FOSSIL ENERGY

I-12560

DR-2002-8

DOE/BC/10640-21

COMPUTER SIMULATION OF RECOVERY OF HEAVY CRUDE OIL USING CARBON DIOXIDE DRIVE OR HUFF-N-PUFF

Final Report

Work Performed for the Department of Energy
Under Contract No. DE-AS19-81BC10640

Date Published—November 1983

DO NOT MICROFILM COVER

Colorado State University
Fort Collins, Colorado

U. S. DEPARTMENT OF ENERGY

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DOE/BC/10640-21
Distribution Category UC-92a

COMPUTER SIMULATION OF RECOVERY OF HEAVY CRUDE OIL USING CARBON DIOXIDE DRIVE OR HUFF-N-PUFF

Final Report

DOE/BC/10640--21

DE84 002484

By

Roland Begin
and
David A. Krueger, *Principal Investigator*
Department of Physics
Colorado State University
Fort Collins, Colorado 80523

Kenneth L. Spence, *Technical Project Officer*
Bartlesville Energy Technology Center
P.O. Box 1398
Bartlesville, Oklahoma 74005

Work Performed for the Department of Energy
Under Contract No. DE-AS19-81BC10640

Date Published—November 1983

UNITED STATES DEPARTMENT OF ENERGY

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Table of Contents

I. Introduction 1

II. Overview of Model 5

III. Mathematical Development of the Model 11
- A. Governing Equations 11
- B. Pressure Equation Derivation 15
- C. Finite Differencing of Equations 19

IV. Solution Method 25

V. Phase Properties 29
- A. Thermodynamic Equilibria 29
- B. Density 32
- C. Viscosity 35
- D. Saturation 36
- E. Compressibility 37
- F. Relative Permeability 39

VI. Sample Runs 42

VII. Discussion and Conclusions 48

VIII. Nomenclature 51

IX. References 53

X. Appendix 55
- A. Sample Input Format 55
- B. Sample Output Format 58
- C. Source Code for CO2CYCL 62

List of Figures and Tables

Figure II-1. Model geometry 8

Figure III-1. Flow geometry 12

Figure III-2. Grid detail 20

Figure IV-1. Solution flow chart 28

Figure V-1. Fluid phase behavior 30

Figure V-2. Relative permeability 40

Table VI-1. Results of Sample Runs 43

Table VI-2. Gas front location during sample runs 44

Acknowledgment

The authors thank F. C. Orr, Jr. for providing a copy of his computer program for simulating one-dimensional convection of four phase, four component mixtures.

I. Introduction

It has been estimated by the U.S. Bureau of Mines that the total amount of heavy oil deposits in this country exceeds 150 billion barrels (bbl).[1] This resource includes 55.5 billion bbl of oil with an API gravity less than 20° and 21.3 billion bbl that lie in shallow reservoirs which are less than 1500 feet deep. Forty percent of the proven heavy crude deposits are in Kansas, Missouri, and Oklahoma in reservoirs that are shallower than 500 feet deep.[2] Most of the remaining deposits are located in deeper reservoirs located in California and Louisiana. Given the great variety of conditions that can exist in different reservoirs and the importance of the economic and environmental considerations involved in tertiary recovery operations, the development of a number of alternative technologies to recover this oil is crucial.

Many enhanced recovery techniques exist.[3,4] Waterflooding, for example, has been a standard secondary recovery practice for decades. However, by itself it is seldom useful in the recovery of highly viscous crude oils. This is due to the adverse mobility ratio of water to such crudes. In general, a mobility ratio, $(K_w/\mu_w)/(K_o/\mu_o)$, greater than one is considered unfavorable since water will usually finger through the oil at such ratios resulting in poor oil recovery.

Immiscible gas displacement processes (e.g., using air as the displacing fluid), like waterfloods, have been utilized in recovery operations for many years. Although such processes have even less favorable mobility ratios than do waterfloods, they have the advantage that the injected gas, being lighter than oil or water, will generally form a gas cap, thereby forcing the oil to migrate downdip to a well.

