}\,\Delta x\Delta y + q(\Delta x\Delta y\Delta z) = \Delta x\Delta y\Delta z\,\frac{C_{t+\Delta t}-C_t}{\Delta t}\end{aligned} \tag{A-2}$$

Here F_x, F_y, and F_z are in the positive x,y, and z directions, respectively, and we have divided both sides by Δt.

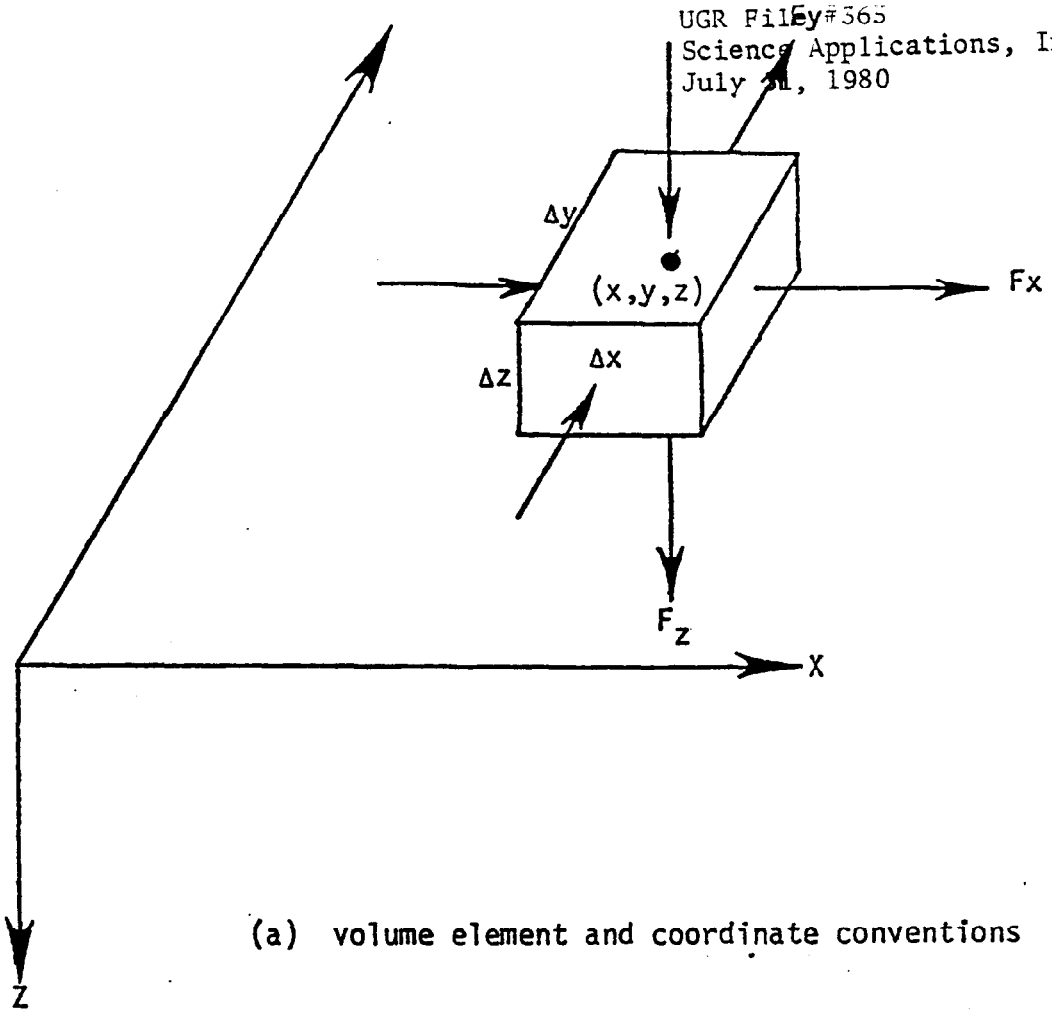
Assuming the x,y, and z axes are the principal axes of conductivity, fluid flow in porous media, heat conduction in solids, and diffusion all have flux components of the form

$$F_sA_s=q_s=-K_sA_s\,\frac{\Delta U_s}{\Delta s} \tag{A-3}$$

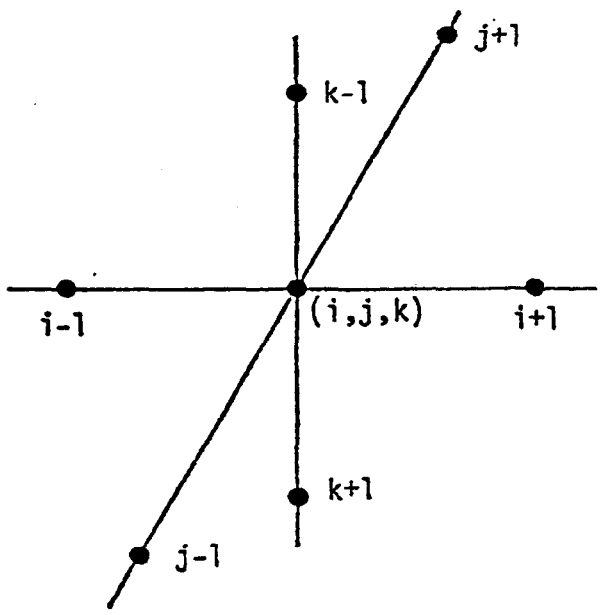
where s represents one of the coordinate axes and K_s is a general conductivity term. Using the differential form

$$F_s=-K_s\,\frac{\partial U_s}{\partial s} \tag{A-4}$$

UGR File #365
Science Applications, Inc.
July 31, 1980

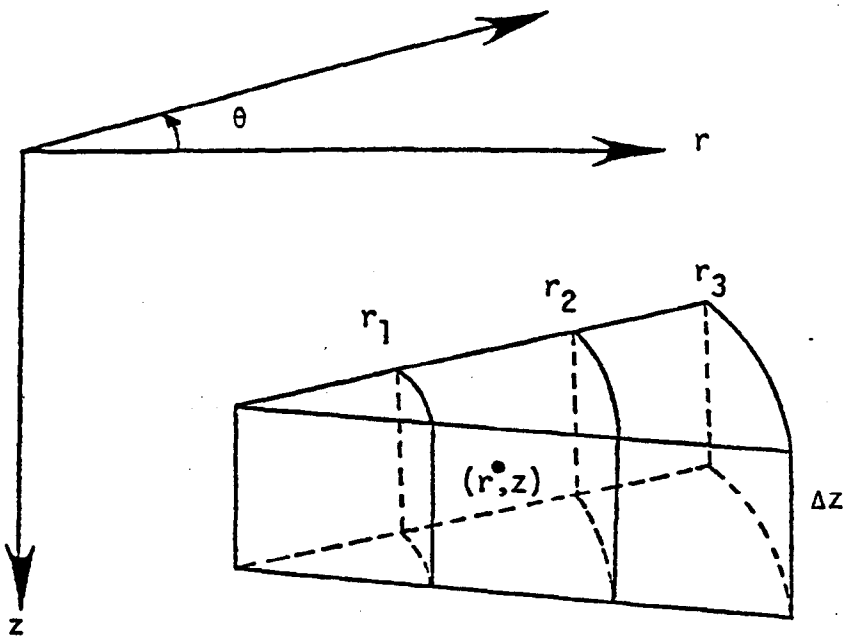


(a) volume element and coordinate conventions



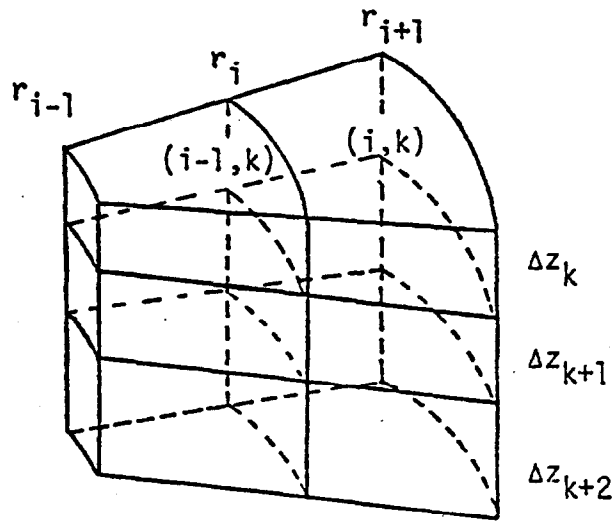
(b) finite-difference notation

Figure A-1. Volume Element Representations-Cartesian Coordinates



$$q_r = \frac{2\pi K \Delta z}{\ln(r_2/r_1)} \left[U(r_2) - U(r_1)\right] (\theta/360)$$

(a) Coordinate Convention



(b) Finite-Difference Notation

Figure A-2. Volume Element Representations -- Polar Cylindrical Coordinates

in (3), dividing by x y z, and expressing C in terms of U gives equation of the form

$$\nabla \cdot K \nabla U = 2 \frac{\partial U}{\partial t} \tag{A-5}$$

The most direct approach to developing a finite-difference model, however, takes a different route.

Defining the coefficient of U_s in Eq. A-4 as the transmissibility across the appropriate face of a volume element as illustrated in Figures 1 and 2

$$TS = \frac{K_s A_s}{\Delta s} \tag{A-6}$$

we see equation 3, that

$$(F_x)_{x-\frac{\Delta x}{2}} A_{x-\frac{\Delta x}{2}} = TX_{x-\frac{\Delta x}{2}} (U_{x-\Delta x} - U_x) \tag{A-7a}$$

$$-(F_x)_{x+\frac{\Delta x}{2}} A_{x+\frac{\Delta x}{2}} = TX_{x+\frac{\Delta x}{2}} (U_{x+\Delta x} - U_x). \tag{A-7b}$$

Similar results hold for the y and z terms. Setting $Q = q(\Delta x \Delta y \Delta z)$ and $V = \Delta x \Delta y \Delta z$, and using the finite difference notation of Figure 1, equation 3 thus becomes

$$\begin{aligned} &TZ_{k-1/2} (U_{k-1} - U_k) + TY_{j-1/2} (U_{j-1} - U_j) + \\ &TX_{i-1/2}(U_{i-1} - U_i) + TX_{i+1/2} (U_{i+1} - U_i) + \\ &TY_{j+1/2}(U_{j+1} - U_j) + TZ_{k+1/2} (U_{k+1} - U_k) + Q = \Delta V \left\{\frac{C^{n+1} - C^n}{\Delta t}\right\}. \end{aligned} \tag{A-8}$$

Here, most subscripts i, j, and k have been suppressed for brevity and a superscript has been added to denote the time level on the RHS. All U's on the left hand side are taken at time level n+1. Each block to block transmissibility is the harmonic mean of the two values.

For Darcy flow of a gas through a porous media of permeability k and porosity ϕ we have

$$K = \frac{k}{\mu B_g} \tag{A-9}$$

$$C = \phi \rho \tag{A-10}$$

where μ is gas viscosity and ρ is gas density given by

$$\rho = \frac{M}{RT} \cdot \frac{p}{z} \tag{A-11}$$

Since our dependent variable is pressure and

$$B_g = \frac{p_{sc}T}{T_{sc}} \cdot \frac{z}{p} \tag{A-12}$$

equation (8) becomes

$$\begin{aligned} &TZ_{k-\frac{1}{2}} (p_{k-1} - p_k) + TY_{j-\frac{1}{2}} (p_{j-1} - p_j) + \\ &TX_{i-t} (p_{i-1} - p_i) + TX_{i+\frac{1}{2}} (p_{i+1} - p_i) + \\ &TY_{j+\frac{1}{2}} (p_{j+1} - p_j) + TZ_{k+\frac{1}{2}} (p_{k+1} - p_k) + \\ &\qquad Q_{sc} + Q_m = \phi \Delta V \frac{T_{sc}}{p_{sc}T} \left\{\frac{(p/z)^{n+1} - (p/z)^n}{\Delta t}\right\} \end{aligned} \tag{A-13}$$

where each term has units of SCF/day. The nonlinear term on the RHS may be expanded as follows

$$\frac{p^{n+1}}{z^{n+1}} - \frac{p^n}{z^n} \doteq p^n \left(\frac{1}{z^*} - \frac{1}{z^n}\right) + \frac{1}{z^*} \left(p_i^{n+1} - p_i^n\right) \tag{A-14}$$

where z^* represents an approximation to z^{n+1}.

Using $z^* = z^n$ to start, equation 13 may be solved by any of the standard techniques (e.g. LSOR, APIP, direct methods). Then using the new pressures new z^*'s may be calculated and the process repeated until no further change is observed. Normally only two to four iterations are required for convergence.

Matrix Element

Gas flow through a "matrix" element, including the Klinkenberg effect, can be described by

$$\nabla \left[\frac{k_m}{\mu} \frac{p_m}{z} \left(1 + \frac{b}{p_m}\right) \nabla p_m\right] = \left(\frac{dc_D}{dp}\right) RT \frac{\partial p_m}{\partial t} + \phi_m \frac{\partial}{\partial t} \left(\frac{p_m}{z}\right) \tag{A-15}$$

where c_D is adsorbed gas concentration, moles/cm^3, T is in degrees K, R is the gas constant in cm^3-atm/g mole degrees K, and other terms are in Darcy units.

Solution of Eq. 15 may be accomplished by first writing the finite-difference analog. Subscript "ℓ" will be used to distinguish the matrix equations from the reservoir finite-difference equations. The Klinkenberg term can be accounted for by first calculating effective gas permeability

$$k_{gm} = k_m \left(1 + b/p^n\right)$$

and then determining transmissibilities in the normal manner. The additional term due to desorption in the "matrix" may easily be handled. Thus, the finite difference analog of Eq. 15 is

$$TR^{n+1}_{\ell-\frac{1}{2}} \left(p^{n+1}_{\ell-1} - p^{n+1}_{\ell}\right) + TR^{n+1}_{\ell+\frac{1}{2}} \left(p^{n+1}_{\ell+1} - p^{n+1}_{\ell}\right) = \tag{A-16}$$

$$\Gamma_\ell \; p^n_\ell \left[\left(\frac{1}{z^*_\ell} - \frac{1}{z^n_\ell}\right) + \frac{1}{z^*_\ell}\left(p^{n+1}_\ell - p^n_\ell\right)\right] +$$

$$\frac{V_\ell}{\Delta t}\left(\frac{dc}{dp}\right)^{**}_{p_\ell} \left(p^{n+1}_\ell - p^n_\ell\right)$$

where $V_\ell = 2\pi k \bar{r}_\ell \left(r_\ell + 1 - r_\ell\right)$ and ** denotes the average of p_ℓ and p^n_ℓ.

In Eq. 16 TR is the interblock transmissibility which, for a cylindrical element, is defined as follows:

$$TR^{n+1}_{\ell-\frac{1}{2}} = \pi h \frac{T_{sc}}{p_{sc}T} \; TM_{\ell-\frac{1}{2}} \left(p^{n+1}_{\ell-1} - p^{n+1}_{\ell}\right) \tag{A-17}$$

where
$$TM_{\ell-\frac{1}{2}} = \frac{k_{\ell-1}\ k_{\ell}}{k_{\ell}\ (\mu z)_{\ell-1}\ \ell_n\ (r_{\ell}/\bar{r}_{\ell-1} + k_{\ell-1}\ (\mu z)_{\ell}\ \ell_n(\bar{r}_{\ell}/r_{\ell})} \tag{A-18}$$

In this case we have only a one-dimensional problem and it is easy to see that a tri-diagonal system is present; that is,

$$a_{\ell}\ p_{\ell-1}^{n+1} + b_{\ell}\ p_{\ell}^{n+1} + c_{\ell}\ p_{\ell+1}^{n+1} = d_{\ell} \tag{A-19}$$

where
$$a_{\ell} = TR^{*}_{\ell-\frac{1}{2}} \tag{A-20a}$$

$$b_{\ell} = -\ TR^{*}_{\ell-\frac{1}{2}} - TR^{*}_{\ell+\frac{1}{2}} - \frac{\Gamma_{\ell}}{z^{*}_{\ell}} - \frac{V_{\ell}}{\Delta t}\left(\frac{dc}{dp}\right)^{**}_{p_{\ell}} \tag{A-20b}$$

$$c_{\ell} = TR^{*}_{\ell+\frac{1}{2}} \tag{A-20c}$$

and
$$d_{\ell} = -\left[\frac{\Gamma_{\ell}}{z^{n}_{\ell}} + \frac{V_{\ell}}{\Delta t}\left(\frac{dc}{dp}\right)^{**}_{p_{\ell}}\right] p^{n}_{\ell} \tag{A-20d}$$

The source term, Q_m, in Eq. 13 may be written implicitly in terms of matrix and fracture pressures as

$$Q_m^{n+1} = TR^{n}_{L-\frac{1}{2}}\left(p^{n}_{m,\ L-1} - p^{n+1}_{m,\ L}\right) \tag{A-21}$$

To maintain pressure continuity at the matrix element boundary we must have

$$p^{n+1}_{m,\ L} = p^{n+1}_{i,j,k.} \tag{A-22}$$

APPENDIX B

PROGRAM DESCRIPTION

Introduction

SUGAR-MD is a general purpose two-dimensional reservoir simulator for gas reservoirs. It is extremely versatile in that it can be used to study fractured formations and will efficiently solve one- or two-dimensional problems in either cartesian or polar cylindrical coordinates. Although three-dimensional arrays are used in the code in its present form, all testing and debugging has been done using only two dimensions. Significant changes would be necessary (such as the determination of bottomhole pressures and/or rate allocations for multi-node wells) in order to extend the model to have complete three-dimensional capability. Hence, SUGAR-MD must be considered a 2D model and no attempt should be made to run with more than one layer in the z-direction.

Boundary conditions are completely flexible in that any desired pressure or rate (as a function of time) may be imposed at any interior or boundary block within the finite-difference grid. In the radial mode, SUGAR-MD may be used as a well simulator. For example, it may be used for history-matching well test or production data, studying the effects of dual-porosity systems, or forecasting production performance of isolated wells. In the rectangular mode, the reservoir may be virtually any shape by use of "zero permeability blocks". Complete heterogeneity of reservoir

properties is permitted in the sense that each finite-difference grid-block may have a unique porosity value as well as unique permeability values in each of the coordinate directions.

In either case, a fractured reservoir may be simulated by specifying a nonzero value of the input variable MCODE along with "matrix" porosity and permeability values and element size (fracture spacing). Also desorption of gas from pore walls of the "matrix" can be considered by inputting an appropriate desorption isotherm. A specialized non-linear least squares technique is used to develop an equation which fits the input table values in the "least squares" sense.

Hereafter, the term "matrix" will be used to denote the less permeable portion of the formation (e.g. Devonian shale) which delivers its gas contents into an existing natural fracture system. The fracture system will sometimes be denoted by "reservoir" since the model is a gas flow simulator in which all flow into wells occurs in the primary or fracture system porosity. The "matrix" acts as a uniformly distributed source within the fracture system.

Two user specified solution algorithms are available in SUGAR-MD. For most problems, LSOR (Line Successive Over-Relaxation)[14] is recommended as it is very powerful and requires very little array storage. For one-dimensional problems, specification of the acceleration parameter (OMEGA) equal to unity will give a direct solution the first iteration using the tridiagonal algorithm. For very difficult problems (e.g. LSOR converges very slowly) a direct solution algorithm (D4GM), which uses the D4 ordering scheme[15] may be used if sufficient core storage

is available. A special subroutine (STOR) is included in the code to approximate the amount of memory needed for arrays used for storage of fracture system parameters.

Description of the Code

The model consists of 25 subroutines which are invoked as needed to perform the necessary I/O operations and specialized calculation procedures. A brief description of each routine is given in Table 1.

TABLE 1 - DESCRIPTION OF SUBROUTINES IN SUGAR-MD

No.	Name	Function
1	BDRY	Determines shape of boundary by zeroing permeabilities in specified grid-blocks
2	CPRES	Calculates internal injection rates at specified nodes (variable pressure and history nodes)
3	D4SM	Solves for new pressure distribution (direct method using D4 ordering)
4	FRIC	Calculates friction factors for bottom-hole pressure calculations
5	SUGRMD	Main program - also in subroutine form to facilitate "dynamic dimensioning"
6	GRDMD	Establishes geometry and grid-block dimensions based on input data
7	IDXSMD	Allocates storage into working arrays based on user specified dimensions and calls main simulator (SUGRMD)
8	LSOR	Calculates new pressure distribution each time-step (iterates line by line in x-direction)
9	NLLSQS	Determines non-linear least squares coefficients for isotherm data
10	ORDER	Calculates parameters for D4SM regarding D4 grid-block numbering scheme

11	PARM	Establishes porosity and permeability values for each grid-block
12	PBH	Calculates bottom-hole pressure at nodes with specified wellhead pressure
13	PWH	Calculates wellhead pressure at nodes with specified rate
14	SOURCE	Establishes pressure and rate boundary conditions based on input data; fractured wells may be simulated by treating each node penetrated by the fracture separately
15	RVOL	Establishes pore volume of each node; reads parameters regarding solution technique, convergence tolerances, and output format
16	SPECN	Writes on UNIT 6 and/or 7 pressure and calculated injection rates at specified "history nodes"
17	STOR	Calculates storage requirements for "reservoir" arrays based on user specified dimensions
18	TRANS2	Calculates inter-block transmissibilities between "reservoir" nodes
19	TRNRD1	Calculates radial transmissibilities for solution of pressure distribution within "matrix" elements
20	SYMM	Solves linear system of equations with symmetric matrix (called by D4SM)
21	UINIT	Establishes initial pressure at each grid-block
22	RDATA	Reads gas property data and desorption isotherm table
23	PRTSMD	Prints reservoir pressure distribution, production rates, etc. at pre-selected time-steps
24	PCGINV	Prints summary of resource recovery at pre-selected time-steps
25	INTRP1	Linearly interpolates gas viscosity and z-factor table to update values at each "reservoir" and "matrix" node

Details of the fundamental equations used is SUG3D-MD, including a derivation of the finite-difference equations, are given in Appendix A. Appendix B1 contains a listing of all subroutines which have many comment statements to facilitate understanding and/or user modification.

Data Preparation

The code is written in FORTRAN IV and should be easily adapted to any computer system which supports FORTRAN and has sufficient speed and available memory to be practical. As described in Table 1, input data is read by various subroutines and stored for subsequent execution.

Unless otherwise specified, integer and real variables are read with I5 and F10.0 format specifications, respectively. Detailed instructions follow for preparing input data. Where the designation "Title Card" appears, the card may contain any desired information. It is intended primarily for user identification of data.

INPUT DATA FOR SUBROUTINE GRID

1. Title card

2. Reservoir dimensions(ft) ------- FORMAT(3F10.0,3I5)
   LX----------Distance in x-direction
   LY----------Distance in y-direction
   LZ----------Distance in z-direction
   KLX---------x-direction code(nonzero for radial or polar coord.)
   KLY---------y-direction code(nonzero for radial or polar coord.)
   KLZ---------z-direction code

3. Number of grid-blocks and geometric codes-----FORMAT(6I5)
   II----------Number of x-direction grid-blocks
   JJ----------Number of y-direction grid-blocks (JJ=1 for KRAD=1)

```
KK----------Number of z-direction grid-blocks (SEE REMARKS)
KRAD--------Geometric selector:
             specify "0" for rectangular coordinates;
             specify "1" for radial coordinates (r-z);
             specify "2" for polar  coordinates (r-0-z)
JJ1---------Pseudo-2D selector:
             nonzero for variable width in y-direction
             (use only when KRAD=0, KLY=0, and JJ=1)
KK1---------Pseudo-3D selector:
             nonzero for variable thickness
             (use only when KRAD=0, KLZ=0, and KK=1)

Uniform grids are obtained for the following conditions:
If KLX=0, grid-block size in the x-direction is LX/II.
If KLY=0, grid-block size in the y-direction is LY/JJ.
IF KLZ=0, grid-block size in the z-direction is LZ/KK.


4.  X-direction grid-block dimensions---FORMAT(I5,F10.0)
    I-----------Grid-block index
    DX(I)-------Grid-block size (rectangular coordinates);
                 Inner radius of grid block (polar coordinates);
                 II cards are read when KLX in nonzero & KRAD=0.
                 II+1 cards are read when KRAD is nonzero.


5.  Y-direction grid-block dimensions---FORMAT(I5,F10.0)
    J-----------Grid-block index
    DY(J)-------Grid-block size:
                 feet    - rectangular coordinates;
                 degrees - polar coordinates
                 For radial coordinates specify JJ=1 and DY(1)=360.
                 If JJ=1 & JJ1 is nonzero, specify DY(1) nonzero.
                 JJ cards are read when KLY is nonzero.


6.  Z-direction grid-blocks dimensions---FORMAT(I5,F10.0)
    K-----------Grid-block index (SEE REMARKS BELOW)
    DZ(K)-------Grid-block size:
                 If KK=1 & KK1 is nonzero, specify DZ(1) nonzero.
                 KK cards are read when KLZ is nonzero.

7.  Y-direction variation in x-direction grid-blocks---FORMAT(I5,F10.0)
    I-----------Grid-block index
    DY1(I)------Grid-block size (y-direction)
                 KK1 must be 0 for this option
                 II cards are read when JJ=1 and JJ1 is nonzero.

8.  Z-direction variation in grid-blocks-----FORMAT(I5,10F8.3)
    HZ(I,J)-----Thickness values for Jth row;
                 II values must be read; continue on additional
                 cards as necessary.  The row order must be JJ,
                 JJ-1, JJ-2, --- ,1.

    REMARKS: (a)  Option 7 is particularly useful for cross-sectional
                  or symmetry element studies

             (b)  Option 8 is invoked whenever JJ>1, KK=1, JJ1=0, and
```

KK1 is nonzero. It provides the capability for variable thickness simulations without having to incur the time and expense of a 3D run.

(c) DY(1) and DZ(1) must be nonzero for options 7 and 8.

(d) For radial or polar coordinates the x-direction becomes the r-direction; the y-direction is the O-direction for polar coordinates.

(e) Multi-layer capability is completely untested for dual porosity simulations --- I.E. SPECIFY KK = 1 UNTIL 3D CAPABILITY HAS BEEN VALIDATED.

INPUT DATA FOR SUBROUTINE PARM

1. Title card

2. Porosity and permeabiliy distributions-----FORMAT(4F10.0,8I5)
   PHIC---------Initial porosity to assign to all grid-blocks
   KXC----------Initial permeability (KX) to assign to all blocks
   KYC----------Initial permeability (KY) to assign to all blocks
   KZC----------Initial permeability (KZ) to assign to all blocks
   (FOR LATER USE WHEN MODEL IS TESTED FOR 3D)
   NUMP---------Number of nodes to alter initial porosity
   NUMKX--------Number of nodes to alter initial permeability (KX)
   NUMKY--------Number of nodes to alter initial permeability (KY)
   NUMKZ--------Number of nodes to alter initial permeability (KZ)
   KPH----------Porosity code
   KKX----------Permeability code (KX)
   KKY----------Permeability code (KY)
   KKZ----------Permeability code (KZ)
   (FOR LATER USE WHEN MODEL IS TESTED FOR 3D)

3. Special porosities-----FORMAT(3I5,F10.0)
   I-----------x-direction index
   J-----------y-direction index
   K-----------z_z-direction index
   PHI(I,J,K)--New porosity value
   NUMP cards are read.

4. Porosity distribution-----FORMAT(10F8.0)
   PHI----------Porosity distribution for Jth row.
   II values must be read; continue on additional cards as necessary. The row order must be JJ, JJ-1, JJ-2, --- 1.
   [II/10] x II cards are read for each layer.
   [a] means the least positive integer greater than a.

REMARKS: Distribution read only if KPH is nonzero and NUMP=0

5. Special permeabilities (KX) -----FORMAT(3I5,F10.0)

```
I-----------x-direction index
J-----------y-direction index
K-----------z-direction index
KX(I,J,K)---New permeability value
            NUMKX cards are read.
```

6. Permeability distribution (KX) -----FORMAT(10F8.0)

```
KX----------KX distribution for Jth row
            II values must be read; continue on additional
            cards as necessary.  The row order must be JJ,
            JJ-1, JJ-2, --- 1.
            [JJ/10] x JJ cards are read for each layer.
```

REMARKS: Distribution read only if KKX is nonzero and NUMKX=0

7. Special permeabilities (KY) -----FORMAT(3I5,F10.0)

```
I-----------x-direction index
J-----------y-direction index
K-----------z-direction index
KY(I,J,K)---New permeability value
            NUMKY cards are read.
```

8. Permeability distribution (KY) -----FORMAT(10F8.0)

```
KY----------KY distribution for Jth row
            II values must be read; continue on additional
            cards as necessary.  The row order must be JJ,
            JJ-1, JJ-2, --- , 1.
            [II/10] cards are read for each layer.
```

REMARKS: Distribution read only if KKY in nonzero and NUMKY=0

9. Special permeabilities (KZ) -----FORMAT(3I5,F10.0)
(FOR LATER USE)

```
I-----------x-direction index.
J-----------y-direction index
K-----------z-direction index
KZ(I,J,K)---New permeability value
            NUMKZ cards are read.
```

10. Permeability distribution (KZ) -----FORMAT(10F8.0)
(FOR LATER USE)

```
KZ----------KZ distribution for Jth row
            II values must be read; continue on additional
            cards as necessary.  The row order must be JJ,
            JJ-1, JJ-2, --- , 1.
            [II/10] cards are read for each layer.
```

REMARKS: Distribution read only if KKZ is nonzero and NUMKZ=0

INPUT DATA FOR SUBROUTINE BDRY

1. Title card

2. Number of "zero permeability blocks"----------FORMAT(I5)
   IBCODE------Number of blocks to have all
   transmissibilities zero
   (nonzero only for rectangular coordinates)

3. "Zero permeability nodes" -----FORMAT(4I5)
   I----------x-direction index
   J----------y-direction index
   K----------z-direction index
   IBK(I,J,K)--Specify "1" for zero permeability
   IBKODE cards are read.

INPUT DATA FOR SUBROUTINE RDATA

1. Title card

2. Parameters---------FORMAT(6F10.0)
   TR---------Reservoir temperature, deg F
   TSC--------Standard temperature, deg F
   PSC--------Standard pressure, psia
   GR---------Gas gravity
   TA---------Average wellbore temperature, deg F
   RHOSH------Density of "matrix", gm/cu cm

3. Title card

4. Parameters---------FORMAT(3F10.0)
   EE---------Tubing roughness, inches
   DFTV-------Default value for friction factor
   DPW--------Convergence tolerance for BHP/WHP calculations, psia

5. Title card

6. Number of gas property cards to be read---FORMAT(I5)
   NT1

7. Viscosity and z-factor table-----FORMAT(3F10.0)

PT1--------Pressure, psia
VIS1-------Viscosity, cp
ZT---------z-factor
NT1 cards are read.

8. Title card

9. Number of values in Isotherm Table------FORMAT(I5)
NTO

10. Desorption Isotherm Table------FORMAT(2F10.0)
PT---------Pressure, psia
CT---------Concentration, cc @ STP/gm
NTO cards will be read.

11. Title card

12. Initial guess for least squares coefficients-----FORMAT(2F10.0)
B1---------Coefficients for the following equation:
B2--------- C = B1 * P/(1 + B2*P)

REMARKS: If B1 = 0.0, no desorption will occur. If B1 is nonzero, B1 and B2 will be used as initial guesses and Subroutine NLLSQS will be used to determine (i.e., redefine) new values so that the above equation is satisfied (using the input data) in the least squares sense.

INPUT DATA FOR SUBROUTINE UINIT

1. Title card

2. Initial distribution constants -----FORMAT(F10.0,3I5)
PI----------Initial pressure assigned to all nodes
KPI----------Code
NPRES--------Number of nodes to be reassigned initial value
FCODE--------Code for determining format for reading initial distribution (if KPI is nonzero)

3. Special initial pressures (when KPI = 0) -----FORMAT(3I5,F10.0)
I----------x-direction index
J----------y-direction index
K----------z-direction index
P(I,J,K)----Initial value to reassign to node (I,J,K)

NPRES cards are read.

4. Initial parameters (if KPI = 0) -----FORMAT(F20.12,F30.12,F30.12)
   ETO---------Prior simulation timer, days
   CUMO--------Prior cumulative production, MCF
   MBEO--------Prior material balance error, %

5. Initial distribution (if KPI=0 & FCODE=0)-----FORMAT(4F19.12)
   P----------Four values are read per card beginning with row JJ. The row order must be JJ, JJ-1, JJ-2, ---, 1.
   [II/4] cards are read.

6. Initial distribution (if KPI is nonzero & FCODE is nonzero) -----FORMAT(16I5)
   P----------Sixteen (or II) values are read per card until all values are read for the Jth row; continue on additional cards as necessary. The row order must be JJ, JJ-1, JJ-2, --- , 1.
   [II/16] cards are read.

At the beginning of each run the "matrix" pressure in each reservoir grid-block is assigned the value determined in Subroutine UINIT.

INPUT DATA FOR SUBROUTINE SOURCE

1. Title card

2. Number of variable rate nodes ----- FORMAT(I5)
   NVQN

3. Title card

4. Grid-blocks with(specified) rates---FORMAT(3I5,3F10.0,2I5,2F10.0)
   QN1---------x-direction index
   QN2---------y-direction index
   QN3---------z-direction index
   QV----------Production rate (MCFD) when IP is nonzero
   PI----------Productivity Index
   PWF---------Bottomhole or wellhead pressure (depending on IVQ)
   IP----------Rate-pressure code
   IVQ---------Bottomhole/wellhead pressure code
   XWDQ--------Well depth, ft (for BHP/WHP calculation)

TIDQ---------Tubing ID, in (for BHP/WHP calculation)

5. Time schedule for variable rate nodes---FORMAT(8F10.0)

D1----------Day
H1----------Hour
M1----------Miniute
SEC1---------Second

"on time" for specified rate or pressure

D2----------Day
H2----------Hour
M2----------Minute
SEC2---------Second

"off time" for specified rate or pressure

REMARKS: Whenever "on time" < simulation time at end of step ≤ "off time", either rate QV or pressure PV is applied at the specified node (QN1,QN2,QN3). Several options are available depending on the values of PI, IP, and IVQ. These options are explained in the following table, where "S" means "specified" and "C" means "calculated".

TABLE 2 - VARIABLE RATE NODE OPTIONS

Case	PI	IP	QV	PWF	IVQ	Remarks
1	< 0.0	1	S	-	-	No BHP or WHP calc.
2	> 0.0	1	S	C	0	BHP calculated; array PWF contains calculated BHP
3	> 0.0	1	S	C	1	BHP & WHP calculated; array PWF contains BHP array DUMQ " WHP
4	> 0.0	0	C	S	0	PWF is (specified) BHP; Rate is calculated
5	> 0.0	0	C	S	1	PWF is (specified) WHP; Rate & BHP calculated; array DUMQ contains BHP

6. Title card

7. Number of variable pressure nodes-----FORMAT(I5)

8. Title card

9. Variable pressure nodes-----FORMAT(3I5,F10.0,I5,2F10.0)
   PN1----------x-direction index
   PN2----------y-direction index
   PN3----------z-direction index
   PV----------Pressure (psia)
   IVP----------Bottomhole/wellhead pressure code
   XWDP---------Well depth, ft (for BHP/WHP calculations)
   TIDP---------Tubing ID, in (for BHP/WHP calculations)

10. Time schedule for variable pressure nodes---FORMAT(8F10.0)

D1----------Day		
H1----------Hour	"on time" for	
M1----------Miniute	specified pressure	
SEC1---------Second		
D2----------Day		
H2----------Hour	"off time" for	
M2----------Minute	specified pressure	
SEC2---------Second		

REMARKS: Whenever "on time" < simulation time at end of step ≤ "off time", pressure PV is applied at the specified node (PN1,PN2,PN3). Two options are available as shown below:

Case	PV	IVP	Remarks
1	S	0	PV is (specified) BHP; Rate is calculated
2	S	0	PV is (specified) WHP; array DUMP contains calculated BHP; rate is calculated at end of time-step using DUMP & latest trans-missibilities

KLINKENBERG PARAMETERS

1. Title card

2. Klinkenberg parameters-----FORMAT(3F10.0)
   BK---------"Klinkenberg factor"
   C1---------Parameter for equation
   C2---------Parameter for equation

   REMARKS: If BK = 0.0, b=0; if BK is positive, b=BK; if BK is negative, b is determined from the equation

$$b = C1 * k ** -C2$$

   where k is the true (liquid) permeability. For nodes with different permeabilities in different directions k should perhaps be some type of average value. At this writing, k is the x-direction permeability but can easily be changed by the insertion of one or more cards in the 994 LOOP(see program listing in Appendix B).

   Since only a single value is permitted for "matrix" permeability, there can be no ambiguity in the value of b here. Of course, "matrix" permeability is usually believed to be several orders of magnitude lower that reservoir (fracture) permeability and thus effects of gas slippage will be much greater in the "matrix".

"MATRIX" ELEMENT GRID DIMENSIONS

1. Title card

2. Matrix element parameters------FORMAT(I5,F10.0)
   NRM---------Number of matrix element grid-blocks
   ARAD--------Cylindrical matrix element radius, cm

3. Title card

4. Normalized cumulative radii------FORMAT(I5,F10.0)
   J----------Matrix element grid-block index
   RAN(J)------Cumulative radius, cm

   REMARKS: A block-centered grid is used; hence the outer radius must have index NRM+1 and NRM+1 cards must be read.

5. Title card

6. Matrix element properties------FORMAT(F10.0,E11.4)

PHIM---------Porosity, fraction
KLM----------Permeability, md

MASTER CODE

1. Master code----------FORMAT(I5)
   MCODE---------(See below)

REMARKS: If MCODE = 0, all matrix element calculations are bypassed giving a 'single porosity' gas simulator.

If MCODE = 1 or 2 a normal (dual porosity) run will occur. The overall iterative procedure is accomplished using explicit rates (source term Qm in Appendix A) by material balance on a matrix element as shown below:

Qm = {[decrease in gas content over time-step] + [amount of gas desorbed over time-step]}/dt

Here dt is the time-step size and the equation holds at any reservoir node (i,j,k). Of course, if Nm is the number of matrix elements in node (i,j,k), then Qm, as given above, would have to be multiplied by Nm. It should be mentioned that the pressure distribution within the matrix element was determined to be correct by comparison with both analytical and numerical solutions.

The fracture pressure is used as a boundary pressure for the matrix element and cycles are performed until the maximum change in fracture pressure is less than ERRF psia. MCODE = 2 is equivalent to MCODE = 1 with the following exception: If all wells have not been shut in, matrix pressures are not allowed to increase with time. This was found to be necessary in cases involving fracture porosities less than 0.0025 to avoid physically unreal desorption rates due to intermediate calculated pressures slightly higher (one to two psi) than the initial pressure.

If MCODE = 3, implicit rates are used each cycle. This completely avoids any oscillation problems between "matrix" and fracture pressures by use of the following equation:

$$QSCM^{n+1}_{i,j,k} = TRMB^{n}_{i,j,k} * [\, PM^{n}_{i,j,k,L-1} - P^{n+1}_{i,j,k} \,]$$

Here, TRMB is the transmissibility between the outer

two matrix element nodes and [illegible] pressure in the second grid-block of the matrix element. [Of course, PM(i,j,k,L) = P(i,j,k)]. In this case the above equation for QSCM is used instead of explicit rates. Once convergence is obtained, the new pressures (i.e. @ time level n+1) are used both to calculate rates (using the above equation) and as boundary conditions to solve for new matrix pressures.

If MCODE = 4, implicit rates (as above) are used for all cycles. However, explicit rates are used in solving for the "matrix" pressure distributions for the first cycle to assure some pressure drawdown. Thereafter, pressure boundary conditions are used as for MCODE = 3.

MCODE = 4 is the most powerful convergence method and has been successfully used with extrememly low fracture porosities (e.g. 1.0E-08)!

INPUT DATA FOR SUBROUTINE RVOL

1. Title card

2. Print codes----------FORMAT(5I5)

3. KT1---------Every KT1th step pressure distribution is printed
   LT1---------Every LT1th step pressure distribution is written on UNIT 7 if KPU2 is nonzero
   KPR---------Solution matrix is printed every KT1th step if KPR is nonzero
   KSN---------Every KSNth step variable rate node data, variable pressure node data, and history node data are printed
   KBV---------Every KT1th step formation volume factor, viscosity, z-factor, RHS coefficient (GAMMA), and source terms (Q) are printed is KBV is nonzero

4. Title card

5. Number of "history nodes"-----FORMAT(I5)
   NSPN

6. ISP--------x-direction index
   JSP--------y-direction index
   KSP--------z-direction index

NSPN cards are read.

REMARKS: A "history node" is simply a grid-block which is of particular interest, usually because it contains a well or segment of an induced fracture. By coding a value of "1" for KSN, pressure for the current and previous step, elapsed time, and flow rate into the specified node will be printed each time-step.

Also, the distribution of pressure within the "matrix" element and the rates of flow from the element will be printed at "history nodes".

Table 3 in the next section describes all output which is not labeled on the output listing.

7. Title card

8. Punch codes--------FORMAT(3I5)
KPU-------Final pressure distribution will be written on UNIT 7 if KPU is nonzero.
KPU1------Data at "history nodes" will be written on UNIT 7 erery KSNth step is KPU1 is nonzero.
KPU2------Every LT1th step pressure distribution will be written on UNIT 7.

9. Title card

10. Solution parameters
KSOL------Solution algorithm selector;
specify "3" for LSOR (recommended)
specify "4" for D4SM (use only for very difficult problems when LSOR will not converge)

MITER-----Maximum permissible number of iterations (LSOR)
OMEGA-----Starting value of relaxation parameter (LSOR)
TOL-------Convergence tolerance, psia (LSOR)
TOL1------Parameter for determining when to change (i.e. optimize) OMEGA

REMARKS: When TOL1 is zero, the input value of OMEGA will not be changed; if optimization is desired use TOL1 < .0005.

For one-dimensional simulations set OMEGA = 1.000 and TOL1 = 0.0 to get a direct solution in one iteration.

11. Title card

12. Run time parameters-----FORMAT(I5,4F10.0,I5)
    NN--------Maximum number of time-steps per run
    TMAX-------Maximum simulation time per run
    ERR--------Cycle tolerance, psia, for calculating fracture pressure distribution (KCY LOOP)
    ERRM-------Cycle tolerance, psia, for solving for "matrix" pressure distribution (KCYM LOOP)
    ERRF-------Cycle tolerance, psia, for overall iteration procedure-both fracture and "matrix" pressures are calculated until the maximum fracture pressure change is ERRF psia.
    KPRT-------Supplementary print code; e.g. initial "matrix" gas, number of iterations for convergence in LSOR, etc. are printed if KPRT is nonzero.

    REMARKS: A simulation run may be terminated normally by either exceeding NN or TNAX.

    A suggested range for ERR & ERRM is one to ten psia. ERRF should be at least as small as 0.05. When using MCODE = 1 or 2, use ERRF = 0.001.

TIME-STEP CARDS

1. Title card

2. Time-step size cards--------FORMAT(4F10.0)
    DAY-------Step size in days, hours, minutes, and seconds
    HOUR
    MIN
    SEC

    REMARKS: As many cards as desired may be read. One card is read each time-step, if available. If N cards are read, all time-steps after step N will be the same as step N.

    For internal use, each step size in converted to days.

Description of Output

Although most SUGAR-MD output is described by appropriate labeling, some information appears without headings to conserve CPU time and space on the output listing. Specifically, Subroutine SPECN

produces unlabeled output after each KSNth step as described in Table 3. Note that UNIT 7 can be any user specified output device (e.g. tape, disk, terminal). Thus by choice of the "punch codes" defined above, pertinent information may be saved for subsequent use such as plotting, etc. The repetiton in output data such as FT and CEXPRD is intentional. It provides the capability for reading only records for a specific node and still obtaining time, rate, and cumulative production data.

TABLE 3 - DESCRIPTION OF SPECN OUTPUT

Line	Variables Printed	Format
1	FT ------Elapsed time	1x,F25.12
	DELT-----Time-step size	F25.12
	EXTINJ---External (well) injection	F25.12
	EXTPRD---External (well) production	F25.12
2	TEXPRD---Net external production this time-step	1x,F25.12
	CEXPRD---Cumulative net external production	F25.12
	PAVG-----Volumetric average P/Z	F25.12
	MBE------Cumulative material balance error	F25.12
NSPN lines	ISP------x-dir. history node index	I3
	JSP------y-dir. history node index	I3
	KSP------z-dir. history node index	I3
	P--------History node pressure	F8.2
	PN-------History node pressure previous time-step	F8.2
	FT-------Elapsed time, days	E15.6
	QSP------Internal flow rate "into" history node	E15.6

CEXPRD---Cumulative net external production E15.6

Card	Variables Written to UNIT 7	Format
1	FT, DELT, EXTINJ, EXTPRD, CEXPRD, PAVG	F12.6, F13.7, F13.3 F13.3, F16.3, F13.3
NSPN Records	ISP, JSP, KSP, P, PN, FT, QSP, CEXPRD	I3, I3, I3, F8.1, F8.1 E15.6, E15.6, E15.6

The Calling Program

To execute the 25 subroutine code SUGAR-MD, it is necessary to write a short FORTRAN calling program to allocate storage to four "working arrays." Also, each array should be initialized to either zero or one to affect proper start-up.

To allocate the proper storage requires specification of maximum dimensions required in the study to be undertaken. These dimensions are as follows:

IM------maximum number of grid-blocks in x-direction

JM------maximum number of grid-blocks in y-direction

KM------maximun number of grid-blocks in z-direction

NSPM----maximum number of variable rate nodes, variable pressure nodes, or history nodes

NTM-----maximum number of entries in gas property table

These dimensions are first passed to subroutine STOR which calculates the required dimensions of the four working arrays which are typically called WK1, WK2, WK3, and WK4. Subsequently, the calculated values are inserted in the dimension statements in the calling program. To make a simulation run, it is then only necessary to provide the required input data. A sample calling program is given below.

```
C---- CALLING PROGRAM FOR SUGAR-MD
      INTEGER*2 WK1(269)
      REAL*4    WK2(696)
      REAL*8    WK3(3186)
      INTEGER*4 WK4(306)
C
      DATA WK1/269*1/,WK2/696*0.0/,WK3/3186*0.0/,WK4/306*0/
C
      DATA IM,JM,KM,NSPM,NTM/10,10,1,5,25/
C
      DATA KSTOR,KSTOP/1,0/
C
      IF(KSTOR .NE. 0) CALL STOR(IM,JM,KM,NSPM,NTM)
      IF(KSTOP .NE. 0) STOP
C
      IMM=IM+1
      JMM=JM+1
      KMM=KM+1
C
      CALL INDEX(WK1,WK2,WK3,WK4,IM,JM,KM,IMM,JMM,KMM,NSPM,NTM)
C
      STOP
      END
```

Note that by specifying KSTOP=1, Subroutine STOR is called and execution terminated immediately upon return to the calling program. This is convenient to get dimensions for the working arrays. Then, the proper dimensions are inserted, KSTOP set to zero, and a normal run may be made.

Due to problems with the number of calling arguments permitted (63) on the METC VAX/1170 Computing System, complete dynamic dimensioning is not currently possible. This problem can be avoided by "hard coding" all arrays pertaining to variable rate nodes, variable pressure nodes, and history nodes in the main program (SUGRMD).

All arrays for calculation of "matrix" element pressures have intentionally been "hard coded". Presently, "matrix" arrays are dimensioned 10 x 10 x 1 x 10 which restricts simulations to a 10 x 10 reservoir grid with a maximum of 10 sub-elements to describe the "matrix". Hence IM, JM, and KM can not exceed 10, 10, and 1, respectively, without going into the source code (SUGRMD) and increasing one or more dimensions. However, this is a very easy task and all re-dimensioning and re-initializing can be done in a few minutes to accomodate any size "reservoir" and "matrix" grid. (As mentioned previously, the code has not been tested for three dimensions and should not be used for KM > 1 until extensive testing is conducted and any necessary changes made!)