In contrast to immiscible displacements where one fluid phase displaces another fluid phase without mixing between the phases, in miscible displacements no distinct interface exists between the displaced and displacing fluids. That is, miscibility implies that the two fluids can mix in all proportions. Generally, the recovery of heavy oil requires the application of such miscible displacement processes. Chemical solvents, dry gas, enriched gas, and carbon dioxide are typical displacing agents in miscible floods. Often, to improve the recovery economics, a small quantity of miscible fluid is driven by a large slug of immiscible fluid. Dry gas displacements are generally applicable only in reservoirs where pressures are in excess of 2500 psi, whereas enriched gas and chemical solvent displacements are useful down to roughly 1200 psi.[4]

Displacement processes that use CO_2 as the displacing agent can be classed as either miscible or partially miscible. Complete miscibility of CO_2 with oil is not generally achieved unless the reservoir pressure is greater than about 1500 psi. This minimum miscibility pressure (MMP) will vary considerably depending upon both the oil composition and the reservoir temperature. The displacement mechanisms most important in CO_2 processes are the reduction of the oil viscosity and swelling of the oil. These will be active to some degree even at pressures below the MMP. Various techniques have been tried using CO_2. These include carbonated water-flooding, water-alternated-with-gas (WAG) injection, CO_2 mixed with flue exhaust gases, and continuous CO_2 injection. Because the solubility of CO_2 in oil increases with pressure, the application of CO_2 processes have so far been generally limited to relatively deep reservoirs (>2000 feet).

The recovery of heavy crude from shallow deposits such as those described above presents some unique problems to be investigated. At these depths the maximum pressure that the reservoir can withstand without fracturing is usually less than 500 psi (assuming a pressure gradient of 1 psi/foot of depth). Since the solubility of CO_2 in oil increases with pressure this restriction is an important one. This situation could be improved somewhat if the injectivity of the reservoir were increased by hydraulically fracturing the formation. It is plausible to assume that injection of CO_2 into such fractures would serve to stimulate the oil in the near vicinity of the fracture. The resultant increase in swelling and decreased viscosity of the oil phase near the fracture should help displace the oil toward the fracture. As injection of CO_2 continues and the oil and water become saturated, a mobile gas phase will develop. The appearance of gas in the pores blocks the flow of oil and water reducing their relative permeabilities drastically. However, assuming a favorable ratio of vertical to horizontal absolute permeability, then since the gas is less dense than the oil it will tend to percolate up through the reservoir eventually forming a secondary gas cap at the top of the formation. This gas cap will then aid in the displacement of oil toward the fracture and the well bore.

Because of the enormously complex nature of the system described above, the present study is limited to an investigation of the behavior of the flows in and near a fracture. The numerical model simulates multicomponent, multiphase, compressible flow through a horizontal two-dimensional porous medium which is bounded on one side by a one-dimensional fracture. The absolute permeability of the fracture is assumed to be much greater (five orders of magnitude or more) than that

of the reservoir matrix, resulting in the simplification that flow in the matrix is predominately perpendicular to the fracture face. Flow in the matrix parallel to the fracture is ignored in this model.

The fluid system consists of three components in three phases. Component 1 is the injected fluid, e.g., CO_2, and can exist in any of the three phases - gas, oil, and water. Component 2 is the hydrocarbon component and is assumed to be heavy enough so that it exists only in the oil phase. Component 3 is the aqueous component and is restricted to the water phase. All fluid properties, except viscosity and relative permeability, are assumed to be linear functions of pressure and composition. The temperature of the system is taken as constant and effects due to capillary pressure and gravity are not included.

The mathematical formulation is based on a set of molar continuity equations (one per component), the phase equilibrium condition, and a volume conservation equation. Reduction of this system of equations to a single equation in pressure leads to a sequential (implicit pressure-explicit moles) method of solution. In-step iterations are performed to increase the implicitness of the method. The algorithm includes time step control and a volume balance check.

A flow chart of the computer code as well as the source code and input and output formats are included in the Appendix. Three sample problems are examined.

II. Overview of Model

The fundamental relation describing flow in a porous medium is Darcy's Law which simply states that the volumetric flow rate, $\vec{Q}$, through a porous medium is directly proportional to the pressure gradient. For single phase, single component flow this relation has the form:

$$\vec{v} = -(K/\mu)(\nabla P - \rho g \nabla D) \qquad \text{(II-1)}$$

where:

$\vec{v} = \vec{Q}/A$ = the apparent macroscopic velocity (also known as the superficial or Darcy velocity),

K = absolute permeability of the medium,

μ = viscosity of fluid,

P = pressure,

ρ = fluid density,

g = acceleration due to gravity,

D = depth,

A = contact area.