APPENDIX B1

PROGRAM LISTINGS

```
      INTEGER*2 WK1(464)
      REAL*4    WK2(891)
      REAL*8    WK3(1798)
      INTEGER*4 WK4(1)
      DATA WK1/464*1/,WK2/891*0.0/,WK3/1798*0.0/,WK4/1*0/
      DATA IM,JM,KM,NSPM,NTM/10,10,1,20,25/
      DATA KSTOR,KSTOP/1,0/
      IF (KSTOR .NE. 0) CALL STOR(IM,JM,KM,NSPM,NTM)
      IF (KSTOP .NE. 0) STOP
      IMM=IM+1
      JMM=JM+1
      KMM=KM+1
      CALL IDXSMD(WK1,WK2,WK3,WK4,IM,JM,KM,IMM,JMM,KMM,NSPM,NTM)
      STOP
      END
      SUBROUTINE SUGRMD(IVQ,IVP,SE,MEP,IR1,JR1,
     &IBK,ISP,JSP,KSP,PFLAG,QN1,QN2,QN3,PN1,PN2,PN3
     &,IP,PV,DX,DY,DZ,
     &XWDQ,XWDP,TIDQ,TIDP,DUMQ,DUMP,PBHQ,PBHP,QWVQ,QWVP,
     &PHI,VP,KX,KY,KZ,A1,A2,A3,PI,PWF,VIS,DY1,HZ,BKLF,Q,QS,P,
     &BG,E,GAMMA,
     &TX,TY,TZ,QV,TQ1,TQ2,TP1,TP2,QSP,AZ,BZ,CZ,D1,EZ,FZ,UZ,
     &UM,PT1,VIS1,ZT,PN,ZN,PG,BBB,X,CCC,AAA,IR,JR,KR,IC,JC,KC,
     &IM,JM,KM,IMM,JMM,KMM,NSPM,NTM)
      INTEGER*2 HEADIN(40),IBK(IM,JM,1),KP(26),ISP(1),JSP(1),KSP(1)
     &,PFLAG(1),QN1(1),QN2(1),QN3(1),PN1(1),PN2(1),PN3(1),IP(1)
     &,SE(1),MEP(1),IR1(1),JR1(1),IVQ(1),IVP(1)
C
      INTEGER*4 IR(1),JR(1),KR(1),IC(1),JC(1),KC(1)
C
      REAL*4 LX,LY,LZ,PV(1),DX(1),DY(1),DZ(1),PI(1),PWF(1)
     $,KX(IM,JM,1),KY(IM,JM,1)
     &,KZ(IM,JM,1),A1(IM,JM,1),A2(IM,JM,1),A3(IM,JM,1),VIS(IM,JM,1)
     &,DY1(1),HZ(IM,1),BKLF(IM,JM,1)
     &,XWDQ(1),XWDP(1),TIDQ(1),TIDP(1),DUMQ(1),DUMP(1),PBHQ(1),
     &PBHP(1),QWVQ(1),QWVP(1)
C
      REAL*8 SUM,MCFINT,MCFCAL,CUMPRD,Q(IM,JM,1),QS(IM,JM,1)
     &,P(IM,JM,1),BG(IM,JM,1),E(IM,JM,1),GAMMA(IM,JM,1)
     &,TX(IMM,JM,1),TY(IM,JMM,1),TZ(IM,JM,1),TOTPRD,OMEGA,SM,PAVG,SMN,
     &RESVOL,MBE,MBEO,CUMO,MCFACT,ETI,ETO,DELT,FT,DAY,HOUR,
     &MIN,SEC,TQ1(1),TQ2(1),QV(1),TP1(1),TP2(1),TOL,TOL1,DW,QSP(1)
     &,DELTO,CON,CON1,POZAVG,NSC,IMBE,ZZ,VS
     &,UM(1),AZ(1),BZ(1),CZ(1),D1(1),EZ(1),FZ(1),UZ(1),DIV,DIVO
     &,BBB(1),X(1),CCC(1),AAA(1),PHI(IM,JM,1),VP(IM,JM,1)
     &,PT1(1),VIS1(1),ZT(1),PN(IM,JM,1),ZN(IM,JM,1),PG(IM,JM,1)
```

```
     &,PPP,A11,A22,QQ,PBH1,DPW,PBH2,GR,XWDQJ,XWDPJ,EE,RE,F,DFTV
     &,C11,C22
C
C@@@@ ARRAYS AND VARIABLES FOR MATRIX (DUAL-POROSITY)
C
      REAL*8 KLM,KGM(10),MCFDI,MCFMA,TPI(10,10,1),
     &NE(10,10,1),INJN,MCFDA,INJ(10,10,1),CUMINJ,CUMDSP,
     &GENQ,MCFMI,RAN(11),RA(11),PM(10,10,1,10),SQ,QSC(10,10,1),
     &PMN(10,10,1,10),VISM(10),ZM(10,10,1,10),
     &ZMN(10,10,1,10),TRM(11),GAMJ(10,10,1,10),AZM(10),BZM(10),CZM(10),
     &DZM(10),TRMB(10,10,1),A(2,2),FF(2,50),BB(2),QSCD(10,10,1),
     &GASM1(10,10,1),PRD(10,10,1),POLD(10,10,1),
     &GMCAL(10,10,1),GMN(10,10,1),EZM(11),FZM(11),UZM(10)
     &,PT(50),CT(50),YE(50),DEV(50),DUM(50),DUM1(50),DUM2(50),
     &QSCM(10,10,1),QQQ(10,10,1),PF(10,10,1)
     &,GSM,QDD,AMT,STAM,PP,VOLM(10,10,1,10),VPM(10,10,1,10),
     &RVOLM(10,10,1),BGMM,GASM,TGAS,FGF,FGM,FGD,SQSC,SQSCM
     &,FGR,TGAS1,PPM,ZZM,QWELL(10,10,1),EXTINJ,EXTPRD,TEXPRD,CEXPRD
     &,GPZ,STAT,SQWELL,SCM,SQSCD,DPF,ERRF
C
C*****THIS IS A GENERAL PURPOSE MODEL  FOR SIMULATING THREE DIMENSIONAL
C     FLOW OF A COMPRESSIBLE GAS IN  HETEROGENEOUS, ANISOTROPIC,
C     FRACTURED GAS RESERVOIR
C
C@@@@ INITIALIZATION OF VARIABLES AND ARRAYS FOR MATRIX ELEMENT
      DATA TRM,GAMJ,CUMINJ,CUMDSP/11*0.0,1000*0.0,0.0,0.0/
      DATA QSCM,RVOLM,MCFMI,MCFDI,MCFMA/100*0.0,100*0.0,0.0,0.0,0.0/
      DATA NRMMAX/10/
C
      DATA MBEO,CUMO,ETO/0.0,0.0,0.0/,ETI,CUMPRD,IND/0.0,0.0,0/
     &,MCFCAL/0.0/,CEXPRD/0.0/
C
      FK(X,A,B)=A*X**B
C@@@@ FUNCTIONS FOR DESORPTION
      FC(X,B1,B2) = B1*X/(1. + B2*X)
      DFDX(X,Y,Z) = Y/((1. + Z*X) * (1. +Z*X))
C
      CALL TITLE2
C
C*****ESTABLISH RESERVOIR LENGTH,WIDTH,AND THICKNESS AND GRID
C     BLOCK SIZES
C
      CALL GRDRD(IM,II,JJ,KK,JJ1,KK1,LX,LY,LZ,DX,DY,DZ,DY1,HZ,KRAD,
     &STHETA)
C
C*****ESTABLISH POROSITY & PERMEABILITY DISTRIBUTIONS
C
      CALL PARM(II,JJ,KK,PHI,KX,KY,KZ,IM,JM)
C
      WRITE(6,56)
C
C*****ESTABLISH IRREGULARITIES IN RESERVOIR BOUNDARY
```

```
C
      CALL BDRY(IM,JM,IBK,IBKODE)
C
C*****ESTABLISH GAS PROPERTY DATA (VISCOSITY & Z-FACTORS)
C**** READ PARAMETERS FOR BOTTOMHOLE PRESSURE CALCULATIONS
C@@@@ READ ISOTHERM TABLE
C
      CALL RDATA(TR,TSC,PSC,GR,TA,RHOSH,C11,C22,EE,DFTV,
     &DPW,PARMM,NT1,PT1,VIS1,ZT,NTM,NTO,PT,CT,B1,B2)
      BB(1)=B1
      BB(2)=B2
      IK=2
      IF(B1.GT.1.E-05)
     &CALL NLLSQS(IK,NTO,PT,CT,A,BB,CZ,FF,EZ,YE,DEV,DUM,DUM1)
      B1=BB(1)
      B2=BB(2)
C
C
      CON=PSC*TR*1000./TSC
      CON1=1./CON
      STAM=3.14159 * .006328 * CON1
C
C*****ESTABLISH INITIAL PRESSURE DISTRIBUTION
C
      CALL UINIT(II,JJ,KK,PI,PN,P,IM,JM,MBEO,CUMO,ETO)
C
      IF(ETO.GT.0.0) ETI=ETO
      IF(ETO.GT.0.0) CUMPRD=CUMO
C
C
C*****ESTABLISH RATE & PRESSURE NODES
C
      CALL SOURCE(P,KX,KY,KZ,IM,JM,NVQN,QN1,QN2,QN3,
     &TQ1,TQ2,QV,NVPN,PN1,PN2,PN3,TP1,TP2,PV,PN,PI,PWF,IP,
     &IVQ,IVP,XWDQ,XWDP,TIDQ,TIDP)
C
C
C
C*****READ KLINKENBERG FACTOR
C
      READ(5,69) HEADIN
      READ(5,1)BK,C1,C2
      IF(BK.LT.0.0) WRITE(6,3109) C1,C2
      IF(BK.GE.0.0) WRITE(6,3111) BK
C
      DO 994 K=1,KK
      DO 994 J=1,JJ
      DO 994 I=1,II
      XK=KX(I,J,K)
      IF(BK.LT.0.0) BKLF(I,J,K)=FK(XK,C1,C2)
      IF(BK.GE.0.0) BKLF(I,J,K)=BK
994   CONTINUE
```

```
C
C@@@@ ESTABLISH MATRIX ELEMENT GRID DIMENSIONS
      READ(5,69) HEADIN
      READ(5,3) NRM,ARAD
      NRM1=NRM+1
      WRITE(6,19) ARAD
C@@@@ READ NORMALIZED CUMULATIVE RADII
      READ(5,69) HEADIN
      READ(5,3) (J,RAN(J),J=1,NRM1)
      DO 120 J=1,NRM1
120   RA(J)=RAN(J)*ARAD
      WRITE(6,21)
      WRITE(6,23) (J,RAN(J),RA(J),J=1,NRM1)
C@@@@ NOW CONVERT TO FEET
      DO 126 J=1,NRM1
126   RA(J)=RA(J)/30.48
      ARAD=ARAD/30.48
      ARAD2=ARAD*ARAD
C
      DO 1266 L=1,NRM
1266  DUM(L) = 3.14159 * (RA(L+1)*RA(L+1) - RA(L)*RA(L))
C
C@@@@ READ MATRIX GRID DATA (PHIM & KLM)
      READ(5,69) HEADIN
      READ(5,1377) PHIM,KLM
      WRITE(6,1777)PHIM,KLM
C
      IF(BK.LT.0.0) BKL = FK(KLM,C1,C2)
      IF(BK.GE.0.0) BKL = BK
C
C@@@@ READ MASTER CODE (IF MCODE = 0, ALL MATRIX CALCULATIONS
C                       WILL BE BYPASSED GIVING A SINGLE POROSITY
C                       MODEL)
      READ(5,69) HEADIN
      READ(5,2) MCODE
      WRITE(6,3115) MCODE
C
C*****ESTABLISH RESERVOIR VOLUME & INITIAL GAS IN PLACE
C
      CALL RVOL(RESVOL,KX,DY,DZ,VIS,KY,KZ,BG,KSOL,MITER,VP,
     &P,PHI,DX,IBK,SM,MCFINT,KT1,LT1,II,JJ,KK,JJ1,KK1,IM,JM,KRAD,
     &KPR,KBV,LT,KT,ISP,JSP,KSP,NSPN,NTM,KPU,KPU1,KPU2,KSN,KS,
     &TMAX,OMEGA,TOL,TOL1,DW,NN,CON,PT1,VIS1,ZT,DY1,HZ,ERR,NT1,
     &ERRM,ERRF,KPRT)
      TGAS = MCFINT
C
      IF(MCODE.EQ.0) GO TO 258
C@@@@ INITIALIZE MATRIX PRESSURES & Z-FACTORS;
C@@@@ CALCULATE NUMBER OF MATRIX ELEMENTS IN EACH GRID-BLOCK;
C@@@@ CALCULATE ADSORBED GAS CONTENT
      DO 218 K=1,KK
      DO 218 J=1,JJ
```

```
      DO 218 I=1,II
      PF(I,J,K) = P(I,J,K)
      QQQ(I,J,K)=0.0
      QSCM(I,J,K)=0.0
      TRMB(I,J,K)=0.0
      GMCAL(I,J,K) = 0.0
      PP=P(I,J,K)
      CALL INTRP1(PT1,ZT,NT1,PP,ZZ)
      DELZ=DZ(K)
      IF(JJ1.EQ.0 .AND. KK1.NE.0) DELZ = HZ(I,J)
      TPI(I,J,K) = STAM * DELZ
      VOL=VP(I,J,K)/PHI(I,J,K)
      NE(I,J,K) = VOL * (1. - PHI(I,J,K))/(3.14159*ARAD2*DELZ)
      IF(NE(I,J,K).LT.1.0) NE(I,J,K) = 0.0
      GMD=.001*(1-PHI(I,J,K)) * VOL *B1 * PP/(1. +B2*PP)
      MCFDI=MCFDI + GMD
      DO 218 L=1,NRM
      VOLM(I,J,K,L) = DUM(L) * DELZ
      VPM(I,J,K,L)  = PHIM   * VOLM(I,J,K,L)
      RVOLM(I,J,K) = RVOLM(I,J,K) + VPM(I,J,K,L)
      ZM(I,J,K,L)=ZZ
      PM(I,J,K,L)=P(I,J,K)
218   CONTINUE
C
C@@@@ CALCULATE FREE MATRIX GAS CONTENT
C     ----- ASSUME INITIAL MATRIX PRESSURE = LOCAL FRACTURE PRESSURE
      IF(KPRT.NE.0) WRITE(6,117)
      DO 256 K=1,KK
      DO 256 J=1,JJ
      DO 256 I=1,II
      PP=P(I,J,K)
      CALL INTRP1(PT1,ZT,NT1,PP,ZZ)
      BGMM = CON * ZZ/PP
      GASM = RVOLM(I,J,K)/BGMM
      GASM1(I,J,K)= GASM * NE(I,J,K)
      MCFMI= MCFMI + GASM1(I,J,K)
      IF(KPRT.NE.0)WRITE(6,111) I,J,K,BGMM,GASM,NE(I,J,K),GASM1(I,J,K)
     &,MCFMI
256   CONTINUE
C
C@@@@ PRINT INITIAL INVENTORY OF GAS IN PLACE
      WRITE(6,3123)
      WRITE(6,3117)
      WRITE(6,3119) MCFINT,MCFMI,MCFDI
      TGAS = MCFINT + MCFMI + MCFDI
      FGF = MCFINT/TGAS * 100.
      FGM = MCFMI/TGAS * 100.
      FGD = MCFDI/TGAS * 100.
      WRITE(6,3121) TGAS,FGF,FGM,FGD
      WRITE(6,3123)
C
C
```

```
258   CONTINUE
      DO 995 K=1,KK
      DO 995 J=1,JJ
      DO 995 I=1,II
      QSC(I,J,K) = 0.0
      VMOD=1. + BKLF(I,J,K)/P(I,J,K)
      VIS(I,J,K)=VIS(I,J,K)/VMOD
995   CONTINUE
C
      READ(5,69) HEADIN
C
C*****THE FOLLOWING LOOP CALCULATES A NEW RESERVOIR PRESSURE
C     DISTRIBUTION FOR EACH TIME STEP
C
      NRM1=NRM-1
C
      DO 1000 N=1,NN
      IF(N.EQ.1.OR.N.EQ.NN.OR.N.EQ.KT)WRITE(6,9) N
      ISH=1
      IPP=0
      IPPP=0
      KCY=0
      IF(N.GT.1)DELTO=DELT
      IF(N.GT.1)ETI=ETI+DELT
      IF(IND.EQ.0)READ(5,4,END=2049) DAY,HOUR,MIN,SEC
      IF(IND.EQ.0)DELT = DAY + HOUR/24. + MIN/1440. + SEC/86400.
      IF(N.EQ.1)DELTO=DELT
1049  FT=ETI+DELT
      IF(ETI+DELT*.5.GE.TMAX)GO TO 1001
      DIV=1.0/DELT
      DIVO=1.0/DELTO
C
C
C**** STORE 'N-LEVEL VALUES' AND INITIALIZE SOURCE TERMS
C
C
      DO 70 J=1,JJ
      DO 70 I=1,II
      DO 70 K=1,KK
      QWELL(I,J,K) = 0.0
      Q(I,J,K)=0.0
      QSCD(I,J,K)=0.0
      GAMMA(I,J,K)=CON1*VP(I,J,K)*DIV
      PN(I,J,K)=P(I,J,K)
      PP=P(I,J,K)
      CALL INTRP1(PT1,ZT,NT1,PP,ZZ)
      ZN(I,J,K)=ZZ
70    CONTINUE
C
      IF(MCODE.EQ.0) GO TO 560
C
        DO 292 K=1,KK
```

```
      DO 292 J=1,JJ
      DO 292 I=1,II
      DO 292 L=1,NRM
       GAMJ(I,J,K,L)=CON1 * VPM(I,J,K,L) * DIV
      PMN(I,J,K,L) = PM(I,J,K,L)
292   ZMN(I,J,K,L) = ZM(I,J,K,L)
C
C
      KCYPF = 0
5400  KCYPF = KCYPF + 1
      IF(KCYPF.GT.99) WRITE(6,4707) DPF
      IF(KCYPF.GT.99) STOP
      IF(KCYPF.EQ.1.AND.MCODE.LE.2) GO TO 560
C
C@@@@ STORE FRACTURE PRESSURES & RATES(QSC) FROM LAST CYCLE
C     FOR MCODE .LE. 2
      DO 5410 K=1,KK
      DO 5410 J=1,JJ
      DO 5410 I=1,II
      IF(MCODE.GT.2) GO TO 5410
      QQQ(I,J,K)=QSC(I,J,K)
      POLD(I,J,K)=PF(I,J,K)
      P(I,J,K)=0.5 * (PF(I,J,K) + P(I,J,K))
C     IF(KCYPF.GT.10) P(I,J,K) = 0.5*(P(I,J,K) + POLD(I,J,K))
C@@@@ THIS GAVE GREATER RELIABILITY OF CONVERGENCE FOR DIFFICULT
C@@@@ PROBLEMS IN THE EARLY DEVELOPMENT OF THE RADIAL MODEL(SUGAR)
C@@@@ BUT ACTUALLY SLOWS CONVERGENCE ON MOST PROBLEMS; HENCE IT
C@@@@ SHOULD NOT BE USED UNLESS PROBLEM CANNOT BE SOLVED WITH
C@@@@ MCODE = 3 OR 4 AND WILL NOT CONVERGE WITH MCODE = 2
5410  PF(I,J,K)=P(I,J,K)
C
C@@@@ RESET MATRIX PRESSURES
      DO 5420 K=1,KK
      DO 5420 J=1,JJ
      DO 5420 I=1,II
      DO 5420 L=1,NRM
5420  PM(I,J,K,L)=PMN(I,J,K,L)
C
C@@@@ CALCULATE NEW MATRIX PRESSURES & GAS INFLUX IN EACH RESERVOIR
C     GRID-BLOCK
      DO 3100 K=1,KK
      DO 3100 J=1,JJ
      DO 3100 I=1,II
      IF(NE(I,J,K).LT.1.E-05) GO TO 3100
      KCYM = 0
      IF(KCYPF.EQ.1.AND.MCODE.EQ.3) GO TO 5602
      IF(KCYPF.LE.2.AND.MCODE.EQ.4) GO TO 5602
      AZM(NRM)=0.0
      BZM(NRM)=1.0
      CZM(NRM)=0.0
      DZM(NRM)=P(I,J,K)
5600  KCYM=KCYM+1
```

```
5602    CONTINUE
C
C@@@@   FIRST CALCULATE KLINKENBERG PERMEABILITIES
        DO 2998 L=1,NRM
2998    KGM(L) = KLM * (1. + BKL/PM(I,J,K,L))
C
C@@@@   NOW CALCULATE MATRIX TRANSMISSIBILITIES
        DO 3066 L=1,NRM
        PP=PM(I,J,K,L)
        CALL INTRP1(PT1,ZT,NT1,PP,ZZ)
        CALL INTRP1(PT1,VIS1,NT1,PP,VS)
        ZM(I,J,K,L)=ZZ
        DUM1(L)=ZZ
        VISM(L)=VS
        DUM(L)=PP
3066    CONTINUE
C
        CALL TRNRD1(IM,JM,NRM,DUM,TRM,KGM,RA,DUM1,VISM,TPI,I,J,K)
C
        TRMB(I,J,K)=TRM(NRM)
        IF(MCODE.LE.2) GO TO 3108
        IF(KCYPF.EQ.1) TRMB(I,J,K) = TRMB(I,J,K) * PARMM
        IF(KCYPF.EQ.1) GO TO 3100
C
C@@@@   DETERMINE ISOTHERM SLOPE FOR EACH MATRIX ELEMENT NODE
C
3108    CONTINUE
C
        DO 30100 L=1,NRM
        IF(B1.LE.1.E-05) DUM1(L) = 0.0
        IF(B1.LE.1.E-05) GO TO 30100
        PPM = 0.5 * (PM(I,J,K,L) + PMN(I,J,K,L))
        DUM(L) = DFDX(PPM,B1,B2)
        DUM1(L) = .001*VOLM(I,J,K,L)*DIV*DUM(L)
30100   CONTINUE
C
C@@@@   SET UP TRIDIAGONAL MATRIX USING RESERVOIR NODE PRESSURE AS
C       BOUNDARY FOR MATRIX ELEMENT
        DO 3012 L=1,NRM1
        AZM(L)=TRM(L)
        CZM(L)=TRM(L+1)
        BZM(L)=-AZM(L)-CZM(L)-GAMJ(I,J,K,L)/ZM(I,J,K,L) - DUM1(L)
3012    DZM(L)=-(GAMJ(I,J,K,L)/ZMN(I,J,K,L) + DUM1(L)) * PMN(I,J,K,L)
C
        IF(MCODE.LE.2) GO TO 4014
        IF(KCYPF.GT.2 .OR. MCODE.NE.4) GO TO 4014
C
        AZM(NRM)=TRM(NRM)
        CZM(NRM)=0.0
        BZM(NRM)=-AZM(NRM)-GAMJ(I,J,K,NRM)/ZM(I,J,K,NRM) - DUM1(NRM)
        QELEM=QSCM(I,J,K)/NE(I,J,K)
        DZM(NRM)=QELEM-(GAMJ(I,J,K,NRM)/ZMN(I,J,K,NRM)+DUM1(NRM))*
```

```
     &PMN(I,J,K,NRM)
C
C
C@@@@   STORE LATEST PRESSURES & SOLVE FOR NEW MATRIX ELEMENT PRESSURES
4014    DO 3014 L=1,NRM
3014    DUM2(L)=PM(I,J,K,L)
        CALL LTRI(NRM,AZM,BZM,CZM,DZM,EZM,FZM,UZM)
C
C
C@@@@   CHECK FOR CONVERGENCE
        DP=DABS(UZM(1) - DUM2(1))
        PM(I,J,K,1)=UZM(1)
        DO 3030 L=2,NRM
        DPM=DABS(UZM(L) - DUM2(L))
        IF(DPM.GT.DP) DP=DPM
        PM(I,J,K,L)=UZM(L)
3030    CONTINUE
C
        IF(KCYM.GT.99 .OR. UZM(NRM).LT.0.0) WRITE(6,9994) KCYM,UZM(NRM)
        IF(KCYM.GT.99 .OR. UZM(NRM).LT.0.0) STOP
9994    FORMAT(T15,'KCYM = ',I3,5X,'UZM(NRM) = ',E15.6,
     & /T5,'!!!!!!!!!! RUN TERMINATED !!!!!!!!!!!')
        IF(DP.LT.ERRM) GO TO 3044
        GO TO 5600
C
3044    CONTINUE
C
C@@@@   NOTE THAT FOLLOWING LOOP USED ONLY BEFORE SHUT-IN
        IF(ISH.EQ.0) GO TO 4056
        DO 4054 L=1,NRM
        IF(MCODE.EQ.2.AND.ISH.EQ.1.AND.PM(I,J,K,L).GT.PMN(I,J,K,L))
     &  PM(I,J,K,L) = PMN(I,J,K,L)
4054    CONTINUE
C
4056    IF(B1.LE.1.E-05) GO TO 3056
        QDD=0.0
        DO 3054 L=1,NRM
        PPM=PM(I,J,K,L)
        PPMN=PMN(I,J,K,L)
        AMT = VOLM(I,J,K,L) * (FC(PPMN,B1,B2) - FC(PPM,B1,B2))
3054    QDD = QDD + AMT
        QSCD(I,J,K) = .001*DIV*QDD*NE(I,J,K)
3056    CONTINUE
C
        IF(KCYPF.EQ.2) GMN(I,J,K) = GMCAL(I,J,K)
        IF(N.EQ.1)     GMN(I,J,K) = GASM1(I,J,K)
        GSM=0.0
        DO 4606 L=1,NRM
        PPM = PM(I,J,K,L)
        CALL INTRP1(PT1,ZT,NT1,PPM,ZZM)
        BGM = CON*ZZM/PPM
4606    GSM = GSM + VPM(I,J,K,L)/BGM
```

```
        GMCAL(I,J,K) = GSM*NE(I,J,K)
        INJ(I,J,K)   = QSCD(I,J,K)*DELT
        PRD(I,J,K)   = GMN(I,J,K) - GMCAL(I,J,K) + INJ(I,J,K)
        QSC(I,J,K) = -PRD(I,J,K) * DIV
        IF(MCODE.GT.2) GO TO 3100
        IF(KCYPF.GT.5)  QSC(I,J,K) = 0.5*(QSC(I,J,K) + QQQ(I,J,K))
        IF(KCYPF.GT.15) QSC(I,J,K) = 0.5*(QSC(I,J,K) + QQQ(I,J,K))
3100    CONTINUE
C
C       -------------------- 5 6 0  L O O P  S T A R T S  H E R E -------
560    KCY=KCY+1
       IF(KCY.GT.20) WRITE(6,3017)
       IF(KCY.GT.20) STOP
3017    FORMAT(T15,'!!! RUN TERMINATED -- KCY LOOP NOT CONVERGING !!!')
C
C**** CALCULATE INTER-BLOCK TRANSMISSIBILITIES & STORE PRESSURES
       DO 310 K=1,KK
       DO 310 J=1,JJ
       DO 310 I=1,II
       PG(I,J,K)=P(I,J,K)
       PP=P(I,J,K)
       CALL INTRP1(PT1,ZT,NT1,PP,ZZ)
       CALL INTRP1(PT1,VIS1,NT1,PP,VS)
       BG(I,J,K)=CON*ZZ/P(I,J,K)
       VMOD=1. + BKLF(I,J,K)/P(I,J,K)
       VIS(I,J,K)=VS/VMOD
310    CONTINUE
C
       CALL TRANS2(II,JJ,KK,IM,JM,TX,TY,TZ,A1,A2,A3,DX,DY,DZ,
      &BG,IBK,IBKODE,IMM,JMM,VIS,KRAD,STHETA)
C
C*****CALCULATE MAIN DIAGONAL AND RHS VECTOR
       DO 80 J=1,JJ
       DO 80 I=1,II
       DO 80 K=1,KK
       IF(MCODE.GT.2) STAT = TRMB(I,J,K) *NE(I,J,K)
       PP=P(I,J,K)
       CALL INTRP1(PT1,ZT,NT1,PP,ZZ)
       SUM=TZ(I,J,K)+TY(I,J,K)+TX(I,J,K)+TX(I+1,J,K)+TY(I,J+1,K)+TZ(
      &I,J,K+1)
       E(I,J,K)=-SUM-GAMMA(I,J,K)/ZZ
       IF(MCODE.GT.2) E(I,J,K) = E(I,J,K) - STAT
       GPZ = -GAMMA(I,J,K)*PN(I,J,K)/ZN(I,J,K)
       IF(MCODE.LE.2) QS(I,J,K) = GPZ + QSC(I,J,K)
       IF(MCODE.GT.2) QS(I,J,K) = GPZ - STAT*PM(I,J,K,NRM1)
80     CONTINUE
C
C---   ******** VARIABLE RATE NODE LOGIC *********
C---   MODIFY MATRIX ELEMENTS ACCORDING TO SPECIFIED WELL PRODUCTION
C---   RATES OR PRESSURES (BOTH NON-FRACTURED AND FRACTURED WELLS MAY
C---   BE HANDLED; SEE USER'S MANUAL FOR DETAILS)
       IF(NVQN.EQ.0)GO TO 2028
```

```
      DO 2029 J=1,NVQN
2029  QTOT = DABS(QV(J))
      IPP=0
      DO 2027 J=1,NVQN
      DUMQ(J)=PWF(J)
      QWVQ(J)=QV(J)
      IF(FT.LE.TQ1(J).OR.FT.GT.TQ2(J)) GO TO 2027
      IPP = 1
      JQ1 = QN1(J)
      JQ2 = QN2(J)
      JQ3 = QN3(J)
      IF(IP(J) .EQ. 0) GO TO 2025
      IF(KCY.GT.1) GO TO 2023
      QWELL(JQ1,JQ2,JQ3) = QV(J)
 2023 QS(JQ1,JQ2,JQ3) = QS(JQ1,JQ2,JQ3) + QV(J)
      GO TO 2027
2025  FAC=PI(J)/(VIS(JQ1,JQ2,JQ3)*BG(JQ1,JQ2,JQ3))
      E(JQ1,JQ2,JQ3) = E(JQ1,JQ2,JQ3) - FAC
      IF(IVQ(J).EQ.0. OR .KCY.EQ.1) GO TO 2026
C
C.....CALCULATE BOTTOM-HOLE PRESSURE
C.... NOTE: IN THIS CASE "PWF" IS SPECIFIED WELLHEAD PRESSURE!!
      XWDQJ=XWDQ(J)
      QQ=DABS(FAC*(P(JQ1,JQ2,JQ3) - DUMQ(J)))
      VS=VIS(JQ1,JQ2,JQ3)
      CALL FRIC(QQ,GR,VS,XWDQJ,EE,RE,F,IF,DFTV)
      A11=C11*XWDQ(J)/TIDQ(J) ** 5
      A22=C22*XWDQ(J)
      PBH1=PWF(J)
      CALL PBH(A11,A22,QQ,F,PBH1,DPW,PT1,ZT,NT1,PBH2)
      DUMQ(J) = PBH2
2026  QS(JQ1,JQ2,JQ3)=QS(JQ1,JQ2,JQ3) - FAC*DUMQ(J)
2027  CONTINUE
2028  CONTINUE
C
      IF(N.NE.1.AND.N.NE.KT.AND.N.NE.NN) GO TO 352
      IF(KPR.EQ.0) GO TO 352
      WRITE(6,16)
      WRITE(6,17)(((I,J,K,TZ(I,J,K),TY(I,J,K),TX(I,J,K),E(I,J,K),
     $TX(I+1,J,K),TY(I,J+1,K),TZ(I,J,K+1),QS(I,J,K),I=1,II),J=1,JJ),
     $K=1,KK)
      WRITE(6,56)
352   CONTINUE
C####     ##### VARIABLE PRESSURE NODE LOGIC (BHP CALCULATIONS)#####
C###  CALCULATE BHP'S AT VARIABLE PRESSURE NODES
      IF(NVPN.EQ.0) GO TO 3522
      IPPP=0
      DO 3027 J=1,NVPN
      DUMP(J) = PV(J)
      IF(FT.LE.TP1(J). OR. FT.GT.TP2(J)) GO TO 3027
      IPPP=1
      IF(IVP(J).EQ.0 .OR. KCY.EQ.1) GO TO 3027
```

```
C
C.....CALCULATE BOTTOM-HOLE PRESSURE
      XWDPJ=XWDP(J)
      JP1=PN1(J)
      JP2=PN2(J)
      JP3=PN3(J)
      PPP = P(JP1,JP2,JP3)
      QQ = TX(JP1,JP2,JP3) * (P(JP1-1,JP2,JP3) - PPP) +
     &     TX(JP1+1,JP2,JP3)*(P(JP1+1,JP2,JP3) - PPP) +
     &     TY(JP1,JP2,JP3) * (P(JP1,JP2-1,JP3) - PPP) +
     &     TY(JP1,JP2+1,JP3)*(P(JP1,JP2+1,JP3) - PPP) +
     &     TZ(JP1,JP2,JP3) * (P(JP1,JP2,JP3-1) - PPP) +
     &     TZ(JP1,JP2,JP3+1)*(P(JP1,JP2,JP3+1) - PPP)
      QQ=DABS(QQ)
      VS=VIS(JP1,JP2,JP3)
      CALL FRIC(QQ,GR,VS,XWDPJ,EE,RE,F,IF,DFTV)
      A11=C11*XWDP(J)/TIDP(J) ** 5
      A22=C22*XWDP(J)
      PBH1=PV(J)
      CALL PBH(A11,A22,QQ,F,PBH1,DPW,PT1,ZT,NT1,PBH2)
      DUMP(J) = PBH2
3027  CONTINUE
C
3522  IF(IPP.EQ.0 .AND. (IPPP.EQ.0.OR.QTOT.LE.1.E-04))ISH=0
C
      IF(KSOL.EQ.3)
     &CALL LSOR(II,JJ,KK,IM,JM,IMM,JMM,TX,TY,TZ,E,QS,OMEGA,TOL,TOL1,
     &NITER,MITER,DELT,DELTO,P,PN,UM,AZ,BZ,CZ,D1,EZ,FZ,UZ,
     &NVPN,TP1,TP2,PN1,PN2,PN3,DUMP,PFLAG,FT,IBK,KRAD,STHETA,KT,N,KPRT)
C
      IF(KSOL.EQ.4)
     &CALL D4SM(N,II,JJ,KK,IM,JM,IMM,JMM,TX,TY,TZ,E,QS,P,
     &SE,MEP,IR1,JR1,IR,JR,KR,IC,JC,KC,AAA,BBB,X,CCC,
     &NVPN,PN1,PN2,PN3,TP1,TP2,DUMP,PFLAG,FT,IBK,KT)
C
C**** CHECK FOR CONVERGENCE
      DP=DABS(P(1,1,1) - PG(1,1,1))
      IF(N.EQ.1.AND.KPR.NE.0)WRITE(6,3013)
      DO 440 K=1,KK
      DO 440 J=1,JJ
      DO 440 I=1,II
      IF(MCODE.EQ.0) GENQ = QWELL(I,J,K)
      IF(MCODE.GE.1) GENQ = QSC(I,J,K)
      IF(MCODE.GE.3) GENQ = -QSCM(I,J,K)
      DM=DABS(P(I,J,K) - PG(I,J,K))
      IF(DM.GT.DP) DP=DM
      IF(N.EQ.1.AND.KPR.NE.0)WRITE(6,3009)I,J,K,GAMMA(I,J,K),PG(I,J,K),
     &P(I,J,K),VIS(I,J,K),ZN(I,J,K),PN(I,J,K),GENQ
440   CONTINUE
C
      IF(DP.LT.ERR) GO TO 460
      GO TO 560
```

```
C       -------------------- 5 6 0  L O O P   E N D S   H E R E -------
460    IF(KPRT.EQ.0) GO TO 462
       IF(N.EQ.1.OR.N.EQ.NN.OR.N.EQ.KT) WRITE(6,161) KCY,N,DP
161    FORMAT(T2,'***** KCY CYCLE ',I3,'   FOR TIME-STEP',
      &I4,' MAXIMUM PRESSURE CHANGE = ',E15.6)
462    IF(KSOL.EQ.4.AND.NVPN.NE.0)
      &CALL TRANS2(II,JJ,KK,IM,JM,TX,TY,TZ,A1,A2,A3,DX,DY,DZ,
      &BG,IBK,IBKODE,IMM,JMM,VIS,KRAD,STHETA)
C
C@@@@ CALCULATE QSCM'S USING OUTER MATRIX TRANSMISSIBILITIES
       IF(MCODE.EQ.0) GO TO 4612
       DO 4614 K=1,KK
       DO 4614 J=1,JJ
       DO 4614 I=1,II
4614   QSCM(I,J,K) = TRMB(I,J,K)*NE(I,J,K)*(PM(I,J,K,NRM1)-P(I,J,K))
       KCY = 0
       IF(KCYPF.EQ.1) GO TO 5400
C@@@@ CHECK ON CONVERGENCE OF FRACTURE PRESSURES
       DPF = DABS(P(1,1,1) - PF(1,1,1))
       DO 4610 K=1,KK
       DO 4610 J=1,JJ
       DO 4610 I=1,II
       DM = DABS(P(I,J,K) - PF(I,J,K))
       IF(DM.GT.DPF) DPF = DM
4610   CONTINUE
C
       IF(DPF.LT.ERRF) GO TO 4612
       GO TO 5400
4612   CONTINUE
C---    ******** VARIABLE RATE NODE LOGIC ********
C---   CALCULATE(IMPLICIT) RATES OR PRESSURES AT VARIABLE RATE NODES
       IF(NVQN.EQ.0 .OR. IPP.EQ.0) GO TO 2054
       DO 2050 J=1,NVQN
       IF(FT.LE.TQ1(J) .OR. FT.GT.TQ2(J)) GO TO 2050
       IF(PI(J).LE.0.0) GO TO 2050
       JQ1 = QN1(J)
       JQ2 = QN2(J)
       JQ3 = QN3(J)
       FAC = PI(J)/(VIS(JQ1,JQ2,JQ3)*BG(JQ1,JQ2,JQ3))
       IF(IP(J) .EQ. 0) QWVQ(J) = FAC * (P(JQ1,JQ2,JQ3) - DUMQ(J))
       IF(IP(J) .EQ. 0) QWELL(JQ1,JQ2,JQ3) = QWVQ(J)
       IF(IP(J) .NE. 0) PWF(J) = -QV(J)/FAC + P(JQ1,JQ2,JQ3)
C
       IF(IP(J).NE.1 .OR. IVQ(J).NE.1) GO TO 2050
C
C.....CALCULATE WELLHEAD PRESSURE HERE
       QQ=DABS(QV(J))
       VS=VIS(JQ1,JQ2,JQ3)
       XWDQJ=XWDQ(J)
       CALL FRIC(QQ,GR,VS,XWDQJ,EE,RE,F,IF,DFTV)
       A11=C11*XWDQ(J)/TIDQ(J)**5
       A22=C22*XWDQ(J)
```

```
      PWH1=PWF(J)
      CALL PWH(A11,A22,QQ,F,PWH1,DPW,PT1,ZT,NT1,PWH2)
      DUMQ(J)=PWH2
C
2050  CONTINUE
      IF(IPP.EQ.0 .OR. (N.NE.1.AND.N.NE.KS) ) GO TO 2054
C
C---- NOTE THAT DUMQ(J) = PWF(J) IF NO BHP CALCULATIONS ARE REQUIRED
      WRITE(6,591) FT
      WRITE(6,592) (QN1(J),QN2(J),QN3(J),QWVQ(J),PI(J),PWF(J),DUMQ(J),
     &             P(QN1(J),QN2(J),QN3(J)),VIS(QN1(J),QN2(J),QN3(J)),
     &            BG(QN1(J),QN2(J),QN3(J)),IP(J),IVQ(J), J=1,NVQN)
2054  CONTINUE
C
C#### VARIABLE PRESSURE NODE LOGIC #####
C**** CALCULATE INJECTION RATES AT VARIABLE PRESSURE & SPECIAL NODES
C
      IF(NVPN.EQ.0.AND.NSPN.EQ.0)GO TO 2006
      CALL CPRES(IM,JM,IMM,JMM,N,NN,KT,QWELL,P,TX,TY,TZ,DUMP,
     &FT,NVPN,PN1,PN2,PN3,TP1,TP2,PV,KS,NSPN,ISP,JSP,KSP,QSP)
C
C@@@@ BOTH SQSCM & SQSC ARE TOTAL FLOW FROM MATRIX INTO FRACTURE SYSTEM
C@@@@  SQSC USED FOR MCODE = 1 OR 2;  SQSCM USED FOR MCODE = 3 OR 4
C
2006  SQSCM=0.0
      SQSC=0.0
      SQSCD=0.0
      DSP = 0.0
      DO 4650 K=1,KK
      DO 4650 J=1,JJ
      DO 4650 I=1,II
      IF(MCODE.EQ.0) Q(I,J,K) = QWELL(I,J,K)
      IF(MCODE.EQ.0) GO TO 4650
      IF(MCODE.LE.2) Q(I,J,K) = QWELL(I,J,K) + QSC(I,J,K)
      IF(MCODE.GT.2) Q(I,J,K) = QWELL(I,J,K) - QSCM(I,J,K)
      DSP = DSP + INJ(I,J,K)
      SQSCD=SQSCD + QSCD(I,J,K)
      SQSCM = SQSCM + QSCM(I,J,K)
      SQSC  = SQSC  - QSC(I,J,K)
4650  CONTINUE
      IF(MCODE.EQ.0) GO TO 4632
      WRITE(6,1661) KCYPF,N,DPF,SQSCM,SQSC,P(1,1,1)
C
      IF(MCODE.GT.2) INJN = SQSCM * DELT
      IF(MCODE.LE.2) INJN = SQSC * DELT
C
C**** CALCULATE GAS REMAINING, VOLUME FACTORS, MBE,S, ETC.
C
4632  POZAVG=0.0
      TOTPRD=0.0
      TEXPRD=0.0
      SM=0.0
```

```
      SQWELL=0.0
      EXTINJ=0.0
      EXTPRD=0.0
      SQ = 0.0
      DO 3006 K=1,KK
      DO 3006 J=1,JJ
      DO 3006 I=1,II
      SQ = SQ + Q(I,J,K)
      SQWELL=SQWELL+QWELL(I,J,K)
      PP=P(I,J,K)
      CALL INTRP1(PT1,ZT,NT1,PP,ZZ)
      BG(I,J,K)=CON*ZZ/P(I,J,K)
      TOTPRD=TOTPRD+Q(I,J,K)*DELT
      TEXPRD=TEXPRD + QWELL(I,J,K) * DELT
      IF(QWELL(I,J,K).GT.0.0) EXTPRD = EXTPRD + QWELL(I,J,K)*DELT
      IF(QWELL(I,J,K).LT.0.0) EXTINJ = EXTINJ +DABS(QWELL(I,J,K))*DELT
      IF(IBK(I,J,K).EQ.0)GO TO 3006
      SM=SM+VP(I,J,K)/BG(I,J,K)
      POZAVG=POZAVG+P(I,J,K)*VP(I,J,K)/ZZ
3006  CONTINUE
      POZAVG=POZAVG/RESVOL
      SMN=MCFCAL
      IF(N.EQ.1)SMN=MCFINT
      MCFCAL=SM
      NSC = MCFCAL - SMN
      IMBE=NSC + TOTPRD
C
      CEXPRD = CEXPRD + TEXPRD
      CUMPRD=CUMPRD+TOTPRD
      FGR = CEXPRD/TGAS
      CUMINJ = CUMINJ + INJN
      CUMDSP = CUMDSP + DSP
C
C     NOTE THAT CUMPRD IS NET PRODUCTION - CONSIDERING INFLUX FROM
C     MATRIX (HENCE IS CORRECT TO USE FOR MATERIAL BALANCE ERROR IN
C             FRACTURE SYSTEM)
      MCFACT=MCFINT-CUMPRD-CUMO
      MBE=(MCFCAL/MCFACT-1.0)*100.0 + MBEO
C
      IF(MCODE.EQ.0) GO  TO 4702
      MCFMA = MCFMI - CUMINJ + CUMDSP
      MCFDA = MCFDI - CUMDSP
      TGAS1 = MCFACT + MCFMA + MCFDA
      FGF = MCFACT/TGAS1 * 100.
      FGM = MCFMA/TGAS1 * 100.
      FGD = MCFDA/TGAS1 * 100.
      FGGF= (MCFINT - MCFACT)/MCFINT
      FGGM= (MCFMI- MCFMA)/MCFMI
      IF(MCFDI.NE.0.0) FGGD = (MCFDI - MCFDA)/MCFDI
      IF(MCFDI.EQ.0.0) FGGD = 0.0
C@@@@
4702  CONTINUE
```

```
      IF(N.NE.1.AND.N.NE.NN.AND.N.NE.KT)GO TO 999
      IF(KBV.EQ.0) GO TO 3008
      WRITE(6,3010)DELT,POZAVG
      WRITE(6,3011)
      DO 3007 K=1,KK
      DO 3007 J=1,JJ
      DO 3007 I=1,II
      PP=P(I,J,K)
      CALL INTRP1(PT1,ZT,NT1,PP,ZZ)
      CALL INTRP1(PT1,VIS1,NT1,PP,VS)
      VMOD=1. + BKLF(I,J,K)/P(I,J,K)
      VIS(I,J,K)=VS/VMOD
      WRITE(6,3009)I,J,K,P(I,J,K),BQ(I,J,K),ZZ,VS,VMOD,VIS(I,J,K)
     &,GAMMA(I,J,K),Q(I,J,K)
3007  CONTINUE
3008  CONTINUE
C
C*****WRITE NEW DISTRIBUTION & SPECIAL NODE DATA
C
      CALL PRTSMD(II,JJ,KK,IM,JM,N,DELT,TOTPRD,CUMINJ,CUMDSP,
     &P,CUMPRD,MCFINT,FT,KP,NN,KT,MCFCAL,PN,MBE,MCFACT,
     &TGAS,NSC,IMBE,EXTINJ,EXTPRD,TEXPRD,CEXPRD,FGR,INJN,DSP)
C
C@@@@ PRINT PRODUCTION RATE SUMMARY
      IF(MCODE.EQ.0) GO TO 999
      IF(MCODE.LE.2) SCM=SQSC
      IF(MCODE.GT.2) SCM=SQSCM
      WRITE(6,1577) SQWELL,SCM,SQSCD,SQ
C
999   IF(NSPN.EQ.0)GO TO 998
      IF(N.NE.1.AND.N.NE.NN.AND.N.NE.KS)GO TO 998
C
      CALL SPECN(IM,JM,IMM,JMM,KPU1,NSPN,ISP,JSP,
     &KSP,P,PN,VP,QSP,TX,TY,TZ,FT,DELT,EXTINJ,EXTPRD,TEXPRD,
     &CEXPRD,POZAVG,MBE)
      IF(N.EQ.KS)KS=KS+KSN
C
C@@@@
998   CONTINUE
      IF(MCODE.EQ.0) GO TO 9998
      IF(N.NE.1.AND.N.NE.NN.AND.N.NE.KT) GO TO 9998
C
C@@@@ NOTE THAT  QSCM AND QSC ARE COMPARABLE
      WRITE(6,1571)
      DO 4500 J=1,NSPN
      J1=ISP(J)
      J2=JSP(J)
      J3=KSP(J)
      IF(NE(J1,J2,J3).LE.1.E-05) GO TO 4500
      QSCP = -QSC(J1,J2,J3)
      WRITE(6,1575) J1,J2,J3
      WRITE(6,1573)(PM(J1,J2,J3,L),L=1,NRM),QSCM(J1,J2,J3),
```

```
     &QSCP,Q(J1,J2,J3),QSCD(J1,J2,J3)
4500  CONTINUE
      CALL PCGINV(MCFACT,MCFMA,MCFDA,TGAS1,FGF,FGM,FGD,FGGF,
     &            FGGM,FGGD)
9998  IF((N.NE.1.AND.N.NE.NN.AND.N.NE.LT).OR.KPU2.EQ.0)GO TO 9985
      WRITE(6,56)
      WRITE(7,2) II,JJ,KK,FT
      DO 996 K=1,KK
      J=JJ
997   WRITE(7,1129)(P(I,J,K),I=1,II)
      J=J-1
      IF(J.EQ.0)GO TO 996
      GO TO 997
996   CONTINUE
9985  IF(N.EQ.LT)LT=LT+LT1
      IF(N.EQ.KT)KT=KT+KT1
1000  CONTINUE
1001  CONTINUE
      IF(KPU.EQ.0)GO TO 9999
      WRITE(6,8051)FT,CUMPRD,MBE
      WRITE(7,8051)FT,CUMPRD,MBE
      DO 9000 K=1,KK
      J=JJ
8050  WRITE(6,58)(P(I,J,K),I=1,II)
      WRITE(7,58)(P(I,J,K),I=1,II)
      J=J-1
      IF(J.EQ.0)GO TO 9000
      GO TO 8050
9000  CONTINUE
8051  FORMAT(F20.12,F30.12,F20.12)
591   FORMAT(//T18,'ELAPSED TIME = ',F11.6,5X,'VARIABLE RATE NODES ARE '
     &,'SET AS FOLLOWS',//,8X,'NODE',7X,'RATE',11X,'PI',13X,'  PWF  ',
     &4X,'  DUMQ   ',5X,'PRES            VIS',9X,'BG         IP  IVQ')
592   FORMAT(5X,3I3,2X,E12.6,2X,E12.6,2X,3F12.3,F14.5,2X,E12.6,2I5/)
92    FORMAT(F20.0)
1     FORMAT(3F10.0,3I5)
2     FORMAT(3I5,F15.7)
3     FORMAT(I5,F10.0)
4     FORMAT(4F10.0,4I5)
5     FORMAT(I5,F10.5)
6     FORMAT(F10.7)
7     FORMAT(3I5,4F10.0)
9     FORMAT(10(/),T30,'B E G I N N I N G   T I M E - S T E P ',I4,////)
1571  FORMAT(///T10,'----- MATRIX PRESSURE DISTRIBUTIONS, SOURCE ','RATE
     &S --                "QSCM"        "QSC"        "Q"      "QSCD"')
1573  FORMAT(1X,10F12.4)
1575  FORMAT(/T5,'RESERVOIR "HISTORY NODE" ',3I3)
1577  FORMAT(///T15,'........... SUMMARY OF FLOW RATES THIS STEP',
     &' (MCFD) ...........'//T5,'  WELL PRODUCTION   ',
     &T35,'INJECTION FROM MATRIX',T66,'DESORPTION INTO MATRIX',
     &T95,'NET FRACTURE SYSTEM PRODUCTION'/,
     &T5,'  ---- ----------   ',T35,'--------- ---- ------',T66,
```

```
     &'---------- ---- ------',T95,'--- -------- ------ ----------',
     &/T4,F12.3,T36,F12.3,T73,F15.1,F24.3///)
12    FORMAT(13X,3I4,2F20.3)
1119  FORMAT(1X,16F5.0)
117   FORMAT(//T5,'INITIAL FREE MATRIX GAS: I, J, K, BGMM, ',
     &'GASM, NE, GASM1, MCFMI'/)
1129  FORMAT(16F5.0)
16    FORMAT(///T5,'I     J     K       TZ K-1/2       TY J-1/2',
     &'       TX I-1/2',
     &6X,'E',9X,'TX I+1/2       TY J+1/2       TZ K+1/2          QS'/)
17    FORMAT(3I5,3F14.3,2X,E12.6,3F14.3,2X,E12.6)
19    FORMAT(5(/),T15,'RADIUS OF CYLINDRICAL MATRIX ELEMENT(CM) = ',
     &F6.1)
21    FORMAT(///T15,'CUMULATIVE GRID-BLOCK RADII(CM)'/)
23    FORMAT(15X,I5,2F12.6)
1661  FORMAT(T2,'++  PF CYCLE ',I2,2X,' STEP',1X,I3,2X,'DPF =',
     &D11.4,3X,'SQSCM =',D11.4,' SQSC =',D11.4,3X,'P(1,1,1) =',
     &F7.1)

1377  FORMAT(F10.0,E11.4)
1777  FORMAT(///T15,' MATRIX POROSITY & PERMEABILITY(PHIM, KLM) ARE:  ',
     &F12.4,E15.6)
3115  FORMAT(///T15,'>>>>>> YOU HAVE SPECIFIED MCODE = ',I2//)
3117  FORMAT(///T15,'$$$$$$$$$$ INVENTORY OF INITIAL GAS IN PLACE',
     &'(MCF) FOLLOWS $$$$$$$$$$'//T5,'RESERVOIR GAS',
     &T35,'"FREE" MATRIX GAS',T75,'"ADSORBED" MATRIX GAS'/,
     &T5,'--------- ---',T35,'------ ------ ---',T75,
     &'---------- ------ ---'/)
3121  FORMAT(//T15,'>>>>>>>> TOTAL GAS IN PLACE(MCF)                 = '
     &,F13.3// T15,'........ PERCENT GAS IN FRACTURE SYSTEM          = '
     &,F8.2//  T15,'........ PERCENT GAS IN MATRIX AS FREE GAS       = '
     &,F8.2//  T15,'........ PERCENT GAS IN MATRIX AS ADSORBED GAS = '
     &,F8.2)
3123  FORMAT(/////T5,125('*')//)
41    FORMAT(20F4.2)
44    FORMAT(16F5.3)
53    FORMAT(I5)
5701  FORMAT(15X,3I5,5X,F7.1)
56    FORMAT(//)
58    FORMAT(4F19.12)
61    FORMAT(15X,I3,3X,F12.7)
63    FORMAT(F10.0,I5)
66    FORMAT(13X,3I4,F20.3)
69    FORMAT(40A2)
71    FORMAT(3I5,F10.0)
72    FORMAT(15F8.4)
7209  FORMAT(2F10.0)
7211  FORMAT(3F10.0)
7213  FORMAT(6F10.0)
79    FORMAT(12X,3I3,F12.5,F14.2,F12.3,F16.2)
91    FORMAT(2I5,F10.0)
3109  FORMAT(///T5,'SINCE YOU HAVE SPECIFIED A NEGATIVE VALUE FOR'
```

```
     &,' THE KLINKENBERG NUMBER IT WILL BE DETERMINED FROM THE FUNCTION'
     &,/T5,'B = C1 * K ** C2 USING THE VALUES OF C1 & C2 YOU HAVE ',
     &'SPECIFIED -- ','C1 = ',E11.4,'    &  C2 = ',E11.4/)
3111  FORMAT(///T15,'KLINKENBERG FACTOR HAS BEEN SPECIFIED AS --- ',
}    &' B = ',F8.4)
111   FORMAT(1X,3I3,8E15.6)
3119  FORMAT(T4,F12.3,T36,F12.3,T78,F15.0)
3009  FORMAT(1X,3I4,8E15.6)
3010  FORMAT(//T15,'SIZE OF CURRENT TIME-STEP =',F11.6/
     &T15,'AVERAGE P/Z                        =',F11.6//)
3011  FORMAT(//T5,'PARAMETERS ARE: I,J,K, P(I,J,K), BG(I,J,K), ZZ,',
     &' VS, VMOD, VIS(I,J,K),   GAMMA(I,J,K),   Q(I,J,K)'/)
3013  FORMAT(//T5,'PARAMETERS ARE: I,J,K, GAMMA, PG, P, VIS, ZN, PN, Q')
4707  FORMAT(T5,'.....PF CYCLE LOOP WILL NOT CONVERGE---DPF= ',E12.6)
9091  FORMAT(1X,16F5.0)
9999  CONTINUE
      STOP
2049  IND=1
      GO TO 1049
      END
      SUBROUTINE BDRY(IM,JM,IBK,IBKODE)
      INTEGER*2 IBK(IM,JM,1),HEADIN(40)
      READ(5,69) HEADIN
      READ(5,2)IBKODE
      IF(IBKODE.EQ.0)RETURN
      WRITE(6,76)
      DO 2410 L=1,IBKODE
      READ(5,2)I,J,K,IBK(I,J,K)
      WRITE(6,74)I,J,K,IBK(I,J,K)
2410  CONTINUE
69    FORMAT(40A2)
2     FORMAT(4I5)
74    FORMAT(15X,4I5)
76    FORMAT(T15,'NODES OUTSIDE RESERVOIR BOUNDARY OR IMPERMEABLE ',
     &'NODES'/)
      RETURN
      END
      SUBROUTINE CPRES(IM,JM,IMM,JMM,N,NN,KT,QWELL,P,TX,TY,TZ,DUMP
     &,FT,NVPN,PN1,PN2,PN3,TP1,TP2,PV,KS,NSPN,ISP,JSP,KSP,QSP)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2 PN1(1),PN2(1),PN3(1)
     &,ISP(1),JSP(1),KSP(1)
      REAL*4 PV(1),DUMP(1)
      DIMENSION QWELL(IM,JM,1),TX(IMM,JM,1),TY(IM,JMM,1),TZ(IM,JM,1)
     &,P(IM,JM,1),TP1(1),TP2(1),QSP(1)
      IF(NSPN.EQ.0)GO TO 2010
      DO 2002 J=1,NSPN
      QSP(J)=
     &TZ(ISP(J),JSP(J),KSP(J))*P(ISP(J),JSP(J),KSP(J)-1)+
     &TY(ISP(J),JSP(J),KSP(J))*P(ISP(J),JSP(J)-1,KSP(J))+
     &TX(ISP(J),JSP(J),KSP(J))*P(ISP(J)-1,JSP(J),KSP(J))+
     &TX(ISP(J)+1,JSP(J),KSP(J))*P(ISP(J)+1,JSP(J),KSP(J))+
```

```
     &TY(ISP(J),JSP(J)+1,KSP(J))*P(ISP(J),JSP(J)+1,KSP(J))+
     &TZ(ISP(J),JSP(J),KSP(J)+1)*P(ISP(J),JSP(J),KSP(J)+1)-
     &P(ISP(J),JSP(J),KSP(J))*
     &(TZ(ISP(J),JSP(J),KSP(J))+
     &TY(ISP(J),JSP(J),KSP(J))+
     &TX(ISP(J),JSP(J),KSP(J))+
     &TX(ISP(J)+1,JSP(J),KSP(J))+
     &TY(ISP(J),JSP(J)+1,KSP(J))+
     &TZ(ISP(J),JSP(J),KSP(J)+1))
2002  CONTINUE
2010  IF(NVPN.EQ.0)RETURN
      IPP=0
      DO 2020 J=1,NVPN
      IF(IPP.EQ.0 .AND. (N.EQ.1.OR.N.EQ.KS) )
     &WRITE(6,70) FT
      IPP=1
      QWELL(PN1(J),PN2(J),PN3(J)) =
     &TZ(PN1(J),PN2(J),PN3(J))*P(PN1(J),PN2(J),PN3(J)-1)+
     &TY(PN1(J),PN2(J),PN3(J))*P(PN1(J),PN2(J)-1,PN3(J))+
     &TX(PN1(J),PN2(J),PN3(J))*P(PN1(J)-1,PN2(J),PN3(J))+
     &TX(PN1(J)+1,PN2(J),PN3(J))*P(PN1(J)+1,PN2(J),PN3(J))+
     &TY(PN1(J),PN2(J)+1,PN3(J))*P(PN1(J),PN2(J)+1,PN3(J))+
     &TZ(PN1(J),PN2(J),PN3(J)+1)*P(PN1(J),PN2(J),PN3(J)+1)-
     &P(PN1(J),PN2(J),PN3(J))*
     &(TZ(PN1(J),PN2(J),PN3(J))+
     &TY(PN1(J),PN2(J),PN3(J))+
     &TX(PN1(J),PN2(J),PN3(J))+
     &TX(PN1(J)+1,PN2(J),PN3(J))+
     &TY(PN1(J),PN2(J)+1,PN3(J))+
     &TZ(PN1(J),PN2(J),PN3(J)+1))
      IF(IPP.EQ.0 .OR. (N.NE.1.AND.N.NE.KS) ) GO TO 2020
      WRITE(6,69) PN1(J),PN2(J),PN3(J),QWELL(PN1(J),PN2(J),PN3(J)),
     &P(PN1(J),PN2(J),PN3(J)),PV(J),DUMP(J)
2020  CONTINUE
C
C***  NOTE THAT DUMP(J) = PV(J) IF NO BHP CALCULATIONS;
C     OF COURSE P(PN1(J),PN2(J),PN3(J)) = PV(J)
C>>>  NOTE THAT BOTH QSP AND QWELL ARRAYS ARE POSITIVE FOR PRODUCTION!!!
C
69    FORMAT(5X,3I3,4E15.6/)
70    FORMAT(/T18,'ELAPSED TIME =',F11.6,5X,'VARIABLE PRESSURE',
     &' NODES ARE SET AS FOLLOWS',//8X,'NODE',7X,'RATE',11X,'PRES',
     &12X,'PV',13X,'DUMP')
      RETURN
      END
      SUBROUTINE D4SM(NSTEP,II,JJ,KK,IM,JM,IMM,JMM,TX,TY,TZ,E,QS,P,
     &SE,MEP,IR1,JR1,IR,JR,KR,IC,JC,KC,A,B,X,C,
     &NVPN,PN1,PN2,PN3,TP1,TP2,PV,PFLAG,FT,IBK,KT)
C****
C**** PROGRAM FOR SOLVING IMPLICIT FIVE-POINT DIFFERENCE EQUATIONS
C**** USING D4 ORDERING(EACH GRID BLOCK IS ASSIGNED A PLANE NUMBER
C**** ACCORDING TO THE SUM OF THE INDICIES I+J+K AND THE EQUATIONS
```

```
C**** ARE REORDERED WITH FIRST ALL ODD PLANE EQUATIONS IN INCREASING ORDER
C**** FOLLOWED BY ALL EVEN PLANE EQUATIONS IN INCREASING ORDER)
C****
      INTEGER*2 SE(1),MEP(1),IR1(1),JR1(1),S1,S2,S3,IFLAG,PN1(1),PN2(1),
     &PN3(1),PFLAG(1),IBK(IM,JM,1)
      INTEGER*4 IR(1),JR(1),KR(1),IC(1),JC(1),KC(1)
      REAL*8 TX(IMM,JM,1),TY(IM,JMM,1),TZ(IM,JM,1),E(IM,JM,1),
     &QS(IM,JM,1),P(IM,JM,1),A(1),B(1),X(1),C(1),TP1(1),TP2(1),FT
      REAL*4 PV(1)
C
C>>>>>>>> NOTE: BEFORE USING ON OTHER THAN IBM SYSTEM MUST MODIFY
C>>>>>>>>       THE FOLLOWING STATEMENT(TO ALLOW FOR DIVISION BY
C>>>>>>>>       ZERO WHEN NUMERATOR IS ALSO ZERO)
C
      CALL ERRSET(209,256,-1,1,0,200)
C     WRITE(6,8)(((I,J,K,TX(I,J,K),TY(I,J,K),TZ(I,J,K),E(I,J,K),
C    &QS(I,J,K),I=1,II),J=1,JJ),K=1,KK)
8     FORMAT(1X,3I5,5F15.6)
C
      IF(NSTEP.GT.1)GO TO 2000
      CALL ORDER(II,JJ,KK,SE,MEP,IR1,JR1,IR,JR,KR)
C*****NEP=TOTAL NUMBER OF EVEN PLANE BLOCKS
      NEP=0
      DO 2 K=1,KK
2     NEP=NEP+SE(K)
      NME=NEP*(NEP+1)/2
C
2000  DO 3 L=1,NME
3     A(L)=0.0
C
C**** MODIFY MATRIX IF NODE IS TO HAVE SPECIFIED VALUE
C
      IF(NVPN.EQ.0)GO TO 3001
      DO 2500 L=1,NVPN
      PFLAG(L)=0
      IF(FT.GT.TP1(L).AND.FT.LE.TP2(L))PFLAG(L)=1
2500  CONTINUE
      DO 3000 K=1,KK
      DO 3000 J=1,JJ
      DO 3000 I=1,II
      DO 1999 L=1,NVPN
      IF(I.NE.PN1(L).OR.J.NE.PN2(L).OR.K.NE.PN3(L).OR.PFLAG(L).EQ.0)
     &GO TO 1999
      QS(I,J,K)=PV(L)
      IF(I.NE.1)QS(I-1,J,K)=QS(I-1,J,K)-TX(I,J,K)*PV(L)
      IF(I.NE.II)QS(I+1,J,K)=QS(I+1,J,K)-TX(I+1,J,K)*PV(L)
      IF(J.NE.1)QS(I,J-1,K)=QS(I,J-1,K)-TY(I,J,K)*PV(L)
      IF(J.NE.JJ)QS(I,J+1,K)=QS(I,J+1,K)-TY(I,J+1,K)*PV(L)
      IF(K.NE.1)QS(I,J,K-1)=QS(I,J,K-1)-TZ(I,J,K)*PV(L)
      IF(K.NE.KK)QS(I,J,K+1)=QS(I,J,K+1)-TZ(I,J,K+1)*PV(L)
      E(I,J,K)=1.0
      TX(I,J,K)=0.0
```

```
      TX(I+1,J,K)=0.0
      TY(I,J,K)=0.0
      TY(I,J+1,K)=0.0
      TZ(I,J,K)=0.0
      TZ(I,J,K+1)=0.0
      GO TO 3000
1999  CONTINUE
3000  CONTINUE
3001  CONTINUE
C
11    FORMAT(1X,4I5)
      DO 4 L=1,NEP
C     WRITE(6,11)L,IR(L),JR(L),KR(L)
      IC(L)=IR(L)
      JC(L)=JR(L)
      KC(L)=KR(L)
      I1=IR(L)
      J1=JR(L)
      K1=KR(L)
      B(L)=QS(I1,J1,K1)-
     &(TX(I1,J1,K1)*QS(I1-1,J1,K1)/E(I1-1,J1,K1) +
     &     TY(I1,J1,K1)*QS(I1,J1-1,K1)/E(I1,J1-1,K1) +
     &     TZ(I1,J1,K1)*QS(I1,J1,K1-1)/E(I1,J1,K1-1) +
     &     TX(I1+1,J1,K1)*QS(I1+1,J1,K1)/E(I1+1,J1,K1) +
     &     TY(I1,J1+1,K1)*QS(I1,J1+1,K1)/E(I1,J1+1,K1) +
     &     TZ(I1,J1,K1+1)*QS(I1,J1,K1+1)/E(I1,J1,K1+1))
4     CONTINUE
C
      N=0
      DO 100 L=1,NEP
      DO 100 M=1,L
      IFLAG=0
      N=N+1
      IF(L.EQ.M)GO TO 10
C
      S1=IC(L)-IR(M)
      S2=JC(L)-JR(M)
      S3=KC(L)-KR(M)
C
      IF(S1.EQ.2.AND.S2.EQ.0.AND.S3.EQ.0)IFLAG=2
      IF(S1.EQ.0.AND.S2.EQ.2.AND.S3.EQ.0)IFLAG=3
      IF(S1.EQ.0.AND.S2.EQ.0.AND.S3.EQ.2)IFLAG=4
      IF(S1.EQ.1.AND.S2.EQ.1.AND.S3.EQ.0)IFLAG=5
      IF(S1.EQ.1.AND.S2.EQ.0.AND.S3.EQ.1)IFLAG=6
      IF(S1.EQ.0.AND.S2.EQ.1.AND.S3.EQ.1)IFLAG=7
      IF(S1.EQ.1.AND.S2.EQ.-1.AND.S3.EQ.0)IFLAG=8
      IF(S1.EQ.-1.AND.S2.EQ.1.AND.S3.EQ.0)IFLAG=9
      IF(S1.EQ.1.AND.S2.EQ.0.AND.S3.EQ.-1)IFLAG=10
      IF(S1.EQ.-1.AND.S2.EQ.0.AND.S3.EQ.1)IFLAG=11
      IF(S1.EQ.0.AND.S2.EQ.1.AND.S3.EQ.-1)IFLAG=12
      IF(S1.EQ.0.AND.S2.EQ.-1.AND.S3.EQ.1)IFLAG=13
C     WRITE(6,101)IFLAG
```

```
C
      GO TO 20
C
10    CONTINUE
      AA=TX(IR(L),JR(L),KR(L))*TX(IR(L),JR(L),KR(L))
      BB=TY(IR(L),JR(L),KR(L))*TY(IR(L),JR(L),KR(L))
      CC=TZ(IR(L),JR(L),KR(L))*TZ(IR(L),JR(L),KR(L))
      DD=TX(IR(L)+1,JR(L),KR(L))*TX(IR(L)+1,JR(L),KR(L))
      EE=TY(IR(L),JR(L)+1,KR(L))*TY(IR(L),JR(L)+1,KR(L))
      FF=TZ(IR(L),JR(L),KR(L)+1)*TZ(IR(L),JR(L),KR(L)+1)
C
      A(N)=E(IR(L),JR(L),KR(L)) - AA/E(IR(L)-1,JR(L),KR(L))
     &                          - BB/E(IR(L),JR(L)-1,KR(L))
     &                          - CC/E(IR(L),JR(L),KR(L)-1)
     &                          - DD/E(IR(L)+1,JR(L),KR(L))
     &                          - EE/E(IR(L),JR(L)+1,KR(L))
     &                          - FF/E(IR(L),JR(L),KR(L)+1)
C
      GO TO 100
C
20    IF(IFLAG.EQ.0)GO TO 100
C
      IF(IFLAG.EQ.2)A(N)=-TX(IC(L),JC(L),KC(L))*TX(IR(M)+1,JR(M),KR(M))/
     &                   E(IR(M)+1,JR(M),KR(M))
      IF(IFLAG.EQ.3)A(N)=-TY(IC(L),JC(L),KC(L))*TY(IR(M),JR(M)+1,KR(M))/
     &                   E(IR(M),JR(M)+1,KR(M))
      IF(IFLAG.EQ.4)A(N)=-TZ(IC(L),JC(L),KC(L))*TZ(IR(M),JR(M),KR(M)+1)/
     &                   E(IR(M),JR(M),KR(M)+1)
C
      I1=MINO(IR(M),IC(L))
      J1=MINO(JR(M),JC(L))
      K1=MINO(KR(M),KC(L))
C
      IF(IFLAG.EQ.5)A(N)=-TX(IR(M)+1,JR(M),KR(M))*TY(IC(L),JC(L),KC(L))/
     &        E(MINO(IR(M)+1,IC(L)),J1,K1)
     &                  -TY(IR(M),JR(M)+1,KR(M))*TX(IC(L),JC(L),KC(L))/
     &        E(I1,MINO(JR(M)+1,JC(L)),K1)
C
      IF(IFLAG.EQ.6)A(N)=-TX(IR(M)+1,JR(M),KR(M))*TZ(IC(L),JC(L),KC(L))/
     &        E(MINO(IR(M)+1,IC(L)),J1,K1)
     &                  -TZ(IR(M),JR(M),KR(M)+1)*TX(IC(L),JC(L),KC(L))/
     &        E(I1,J1,MINO(KR(M)+1,KC(L)))
C
      IF(IFLAG.EQ.7)A(N)=-TY(IR(M),JR(M)+1,KR(M))*TZ(IC(L),JC(L),KC(L))/
     &        E(I1,MINO(JR(M)+1,JC(L)),K1)
     &                  -TZ(IR(M),JR(M),KR(M)+1)*TY(IC(L),JC(L),KC(L))/
     &        E(I1,J1,MINO(KR(M)+1,KC(L)))
C
      IF(IFLAG.EQ.8)A(N)=-TX(IR(M)+1,JR(M),KR(M))*TY(IC(L),JC(L)+1,KC(L)
     &)/E(MINO(IR(M)+1,IC(L)),MINO(JR(M),JC(L)+1),K1)
     &-TY(IR(M),JR(M),KR(M))*TX(IC(L),JC(L),KC(L))/
     &E(I1,J1,K1)
```

```
C
      IF(IFLAG.EQ.9)A(N)=-TY(IR(M),JR(M)+1,KR(M))*TX(IC(L)+1,JC(L),KC(L)
     &    )/E(MINO(IR(M),IC(L)+1),MINO(JR(M)+1,JC(L)),K1)
     &                    -TX(IR(M),JR(M),KR(M))*TY(IC(L),JC(L),KC(L))/
     &         E(I1,J1,K1)
C
      IF(IFLAG.EQ.10)A(N)=-TX(IR(M)+1,JR(M),KR(M))*TZ(IC(L),JC(L),KC(L)+
     &   1)/E(MINO(IR(M)+1,IC(L)),J1,MINO(KR(M),KC(L)+1))
     &                    -TZ(IR(M),JR(M),KR(M))*TX(IC(L),JC(L),KC(L))/
     &         E(I1,J1,K1)
C
      IF(IFLAG.EQ.11)A(N)=-TZ(IR(M),JR(M),KR(M)+1)*TX(IC(L)+1,JC(L),KC(L
     &   ))/E(MINO(IR(M),IC(L)+1),J1,MINO(KR(M)+1,KC(L)))
     &                    -TX(IR(M),JR(M),KR(M))*TZ(IC(L),JC(L),KC(L))/
     &         E(I1,J1,K1)
C
      IF(IFLAG.EQ.12)A(N)=-TY(IR(M),JR(M)+1,KR(M))*TZ(IC(L),JC(L),KC(L)+
     &   1)/E(I1,MINO(JR(M)+1,JC(L)),MINO(KR(M),KC(L)+1))
     &                    -TZ(IR(M),JR(M),KR(M))*TY(IC(L),JC(L),KC(L))/
     &         E(I1,J1,K1)
C
      IF(IFLAG.EQ.13)A(N)=-TZ(IR(M),JR(M),KR(M)+1)*TY(IC(L),JC(L)+1,KC(L
     &   ))/E(I1,MINO(JR(M),JC(L)+1),MINO(KR(M)+1,KC(L)))
     &                    -TY(IR(M),JR(M),KR(M))*TZ(IC(L),JC(L),KC(L))/
     &         E(I1,J1,K1)
100    CONTINUE
C
1001  IF(NSTEP.EQ.1.OR.NSTEP.EQ.KT) WRITE(6,9999)NEP,NME
9999  FORMAT(T5,'SOLUTION TECHNIQUE IS D4SM  ',
     &             'NUMBER OF EVEN PLANES = ',I5,
     &             '  NUMBER OF MATRIX ELEMENTS USED = ',I5)
C     WRITE(6,101)N,NME,NEP
C     WRITE(6,102)(I,A(I),I=1,N)
C     WRITE(6,102)(I,B(I),I=1,NEP)
      CALL SYMM(NEP,NME,A,B,X,C)
C     WRITE(6,102)(I,X(I),I=1,NEP)
101   FORMAT(1X,25I5)
102   FORMAT(1X,I5,E15.6)
      DO 200 L=1,NEP
200   P(IR(L),JR(L),KR(L))=X(L)
C
      DO 300 K=1,KK
      DO 300 J=1,JJ
      DO 300 I=1,II
      NS=I+J+K
      NA=MOD(NS,2)
      IF(NA.EQ.0)GO TO 300
C
C
      IF(NVPN.EQ.0)GO TO 210
      DO 205 L=1,NVPN
      IF(I.EQ.PN1(L).AND.J.EQ.PN2(L).AND.K.EQ.PN3(L).AND.PFLAG(L).EQ.1)
```

```
     &GO TO 295
C****  THIS ODD PLANE NODE HAS A SPECIFIED VALUE
205   CONTINUE
210   P(I,J,K)=1.0/E(I,J,K)*(QS(I,J,K)-TZ(I,J,K)*P(I,J,K-1)-
     &                                TY(I,J,K)*P(I,J-1,K)-
     &                                TX(I,J,K)*P(I-1,J,K)-
     &                                TX(I+1,J,K)*P(I+1,J,K)-
     &                                TY(I,J+1,K)*P(I,J+1,K)-
     &                                TZ(I,J,K+1)*P(I,J,K+1))
      GO TO 300
295   P(I,J,K)=PV(L)
C
300   CONTINUE
      RETURN
      END



      SUBROUTINE GRDRD(IM,II,JJ,KK,JJ1,KK1,LX,LY,LZ,DX,DY,DZ,DY1,HZ,KRAD
     &,STHETA)
      INTEGER*2 HEADIN(40)
      REAL*4 LX,LY,LZ
      DIMENSION DX(1),DY(1),DZ(1),DY1(1),HZ(IM,1)
      READ(5,69) HEADIN
      READ(5,1)LX,LY,LZ,KLX,KLY,KLZ
      READ(5,52)II,JJ,KK,KRAD,JJ1,KK1
C**** USE KRAD = 1 FOR RADIAL SYSTEM & KRAD = 2 FOR POLAR SYSTEM
      WRITE(6,21)II,JJ,KK
      IF(KRAD.NE.0)IRMAX=II+1
      IF(KRAD.NE.0)WRITE(6,222)
      IF(KLX.NE.0)GO TO 100
      DO 95 I=1,II
95    DX(I)=LX/DFLOAT(II)
      GO TO 105
100   READ(5,3)(I,DX(I),K=1,II)
      IF(KRAD.NE.0)READ(5,31)DX(IRMAX)
      LX=0.0
      DO 106 I=1,II
      IF(KRAD.NE.0)LX=LX+(DX(I+1)-DX(I))
      IF(KRAD.NE.0)GO TO 106
      LX=LX+DX(I)
106   CONTINUE
      IF(KRAD.NE.0)WRITE(6,220)(I,DX(I),I=1,IRMAX)
105   IF(KRAD.EQ.0)WRITE(6,22)(I,DX(I),I=1,II)
      IF(KLY.NE.0)GO TO 115
C**** IF RADIAL COORDINATES ARE TO BE USED SET JJ = 1, KLY = 1
C**   AND READ DY(1) = 360.0
C**** FOR A POLAR SYSTEM SET KLY = 1 AND READ EACH DY(J) IN
C**   DEGREES BEING CERTAIN THAT SUM OF DY'S DOES NOT EXCEED 360.
      DO 110 J=1,JJ
110   DY(J)=LY/DFLOAT(JJ)
```

```
      GO TO 120
115   READ(5,3)(J,DY(J),K=1,JJ)
      LY=0.0
      DO 122 J=1,JJ
122   LY=LY+DY(J)
      IF (KRAD.EQ.2)WRITE(6,230)(J,DY(J),J=1,JJ)
      IF (KRAD.EQ.2)GO TO 121
120   IF(JJ1.EQ.0) WRITE(6,23)(J,DY(J),J=1,JJ)
121   IF(KLZ.NE.0)GO TO 130
      DO 125 K=1,KK
125   DZ(K)=LZ/DFLOAT(KK)
      GO TO 135
130   READ(5,3)(K,DZ(K),L=1,KK)
      LZ=0.0
      DO 137 K=1,KK
137   LZ=LZ+DZ(K)
135   IF(KK1.EQ.0) WRITE(6,24)(K,DZ(K),K=1,KK)
      IF(KRAD.EQ.0)WRITE(6,51)LX,LY,LZ
      IF(KRAD.NE.0)WRITE(6,510)DX(1),DX(IRMAX),LX,LZ
      STHETA=0.0
      DO 520 J=1,JJ
520   STHETA = STHETA+DY(J)
C
      IF(JJ1.EQ.0.AND.KK1.EQ.0) GO TO 999
C
      IF(JJ.EQ.1.AND.JJ1.NE.0) WRITE(6,112)
      IF(JJ.EQ.1.AND.JJ1.NE.0) READ(5,3) (I,DY1(I),K=1,II)
      IF(JJ.EQ.1.AND.JJ1.NE.0) WRITE(6,113) (I,DY1(I),I=1,II)
C
      IF(JJ.GT.1.AND.KK.EQ.1.AND.KK1.NE.0) GO TO 333
      GO TO 999
C
333   WRITE(6,215)
C
      J=JJ
334   READ(5,13) (HZ(I,J),I=1,II)
      WRITE(6,116) (HZ(I,J),I=1,II)
      J=J-1
      IF(J.EQ.0) RETURN
      GO TO 334
C
112   FORMAT(//T5,'X-DIRECTION BLOCKS HAVE VARYING WIDTH AS FOLLOWS',
     &/T15,'I',8X,'DY1(I)'/)
215   FORMAT(//T5,'TWO-DIMENSIONAL RESERVOIR-THICKNESS VARIES AS FOLLOWS'
     &'/)
113   FORMAT(I15,F15.3)
13    FORMAT(10F8.3)
116   FORMAT(13F10.3)
C
1     FORMAT(3F10.0,3I5)
3     FORMAT(I5,F10.0)
69    FORMAT(40A2)
```

```
52    FORMAT(16I5)
51    FORMAT(//T15,'DIMENSIONS OF RESERVOIR---LENGTH(FT)      ='
     &,F6.1/T41,'WIDTH(FT)       =',F6.1/T41,'THICKNESS(FT)  =',F6.1/)
510   FORMAT(//T15,'DIMENSIONS OF POLAR SYSTEM(FT)----INNER RADIUS = ',
     &F13.6/,  T15,'                                    OUTER RADIUS = ',
     &F13.6/,  T15,'                                    DIFFERENCE   = ',
     &F13.6/,  T15,'                                 CYLINDER LENGTH = ',
     &F13.6//)
21    FORMAT(T15,'NUMBER OF NODES IN X-DIRECTION =',I3,/T15,'NUMBER OF',
     &' NODES ',
     &'IN Y-DIRECTION =',I3/T15,'NUMBER OF NODES IN Z-DIRECTION =',I3//)