In multiphase, multi-component flow each fluid phase, e.g., gas, oil, and water, is assumed to be individually governed by Darcy's Law and each component is subject to the constraint of mass conservation. Darcy's Law for some phase j may be written:

$$\vec{v}_j = -(k_{rj} K/\mu_j)(\nabla P_j - \rho_j g \nabla D) \qquad \text{(II-2)}$$

where

k_{rj} = relative permeability of phase j (dimensionless and $0 < k_{rj} < 1$),

μ_j, ρ_j = viscosity and density of phase j.

In general, the pressures in the phases will not be equal due to the presence of capillary pressure, i.e., the pressure difference across the interface separating two fluid phases. If, however, the capillary pressure is small compared to the externally imposed pressure gradient then the pressure in the different phases are approximately the same and $P_j \rightarrow P$.

Conservation of mass of component i can be expressed:

$$D_t(\phi \Sigma_j x_{ij} \rho_j S_j) = -\nabla \cdot \Sigma_j x_{ij} \rho_j \vec{v}_j \qquad \text{(II-3)}$$

where

x_{ij} = the mole fraction of component i in phase j,

ϕ = porosity,

S_j = saturation of phase j,

D_t = the partial derivative operator with respect to t.

In this model, we assume that the system is a horizontal plane and we can therefore neglect gravity in our formulation. We further assume that the capillary pressure is zero.

There will be an equation like Eq. (II-3) for each component of the system. These equations must be discretized and solved for each grid block in the system (see Sec. III). The geometry of the system and its corresponding discrete representations are illustrated in Fig. II-1.

Finally, we observe that if the permeability of the fracture is much larger than that of the rock matrix, and the well is located as indicated in the figure, then although the flow in the fracture will be both parallel and perpendicular to the fracture axis the flow in the matrix will be primarily perpendicular to the fracture. That is, even though we are considering a two-dimensional geometry the flow in each matrix "branch" will be essentially one-dimensional. This results in a substantial simplification of the solution algorithm.

The conditions leading to this simplified flow geometry can be made more quantitative. The basic constraint is that the flow in the reservoir matrix perpendicular to the fracture (y-direction) be much greater than that parallel to the fracture (z-direction):

$$q_{Ry} \gg q_{Rz} \quad \text{or} \quad \Delta z D_y(P_R) \gg \Delta y D_z(P_R) \tag{II-4}$$

This implies that the pressure gradients in the z-direction in both the matrix and the fracture are approximately the same:

$$D_z(P_R) \approx D_z(P_F) \tag{II-5}$$

where

P_R = pressure in the rock matrix,

P_F = pressure in the fracture.