22    FORMAT(//T15,'GRID BLOCK SIZES IN X-DIRECTION'/(100(/I20,F17.3)))
220   FORMAT(//T15,'INNER & OUTER NODE RADII IN R-DIRECTION',
     &/(100(/I20,F17.5)))
222   FORMAT(/T15,'POLAR CYLINDRICAL COORDINATES HAVE BEEN SELECTED',
     &'--HENCE X=DIRECTION IS R=DIRECTION & Y=DIRECTION IS THETA',
     &' DIRECTION')
23    FORMAT(T15,'GRID BLOCK SIZES IN Y-DIRECTION'/(100(/I20,F17.3)))
230   FORMAT(T15,'THETA VALUES (IN DEGREES)'/(100(/I20,F17.3)))
24    FORMAT(T15,'GRID BLOCK SIZES IN Z-DIRECTION'/(100(/I20,F17.3)))
31    FORMAT(5X,F10.0)
999   CONTINUE
      RETURN
      END
      SUBROUTINE IDXSMD(WK1,WK2,WK3,WK4,IM,JM,KM,IMM,JMM,KMM,
     &NSPM,NTM)
      INTEGER*2 WK1(1)
      INTEGER*4 WK4(1)
      REAL*4    WK2(1)
      REAL*8    WK3(1)
      NPR2=IM*JM*KM
      NPR3=IMM*JM*KM
      NPR4=IM*JMM*KM
      NPR5=IM*JM*KMM
      NPR6=IM*JM/2+1
      NEP=NPR2/2+1
      IISP=1
      IJSP=IISP+NSPM
      IKSP=IJSP+NSPM
      IPFLAG=IKSP+NSPM
      IQN1=IPFLAG+NSPM
      IQN2=IQN1+NSPM
      IQN3=IQN2+NSPM
      IPN1=IQN3+NSPM
      IPN2=IPN1+NSPM
      IPN3=IPN2+NSPM
      IIP=IPN3+NSPM
      IIBK=IIP+NSPM
      ISE=IIBK+NPR2
      IMEP=ISE+KM
      IIR1=IMEP+KM
      IJR1=IIR1+NPR6
```

```
      IIVQ=IJR1+NPR6
      IIVP=IIVQ+NSPM
C
      IIR=1
      IJR=IIR+NEP
      IKR=IJR+NEP
      IIC=IKR+NEP
      IJC=IIC+NEP
      IKC=IJC+NEP
C
      IPV=1
      IDX=IPV+NSPM
      IDY=IDX+IM
      IDZ=IDY+JM
      IKX=IDZ+KM
      IKY=IKX+NPR2
      IKZ=IKY+NPR2
      IPI=IKZ+NPR2
      IPWF=IPI+NSPM
      IVIS=IPWF+NSPM
      IDY1=IVIS+NPR2
      IHZ=IDY1+IM
      IBKLF=IHZ+IM*JM
      IXWDQ=IBKLF+NPR2
      IXWDP=IXWDQ+NSPM
      ITIDQ=IXWDP+NSPM
      ITIDP=ITIDQ+NSPM
      IDUMQ=ITIDP+NSPM
      IDUMP=IDUMQ+NSPM
      IPBHQ=IDUMP+NSPM
      IPBHP=IPBHQ+NSPM
      IQWVQ=IPBHP+NSPM
      IQWVP=IQWVQ+NSPM
C
      ITP1=1
      ITP2=ITP1+NSPM
      IQV=ITP2+NSPM
      ITQ1=IQV+NSPM
      ITQ2=ITQ1+NSPM
      IQSP=ITQ2+NSPM
      IQ=IQSP+NSPM
      IQS=IQ+NPR2
      IGAMMA=IQS+NPR2
      IP=IGAMMA+NPR2
      IE=IP+NPR2
      ITX=IE+NPR2
      ITY=ITX+NPR3
      ITZ=ITY+NPR4
      IBG=ITZ+NPR5
      IAZ=IBG+NPR2
      IBZ=IAZ+IM
      ICZ=IBZ+IM
```

```
      ID1=ICZ+IM
      IEZ=ID1+IM
      IFZ=IEZ+IMM
      IUZ=IFZ+IMM
      IUM=IUZ+IM
      IPHI=IUM+IM
      IVP=IPHI+NPR2
      IPT1=IVP+NPR2
      IVIS1=IPT1+NTM
      IZT=IVIS1+NTM
      IPN=IZT+NTM
      IZN=IPN+NPR2
      IPG=IZN+NPR2
      IBB=IPG+NPR2
      IX=IBB+NEP
      ICC=IX+NEP
      IAA=ICC+NEP
      CALL SUGRMD(WK1(IIVQ),WK1(IIVP),
     &WK1(ISE),WK1(IMEP),WK1(IIR1),WK1(IJR1),
     &WK1(IIBK),WK1(IISP),WK1(IJSP),WK1(IKSP),WK1(IPFLAG),WK1(IQN1),
     &WK1(IQN2),WK1(IQN3),WK1(IPN1),WK1(IPN2),WK1(IPN3),WK1(IIP),
     &WK2(IPV),WK2(IDX),WK2(IDY),WK2(IDZ),
     &WK2(IXWDQ),WK2(IXWDP),WK2(ITIDQ),WK2(ITIDP),WK2(IDUMQ),WK2(IDUMP),
     &WK2(IPBHQ),WK2(IPBHP),WK2(IQWVQ),WK2(IQWVP),
     &WK3(IPHI),WK3(IVP),WK2(IKX),WK2(IKY),WK2(IKZ),
     &WK2(IKX),WK2(IKY),WK2(IKZ),WK2(IPI),WK2(IPWF),WK2(IVIS),WK2(IDY1),
     &WK2(IHZ),WK2(IBKLF),WK3(IQ),WK3(IQS),WK3(IP),WK3(IBG),
     &WK3(IE),WK3(IGAMMA),WK3(ITX),WK3(ITY),WK3(ITZ),WK3(IQV),WK3(ITQ1),
     &WK3(ITQ2),WK3(ITP1),WK3(ITP2),WK3(IQSP),
     &WK3(IAZ),WK3(IBZ),WK3(ICZ),WK3(ID1),WK3(IEZ),WK3(IFZ),WK3(IUZ),
     &WK3(IUM),WK3(IPT1),WK3(IVIS1),WK3(IZT),
     &WK3(IPN),WK3(IZN),WK3(IPG),
     &WK3(IBB),WK3(IX),WK3(ICC),WK3(IAA),
     &WK4(IIR),WK4(IJR),WK4(IKR),WK4(IIC),WK4(IJC),WK4(IKC),
     &IM,JM,KM,IMM,JMM,KMM,NSPM,NTM)
      RETURN
      END
      SUBROUTINE ORDER(IMAX,JMAX,KMAX,SE,MEP,IR,JR,IRR,JRR,KRR)
      INTEGER*2 SE(1),MEP(1),IR(1),JR(1)
      INTEGER*4 IRR(1),JRR(1),KRR(1)
C
C*****MINIMUM DIMENSIONS ARE AS FOLLOWS:
C     SE & MEP -----------KMAX
C     IR & JR ------------IMAX*JMAX/2 + 1
C     IRR & JRR & KRR-----IMAX*JMAX*KMAX/2
C
C*****SE(K)=NUMBER OF EVEN PLANE BLOCKS FOR LAYER K
C*****MEP(K)=MAXIMUM PLANE NUMBER IN LAYER K
C
      MK=0
      DO 50 K=1,KMAX
      SE(K)=0
```

```
      NS=IMAX+JMAX+K
      NA=MOD(NS,2)
      IF(NA.EQ.0)MEP(K)=NS
      IF(NA.EQ.1)MEP(K)=NS-1
      J=JMAX
5     DO 10 I=1,IMAX
      NSUM=I+J+K
C*****CHECK FOR EVEN PLANE
      IF(MOD(NSUM,2).EQ.1)GO TO 10
      SE(K)=SE(K)+1
      IR(SE(K))=I
      JR(SE(K))=J
10    CONTINUE
      J=J-1
      IF(J.GT.0)GO TO 5
C
C*****NOW REORDER EVEN PLANE BLOCKS IN LAYER K ACCORDING TO
C     INCREASING PLANES & INCREASING I IN EACH PLANE
C
      LL=SE(K)
      MM=MEP(K)
      DO 30 M=4,MM,2
      DO 30 L=1,LL
      NSUM=IR(L)+JR(L)+K
      IF(NSUM.NE.M)GO TO 30
      MK=MK+1
      IRR(MK)=IR(L)
      JRR(MK)=JR(L)
      KRR(MK)=K
30    CONTINUE
50    CONTINUE
      RETURN
      END
      SUBROUTINE PARM(II,JJ,KK,PHI,KX,KY,KZ,IM,JM)
      INTEGER*2 HEADIN(40)
      REAL*4 KXC,KYC,KZC,KX(IM,JM,1),KY(IM,JM,1),KZ(IM,JM,1)
      REAL*8 PHI(IM,JM,1)
      READ(5,69) HEADIN
      READ(5,4)PHIC,KXC,KYC,KZC,NUMP,NUMKX,NUMKY,NUMKZ,KPH,KKX,KKY,KKZ
      WRITE(6,56)
      IF(KPH.NE.0)GO TO 145
      DO 140 K=1,KK
      DO 140 J=1,JJ
      DO 140 I=1,II
140   PHI(I,J,K)=PHIC
      WRITE(6,26)PHIC

      IF(NUMP.EQ.0)GO TO 165
      WRITE(6,27)PHIC
      DO 240 L=1,NUMP
      READ(5,71)I,J,K,PHI(I,J,K)
      WRITE(6,32)I,J,K,PHI(I,J,K)
```

```
240   CONTINUE
      GO TO 165
145   WRITE(6,39)
      DO 160 K=1,KK
      WRITE(6,38)K
      J=JJ
155   READ(5,44)(PHI(I,J,K),I=1,II)
      WRITE(6,72)(PHI(I,J,K),I=1,II)
      J=J-1
      IF(J.EQ.0)GO TO 160
      GO TO 155
160   CONTINUE
165   CONTINUE
      WRITE(6,56)
C
C*****ESTABLISH PERMEABILITY (KX) DISTRIBUTION
C
      IF(KKX.NE.0)GO TO 180
      DO 175 K=1,KK
      DO 175 J=1,JJ
      DO 175 I=1,II
175   KX(I,J,K)=KXC
      WRITE(6,29)KXC
      IF(NUMKX.EQ.0)GO TO 195
      WRITE(6,31)KXC
      DO 275 L=1,NUMKX
      READ(5,71)I,J,K,KX(I,J,K)
      WRITE(6,32)I,J,K,KX(I,J,K)
275   CONTINUE
      GO TO 195
180   WRITE(6,43)
      DO 190 K=1,KK
      WRITE(6,38)K
      J=JJ
185   READ(5,44)(KX(I,J,K),I=1,II)
      WRITE(6,72)(KX(I,J,K),I=1,II)
      J=J-1
      IF(J.EQ.0)GO TO 190
      GO TO 185
190   CONTINUE
195   CONTINUE
      WRITE(6,56)
C
C*****ESTABLISH PERMEABILITY (KY) DISTRIBUTION
C
      IF(KKY.NE.0)GO TO 205
      DO 200 K=1,KK
      DO 200 J=1,JJ
      DO 200 I=1,II
200   KY(I,J,K)=KYC
      WRITE(6,33)KYC
      IF(NUMKY.EQ.0)GO TO 220
```

```
      WRITE(6,34)KYC
      DO 300 L=1,NUMKY
      READ(5,71)I,J,K,KY(I,J,K)
      WRITE(6,32)I,J,K,KY(I,J,K)
300   CONTINUE

      GO TO 220
205   WRITE(6,47)
      DO 215 K=1,KK
      WRITE(6,38)K
      J=JJ
210   READ(5,44)(KY(I,J,K),I=1,II)
      WRITE(6,72)(KY(I,J,K),I=1,II)
      J=J-1
      IF(J.EQ.0)GO TO 215
      GO TO 210
215   CONTINUE
220   CONTINUE
      WRITE(6,56)
C
C*****ESTABLISH PERMEABILITY (KZ) DISTRIBUTION
C
      IF(KKZ.NE.0)GO TO 230
      DO 225 K=1,KK
      DO 225 J=1,JJ
      DO 225 I=1,II
225   KZ(I,J,K)=KZC
      WRITE(6,36)KZC
      IF(NUMKZ.EQ.0)GO TO 245
      WRITE(6,37)KZC
      DO 325 L=1,NUMKZ
      READ(5,71)I,J,K,KZ(I,J,K)
      WRITE(6,32)I,J,K,KZ(I,J,K)
325   CONTINUE
      GO TO 245
230   WRITE(6,48)
      DO 2401 K=1,KK
      WRITE(6,38)K
      J=JJ
235   READ(5,44)(KZ(I,J,K),I=1,II)
      WRITE(6,72)(KZ(I,J,K),I=1,II)
      J=J-1
      IF(J.EQ.0)GO TO 2401
      GO TO 235
2401  CONTINUE
245   CONTINUE
69    FORMAT(40A2)
56    FORMAT(//)
71    FORMAT(3I5,F10.0)
72    FORMAT(13E10.3)
4     FORMAT(4F10.0,8I5)
26    FORMAT(T15,'POROSITY (PHI) IS INITIALLY SET AT',F8.5,' FOR ALL',
```

```
     &' NODES'//)
27    FORMAT(T15,'NODES AT WHICH PHI IS TO BE OTHER THAN',F8.5,
     &' ARE AS FOLLOWS'/)
29    FORMAT(T15,'PERMEABILITY (KX) IS INITIALLY',
     &' SET AT',F16.9//)
31    FORMAT(T15,'NODES AT WHICH KX IS TO BE OTHER THAN',F16.9,
     &' ARE AS FOLLOWS'/)
32    FORMAT(15X,3I5,5X,F20.10)
33    FORMAT(T15,'PERMEABILITY (KY) IS INITIALLY',
     &' SET AT',F16.9//)
34    FORMAT(T15,'NODES AT WHICH KY IS TO BE OTHER THAN',F16.9,
     &' ARE AS FOLLOWS'/)
36    FORMAT(T15,'PERMEABILITY (KZ) IS INITIALLY',
     &' SET AT',F16.9//)
37    FORMAT(T15,'NODES AT WHICH KZ IS TO BE OTHER THAN',F16.9,
     &' ARE AS FOLLOWS'/)
38    FORMAT(/1X,'K =',I2/)
39    FORMAT(//T15,'**********POROSITY DISTRIBUTION FOLLOWS**********'/)
43    FORMAT(//T15,'**********PERMEABILITY (KX) DISTRIBUTION**********'
     &/)
44    FORMAT(10F8.5)
47    FORMAT(//T15,'**********PERMEABILITY (KY) DISTRIBUTION**********'
     &/)
48    FORMAT(//T15,'**********PERMEABILITY (KZ) DISTRIBUTION**********'
     &/)
      RETURN
      END
      SUBROUTINE PRTSMD(II,JJ,KK,IM,JM,N,DELT,TOTPRD,CUMINJ,CUMDSP,
     &P,CUMPRD,MCFINT,FT,IP,NN,KT,MCFCAL,PN,MBE,MCFACT,
     &TGAS,NSC,IMBE,EXTINJ,EXTPRD,TEXPRD,CEXPRD,FGR,INJN,DSP)
      INTEGER*2 IP(1)
      REAL*8 P(IM,JM,1),TOTPRD,MCFCAL,MCFACT,CUMPRD,MCFINT,MBE
     &,FT,HOUR,DELT,CUMINJ,CUMDSP,INJN,DSP,NSC,IMBE
     &,TGAS,PN(IM,JM,1),EXTINJ,EXTPRD,TEXPRD,CEXPRD,FGR
C
      IF(N.NE.1.AND.N.NE.NN.AND.N.NE.KT)GO TO 10
      HOUR=FT*24.
10    CONTINUE
C
C*****WRITE NEW PRESSURE DISTRIBUTION
C
3008  IF(N.EQ.1.OR.N.EQ.NN.OR.N.EQ.KT)
     &WRITE(6,13) N,DELT,FT,HOUR,NSC,TOTPRD,IMBE,
     &TEXPRD,INJN,DSP,
     &CEXPRD,CUMINJ,CUMDSP
      WRITE(6,15) EXTINJ,EXTPRD,TEXPRD,CEXPRD,
     &TGAS,FGR,TOTPRD,CUMPRD,MCFACT,MCFCAL,MBE
C
      IF(N.NE.1.AND.N.NE.NN.AND.N.NE.KT)RETURN
3011  WRITE(6,9092)
      DO 901 K=1,KK
      WRITE(6,38)K
```

```
      J=JJ
906   WRITE(6,1115)(P(I,J,K),I=1,II)
      J=J-1
      IF(J.EQ.0)GO TO 901
      GO TO 906
901   CONTINUE
      WRITE(6,9093)
      DO 90 K=1,KK
      WRITE(6,38)K
      J=JJ
85    DO 2050 I=1,II
      IF(P(I,J,K).GE.PN(I,J,K))IP(I)=P(I,J,K)-PN(I,J,K)+0.5
      IF(P(I,J,K).LT.PN(I,J,K))IP(I)=P(I,J,K)-PN(I,J,K)-0.5
2050  CONTINUE
      WRITE(6,14)(IP(I),I=1,II)
      WRITE(6,1114)
      J=J-1
      IF(J.EQ.0)GO TO 90
      GO TO 85
90    CONTINUE
      WRITE(6,9094)
13    FORMAT(///T15,'NUMBER OF TIME-STEPS COMPLETED    = ',I3/
     &          T15,'SIZE OF TIME STEP                 =',F14.6,5H DAYS/
     &          T15,'TOTAL ELAPSED TIME                =',F14.6,5H DAYS,
     &3X,'= ',E15.6,' HOURS',/
     &          T15,'SYSTEM CHANGE THIS STEP           =',F14.3,' MCF'/
     &          T15,'SYSTEM OUTPUT THIS STEP           =',F14.3,' MCF'/
     &          T15,'INCREMENTAL ERROR THIS STEP       =',F14.3,' MCF'//
     &          T15,'NET WELL PRODUCTION THIS STEP     =',F14.3,' MCF'/
     &          T15,'NET INJECTION FROM MATRIX         =',F14.3,' MCF'/
     &          T15,'NET DESORPTION INTO MATRIX        =',F14.3,' MCF'//
     &          T15,'CUMULATIVE WELL PRODUCTION        =',F14.3,' MCF'/
     &          T15,'CUMULATIVE INJECTION FROM MATRIX  =',F14.3,' MCF'/
     &          T15,'CUMULATIVE DESORPTION             =',F14.3,' MCF')
15    FORMAT(//,T15,'WELL INJECTION THIS STEP          =',F14.3,' MCF'/
     &          T15,'WELL PRODUCTION THIS STEP         =',F14.3,' MCF'/
     &          T15,'NET WELL PRODUCTION THIS STEP     =',F14.3,' MCF'/
     &          T15,'CUMULATIVE NET WELL PRODUCTION    =',F14.3,' MCF'//
     &          T15,'TOTAL INITIAL GAS IN PLACE        =',F14.0,' MCF'/
     &          T15,'FRACTIONAL RESOURCE RECOVERY      =',F14.6,' %'//
     &          T15,'NET SYSTEM PRODUCTION THIS STEP   =',F14.3,' MCF'/,
     &          T15,'CUMULATIVE NET PRODUCTION         =',F14.3,' MCF'/,
     &          T15,'ACTUAL FRACTURE GAS REMAINING     =',F14.0,' MCF'/,
     &          T15,'CALCULATED FRACTURE GAS REMAINING =',F14.0,' MCF'/,
     &          T15,'CUM. MAT. BAL. ERROR (FRACTURES)  =',F14.6,' %'//)
14    FORMAT(5X,20I5)
1114  FORMAT(' ')
1115  FORMAT(/(5X,16F5.0))
38    FORMAT(/1X,'K =',I2/)
9092  FORMAT(//T25,'NEW RESERVOIR PRESSURE DISTRIBUTION FOLLOWS')
9093  FORMAT(//T25,'NEW RESERVOIR DELTA-PRESSURE DISTRIBUTION',
     &' FOLLOWS')
```

```
9094  FORMAT(///)
      RETURN
      END
      SUBROUTINE LSOR (NX,NY,NZ,NXM,NYM,NXMM,NYMM,TX,TY,TZ,E,QS,
     &OMEGA,TOL,TOL1,NITER,MITER,DELT,DELTO,P,PN,UM,AZ,BZ,CZ,DZ,EZ,FZ,UZ
     &,NVPN,TP1,TP2,PN1,PN2,PN3,PV,PFLAG,FT,IBK,KRAD,STHETA,KT,NSTEP,
     &KPRT)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     THIS PROGRAM SOLVES A LINEAR SYSTEM OF THREE-DIMENSIONAL FINITE-
C     DIFFERENCE EQUATIONS ARISING FROM FLUID FLOW IN POROUS MEDIA, HEAT
C     CONDUCTION IN SOLIDS, OR DIFFUSION.
C
      INTEGER*2 PFLAG(1),IBK(NXM,NYM,1),PN1(1),
     &PN2(1),PN3(1)
      REAL*4 PV(1),STHETA
      DIMENSION TX(NXMM,NYM,1),TY(NXM,NYMM,1),TZ(NXM,NYM,1),
     &E(NXM,NYM,1),QS(NXM,NYM,1),P(NXM,NYM,1),UM(1)
     &,AZ(1),BZ(1),CZ(1),DZ(1),EZ(1),FZ(1),UZ(1),TP1(1),TP2(1)
     &,PN(NXM,NYM,1)
C
      DIV=DELT/DELTO
      NITER=0
      DMAX=1.0
      RHO1=0.0
      THETA=0.0
C
      DO 10 K=1,NZ
      DO 10 J=1,NY
      DO 10 I=1,NX
      UOLD = PN(I,J,K)
10    P(I,J,K) = P(I,J,K) + DIV * (P(I,J,K) - UOLD)
      IF(NVPN.EQ.0)GO TO 90
      IPP=0
      DO 89 J=1,NVPN
      PFLAG(J)=0
      IF(FT.LE.TP1(J).OR.FT.GT.TP2(J))GO TO 89
      P(PN1(J),PN2(J),PN3(J))=PV(J)
      PFLAG(J)=1
      IPP=1
89    CONTINUE
90    CONTINUE
11    CONTINUE
      TW = 1.0 - OMEGA
      DMAXO=DMAX
      THETAO=THETA
      IF(NITER.GT.MITER)WRITE(6,30)NITER,TOL,DMAX
      IF(NITER.GT.MITER)RETURN
      NITER = NITER +1
      DMAX=0.0
C
      IF(KRAD.EQ.2.AND.ABS(STHETA-360.).LE.1.E-3)GO TO 205
```

```
      DO 20 K=1,NZ
      DO 20 J=1,NY
      DO 15 I=1,NX
      UM(I) = P(I,J,K)
C
      IF(NVPN.EQ.0.OR.IPP.EQ.0)GO TO 27
      DO 260 L=1,NVPN
      IF(I.NE.PN1(L).OR.J.NE.PN2(L).OR.K.NE.PN3(L).OR.PFLAG(L).NE.1)
     &GO TO 260
C
      AZ(I)=0.0
      BZ(I)=1.0
      CZ(I)=0.0
      DZ(I)=PV(L)
      GO TO 15
260   CONTINUE
C
C
27    AZ(I) = TX(I,J,K)
      BZ(I)=E(I,J,K)
      CZ(I) = TX(I+1,J,K)
      DZ(I)=QS(I,J,K) - TZ(I,J,K)*P(I,J,K-1)-TY(I,J,K)*
     &P(I,J-1,K)- TY(I,J+1,K)*P(I,J+1,K) - TZ(I,J,K+1)*P(I,J,K+1)
15    CONTINUE
C
      CALL LTRI (NX,AZ,BZ,CZ,DZ,EZ,FZ,UZ)
C
      DO 16 I=1,NX
      GSLSOR=UZ(I)
      P(I,J,K) = TW*UM(I) + OMEGA*GSLSOR
      DM=DABS(P(I,J,K) - UM(I))
      IF(DM.GT.DMAX) DMAX = DM
16    CONTINUE
C
20    CONTINUE
      GO TO 305
205   DO 200 K=1,NZ
      DO 200 J=1,NY
      DO 150 I=1,NX
      UM(I) = P(I,J,K)
      IF(NVPN.EQ.0.OR.IPP.EQ.0)GO TO 270
      DO 2600 L=1,NVPN
      IF(I.NE.PN1(L).OR.J.NE.PN2(L).OR.K.NE.PN3(L).OR.PFLAG(L).NE.1)
     &GO TO 2600
C
      AZ(I)=0.0
      BZ(I)=1.0
      CZ(I)=0.0
      DZ(I)=PV(L)
      GO TO 150
2600   CONTINUE
C
```

```
C
270   AZ(I) = TX(I,J,K)
      BZ(I)=E(I,J,K)
      CZ(I) = TX(I+1,J,K)
      IF(J.EQ.1)TYP1=TY(I,J,K)*P(I,NY,K)
      IF(J.EQ.1)GO TO 281
      TYP1=TY(I,J,K)*P(I,J-1,K)
281   IF(J.EQ.NY)TYP2=TY(I,J+1,K)*P(I,1,K)
      IF(J.EQ.NY)GO TO 282
      TYP2=TY(I,J+1,K)*P(I,J+1,K)
282   DZ(I)=QS(I,J,K) - TZ(I,J,K)*P(I,J,K-1)- TYP1
     & - TYP2 - TZ(I,J,K+1)*P(I,J,K+1)
150    CONTINUE
C
      CALL LTRI (NX,AZ,BZ,CZ,DZ,EZ,FZ,UZ)
C
      DO 160 I=1,NX
      GSLSOR=UZ(I)
      P(I,J,K) = TW*UM(I) + OMEGA*GSLSOR
      DM=DABS(P(I,J,K) - UM(I))
      IF(DM.GT.DMAX) DMAX = DM
160    CONTINUE
C
200    CONTINUE
305   THETA=DMAX/DMAX0
      DELTA=THETA-THETA0
      IF(DABS(DELTA).GT.TOL1)GO TO 25
      OM=OMEGA-1.0
      RHO1=(THETA+OM)*(THETA+OM)/(THETA*OMEGA*OMEGA)
      IF(RHO1.GE.1.0)GO TO 25
      OMEGA=2.0/(1.0+DSQRT(1.0-RHO1))
C
C
25    IF(DMAX.GT.TOL)GO TO 11
      IF(NSTEP.NE.1 .AND. NSTEP.NE.KT) RETURN
      IF(KPRT.NE.0) WRITE(6,40)NITER,OMEGA,DMAX,THETA,RHO1
40    FORMAT(T3,'CONVERGENCE(LSOR) AFTER ',I3,
     &' ITERATIONS--OMEGA,DMAX,THETA,RHO1 ARE: ',4E15.6)
C
30    FORMAT(T15,'CONVERGENCE(LSOR)  WAS NOT REACHED IN ',I5,
     &' ITERATIONS'/T15,'TOL = ',F10.7,10X,'DMAX = ',F15.7)
C
      RETURN
      END
       SUBROUTINE LTRI(N,AZ,BZ,CZ,DZ,EZ,FZ,UZ)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C     THIS PROGRAM SOLVES THE TRIDIAGONAL SYSTEM
C     GENERATED BY THE SYSTEM OF N EQUATIONS
C
C     A(I)*P(I-1) + B(I)*U(I) + C(I)*P(I+1) = D(I)
C
```

```
C     WITH A(1)=C(N) = 0.0
C
      DIMENSION AZ(1),BZ(1),CZ(1),DZ(1),EZ(1),FZ(1),UZ(1)
      BB=1./BZ(1)
      EZ(2)=-CZ(1)*BB
      FZ(2)= DZ(1)*BB
      DO 10 I=2,N
      FF=1./(AZ(I)*EZ(I)+BZ(I))
      EZ(I+1)=-CZ(I)*FF
10    FZ(I+1)=(DZ(I)-AZ(I)*FZ(I))*FF
      UZ(N)=FZ(N+1)
      K=N
      NN=N-1
      DO 20 I=1,NN
      K=K-1
20    UZ(K)=EZ(K+1)*UZ(K+1) + FZ(K+1)
      RETURN
      END
      SUBROUTINE RVOL(RESVOL,KX,DY,DZ,VIS,KY,KZ,BG,KSOL,MITER,VP,
     &P,PHI,DX,IBK,SM,SCFINT,KT1,LT1,II,JJ,KK,JJ1,KK1,IM,JM,KRAD,
     &KPR,KBV,LT,KT,ISP,JSP,KSP,NSPN,NTM,KPU,KPU1,KPU2,KSN,KS,
     &TMAX,OMEGA,TOL,TOL1,DW,NN,CON,PT1,VIS1,ZT,DY1,HZ,ERR,NT1,
     &ERRM,ERRF,KPRT)
      DIMENSION HEADIN(40)
      INTEGER*2 ISP(1),JSP(1),KSP(1),IBK(IM,JM,1)
      REAL*4 KX(IM,JM,1),KY(IM,JM,1),KZ(IM,JM,1),
     &DX(1),DY(1),DZ(1),VIS(IM,JM,1),DY1(1),HZ(IM,1)
      REAL*8 P(IM,JM,1),OMEGA,TOL,TOL1,DW,RESVOL,SCFINT,SM,BG(IM,JM,1)
     &,ERRF,PHI(IM,JM,1),PT1(1),VIS1(1),ZT(1),CON,ZZ,VS,VP(IM,JM,1),PP
      RESVOL=0.0
      SM=0.0
      DO 400 K=1,KK
      DO 400 J=1,JJ
      DO 400 I=1,II
C**** NOTE THAT FOR POLAR COORDINATES:
C*    (1) THE FORMULA FOR TX MUST BE MULTIPLIED BY THETA/360.
C*    (2) THE SAME FORMULA (AS FOR X,Y,Z) IS USED FOR TY & TZ
C*        BUT DIFFERENT RELATIONSHIPS FOR AREA MUST BE USED.
      IF(KRAD.NE.0)VOL=3.14159*(DX(I+1)*DX(I+1)-DX(I)*DX(I))*DZ(K)
     &* DY(J) * 2.777777D-03
      IF(KRAD.NE.0)KY(I,J,K)=KY(I,J,K)*(DX(I+1)-DX(I))*DZ(K)*.006328
      IF(KRAD.NE.0)KX(I,J,K)=KX(I,J,K)*.006328
      IF(KRAD.NE.0)GO TO 395
      KX(I,J,K)=(KX(I,J,K)*DY(J)*DZ(K))*.006328
      IF(JJ1.NE.0.AND.KK1.EQ.0)KX(I,J,K)=KX(I,J,K)*DY1(I)/DY(J)
      IF(JJ1.EQ.0.AND.KK1.NE.0)KX(I,J,K)=KX(I,J,K)*HZ(I,J)/DZ(K)
      KY(I,J,K)=(KY(I,J,K)*DX(I)*DZ(K))*.006328
      IF(JJ1.EQ.0.AND.KK1.NE.0)KY(I,J,K)=KY(I,J,K)*HZ(I,J)/DZ(K)
      KZ(I,J,K)=(KZ(I,J,K)*DX(I)*DY(J))*.006328
      IF(JJ1.NE.0.AND.KK1.EQ.0.AND.KK.GT.1)KZ(I,J,K)=KZ(I,J,K)*DY1(I)/
     &DY(J)
      VOL=DX(I)*DY(J)*DZ(K)
```

```
      IF(JJ1.NE.0.AND.KK1.EQ.0) VOL=DX(I)*DY1(I)*DZ(K)
      IF(JJ1.EQ.0.AND.KK1.NE.0) VOL=DX(I)*DY(J)*HZ(I,J)
395   VP(I,J,K) = PHI(I,J,K) * VOL
      PP=P(I,J,K)
      CALL INTRP1(PT1,ZT,NT1,PP,ZZ)
      BG(I,J,K)=CON*ZZ/P(I,J,K)
      CALL INTRP1(PT1,VIS1,NT1,PP,VS)
      VIS(I,J,K) = VS
      IF(IBK(I,J,K).EQ.0)GO TO 400
      RESVOL=RESVOL+VP(I,J,K)
      SM=SM+VP(I,J,K)/BG(I,J,K)
400   CONTINUE
      WRITE(6,47)
      DO 2401 K=1,KK
      WRITE(6,38)K
      J=JJ
235   WRITE(6,72)(VP(I,J,K),I=1,II)
      J=J-1
      IF(J.EQ.0)GO TO 2401
      GO TO 235
2401  CONTINUE
      WRITE(6,50)RESVOL
      SCFINT=SM
      READ(5,69) HEADIN
      READ(5,18)KT1,LT1,KPR,KSN,KBV
C*****EVERY KT TH STEP NEW DISTRIBUTION WILL BE PRINTED
C*****EVERY KT TH STEP VOLUME FACTOR DIST. WILL BE PRINTED IF KBV IS NOT ZER
O
C*****EVERY LT TH STEP NEW DISTRIBUTION WILL BE PUNCHED IF KPU2 IS NOT ZERO
C*****TRANSMISSIBILITIES WILL BE PRINTED FIRST & LAST STEP IF KPR IS NOT ZER
O
      LT=LT1
      KT=KT1
      KS=KSN
      READ(5,69) HEADIN
      READ(5,52)NSPN
      IF(NSPN.EQ.0)GO TO 3999
      READ(5,52)(ISP(J),JSP(J),KSP(J),J=1,NSPN)
C*****VITAL STATISTICS AT SPECIAL NODES WILL BE PRINTED EVERY KSN'TH STEP
3999  CONTINUE
      READ(5,69) HEADIN
      READ(5,2)KPU,KPU1,KPU2
C*****FINAL DISTRIBUTION WILL BE PUNCHED IF KPU IS NOT ZERO
C*****SPECIAL NODE DATA WILL BE PUNCHED EVERY KSN'TH STEP IF KPU1 IS NOT ZER
O
      READ(5,69) HEADIN
      READ(5,13) KSOL,MITER,OMEGA,TOL,TOL1
      READ(5,69) HEADIN
      READ(5,9092)NN,TMAX,ERR,ERRM,ERRF,KPRT
      WRITE(6,59)NN,TMAX,ERR,ERRM,ERRF,KT1,LT1,KPR,KSN,KBV,KPRT
13    FORMAT(2I5,3F10.0)
59    FORMAT(//T15,'NUMBER OF TIME STEPS THIS RUN(NN)  =',I5/
```

```
     &T15,'MAXIMUM SIMULATION TIME,DAYS(TMAX) = ',F11.6/
     &T15,'CYCLE TOL. FOR RES. PRES.(ERR)     = ',F5.3/
     &T15,'CYCLE TOLERANCE FOR MATRIX(ERRM)   = ',F5.3/
     &T15,'PRESSURE CONTINUITY TOLERANCE(ERRF)= ',F6.4/
     &T15,'SKIP PRINT PARAMETER(KT1)          = ',I3/
     &T15,'SKIP PUNCH PARAMETER(LT1)          = ',I3/
     &T15,'TRANSMISSIBILITY CODE(KPR)         = ',I3/
     &T15,'SPECIAL NODE CODE(KSN)             = ',I3/
     &T15,'EXTRA PRINT CODE(KBV)              = ',I3/
     &T15,'GENERAL CODE(KPRT)                 = ',I3///)
C@@@@ IF KPRT=0, THE FOLLOWING PRINTOUTS WILL NOT BE OBTAINED:
C     ------ MATRIX GAS CONTENT AT INITIAL PRESSURE IN EACH NODE
C     ------ NUMBER OF ITERATIONS TO CONVERGENCE IN LSOR
C
2     FORMAT(3I5,F15.7)
18    FORMAT(20I5)
52    FORMAT(3I5)
69    FORMAT(40A2)
9092  FORMAT(I5,4F10.0,I5)
38    FORMAT(/1X,'K =',I2/)
47    FORMAT(///T15,'********** RESERVOIR GRID-BLOCK PORE VOLUMES',
     &'**********'/)
50    FORMAT(//T15,'TOTAL RESERVOIR PORE VOLUME =',F20.2/)
72    FORMAT(10(1X,E12.6))
      RETURN
      END
      SUBROUTINE SOURCE(P,KX,KY,KZ,IM,JM,NVQN,QN1,QN2,QN3,
     &TQ1,TQ2,QV,NVPN,PN1,PN2,PN3,TP1,TP2,PV,PN,PI,PWF,IP,
     &IVQ,IVP,XWDQ,XWDP,TIDQ,TIDP)
      IMPLICIT REAL*8 (A-H,O-Z)
      INTEGER*2 HEADIN(40),QN1(1),QN2(1),QN3(1),PN1(1),PN2(1),PN3(1),
     &IP(1),IVQ(1),IVP(1)
      DIMENSION P(IM,JM,1),TP1(1),TP2(1),QV(1),TQ1(1),TQ2(1)
     &,PN(IM,JM,1)
      REAL*4 KX(IM,JM,1),KY(IM,JM,1)
     &,       XWDQ(1),XWDP(1),TIDQ(1),TIDP(1)
     &,KZ(IM,JM,1),PV(1),PI(1),PWF(1)
      REAL*8 M1,M2
C
C
C*****ESTABLISH VARIABLE RATE & VARIABLE VALUE NODES
C
      READ(5,69) HEADIN
      READ(5,2)NVQN
      READ(5,69) HEADIN
      IF(NVQN.EQ.0) GO TO 2001
      WRITE(6,67)
      DO 2000 J=1,NVQN
      READ(5,72)QN1(J),QN2(J),QN3(J),QV(J),PI(J),PWF(J),IP(J)
     &,IVQ(J),XWDQ(J),TIDQ(J)
      READ(5,721)D1,H1,M1,SEC1,D2,H2,M2,SEC2
      WRITE(6,70)QN1(J),QN2(J),QN3(J),D1,H1,M1,SEC1,D2,H2,M2,SEC2,QV(J)
```

```
     &,PI(J),PWF(J),IP(J),IVQ(J),XWDQ(J),TIDQ(J)
      TQ1(J) = D1 + H1/24. + M1/1440. + SEC1/86400.
      TQ2(J) = D2 + H2/24. + M2/1440. + SEC2/86400.
2000  CONTINUE
2001  CONTINUE
      READ(5,69) HEADIN
      READ(5,2)NVPN
      READ(5,69) HEADIN
      IF(NVPN.EQ.0)GO TO 2010
      WRITE(6,73)
      WRITE(6,74)
      DO 2005 J=1,NVPN
      READ(5,76)PN1(J),PN2(J),PN3(J),PV(J),IVP(J),XWDP(J),TIDP(J)
      READ(5,721)D1,H1,M1,SEC1,D2,H2,M2,SEC2
      WRITE(6,78)PN1(J),PN2(J),PN3(J),D1,H1,M1,SEC1,D2,H2,M2,SEC2,PV(J)
     &,IVP(J),XWDP(J),TIDP(J)
      TP1(J) = D1 + H1/24. + M1/1440. + SEC1/86400.
      TP2(J) = D2 + H2/24. + M2/1440. + SEC2/86400.
      IF(TP1(J).EQ.0.0) P(PN1(J),PN2(J),PN3(J))=PV(J)
      IF(TP1(J).EQ.0.0) PN(PN1(J),PN2(J),PN3(J))=PV(J)
2005  CONTINUE
2010  CONTINUE
C
2     FORMAT(3I5,8F11.3)
69    FORMAT(40A2)
72    FORMAT(3I5,3F10.0,2I5,2F10.0)
721   FORMAT(8F10.0)
70    FORMAT(1X,3I3,F10.2,3F5.1,5X,F10.2,3F5.1,F12.3,2X,E12.6,F10.3,
     &I3,I5,F8.0,F8.3)
67    FORMAT(//T5,'RESERVOIR CONTAINS THE FOLLOWING VARIABLE RATE'
     &,' NODES'/,
     &T5,'NODE       TIME ON(DAY,HR,MIN,SEC)',6X,'TIME OFF(DAY,HR,MIN',
     &',SEC)       RATE        PI          PWF     IP  IVQ    XWDQ     TIDQ'/)
73    FORMAT(//T5,'RESERVOIR CONTAINS THE FOLLOWING VARIABLE PRESSURE'
     &,' NODES'/)
74    FORMAT(T5,'NODE          TIME ON(DAY,HR,MIN,SEC)',5X,'TIME OFF',
     &'(DAY,HR,MIN,SEC)          PV      IVP    XWDP    TIDP'/)
76    FORMAT(3I5,F10.0,I5,2F10.0)
78    FORMAT(2X,3I4,F10.2,3F5.1,5X,F10.2,3F5.1,F12.3,I5,F8.0,F8.3)
      RETURN
      END
       SUBROUTINE SPECN(IM,JM,IMM,JMM,KPU1,NSPN,ISP,JSP,
     &KSP,P,PN,VP,QSP,TX,TY,TZ,FT,DELT,EXTINJ,EXTPRD,TEXPRD,
     &CEXPRD,PAVG,MBE)
      INTEGER*2 ISP(1),JSP(1),KSP(1)
      REAL*4 VP(IM,JM,1)
      REAL*8 P(IM,JM,1),TX(IMM,JM,1),TY(IM,JMM,1),
     &TZ(IM,JM,1),MBE,FT,DELT,TEXPRD,CEXPRD,PAVG,QSP(1)
     &,EXTINJ,EXTPRD,PN(IM,JM,1)
      WRITE(6,1117) FT,DELT,EXTINJ,EXTPRD,TEXPRD,CEXPRD,PAVG,MBE
      WRITE(6,1118)(ISP(J),JSP(J),KSP(J),P(ISP(J),JSP(J),KSP(J)),
     &PN(ISP(J),JSP(J),KSP(J)),FT,QSP(J),CEXPRD,J=1,NSPN)
```

```
      IF(KPU1.EQ.0)RETURN
      WRITE(7,1119) FT,DELT,EXTINJ,EXTPRD,CEXPRD,PAVG
      WRITE(7,1115)(ISP(J),JSP(J),KSP(J),P(ISP(J),JSP(J),KSP(J)),
     &PN(ISP(J),JSP(J),KSP(J)),FT,QSP(J),CEXPRD,J=1,NSPN)
      WRITE(6,56)
1117  FORMAT(1X,4F25.12)
1118  FORMAT(2X,3I3,2F8.2,3E15.6)
1119  FORMAT(F12.6,F13.7,2F13.3,F16.3,F13.3)
1115  FORMAT(3I3,2F8.1,3E15.6)
56    FORMAT(//)
      RETURN
      END
      SUBROUTINE STOR (IM,JM,KM,NSPM,NTM)
C***** PROGRAM FOR CALCULATING MINIMUM DIMENSIONS OF WORKING ARRAYS FOR
C***** GAS3D
      INTEGER*4 TBYTES,TK
      WRITE(6,1)
    1 FORMAT(//T15,'   IM   JM   KM  NSPM  NTM')
    5 FORMAT(T15,16I5)
      WRITE(6,5)IM,JM,KM,NSPM,NTM
      IMM=IM+1
      JMM=JM+1
      KMM=KM+1
      IMMA=IMM*JM*KM
      JMMA=IM*JMM*KM
      KMMA=IM*JM*KMM
      IJKM=IM*JM*KM
      IOD=IM*JM/2+1
      NEP=IM*JM*KM/2+1
      NME=NEP*(NEP+1)/2
      I2D4=4*(KM+IOD)
      I4D4=24*NEP
      IR8D4=24*NEP+8*NME
      NI2=NSPM*26+IJKM*2
      NI2=NI2+I2D4
      NI4=I4D4
      NR4=NSPM*52 + IJKM*20 + (IM+JM+KM)*4 + IM*JM*4 + IM*4
      NR8=NSPM*48 + IJKM*88 + (IMMA + JMMA + KMMA)*8 + IM*48 + IMM*16
     & + 24*NTM
      NR8=NR8+IR8D4
      TBYTES=NI2+NR4+NR8+NI4
      TK=TBYTES/1024
      WRITE(6,3)TBYTES,TK
    3 FORMAT(//T15,'THESE DIMENSIONS REQUIRE ',I9,' BYTES OR ',I5,
     &'K FOR ARRAY STORAGE'/)
      NI2=.5*NI2
      NI4=.25*NI4
      NR4=.25*NR4
      NR8=.125*NR8
      WRITE(6,4)NI2,NR4,NR8,NI4
    4 FORMAT(//T15,'WK1 MUST BE DIMENSIONED AT LEAST ',I9/T15,
     &             'WK2 MUST BE DIMENSIONED AT LEAST ',I9/T15,
```

```
     &                 'WK3 MUST BE DIMENSIONED AT LEAST ',I9/T15,
     &                 'WK4 MUST BE DIMENSIONED AT LEAST ',I9/)
C
      IWI2=0.5*I2D4-1
      IWI4=0.25*I4D4-1
      IWR8=0.125*IR8D4-1
      WRITE(6,7)IWI4,IWR8
7     FORMAT(///T15,'WHEN NOT USING D4SM YOU MAY SUBTRACT'/T15,
     &                 I9,'  WORDS FROM WK4 AND'/T15,
     &                 I9,'  WORDS FROM WK3'//)
C
C     THE FOLLOWING ARRAYS MUST BE DIMENSIONED FOR GAS3D
C
C     DIMENSION       ARRAY NAMES      SIZE(BYTES/DIMENSION)
C
C       NSPM          ISP,JSP,KSP,IP,   13 * 2 = 26
C                     PFLAG,QN1,QN2,
C                     QN3,PN1,PN2,PN3,
C                     IVQ,IVP
C
C       NSPM          PV,PI,PWF,        13 * 4 = 52
C                     XWDQ,XWDP,TIDQ,
C                     TIDP,DUMQ,DUMP,
C                     PBHQ,PBHP,
C                     QVWQ,QVWP
C
C       NSPM          TP1,TP2,QV,TQ1,
C                     TQ2,QSP            6 * 8 = 48
C
C       NTM           PT1, VIS1, ZT      3 * 8 = 24
C
C       KM            SE,MEP             2 * 2 =  4
C
    6 CONTINUE
C
C
C       IM*JM*KM      IBK                1 * 2 =  2
C       IM*JM*KM      KX=A1,KY=A2,       5 * 4 = 20
C                     KZ=A3,VIS,BKLF
C
C       IM            DY1                1 * 4 = 4
C       IM*JM         HZ                 1 * 4 = 4
C
C
C       IM*JM*KM      Q,P,E           11 * 8 = 88
C                     B,QS,GAMMA
C                     PHI,VP,PN,ZN,PG
60     CONTINUE
C
C       IOD           IR1,JR1            2 * 2 = 4
C
C       NEP           IR,JR,KR,          6 * 4 = 24
```

```
C                       IC,JC,KC
C        NEP            BBB,X,CCC          3 * 8 = 24
C
C        NME            AAA                1 * 8 = 8
C
C        IMM*JM*KM      TX                 1 * 8 = 8
C        IM*JMM*KM      TY                 1 * 8 = 8
C        IM*JM*KMM      TZ                 1 * 8 = 8
C
C        IM             AZ,BZ,CZ,DZ,       6 * 8 = 48
C                       UZ,UM
C        IMM            EZ,FZ              2 * 8 = 16
C
      RETURN
      END
      SUBROUTINE SYMM(N,NME,A,B,X,C)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(1),B(1),X(1),M,C(1)
C*****
C***** PROGRAM FOR SOLVING N LINEAR EQUATIONS IN N UNKNOWNS
C***** WITH SYMMETRIC COEFFICIENT MATRIX
C***** NME=NUMBER OF MATRIX ELEMENTS(=N*(N+1)/2)
C***** A=COEFFICIENT MATRIX STORED IN COLUMN VECTOR
C***** FORM AS SHOWN BELOW
C*****     A(1)   A(2)   A(4) - - -
C*****
C*****            A(3)   A(5) - - -
C*****
C*****                   A(6) - - -

C***** B=RIGHT HAND SIDE VECTOR
C***** X=SOLUTION VECTOR UPON RETURN TO CALLING PROGRAM
C***** C=WORKING ARRAY OF DIMENSION N
C***** PIVOTAL EQUATION IS DENOTED BY "IPE"
C***** II=INDEX OF FIRST ELEMENT IN PIVOTAL EQUATION=
C*****    IPE*(IPE+1)/2
C***** M=MULTIPLIER FOR KTH EQUATION=A(MM)/A(II)
C*****   WHERE MM=K(K+1)/2 - K + IPE
C***** INDEX OF FIRST ELEMENT IN KTH EQUATION IS
C*****  KK=K(K+1)/2
C***** OTHER INDICES MAY BE CALCULATED FROM
C*****  KK=KK+K
C*****  KK=KK+(K+1)
C*****  KK=KK+(K+2)
C*****  "  "    "
C*****
C***** CORRESPONDING ELEMENTS IN PIVOTAL EQUATION
C***** ARE GIVEN BY LL=KK-K+IPE
C*****
      N1=N-1
      DO 30 IPE=1,N1
      IF=IPE+1
```

```
      II=(IPE*(IPE+1))/2
C
      DO 20 K=IF,N
      KOUNT=-1
      KK=(K*(K+1))/2
      MM=KK-K+IPE
      M=A(MM)/A(II)
      IF(DABS(M).LE.1.D-9)GO TO 20
      B(K)=B(K)-M*B(IPE)
      A(KK)=A(KK)-M*A(MM)
      IF(KK.EQ.NME)GO TO 30
10    KOUNT=KOUNT+1
      KK=KK+(K+KOUNT)
      LL=KK-K+IPE
      A(KK)=A(KK)-M*A(LL)
      IF(KOUNT.EQ.(N-K-1))GO TO 20
      GO TO 10
20    CONTINUE
30    CONTINUE
C
C*****BACK SUBSTITUTE TO GET SOLUTION
C
40    K=N
      X(K)=B(K)/A(NME)
70    K=K-1
      IF(K.EQ.0)GO TO 100
      CALL CKK(A,C,K,N)
      SUB=0.0
      J=K+1
      DO 80 I=J,N
80    SUB=SUB+C(I)*X(I)
      X(K)=(B(K)-SUB)/C(K)
      GO TO 70
100   RETURN
      END
       SUBROUTINE CKK(A,C,K,N)
      REAL*8 A(1),C(1)
      KK=(K*(K+1))/2
      C(K)=A(KK)
      IF(K.EQ.N)RETURN
      KOUNT=-1
      J=K+1
      DO 50 I=J,N
      KOUNT=KOUNT+1
      KK=KK+(K+KOUNT)
50    C(I)=A(KK)
      RETURN
      END
       SUBROUTINE TITLE2
      WRITE(6,1)
1     FORMAT(1H1,///,15X,'************ SUGAR-MD --- A GENERAL PURPOSE',
     &' FRACTURED GAS RESERVOIR SIMULATOR  ***********'/)
```

```
      WRITE(6,20)
20    FORMAT(//T30,'--------- RESERVOIR AND GAS PROPERTIES FOLLOW --',
     &'-------'/////)
      RETURN
      END
      SUBROUTINE TRANS2(II,JJ,KK,IM,JM,TX,TY,TZ,A1,A2,A3,DX,DY,DZ,
     &B,IBK,IBKODE,IMM,JMM,V,KRAD,STHETA)
      INTEGER*2 IBK(IM,JM,1)
      REAL*8 TX(IMM,JM,1),TY(IM,JMM,1),TZ(IM,JM,1),B(IM,JM,1)
      REAL*4 A1(IM,JM,1),A2(IM,JM,1),A3(IM,JM,1),DX(1),DY(1),DZ(1)
     &,V(IM,JM,1)
      IF(KRAD.NE.0)GO TO 600
      IF(II.EQ.1)GO TO 501
      DO 50 K=1,KK
      DO 50 J=1,JJ
      DO 50 I=2,II
      TX(I,J,K)=A1(I-1,J,K)*A1(I,J,K)/(DX(I-1)*A1(I,J,K)*V(I-1,J,K)
     &+DX(I)*V(I,J,K)*A1(I-1,J,K)) *(1.0/B(I-1,J,K)+1.0/B(I,J,K))
      IF(IBKODE.EQ.0)GO TO 50
      IF(IBK(I,J,K).EQ.0.OR.IBK(I-1,J,K).EQ.0)TX(I,J,K)=0.0
50    CONTINUE
501   CONTINUE
      IF(JJ.EQ.1)GO TO 551
      DO 55 K=1,KK
      DO 55 J=2,JJ
      DO 55 I=1,II
      TY(I,J,K)=A2(I,J-1,K)*A2(I,J,K)/(DY(J-1)*V(I,J-1,K)*A2(I,J,K)
     &+DY(J)*V(I,J,K)*A2(I,J-1,K)) * (1./B(I,J-1,K)+1./B(I,J,K))
      IF(IBKODE.EQ.0)GO TO 55
      IF(IBK(I,J,K).EQ.0.OR.IBK(I,J-1,K).EQ.0)TY(I,J,K)=0.0
55    CONTINUE
551   CONTINUE
      IF(KK.EQ.1)GO TO 65
      DO 60 K=2,KK
      DO 60 J=1,JJ
      DO 60 I=1,II
      TZ(I,J,K)=A3(I,J,K-1)*A3(I,J,K)/(DZ(K-1)*V(I,J,K-1)*A3(I,J,K)
     &+DZ(K)*V(I,J,K)*A3(I,J,K-1)) * (1./B(I,J,K-1)+1./B(I,J,K))
      IF(IBKODE.EQ.0)GO TO 60
      IF(IBK(I,J,K).EQ.0.OR.IBK(I,J,K-1).EQ.0)TZ(I,J,K)=0.0
60    CONTINUE
65    CONTINUE
      RETURN
600   CONTINUE
      IF(II.EQ.1)GO TO 601
      TPI=2.0*3.14159
      DO 75 K=1,KK
      DO 75 J=1,JJ
      DO 75 I=2,II
      TX(I,J,K)=TPI*DZ(K)*A1(I-1,J,K)*A1(I,J,K)/(A1(I,J,K)*V(I-1,J,K)*
     &ALOG(2.0*DX(I)/(DX(I-1)+DX(I))) + A1(I-1,J,K)*
     &V(I,J,K)*ALOG((DX(I)+DX(I+1))/(2.0*DX(I)))) *
```

```
     & (1./B(I-1,J,K) + 1./B(I,J,K))*0.5*DY(J)*2.777777D-03
75    CONTINUE
601   IF(JJ.EQ.1)GO TO 701
      DO 78 K=1,KK
      DO 78 I=1,II
      DO 78 J=2,JJ
      DYJ=0.5*(DX(I+1)+DX(I))*DY(J)*1745.32925D-05
      DYJ1=0.5*(DX(I+1)+DX(I))*DY(J-1)*1745.32925D-05
      TY(I,J,K)=A2(I,J-1,K)*A2(I,J,K)/(DYJ1*V(I,J-1,K)*A2(I,J,K)
     &+DYJ*V(I,J,K)*A2(I,J-1,K)) * (1./B(I,J-1,K)+1./B(I,J,K))
 78   CONTINUE
C**** NOW MUST DEFINE TY'S AT THETA = 0 FOR EACH I, K.
      IF(ABS(STHETA-360.).GT.1.E-5)GO TO 86
      DO 82 K=1,KK
      DO 82 I=1,II
      DY1=0.5*(DX(I+1)+DX(I))*DY(1)*1745.32925D-05
      DYJJ=0.5*(DX(I+1)+DX(I))*DY(JJ)*1745.32925D-05
      TY(I,1,K)=A2(I,1,K)*A2(I,JJ,K)/(DYJJ*V(I,JJ,K)*A2(I,1,K)+DY1
     &*V(I,1,K)*A2(I,JJ,K))* (1./B(I,1,K)+1./B(I,JJ,K))
      TY(I,JJ+1,K)=TY(I,1,K)
 82   CONTINUE
 86   CONTINUE
701   IF(KK.EQ.1)RETURN
      DO 80 K=2,KK
      DO 80 J=1,JJ
      DO 80 I=1,II
      AREA=3.14159*(DX(I+1)*DX(I+1)-DX(I)*DX(I)) * 6.328*DY(J)
     &*2.777777D-03
      A31=A3(I,J,K-1)*AREA
      A32=A3(I,J,K)*AREA
      TZ(I,J,K)=A31*A32/(DZ(K-1)*V(I,J,K-1)*A32 + DZ(K)*V(I,J,K)*A31) *
     &(1./B(I,J,K-1) + 1./B(I,J,K))*0.5
80    CONTINUE
      RETURN
      END
      SUBROUTINE UINIT(II,JJ,KK,PI,PN,P,IM,JM,MBEO,CUMO,ETO)
      INTEGER*2 HEADIN(40),FCODE
      REAL*8 P(IM,JM,1),MBEO,CUMO,ETO
      DIMENSION PN(IM,JM,1)
      READ(5,69) HEADIN
      READ(5,63)PI,KPI,NPRES,FCODE
      IF(KPI.NE.0)GO TO 5600
      DO 30 J=1,JJ
      DO 30 I=1,II
      DO 30 K=1,KK
      PN(I,J,K)=PI
   30 P(I,J,K)=PI
      WRITE(6,54)PI
      IF(NPRES.EQ.0)GO TO 3015
      WRITE(6,5700)PI
      DO 5705 L=1,NPRES
      READ(5,71)I,J,K,PN(I,J,K)
```

```
      P(I,J,K)=PN(I,J,K)
      WRITE(6,5701)I,J,K,P(I,J,K)
5705  CONTINUE
3015  RETURN
5600  WRITE(6,57)
      READ(5,59)ETO,CUMO,MBEO
      WRITE(6,60)ETO,CUMO,MBEO
      DO 3010 K=1,KK
      WRITE(6,38)K
      J=JJ
3005  IF(FCODE.NE.0)READ(5,61)(P(I,J,K),I=1,II)
      IF(FCODE.NE.0)GO TO 4005
      READ(5,58)(P(I,J,K),I=1,II)
4005  WRITE(6,1115)(P(I,J,K),I=1,II)
      J=J-1
      IF(J.EQ.0)GO TO 3010
      GO TO 3005
3010  CONTINUE
      DO 3016 K=1,KK
      DO 3016 J=1,JJ
      DO 3016 I=1,II
3016  PN(I,J,K)=P(I,J,K)
69    FORMAT(40A2)
59    FORMAT(F20.12,F30.12,F20.12)
60    FORMAT(T15,'ELAPSED TIME                    = ',F30.12,' DAYS'/
     &T15,        'PRIOR INJECTION                 = ',F30.12,' DAYS'/
     &T15,        'PRIOR MATERIAL BALANCE ERROR= ',F30.12,
     &' PERCENT'//)
63    FORMAT(F10.0,3I5)
54    FORMAT(//T15,'INITIAL CONDITION IS',F7.1//)
C
C*****ESTABLISH NUMBER OF TIME STEPS AND TIME STEP SIZES
5700  FORMAT(T15,'NODES AT WHICH INITIAL VALUE IS OTHER THAN',
     &F7.1,' ARE AS FOLLOWS'/)
71    FORMAT(3I5,F10.0)
5701  FORMAT(15X,3I5,5X,F7.1)
57    FORMAT(//T15,'**********INITIAL CONDITIONS FOLLOW',
     &' **********'/)
38    FORMAT(/1X,'K =',I2/)
58    FORMAT(4F19.12)
61    FORMAT(16F5.0)
1115  FORMAT(5X,20F5.0/)
      RETURN
      END


      SUBROUTINE INTRP1(X,Y,N,XO,YO)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(1),Y(1)
      IF(XO.EQ.X(N)) YO=Y(N)
      IF(XO.EQ.X(N)) RETURN
```

```
      DO 10 I=2,N
      IF(XO.GE.X(I)) GO TO 10
      YO=Y(I-1)+(XO-X(I-1))*(Y(I)-Y(I-1))/(X(I)-X(I-1))
      RETURN
10    CONTINUE
      END
      SUBROUTINE FRIC(Q,G,V,D,E,R,F,I,DFTV)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C**** ROUTINE FOR CALCULATING FRICTION FACTOR FOR PIPE FLOW
C     --- FOR LAMINAR FLOW OR TRANSITION OR TURBULENT FLOW ---
C
C**** REFERENCE --- "HANDBOOK OF NATURAL GAS ENGINEERING" BY KATZ
C
C     F = 64/RE FOR LAMINAR FLOW (1000<RE<2100)
C     F = IMPLICIT FUNCTION (EQ. (7-23) OF KATZ) FOR TRANSITION OR
C                                                   TURBULENT FLOW
C
C     NEWTON-RAPHSON METHOD USED FOR IMPLICIT FUNCTION
C
C     A DEFAULT VALUE USED IF RE IS IN THE CRICITAL ZONE
C
C     ALSO THE DEFAULT VALUE IS USED IF THE NEWTON-RAPHSON SCHEME
C     DOES NOT CONVERGE
C
C      Q = RATE -- MCFD
C      G = GAS GRAVITY
C      V = VISCOSITY -- CP
C      D = PIPE ID -- IN
C      E = ABSOLUTE ROUGHNESS -- IN
C      R = REYNOLDS NUMBER
C      F = FRICTION FACTOR (TO BE CALCULATED & RETURNED)
C      I = NUMBER OF ITERATONS TO CONVERGENCE
C      DFTV = DEFAULT VALUE (USED FOR CRICITAL ZONE)
C
      F=0.0
      ICRIT=0
      A=D/E
      R=20. * Q*G/(V*D)
      I=0
      IF(R.LT.1000) RETURN
C**** F=64/RE NOT VALID FOR VERY LOW FLOW RATES BEFORE LAMINAR FLOW IS
C             ESTABLISHED; VALUE OF F RETURNED IN THIS CASE IS 0.0
C
      IF(R.LE.2100.)F=64./R
      IF(R.LE.2100.)RETURN
C
      IF(R.GT.2100. .AND. R.LE. 4000.) GO TO 20
C
      C=9.34*A/R
C
C**** USE KATZ EQ (7-25) WHICH IS FIRST TERM OF IMPLICIT FN AS INITIAL GUESS
```

```
      X=2.*DLOG10(A) + 1.14
2     FN = X - 2.0 * (DLOG10(A/(1.+C*X)) + 0.57)
      IF(DABS(FN).LE.1.E-6) GO TO 3
      DC=1./C
      DFDX = 1. - .8686/(DC + X)
      X = X - FN/DFDX
      I=I+1
      IF(I.GT.99)PRINT 5,I,X,FN,DFDX
      IF(I.GT.99) GO TO 10
      GO TO 2
3     F=1./(X*X)
      RETURN
C
5     FORMAT(////T2,'(ITER, X, FN, DFDX) ',I3,3E15.6)
7     FORMAT(////T5,'REYNOLDS NUMBER IS IN CRITICAL RANGE -- ',F6.0,
     &   /T5,'THEREFORE DEFAULT VALUE OF -- ',F8.4,' IS BEING USED'/)
10    F=DFTV
      RETURN
20    F=DFTV
      WRITE(6,7) R,DFTV
      RETURN
      END
      SUBROUTINE PBH(A1,A2,Q,F,PBH1,DPW,PT1,ZT,NT1,PBH2)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION PT1(1),ZT(1)
      H(X,Z)=(DEXP(X/Z) - 1.) * Z*Z/X
C
C**** PBH1 = FIXED WELLHEAD PRESSURE(STARTING VALUE FOR BHP)
C     NOTE THAT ITERATION IS ONLY PERFORMED ON Z
C
      I=0
      PWH=PBH1
1     PAV=0.5*(PWH+PBH1)
C
      CALL INTRP1(PT1,ZT,NT1,PAV,ZAV)
C
C
      PBH2 = DSQRT(DEXP(A2/ZAV)*PWH*PWH + A1*H(A2,ZAV)*F*Q*Q)
      DPPW=PBH2-PBH1
      I=I+1
      IF(I.GT.99)PRINT 7,PAV,ZAV,PBH2,DPPW
      IF(I.GT.99)RETURN
C
      IF(DABS(DPPW).LT.DPW) RETURN
C
      PBH1=PBH2
      GO TO 1
C
7     FORMAT(///T5,'BOTTOM-HOLE PRESSURE ROUTINE WILL NOT CONVERGE'/
     &T5,'PAV, ZAV, PBH2, DPPW ',3X,4E15.6/)
C
      END
```

```
      SUBROUTINE PWH(A1,A2,Q,F,PWH1,DPW,PT1,ZT,NT1,PWH2)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION PT1(1),ZT(1)
      H(X,Z)=(DEXP(X/Z) - 1.) * Z*Z/X
C
C**** PBH = PWH1 IS STARTING VALUE FOR WELLHEAD PRESSURE
C
      PBH=PWH1
      I=0
1     PAV=0.5*(PBH+PWH1)
      CALL INTRP1(NP1,ZT,NT1,PAV,ZAV)
C
      PWH2=DSQRT((PBH*PBH - A1*H(A2,ZAV)*F*Q*Q)*DEXP(-A2/ZAV))
C
      DPPW=PWH2-PWH1
      I=I+1
      IF(I.GT.20) WRITE(6,71) PAV,ZAV,PWH1,PWH2,DPPW
      IF(I.GT.20) RETURN
C
      IF(DABS(DPPW) .LT. DPW) RETURN
C
      PWH1=PWH2
      GO TO 1
71    FORMAT(//T5,'WELLHEAD PRESSURE ROUTINE WILL NOT CONVERGE'/,
     &T5,'PAV,ZAV,PWH1,PWH2,DPPW ARE: ',5E15.6)
      END
      SUBROUTINE TRNRD1(IM,JM,NR,P,TR,K,DR,Z,V,TPI,I1,J1,K1)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K(1)
      DIMENSION TPI(IM,JM,1),TR(1),DR(1),Z(1),V(1),P(1)
      IF(NR.EQ.1)GO TO 601
      DO 75 I=2,NR
      TR(I)=TPI(I1,J1,K1) * (P(I-1)+P(I)) * K(I-1) * K(I)/
     &(K(I) * V(I-1) * Z(I-1) * DLOG(2.*DR(I)/(DR(I-1)+DR(I))) +
     &K(I-1) * V(I) * Z(I) * DLOG((DR(I)+DR(I+1))/(2.*DR(I))))
75    CONTINUE
601   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NLLSQS(MCOEFF,NPTS,X,Y,A,B,C,FF,E,YE,DEV,PCTDEV,FP)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(1),Y(1),A(MCOEFF,1),B(1),C(1),FF(MCOEFF,1),E(1),
     &YE(1),DEV(1),PCTDEV(1),FP(1)
C
```

```
C     LEAST SQUARES PROGRAM - APPROXIMATING FUNCTION NON-LINEAR
C     IN ITS COEFFICIENTS:  Y = A*X/(1 + B*X)
      F(X,B1,B2)=B1*X/(1. + B2*X)
C
C     INSERT ARITHMETIC STATEMENT FUNCTIONS DESCRIBING THE PARTIALS OF
C     F WITH RESPECT TO EACH OF B1,B2,....,BNCOEFF.
      F1(X,B1,B2)=X/(1.+ B2*X)
      F2(X,B1,B2)=-B1*X*X/((1.+B2*X)**2)
C
      DFDX(X,B1,B2)= B1/( (1.+B2*X) *(1.+B2*X) )
C
C     PRINT INITIAL ESTIMATES OF THE COEFFICIENTS TO BE DETERMINED
      PRINT 1031,B(1),B(2)
1031  FORMAT(//T15,'INITIAL ESTIMATES OF COEFFICIENTS B1 & B2 ARE: ',
     &2E15.6/)
C
      KTR=0
C
C     BUILD MATRIX FF BY LETTING FF(I,J)=FI(X(J),B(1),B(2),....,B(MCOEFF))
C     I=1,2,...,MCOEFF
10    BB1=B(1)
      BB2=B(2)
      DO 1 J=1,NPTS
      XX=X(J)
      FF(1,J)=F1(XX,BB1,BB2)
1     FF(2,J)=F2(XX,BB1,BB2)
C
C     BUILD MATRIX A, TAKING ADVANTAGE OF SYMMETRY WITH RESPECT TO
C     MAIN DIAGONAL
      DO 3 I=1,MCOEFF
      DO 3 K=1,I
      A(K,I)=0.
      DO 2 J=1,NPTS
    2 A(K,I)=A(K,I)+FF(I,J)*FF(K,J)
    3 A(I,K) = A(K,I)
C
C     BUILD MATRIX C
      DO 4 K=1,MCOEFF
      C(K) = 0.0
      DO 4 J=1,NPTS
      XX=X(J)
    4 C(K)=C(K)+(Y(J)-F(XX,BB1,BB2))*FF(K,J)
9000  FORMAT(1X,7E15.6)
C
C**** SOLVE THE SYSTEM OF EQUATIONS AE = C
      E(1) = (C(1)-A(1,2)*C(2)/A(2,2))/(A(1,1)-A(1,2)*A(2,1)/A(2,2))
      E(2) = (C(1)-A(1,1)*E(1))/A(1,2)
C
      ERR=0.0
      KTR=KTR+1
C
C     MAKE CORRECTIONS TO ESTIMATES OF COEFFICIENTS
```

```
      DO 5 I=1,MCOEFF
      B(I)=B(I)+E(I)
    5 ERR=ERR+DABS(E(I))
C
C     IS SUM OF CORRECTIONS TO COEFFICIENTS SMALL ENOUGH
      IF(ERR.GT..1.AND.KTR.LT.25)GO TO 10
      IF(KTR.NE.25)GO TO 9
C
C     IF AFTER 25 TRIALS SUM OF CORRECTIONS IS STILL TOO LARGE, STOP
      WRITE(6,105)
  105 FORMAT(43HNON-LINEAR LEAST SQUARES FAILED TO CONVERGE)
      STOP
C
C     IF CONVERGENCE HAS BEEN REACHED, PRINT OUT THE COEFFICENTS
9     B1=B(1)
      B2=B(2)
      PRINT 119,B1,B2
C
      PCTDEV(1)=0.0
      DO 90 J=1,NPTS
      XX=X(J)
      YE(J)=F(XX,B1,B2)
      DEV(J)=YE(J)-Y(J)
      IF(J.GT.1)PCTDEV(J)=DEV(J)/Y(J) * 100.0
      FP(J)=DFDX(XX,B1,B2)
90    CONTINUE
      PRINT 107
      PRINT 123,(X(J),Y(J),YE(J),DEV(J),PCTDEV(J),FP(J),J=1,NPTS)
119   FORMAT(/T5,'NON-LINEAR LEAST SQUARES COEFFICIENTS TO EQUATION',
     &'  C = AP/(1 + BP)  ARE AS FOLLOWS'/,T15,'A = ',E12.6,5X,
     &'B = ',E12.6/)
107   FORMAT(T15,'      P(PSIA)      SCF/CU FT     EST VALUE      DEV',
     &'         PCTDEV           DCDP')
123   FORMAT(16X,F9.1,F12.3,F12.3,F15.6,F10.2,E15.6)
      RETURN
      END
```

```
      SUBROUTINE PCGINV(MCFACT,MCFMA,MCFDA,TGAS1,FGF,FGM,FGD,FGGF,
     &                  FGGM,FGGD)
      IMPLICIT REAL*8 (A-Z)
C@@@@ PRINT CURRENT GAS INVENTORY
      WRITE(6,4117)
      WRITE(6,3119) MCFACT,MCFMA,MCFDA
      WRITE(6,4121) TGAS1,FGF,FGM,FGD
      WRITE(6,4123) FGGF,FGGM,FGGD
C
3119  FORMAT(T4,F12.3,T36,F12.3,T78,F15.0)
4117  FORMAT(///T15,'$$$$$$$$$$ CURRENT INVENTORY OF GAS IN PLACE',
     &'(MCF) FOLLOWS $$$$$$$$$$'//T5,'RESERVOIR GAS',
     &T35,'"FREE" MATRIX GAS',T75,'"ADSORBED" MATRIX GAS'/,
     &T5,'--------- ---',T35,'------ ------ ---',T75,
     &'---------- ------ ---'/)
4121  FORMAT(//T15,'>>>>>>>> TOTAL REMAINING GAS IN PLACE(MCF)      = '
     &,F13.3// T15,'........ % GAS REMAINING IN FRACTURE SYSTEM     = '
     &,F8.2//  T15,'........ % REMAINING FREE MATRIX GAS            = '
     &,F8.2//  T15,'........ % REMAINING ADSORBED MATRIX GAS        = '
     &,F8.2)
4123  FORMAT(///T15,'FRACTION OF ORIGINAL FRACTURE SYSTEM GAS PRODUCED',
     &' = ',F7.4//,
     &          T15,'FRACTION OF ORIGINAL FREE MATRIX GAS PRODUCED    ',
     &' = ',F7.4//,
     &          T15,'FRACTION OF ORIGINAL ADSORBED GAS PRODUCED       ',
     &' = ',F7.4/)
      RETURN
      END
      SUBROUTINE RDATA(TR,TSC,PSC,GR,TA,RHOSH,C11,C22,EE,DFTV,
     &DPW,PARMM,NT1,PT1,VIS1,ZT,NTM,NTO,PT,CT,B1,B2)
C
      REAL*8 GR,C11,C22,EE,DFTV,DPW,PARMM,PT1(1),VIS1(1),ZT(1),
     &       PT(1),CT(1),B1,B2
C
      READ(5,69) HEADIN
      READ(5,7213) TR,TSC,PSC,GR,TA,RHOSH
      WRITE(6,11) TR,TSC,PSC,GR,TA,RHOSH
      TR=TR+460.0
      TSC=TSC+460.
      TA=TA+460.
      C11=GR*TA/40000.
      C22=.0375*GR/TA
      READ(5,69) HEADIN
      READ(5,7213) EE,DFTV,DPW,PARMM
      WRITE(6,7215) EE,DFTV,DPW,PARMM
C
      READ(5,69) HEADIN
      READ(5,2)NT1
      READ(5,7211)(PT1(J),VIS1(J),ZT(J),J=1,NT1)
      WRITE(6,93)
      WRITE(6,95)
      WRITE(6,97)(PT1(L),VIS1(L),ZT(L),L=1,NT1)
```

```
C
C@@@@ READ ISOTHERM TABLE
      READ(5,69) HEADIN
      READ(5,2) NTO
      READ(5,7209)(PT(J),CT(J),J=1,NTO)
      WRITE(6,99)
      WRITE(6,101)
      WRITE(6,103)(PT(J),CT(J),J=1,NTO)
C@@@@ NOW CONVERT TO DESIRED UNITS(PSIA & SCF/CU FT)
C     CONSTANT 1.0569 CONVERTS FROM 0 DEG C TO 60 DEG F
C     (NOTE THAT CC AT SC/CU CM = SCF/CU FT)
      DO 200 J=1,NTO
      PT(J)=PT(J)*14.7
200   CT(J)=CT(J)*RHOSH*1.0569
C@@@@ DETERMINE LEAST SQUARES COEFFICIENTS
C     READ INITIAL ESTIMATES OF COEFFICIENTS B1 & B2 IN THE EQ.
C     C = B1*P/(1 + B2*P)
C     NOTE: IF B1 IS READ AS 0.0, NO DESORPTION WILL OCCUR
C
      READ(5,69) HEADIN
      READ(5,1) B1,B2
      IF(B1.LT.1.E-05) WRITE(6,1370)
1     FORMAT(3F10.0,3I5)
2     FORMAT(3I5,F15.7)
69    FORMAT(40A2)
7209  FORMAT(2F10.0)
7211  FORMAT(3F10.0)
7213  FORMAT(6F10.0)
11    FORMAT(///T15,'RESERVOIR TEMPERATURE = ',F8.2,' (DEG F)',
     &          /T15,'STANDARD TEMPERATURE  = ',F8.2,' (DEG F)',
     &          /T15,'STANDARD PRESSURE     = ',F8.2,' (PSIA)',
     &          /T15,'GAS GRAVITY           = ',F8.3,
     &          /T15,'AVG WELLBORE TEMP     = ',F8.2,' (DEG F)'/
     &           T15,'SHALE DENSITY         = ',F8.2,' (GM/CC)'//)
7215  FORMAT(T15,'PARAMETERS FOR BOTTOM-HOLE PRESSURE CALCULATIONS'/
     &       T15,'ROUGHNESS FACTOR = ',F12.6/
     &       T15,'DEFAULT FRICTION FACTOR = ',F6.3/
     &       T15,'CONVERGENCE TOLERANCE   = ',F6.3//
     &       T15,'PARAMETER FOR OUTER TRM = ',F6.3//)
93    FORMAT(/////T18,'GAS VISCOSITY & Z-FACTOR TABLE'/)
95    FORMAT(T15,'   P(PSIA)        MU(CP)          Z')
97    FORMAT(13X,F10.2,F14.4,F13.3)
99    FORMAT(5(/),T15,'METHANE ADSORPTION ISOTHERM (@ RES. TEMP.)'/)
101   FORMAT(T15,'   P(ATM)     CC AT STP/GM')
103   FORMAT(13X,F9.1,F12.3)
1370  FORMAT(//T15,'SINCE YOU HAVE SPECIFIED B1 = 0.0  NO DESORPTION',
     &' WILL OCCUR THIS RUN......'/)
      RETURN
      END
```