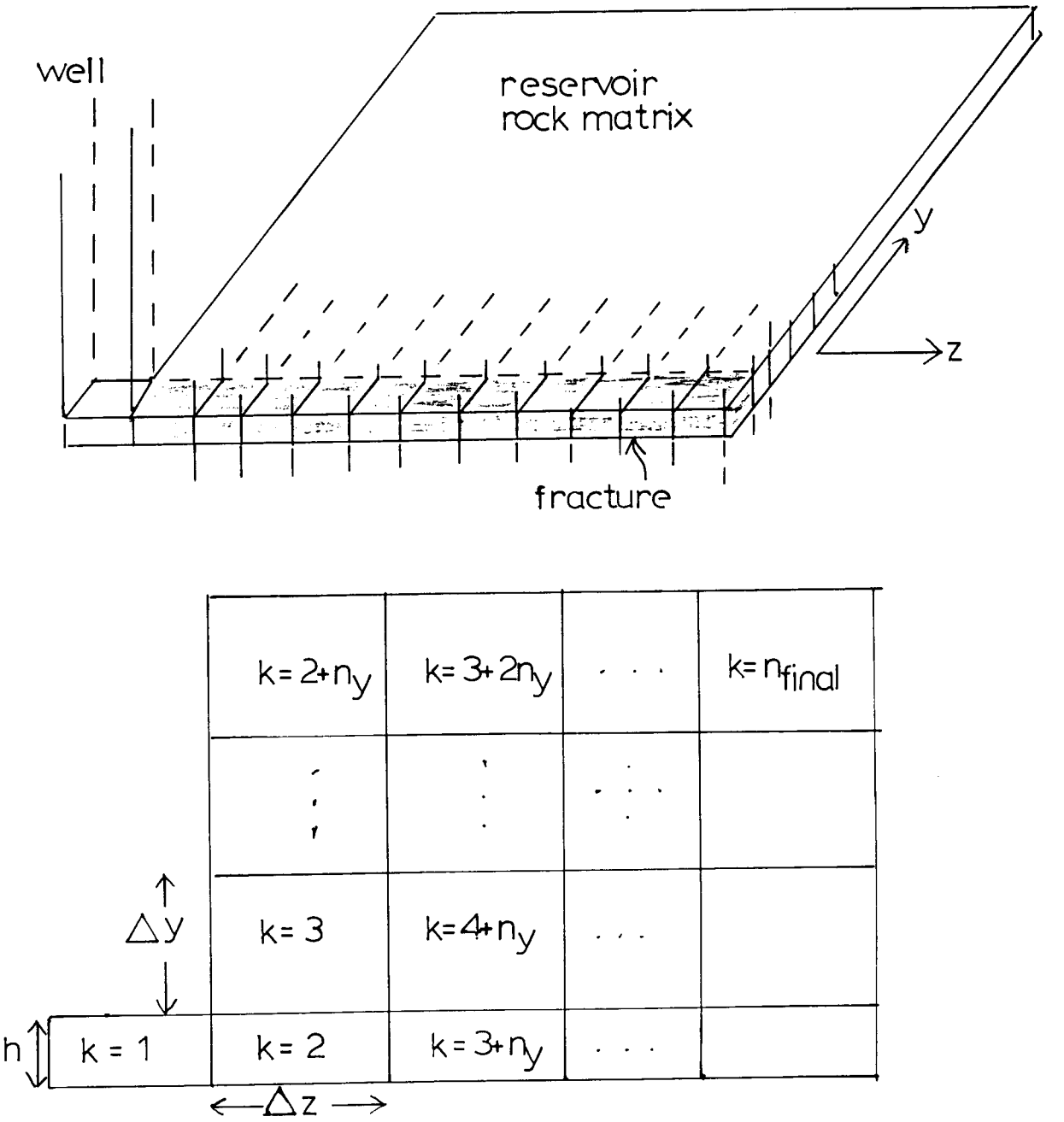

Figure II-1. Model geometry.

If we also assume that the flow into each matrix branch is roughly equal then we get:

$$\Delta z(K_R/\mu)D_y(P_R) \approx h(K_F/\mu)D_z(P_F)/n_z \qquad \text{(II-6)}$$

where

h = fracture width,

n_z = number of grid blocks in fracture.

Combining relations (II-4), (II-5), and (II-6), yield the condition:

$$h/(n_y \Delta y) >> K_R/K_F$$

where

n_y = number of grid blocks in matrix branch .

If the fracture is approximated by two closely spaced parallel plates then its permeability is given by:[5]

$$K_F \approx h^2/12 \quad .$$

Thus, the flow condition reduces to:

$$h^3/12K_R >> \ell_R \quad .$$

If $h \sim 1$ mm $= 10^{-3}$m and $K_R \sim 10^2$ md $= 10^{-9}$ cm^2, then

$$h^3/12K_R = 10^5 \text{ cm} >> \ell_R$$

which is easily satisfied for simulations of flows in laboratory cases. Therefore, for fractures having high permeabilities the simplified flow geometry described above is clearly reasonable.

With the preliminary exposition of the concepts of flow behavior and component mass continuity complete, we are ready to begin the mathematical construction of our model.