APPENDIX B2

SAMPLE INPUT DATA

```
BLOCK SIZES AND NUMBER OF BLOCKS IN X,Y,Z DIRECTIONS
  1000.      1000.     311.          0     0     0
   10    10     1     0     0     0
1--PHI---01---KX---01---KY---01---KZ---01234512345123451234 5
0.038        0.15       0.15       0.1          0     0     0     0
IMPERMEABLE BLOCKS OR BLOCKS OUTSIDE RESERVOIR BOUNDARY
    0
    TR          TSC         PSC        GR         TWB        RHOSH
  108.0        60.        14.7       0.861      84.0       2.714
  EE          DFTV        DPW        PARMM
  .0006       0.05        2.00       1.0
GAS PROPERTY TABLE
   11
        15.     0.0102      0.998
       100.     0.0105      0.985
       200.     0.0109      0.962
       300.     0.0112      0.925
       400.     0.0115      0.902
       500.     0.0117      0.875
       600.     0.0121      0.843
       700.     0.0125      0.830
       800.     0.0131      0.800
       900.     0.0137      0.775
      1000.     0.0143      0.753
ADSORPTION ISOTHERM TABLE(PRESSURE IN ATM, CONC. IN CC @ STP/GM)
   11
  0.0        0.0
  10.        .70
  20.        1.20
  30.        1.65
  40.        2.00
  50.        2.25
  60.        2.40
  70.        2.50
  80.        2.60
  90.        2.65
 100.        2.70
INITIAL ESTIMATES (B1 & B2) FOR THE EQ. -- C = B1*P/(1 + B2*P)
.0129        .001
INITIAL PRESSURE AND BLOCKS WITH DIFFERENT PRESSURE
 305.0
VARIABLE RATE BLOCKS
    1
1---51---51---51---QV---01---PI---01--PWF---01--IP1-IVQ1--XWDQ--01--TIDQ--0
    4     4     1  0.0       .4666        83.         0     1  2500.      6.00
0.0          0.0       0.0        0.0        9999.      0.0        0.0        0.0
```

```
VARIABLE PRESSURE BLOCKS
    0
1---51---51---51---PV---0 IVP 1--XWDP--01--TIDP--0
KLINKENBERG PARAMETERS(BK, C1, C2)
-1.          12.64      -.33
NUMBER OF GRID-BLOCKS IN MATRIX ELEMENT & ELEMENT RADIUS (CM)
    6  50.
NORMALIZED CUMULATIVE RADII OF MATRIX ELEMENT
    1  .05
    2  .10
    3  .20
    4  .40
    5  .80
    6  .9995
    7  1.0
12345678901234567890 1  MATRIX ELEMENT GRID DATA -- PHIM, KLM(MD)
  .015       .1000D-06
SHALE MATRIX CODE (0, 1, 2, 3, OR 4)
    4
PRINT CODES(KT1, LT1, KPR, KSN, KBV)
    5    0    0    5    0
HISTORY BLOCKS
    9
    4    4    1
    3    3    1
    4    3    1
    5    3    1
    3    4    1
    5    4    1
    3    5    1
    4    5    1
    5    5    1
PUNCH CODES
    0    0    0
 KSOLMITER  OMEGA       TOL        TOL1
    3  200  1.200      0.001      0.0000
   NN  TMAX       ERR        ERRM       ERRF       KPRT
   20 9999.       1.00       1.00       0.001        0
TIME STEPS
  5.
  5.
 10.
 20.
 40.
 50.
 60.
 80.
 100.
```

APPENDIX B3

SAMPLE OUTPUT DATA

********** SUGAR-MD --- A GENERAL PURPOSE FRACTURED GAS RESERVOIR SIMULATOR **********

-------- RESERVOIR AND GAS PROPERTIES FOLLOW --------

NUMBER OF NODES IN X-DIRECTION = 10
NUMBER OF NODES IN Y-DIRECTION = 10
NUMBER OF NODES IN Z-DIRECTION = 1

GRID BLOCK SIZES IN X-DIRECTION

1	100.000
2	100.000
3	100.000
4	100.000
5	100.000
6	100.000
7	100.000
8	100.000
9	100.000
10	100.000

GRID BLOCK SIZES IN Y-DIRECTION

1	100.000
2	100.000
3	100.000
4	100.000
5	100.000
6	100.000
7	100.000
8	100.000
9	100.000
10	100.000

GRID BLOCK SIZES IN Z-DIRECTION

1	311.000

```
DIMENSIONS OF RESERVOIR---LENGTH(FT)     =1000.0
                          WIDTH(FT)      =1000.0
                          THICKNESS(FT)  = 311.0


POROSITY (PHI) IS INITIALLY SET AT 0.03800 FOR ALL NODES


PERMEABILITY (KX) IS INITIALLY SET AT     0.149999976


PERMEABILITY (KY) IS INITIALLY SET AT     0.149999976


PERMEABILITY (KZ) IS INITIALLY SET AT     0.099999964


RESERVOIR TEMPERATURE =   108.00 (DEG F)
STANDARD TEMPERATURE  =    60.00 (DEG F)
STANDARD PRESSURE     =    14.70 (PSIA)
GAS GRAVITY           =    0.861
AVG WELLBORE TEMP     =    84.00 (DEG F)
SHALE DENSITY         =     2.71 (GM/CC)

PARAMETERS FOR BOTTOM-HOLE PRESSURE CALCULATIONS
ROUGHNESS FACTOR =     0.000600
DEFAULT FRICTION FACTOR =  0.050
CONVERGENCE TOLERANCE   =  2.000

PARAMETER FOR OUTER TRM =  1.000
```

GAS VISCOSITY & Z-FACTOR TABLE

P(PSIA)	MU(CP)	Z
15.00	0.0102	0.998
100.00	0.0105	0.985
200.00	0.0109	0.962
300.00	0.0112	0.925
400.00	0.0115	0.902
500.00	0.0117	0.875
600.00	0.0121	0.843
700.00	0.0125	0.830
800.00	0.0131	0.800
900.00	0.0137	0.775
1000.00	0.0143	0.753

METHANE ADSORPTION ISOTHERM (@ RES. TEMP.)

P(ATM)	CC AT STP/GM
0.0	0.0
10.0	0.700
20.0	1.200
30.0	1.650
40.0	2.000
50.0	2.250
60.0	2.400
70.0	2.500
80.0	2.600
90.0	2.650
100.0	2.700

INITIAL ESTIMATES OF COEFFICIENTS B1 & B2 ARE: 0.129000D-01 0.100000D-02

NON-LINEAR LEAST SQUARES COEFFICIENTS TO EQUATION C = AP/(1 + BP) ARE AS FOLLOWS
A = 0.181646D-01 B = 0.163263D-02

P(PSIA)	SCF/CU FT	EST VALUE	DEV	PCTDEV	DCDP
0.0	0.0	0.0	0.0	0.0	0.181646D-01
147.0	2.008	2.153	0.145499	7.25	0.118137D-01
294.0	3.442	3.608	0.166291	4.83	0.829292D-02
441.0	4.733	4.657	-0.075540	-1.60	0.614010D-02

588.0	5.737	5.449	-0.287414	-5.01	0.472848D-02
735.0	6.454	6.069	-0.385261	-5.97	0.375309D-02
882.0	6.884	6.566	-0.318086	-4.62	0.305109D-02
1029.0	7.171	6.974	-0.196583	-2.74	0.252910D-02
1176.0	7.458	7.316	-0.142205	-1.91	0.213044D-02
1323.0	7.601	7.605	0.003763	0.05	0.181912D-02
1470.0	7.745	7.854	0.108877	1.41	0.157137D-02

INITIAL CONDITION IS 305.0

RESERVOIR CONTAINS THE FOLLOWING VARIABLE RATE NODES

NODE	TIME ON(DAY,HR,MIN,SEC)	TIME OFF(DAY,HR,MIN,SEC)	RATE	PI	PWF	IP	IVQ	XWDQ	TIDQ
4 4 1	0.0 0.0 0.0 0.0	9999.00 0.0 0.0 0.0	0.0	0.466600E 00	83.000	0	1	2500.	6.000

SINCE YOU HAVE SPECIFIED A NEGATIVE VALUE FOR THE KLINKENBERG NUMBER IT WILL BE DETERMINED FROM THE FUNCTION
B = C1 * K ** C2 USING THE VALUES OF C1 & C2 YOU HAVE SPECIFIED — C1 = 0.1264E 02 & C2 = -0.3300E 00

RADIUS OF CYLINDRICAL MATRIX ELEMENT(CM) = 50.0

CUMULATIVE GRID-BLOCK RADII(CM)

1	0.050000	2.500000
2	0.100000	5.000000
3	0.200000	10.000000
4	0.400000	20.000000
5	0.800000	40.000000
6	0.999500	49.975000
7	1.000000	50.000000

MATRIX POROSITY & PERMEABILITY(PHIM, KLM) ARE: 0.0150 0.100000D-06

>>>>>> YOU HAVE SPECIFIED MCODE = 4

********** RESERVOIR GRID-BLOCK PORE VOLUMES**********

K = 1

```
0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06
0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06
0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06
0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06
0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06
0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06
0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06
0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06
0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06
0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06
0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06 0.118180D 06
```

TOTAL RESERVOIR PORE VOLUME = 11817999.62

```
NUMBER OF TIME STEPS THIS RUN(NN)  =  20
MAXIMUM SIMULATION TIME,DAYS(TMAX) = 9999.000000
CYCLE TOL. FOR RES. PRES.(ERR)     = 1.000
CYCLE TOLERANCE FOR MATRIX(ERRM)   = 1.000
PRESSURE CONTINUITY TOLERANCE(ERRF)= 0.0010
SKIP PRINT PARAMETER(KT1)          =  5
SKIP PUNCH PARAMETER(LT1)          =  0
TRANSMISSIBILITY CODE(KPR)         =  0
SPECIAL NODE CODE(KSN)             =  5
EXTRA PRINT CODE(KBV)              =  0
GENERAL CODE(KPRT)                 =  0
```

****

$$$$$$$$$$ INVENTORY OF INITIAL GAS IN PLACE(MCF) FOLLOWS $$$$$$$$$$

RESERVOIR GAS "FREE" MATRIX GAS "ADSORBED" MATRIX GAS

242985.448	92039.885	1106533.

```
>>>>>>>> TOTAL GAS IN PLACE(MCF)              =  1441558.145

........ PERCENT GAS IN FRACTURE SYSTEM       =    16.86

........ PERCENT GAS IN MATRIX AS FREE GAS    =     6.38

........ PERCENT GAS IN MATRIX AS ADSORBED GAS =   76.76
```

****

B E G I N N I N G T I M E - S T E P 1

ELAPSED TIME = 5.000000 VARIABLE RATE NODES ARE SET AS FOLLOWS

NODE	RATE	PI	PWF	DUMQ	PRES	VIS	BG	IP	IVQ
4 4 1	0.144367E 03	0.466600E 00	83.000	89.479	267.127	0.01020	0.562918D 02	0	1

++ PF CYCLE 4 STEP 1 DPF = 0.1735D-03 SQSCM = 0.1506D 02 SQSC = 0.1553D 02 P(1,1,1) = 305.0

```
NUMBER OF TIME-STEPS COMPLETED   =  1
SIZE OF TIME STEP                =     5.000000 DAYS
TOTAL ELAPSED TIME               =     5.000000 DAYS   =   0.120000D 03 HOURS
SYSTEM CHANGE THIS STEP          =     -646.970 MCF
SYSTEM OUTPUT THIS STEP          =      646.525 MCF
INCREMENTAL ERROR THIS STEP      =       -0.445 MCF
```

NET WELL PRODUCTION THIS STEP	=	721.834	MCF
NET INJECTION FROM MATRIX	=	75.309	MCF
NET DESORPTION INTO MATRIX	=	68.519	MCF
CUMULATIVE WELL PRODUCTION	=	721.834	MCF
CUMULATIVE INJECTION FROM MATRIX	=	75.309	MCF
CUMULATIVE DESORPTION	=	68.519	MCF
WELL INJECTION THIS STEP	=	0.0	MCF
WELL PRODUCTION THIS STEP	=	721.834	MCF
NET WELL PRODUCTION THIS STEP	=	721.834	MCF
CUMULATIVE NET WELL PRODUCTION	=	721.834	MCF
TOTAL INITIAL GAS IN PLACE	=	1441558.	MCF
FRACTIONAL RESOURCE RECOVERY	=	0.000501	%
NET SYSTEM PRODUCTION THIS STEP	=	646.525	MCF
CUMULATIVE NET PRODUCTION	=	646.525	MCF
ACTUAL FRACTURE GAS REMAINING	=	242339.	MCF
CALCULATED FRACTURE GAS REMAINING	=	242338.	MCF
CUM. MAT. BAL. ERROR (FRACTURES)	=	-0.000184	%

NEW RESERVOIR PRESSURE DISTRIBUTION FOLLOWS

K = 1

305.	305.	305.	305.	305.	305.	305.	305.	305.	305.
305.	305.	305.	305.	305.	305.	305.	305.	305.	305.
305.	305.	305.	305.	305.	305.	305.	305.	305.	305.
305.	305.	305.	305.	305.	305.	305.	305.	305.	305.
305.	305.	305.	304.	305.	305.	305.	305.	305.	305.
305.	305.	303.	300.	303.	305.	305.	305.	305.	305.
305.	304.	300.	267.	300.	304.	305.	305.	305.	305.
305.	305.	303.	300.	303.	305.	305.	305.	305.	305.
305.	305.	305.	304.	305.	305.	305.	305.	305.	305.
305.	305.	305.	305.	305.	305.	305.	305.	305.	305.

NEW RESERVOIR DELTA-PRESSURE DISTRIBUTION FOLLOWS

K = 1

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	-1	0	0	0	0	0	0
0	0	-2	-5	-2	0	0	0	0	0
0	-1	-5	-38	-5	-1	0	0	0	0
0	0	-2	-5	-2	0	0	0	0	0
0	0	0	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

..... SUMMARY OF FLOW RATES THIS STEP (MCFD) .....

WELL PRODUCTION	INJECTION FROM MATRIX	DESORPTION INTO MATRIX	NET FRACTURE SYSTEM PRODUCTION
144.367	15.062	13.7	129.305

			5.000000000000		5.000000000000		0.0	721.833953857422
			721.833953857422		721.833953857422		329.261147470238	-0.000183720801
4	4	1	267.13	305.00	0.500000D 01	0.705368D 02	0.721834D 03	
3	3	1	303.41	305.00	0.500000D 01	-0.304824D 01	0.721834D 03	
4	3	1	299.54	305.00	0.500000D 01	-0.105054D 02	0.721834D 03	
5	3	1	303.41	305.00	0.500000D 01	-0.304722D 01	0.721834D 03	
3	4	1	299.54	305.00	0.500000D 01	-0.105054D 02	0.721834D 03	
5	4	1	299.54	305.00	0.500000D 01	-0.105040D 02	0.721834D 03	
3	5	1	303.41	305.00	0.500000D 01	-0.304724D 01	0.721834D 03	
4	5	1	299.54	305.00	0.500000D 01	-0.105040D 02	0.721834D 03	
5	5	1	303.41	305.00	0.500000D 01	-0.304623D 01	0.721834D 03	

—— MATRIX PRESSURE DISTRIBUTIONS, SOURCE RATES —						"QSCM"	"QSC"	"Q"	"QSCD"
RESERVOIR "HISTORY NODE" 4 4 1									
304.9997	304.9994	304.9961	304.9195	301.3385	267.1271	7.4633	7.6796	136.8985	6.7740
RESERVOIR "HISTORY NODE" 3 3 1									
305.0000	305.0000	304.9998	304.9964	304.8379	303.4121	0.3294	0.3385	-0.3294	0.2984
RESERVOIR "HISTORY NODE" 4 3 1									
305.0000	304.9999	304.9994	304.9878	304.4453	299.5362	1.1276	1.1582	-1.1276	1.0211
RESERVOIR "HISTORY NODE" 5 3 1									
305.0000	305.0000	304.9998	304.9964	304.8379	303.4127	0.3293	0.3385	-0.3293	0.2984
RESERVOIR "HISTORY NODE" 3 4 1									
305.0000	304.9999	304.9994	304.9878	304.4453	299.5362	1.1276	1.1582	-1.1276	1.0211
RESERVOIR "HISTORY NODE" 5 4 1									
305.0000	304.9999	304.9994	304.9878	304.4454	299.5369	1.1274	1.1577	-1.1274	1.0207
RESERVOIR "HISTORY NODE" 3 5 1									
305.0000	305.0000	304.9998	304.9964	304.8379	303.4127	0.3293	0.3385	-0.3293	0.2984
RESERVOIR "HISTORY NODE" 4 5 1									
305.0000	304.9999	304.9994	304.9878	304.4454	299.5369	1.1274	1.1577	-1.1274	1.0207
RESERVOIR "HISTORY NODE" 5 5 1									
305.0000	305.0000	304.9998	304.9964	304.8380	303.4132	0.3292	0.3382	-0.3292	0.2982

$$$$$$$$$$ CURRENT INVENTORY OF GAS IN PLACE(MCF) FOLLOWS $$$$$$$$$$

RESERVOIR GAS	"FREE" MATRIX GAS	"ADSORBED" MATRIX GAS
242333.923	92033.095	1106464.

>>>>>>>> TOTAL REMAINING GAS IN PLACE(MCF) = 1440836.312

..... % GAS REMAINING IN FRACTURE SYSTEM = 16.82

..... % REMAINING FREE MATRIX GAS = 6.39

..... % REMAINING ADSORBED MATRIX GAS = 76.79

FRACTION OF ORIGINAL FRACTURE SYSTEM GAS PRODUCED = 0.0027

```
FRACTION OF ORIGINAL FREE MATRIX GAS PRODUCED     =  0.0001

FRACTION OF ORIGINAL ADSORBED GAS PRODUCED        =  0.0001

++  PF CYCLE  4   STEP   2  DPF = 0.2910D-03   SQSCM = 0.2515D 02 SQSC = 0.2544D 02   P(1,1,1) =  304.9
++  PF CYCLE  5   STEP   3  DPF = 0.6433D-04   SQSCM = 0.3728D 02 SQSC = 0.3753D 02   P(1,1,1) =  304.7
++  PF CYCLE  6   STEP   4  DPF = 0.1344D-03   SQSCM = 0.4874D 02 SQSC = 0.4893D 02   P(1,1,1) =  303.6
```

B E G I N N I N G T I M E - S T E P 5

```
ELAPSED TIME =   80.000000     VARIABLE RATE NODES ARE SET AS FOLLOWS

   NODE        RATE          PI              PWF        DUMQ        PRES          VIS         BG         IP  IVQ
  4  4  1  0.102779E 03  0.466600E 00       83.000      83.000     229.200      0.00996  0.666372D 02    0    1

++  PF CYCLE  7   STEP   5  DPF = 0.2954D-03   SQSCM = 0.5758D 02 SQSC = 0.5774D 02   P(1,1,1) =  300.8

NUMBER OF TIME-STEPS COMPLETED    =  5
SIZE OF TIME STEP                 =    40.000000 DAYS
TOTAL ELAPSED TIME                =    80.000000 DAYS   =    0.192000D 04 HOURS
SYSTEM CHANGE THIS STEP           =    -1807.827 MCF
SYSTEM OUTPUT THIS STEP           =     1807.827 MCF
INCREMENTAL ERROR THIS STEP       =        0.001 MCF

NET WELL PRODUCTION THIS STEP     =     4111.152 MCF
NET INJECTION FROM MATRIX         =     2303.324 MCF
NET DESORPTION INTO MATRIX        =     2037.096 MCF

CUMULATIVE WELL PRODUCTION        =     8715.929 MCF
CUMULATIVE INJECTION FROM MATRIX  =     3851.958 MCF
CUMULATIVE DESORPTION             =     3411.451 MCF


WELL INJECTION THIS STEP          =        0.0   MCF
WELL PRODUCTION THIS STEP         =     4111.152 MCF
```

NET WELL PRODUCTION THIS STEP	=	4111.152 MCF
CUMULATIVE NET WELL PRODUCTION	=	8715.929 MCF
TOTAL INITIAL GAS IN PLACE	=	1441558. MCF
FRACTIONAL RESOURCE RECOVERY	=	0.006046 %
NET SYSTEM PRODUCTION THIS STEP	=	1307.827 MCF
CUMULATIVE NET PRODUCTION	=	4863.971 MCF
ACTUAL FRACTURE GAS REMAINING	=	238121. MCF
CALCULATED FRACTURE GAS REMAINING	=	238119. MCF
CUM. MAT. BAL. ERROR (FRACTURES)	=	-0.001154 %

NEW RESERVOIR PRESSURE DISTRIBUTION FOLLOWS

K = 1

304.	304.	304.	304.	304.	304.	305.	305.	305.	305.
304.	304.	304.	304.	304.	304.	304.	305.	305.	305.
303.	303.	303.	303.	303.	303.	304.	304.	305.	305.
302.	302.	301.	300.	301.	302.	303.	304.	304.	305.
301.	299.	296.	294.	296.	300.	302.	303.	304.	304.
299.	296.	288.	278.	289.	296.	301.	303.	304.	304.
298.	293.	277.	229.	278.	294.	300.	303.	304.	304.
298.	295.	288.	277.	288.	296.	301.	303.	304.	304.
300.	298.	295.	293.	296.	299.	302.	303.	304.	304.
301.	300.	298.	298.	299.	301.	302.	303.	304.	304.

NEW RESERVOIR DELTA-PRESSURE DISTRIBUTION FOLLOWS

K = 1

-1	-1	-1	-1	-1	0	0	0	0	0
-1	-1	-1	-1	-1	-1	0	0	0	0
-1	-1	-1	-1	-1	-1	-1	0	0	0
-2	-2	-2	-2	-2	-2	-1	-1	0	0

-3	-3	-4	-4	-3	-3	-2	-1	-1	0
-3	-4	-5	-6	-5	-3	-2	-1	-1	-1
-4	-5	-6	-7	-6	-4	-2	-1	-1	-1
-4	-4	-5	-6	-5	-4	-2	-1	-1	-1
-3	-4	-4	-5	-4	-3	-2	-1	-1	-1
-3	-3	-4	-4	-3	-3	-2	-1	-1	-1

..... SUMMARY OF FLOW RATES THIS STEP (MCFD) .....

WELL PRODUCTION	INJECTION FROM MATRIX	DESORPTION INTO MATRIX	NET FRACTURE SYSTEM PRODUCTION
102.779	57.583	50.9	45.196

80.000000000000	40.000000000000	0.0	4111.151733398437
4111.151733398437	8715.929031372070	323.527847228449	-0.001153820255

4	4	1	229.20	236.15	0.800000D 02	0.949023D 02	0.871593D 04
3	3	1	287.78	293.22	0.800000D 02	-0.294150D 01	0.871593D 04
4	3	1	277.45	283.51	0.800000D 02	-0.393340D 01	0.871593D 04
5	3	1	288.17	293.33	0.800000D 02	-0.283217D 01	0.871593D 04
3	4	1	277.45	283.51	0.800000D 02	-0.393340D 01	0.871593D 04
5	4	1	277.87	283.64	0.800000D 02	-0.381962D 01	0.871593D 04
3	5	1	288.17	293.33	0.800000D 02	-0.283218D 01	0.871593D 04
4	5	1	277.87	283.64	0.800000D 02	-0.381964D 01	0.871593D 04
5	5	1	288.53	293.44	0.800000D 02	-0.273187D 01	0.871593D 04

—— MATRIX PRESSURE DISTRIBUTIONS, SOURCE RATES —

						"QSCM"	"QSC"	"Q"	"QSCD"
RESERVOIR "HISTORY NODE" 4 4 1									
303.4780	303.2848	302.2374	296.0677	259.1798	229.2003	6.4072	6.4120	96.3715	5.6811
RESERVOIR "HISTORY NODE" 3 3 1									
304.7239	304.6874	304.4873	303.2665	295.4065	287.7770	1.7376	1.7412	-1.7376	1.5327
RESERVOIR "HISTORY NODE" 4 3 1									
304.5081	304.4440	304.0951	301.9966	288.9998	277.4505	2.6037	2.6078	-2.6037	2.2981

```
RESERVOIR "HISTORY NODE"   5  3  1
 304.7280    304.6921    304.4952    303.2950    295.5900    288.1700      1.6906      1.6941     -1.6906      1.4911

RESERVOIR "HISTORY NODE"   3  4  1
 304.5081    304.4440    304.0951    301.9966    288.9998    277.45.05     2.6037      2.6078     -2.6037      2.2981

RESERVOIR "HISTORY NODE"   5  4  1
 304.5124    304.4489    304.1033    302.0264    289.1925    277.8678      2.5543      2.5583     -2.5543      2.2544

RESERVOIR "HISTORY NODE"   3  5  1
 304.7280    304.6921    304.4952    303.2950    295.5900    288.1700      1.6906      1.6941     -1.6906      1.4911

RESERVOIR "HISTORY NODE"   4  5  1
 304.5124    304.4439    304.1033    302.0264    289.1925    277.8678      2.5543      2.5583     -2.5543      2.2544

RESERVOIR "HISTORY NODE"   5  5  1
 304.7319    304.6964    304.5026    303.3215    295.7602    288.5333      1.6473      1.6507     -1.6473      1.4530


        $$$$$$$$$$ CURRENT INVENTORY OF GAS IN PLACE(MCF) FOLLOWS $$$$$$$$$$

RESERVOIR GAS              "FREE" MATRIX GAS                  "ADSORBED" MATRIX GAS
-------- ---              ----- ------ --                    --------- ------ --

238121.477                   91599.378                            1103121.

        >>>>>>>> TOTAL REMAINING GAS IN PLACE(MCF)     =   1432842.216

        ........ % GAS REMAINING IN FRACTURE SYSTEM    =    16.62

        ........ % REMAINING FREE MATRIX GAS           =     6.39

        ........ % REMAINING ADSORBED MATRIX GAS       =    76.99


        FRACTION OF ORIGINAL FRACTURE SYSTEM GAS PRODUCED =  0.0200

        FRACTION OF ORIGINAL FREE MATRIX GAS PRODUCED     =  0.0048

        FRACTION OF ORIGINAL ADSORBED GAS PRODUCED        =  0.0031

++  PF CYCLE  8   STEP   6  DPF = 0.2388D-03   SQSCM = 0.6140D 02 SQSC = 0.6152D 02   P(1,1,1) =  297.5
++  PF CYCLE  8   STEP   7  DPF = 0.7147D-03   SQSCM = 0.6325D 02 SQSC = 0.6335D 02   P(1,1,1) =  293.8
++  PF CYCLE  9   STEP   8  DPF = 0.9962D-03   SQSCM = 0.6419D 02 SQSC = 0.6428D 02   P(1,1,1) =  289.7
++  PF CYCLE 11   STEP   9  DPF = 0.7268D-03   SQSCM = 0.6319D 02 SQSC = 0.6327D 02   P(1,1,1) =  285.3
```

B E G I N N I N G T I M E - S T E P 10

ELAPSED TIME = 470.000000 VARIABLE RATE NODES ARE SET AS FOLLOWS

NODE	RATE	PI	PWF	DUMQ	PRES	VIS	BG	IP	IVQ
4 4 1	0.838552E 02	0.466600E 00	83.000	89.479	215.620	0.00987	0.711378D 02	0	1

++ PF CYCLE 11 STEP 10 DPF = 0.6344D-03 SQSCM = 0.6262D 02 SQSC = 0.6270D 02 P(1,1,1) = 281.6

NUMBER OF TIME-STEPS COMPLETED	=	10
SIZE OF TIME STEP	=	100.000000 DAYS
TOTAL ELAPSED TIME	=	470.000000 DAYS = 0.112800D 05 HOURS
SYSTEM CHANGE THIS STEP	=	-2125.337 MCF
SYSTEM OUTPUT THIS STEP	=	2123.251 MCF
INCREMENTAL ERROR THIS STEP	=	-2.086 MCF
NET WELL PRODUCTION THIS STEP	=	8385.518 MCF
NET INJECTION FROM MATRIX	=	6262.267 MCF
NET DESORPTION INTO MATRIX	=	5529.829 MCF
CUMULATIVE WELL PRODUCTION	=	43634.406 MCF
CUMULATIVE INJECTION FROM MATRIX	=	28433.573 MCF
CUMULATIVE DESORPTION	=	25120.057 MCF
WELL INJECTION THIS STEP	=	0.0 MCF
WELL PRODUCTION THIS STEP	=	8385.518 MCF
NET WELL PRODUCTION THIS STEP	=	8385.518 MCF
CUMULATIVE NET WELL PRODUCTION	=	43634.406 MCF
TOTAL INITIAL GAS IN PLACE	=	1441558. MCF
FRACTIONAL RESOURCE RECOVERY	=	0.030269 %
NET SYSTEM PRODUCTION THIS STEP	=	2123.251 MCF
CUMULATIVE NET PRODUCTION	=	15200.833 MCF
ACTUAL FRACTURE GAS REMAINING	=	227785. MCF
CALCULATED FRACTURE GAS REMAINING	=	227778. MCF
CUM. MAT. BAL. ERROR (FRACTURES)	=	-0.003091 %

NEW RESERVOIR PRESSURE DISTRIBUTION FOLLOWS

K = 1

296.	296.	296.	296.	297.	298.	299.	299.	300.	300.
295.	295.	295.	295.	296.	297.	298.	299.	300.	300.
293.	292.	292.	292.	293.	295.	297.	298.	299.	299.
289.	289.	288.	287.	289.	292.	295.	297.	298.	299.
285.	284.	281.	279.	283.	288.	292.	295.	297.	298.
281.	278.	270.	261.	273.	283.	289.	293.	296.	297.
279.	273.	259.	216.	261.	279.	287.	292.	295.	296.
279.	275.	268.	259.	270.	281.	288.	292.	295.	296.
280.	279.	275.	273.	278.	284.	289.	292.	295.	296.
282.	280.	279.	279.	281.	285.	289.	293.	295.	296.

NEW RESERVOIR DELTA-PRESSURE DISTRIBUTION FOLLOWS

K = 1

-2	-2	-2	-2	-2	-2	-2	-2	-1	-1
-2	-2	-2	-2	-2	-2	-2	-2	-1	-1
-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
-3	-3	-3	-2	-2	-2	-2	-2	-2	-2
-3	-3	-3	-3	-3	-2	-2	-2	-2	-2
-3	-3	-3	-3	-3	-3	-2	-2	-2	-2
-3	-3	-3	-2	-3	-3	-2	-2	-2	-2
-4	-4	-3	-3	-3	-3	-3	-2	-2	-2
-4	-4	-4	-3	-3	-3	-3	-2	-2	-2

-4 -4 -4 -3 -3 -3 -3 -2 -2 -2

..... SUMMARY OF FLOW RATES THIS STEP (MCFD) .....

WELL PRODUCTION	INJECTION FROM MATRIX	DESORPTION INTO MATRIX	NET FRACTURE SYSTEM PRODUCTION
83.855	62.623	55.3	21.233

470.000000000000	100.000000000000	0.0	8385.517883300781
8385.517883300781	43634.405746459960	309.477496357885	-0.003090878931

4	4	1	215.62	217.84	0.470000D 03	0.816453D 02	0.436344D 05
3	3	1	267.77	271.13	0.470000D 03	-0.144733D 01	0.436344D 05
4	3	1	258.53	261.65	0.470000D 03	-0.156220D 01	0.436344D 05
5	3	1	270.34	273.31	0.470000D 03	-0.131372D 01	0.436344D 05
3	4	1	258.58	261.65	0.470000D 03	-0.156219D 01	0.436344D 05
5	4	1	261.16	263.85	0.470000D 03	-0.142393D 01	0.436344D 05
3	5	1	270.34	273.31	0.470000D 03	-0.131372D 01	0.436344D 05
4	5	1	261.16	263.85	0.470000D 03	-0.142893D 01	0.436344D 05
5	5	1	272.61	275.26	0.470000D 03	-0.119588D 01	0.436344D 05

—— MATRIX PRESSURE DISTRIBUTIONS, SOURCE RATES —

						"QSCM"	"QSC"	"Q"	"QSCD"
RESERVOIR "HISTORY NODE" 4 4 1									
268.4440	267.8597	264.9423	252.7935	224.8912	215.6198	2.0260	2.0267	81.8291	1.8097
RESERVOIR "HISTORY NODE" 3 3 1									
292.3250	292.0857	290.8847	285.7594	272.8872	267.7731	1.1563	1.1573	-1.1563	1.0226
RESERVOIR "HISTORY NODE" 4 3 1									
287.8961	287.5955	286.0902	279.7291	264.3545	258.5801	1.2983	1.2992	-1.2983	1.1502
RESERVOIR "HISTORY NODE" 5 3 1									
292.9811	292.7575	291.6364	286.8625	274.9949	270.3368	1.0552	1.0561	-1.0552	0.9327
RESERVOIR "HISTORY NODE" 3 4 1									
287.8962	287.5955	286.0903	279.7291	264.3546	258.5801	1.2983	1.2993	-1.2983	1.1503
RESERVOIR "HISTORY NODE" 5 4 1									
288.5472	288.2624	286.8370	280.8280	266.4696	261.1569	1.1968	1.1977	-1.1968	1.0598
RESERVOIR "HISTORY NODE" 3 5 1									

292.9811	292.7575	291.6364	286.8626	274.9950	270.3369	1.0552	1.0561	-1.0552	0.9327

RESERVOIR "HISTORY NODE" 4 5 1

288.5472	288.2624	286.8370	280.8280	266.4697	261.1570	1.1968	1.1977	-1.1968	1.0598

RESERVOIR "HISTORY NODE" 5 5 1

293.5702	293.3607	292.3108	287.8498	276.8709	272.6136	0.9660	0.9668	-0.9660	0.8535

$$$$$$$$$$ CURRENT INVENTORY OF GAS IN PLACE(MCF) FOLLOWS $$$$$$$$$$

RESERVOIR GAS	"FREE" MATRIX GAS	"ADSORBED" MATRIX GAS
227784.615	88726.370	1081413.

>>>>>>>> TOTAL REMAINING GAS IN PLACE(MCF) = 1397923.740

..... % GAS REMAINING IN FRACTURE SYSTEM = 16.29

..... % REMAINING FREE MATRIX GAS = 6.35

..... % REMAINING ADSORBED MATRIX GAS = 77.36

FRACTION OF ORIGINAL FRACTURE SYSTEM GAS PRODUCED = 0.0626

FRACTION OF ORIGINAL FREE MATRIX GAS PRODUCED = 0.0360

FRACTION OF ORIGINAL ADSORBED GAS PRODUCED = 0.0227

```
++ PF CYCLE 11   STEP  11  DPF = 0.5731D-03   SQSCM = 0.6190D 02 SQSC = 0.6197D 02   P(1,1,1) =  278.3
++ PF CYCLE 11   STEP  12  DPF = 0.5286D-03   SQSCM = 0.6104D 02 SQSC = 0.6111D 02   P(1,1,1) =  275.3
++ PF CYCLE 11   STEP  13  DPF = 0.4923D-03   SQSCM = 0.6010D 02 SQSC = 0.6017D 02   P(1,1,1) =  272.5
++ PF CYCLE 11   STEP  14  DPF = 0.4633D-03   SQSCM = 0.5912D 02 SQSC = 0.5919D 02   P(1,1,1) =  269.9
```

B E G I N N I N G T I M E - S T E P 15

```
          ELAPSED TIME =  970.000000     VARIABLE RATE NODES ARE SET AS FOLLOWS

     NODE       RATE          PI              PWF       DUMQ       PRES          VIS        BG        IP  IVQ
   4  4  1  0.750779E 02  0.466600E 00       83.000     89.479     206.838      0.00980  0.744121D 02   0    1

++  PF CYCLE 11   STEP  15  DPF = 0.4517D-03   SQSCM = 0.5814D 02 SQSC = 0.5821D 02   P(1,1,1) =  267.4


          NUMBER OF TIME-STEPS COMPLETED    =  15
          SIZE OF TIME STEP                 =    100.000000 DAYS
          TOTAL ELAPSED TIME                =    970.000000 DAYS   =    0.232800D 05 HOURS
          SYSTEM CHANGE THIS STEP           =     -1695.393 MCF
          SYSTEM OUTPUT THIS STEP           =      1693.454 MCF
          INCREMENTAL ERROR THIS STEP       =        -1.939 MCF

          NET WELL PRODUCTION THIS STEP     =      7507.793 MCF
          NET INJECTION FROM MATRIX         =      5814.338 MCF
          NET DESORPTION INTO MATRIX        =      5145.317 MCF

          CUMULATIVE WELL PRODUCTION        =     82786.791 MCF
          CUMULATIVE INJECTION FROM MATRIX  =     58463.984 MCF
          CUMULATIVE DESORPTION             =     51667.647 MCF


          WELL INJECTION THIS STEP          =         0.0   MCF
          WELL PRODUCTION THIS STEP         =      7507.793 MCF
          NET WELL PRODUCTION THIS STEP     =      7507.793 MCF
          CUMULATIVE NET WELL PRODUCTION    =     82786.791 MCF

          TOTAL INITIAL GAS IN PLACE        =      1441558. MCF
          FRACTIONAL RESOURCE RECOVERY      =      0.057429 %

          NET SYSTEM PRODUCTION THIS STEP   =      1693.454 MCF
          CUMULATIVE NET PRODUCTION         =     24322.806 MCF
          ACTUAL FRACTURE GAS REMAINING     =       218663. MCF
          CALCULATED FRACTURE GAS REMAINING =       218645. MCF
          CUM. MAT. BAL. ERROR (FRACTURES)  =     -0.007948 %


                    NEW RESERVOIR PRESSURE DISTRIBUTION FOLLOWS

K = 1


     286. 286. 286. 287. 288. 289. 290. 292. 292. 293.
```