III. Mathematical Development of the Model

A. Governing Equations

The development of the numerical model begins with the molar balance of each component in each of the system's grid blocks. The molar continuity of component i in the kth fracture grid block is given by:

$$[\Delta(\Sigma_j N_{ij})/\Delta t]_k = [\Sigma_j x_{ij}\rho_j q_{zj}]_{k-\Delta z/2} - [\Sigma_j x_{ij}\rho_j q_{zj}]_{k+\Delta z/2} - [\Sigma_j x_{ij}\rho_j q_{yj}]_{k+h/2} \qquad \text{(III-1)}$$

where:

N_{ij} = number of moles of component i in phase j,

x_{ij} = molar fraction of i in j,

ρ_j = molar density of phase j,

q_{zj}, q_{yj} = volumetric flow rates of phase j along the fracture axis (z-direction) and perpendicular to the fracture (y-direction).

The flow geometry is indicated in Fig. III-1.

For convenience later in reducing the initial system of equations to a single equation in pressure, we transform the discrete form of the continuity equation (Eq. (1)) into the corresponding differential form:

$$[D_t(\Sigma_j N_{ij}]_k = -\Delta z[D_z(\Sigma_j x_{ij}\rho_j q_{zj})]_k - [\Sigma_j x_{ij}\rho_j q_{yj}]_{k+h/2} \quad . \qquad \text{(III-2a)}$$

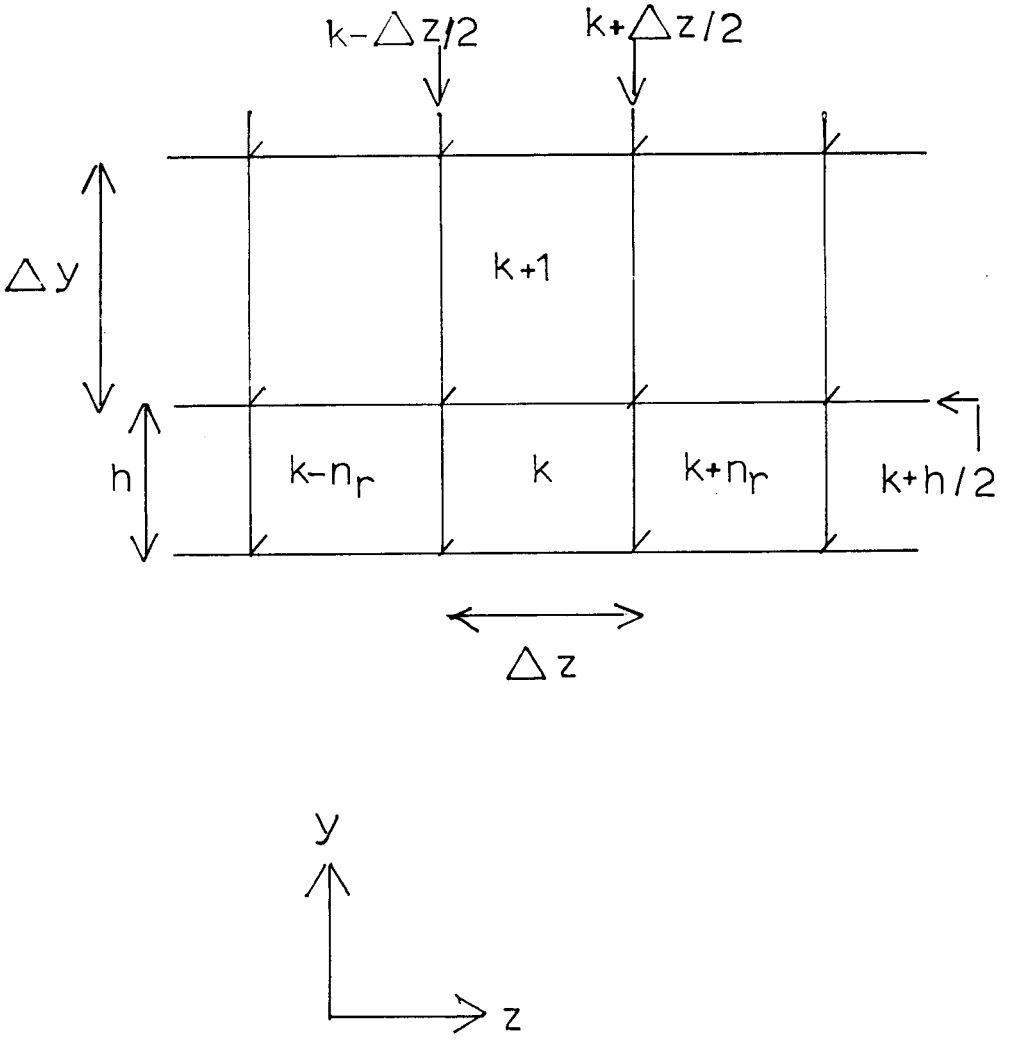

Figure III-1. Flow geometry.

Here D_t and D_z are differential operators representing the partial derivative of the argument with respect to the operator subscript.

Similar reasoning leads to an equation of continuity for component i valid for blocks in the reservoir matrix:

$$[D_t(\Sigma_j N_{ij})]_k = -\Delta y[D_y(\Sigma_j x_{ij}\rho_j q_{yj})]_k \quad . \qquad \text{(III-2b)}$$

There are three continuity equations for each grid block in the system. For example, in the fracture:

$$CO_2(i=1): \quad [D_t(N_{11}+N_{12}+N_{13})]_k =$$

$$-\Delta z[D_z(W_{1z})]_k - [W_{1y}]_{k+h/2} \qquad \text{(III-3)}$$

$$HC(i=2): \quad [D_t(N_{22})]_k = -\Delta z[D_z(W_{2z})]_k - [W_{2y}]_{k+h/2} \qquad \text{(III-4)}$$

$$H_2O(i=3): \quad [D_t(N_{33})]_k = -\Delta z[D_z(W_{3z})]_k - [W_{3y}]_{k+h/2} \qquad \text{(III-5)}$$

where:

$$W_{1z} = \rho_1 q_{1z} + x_{12}\rho_2 q_{2z} + x_{13}\rho_3 q_{3z}$$

$$W_{2z} = x_{22}\rho_2 q_{2z}$$

$$W_{3z} = x_{33}\rho_3 q_{3z}$$

with similar expressions for W_{1y}, W_{2y}, and W_{3y}.

For a compressible fluid system we require that the total volume of the fluids be conserved. If the rock matrix is incompressible then the pore volume is constant and the above condition constrains the total fluid volume to be equal to the pore volume at all times:

$$[V_{pore}]_k = [V_{fluid}]_k = [N_{11}V_{11}+N_{12}V_{12}+N_{13}V_{13}+N_{22}V_{22}+N_{33}V_{33}]_k \quad \text{(III-6)}$$

where:

V_{ij} = volume of a mode of component i in phase j.

Finally, the components are distributed among the three phases so as to insure thermodynamic equilibrium. We consider two possible cases: (a) the oil and water phases are not completely saturated with CO_2, therefore there is no mobile gas phase present; (b) the oil and water are saturated and a mobile gas phase is present. This implies (see Sec. V-A on phase equilibria) for case (a):

$$\frac{x_{12}}{x_{13}} = \frac{\theta_{12}}{\theta_{13}} = \eta = \text{constant} \quad \text{(III-7a)}$$

where $\theta_{12}P$ and $\theta_{13}P$ are the molar solubilities of CO_2 in the oil and water, respectively, and for case (b):

$$x_{12} = \theta_{12}P$$

$$x_{13} = \theta_{13}P \quad \text{(III-7b)}$$

Observe that Eqs. (III-3) to (III-6) and either (III-7a) or (III-7b) can be combined and reduced to a single equation in pressure (see next section). For N grid blocks there will be N such equations in pressure which must be solved simultaneously.

A set of semi-empirical equations relating fluid properties, such as density, viscosity, relative permeability, etc., to the pressure and composition of the system completes the mathematical description of the model.

B. Pressure Equation Derivation

As mentioned above, Eqs. (III-3) to (III-7) can be reduced to a single equation in pressure. This equation has the following form in the fracture:

$$AD_t(P) = -\Delta z[D_z(W_{1z})+\delta_2 D_z(W_{22}) + \delta_3 D_z(W_{3z})] - [W_{1y}+\delta_2 W_{2y}+\delta_3 W_{3y}] \qquad \text{(III-8)}$$

where A, δ_2, and δ_3 are time-lagged coefficients which are dependent upon the particular phase equilibrium condition considered.

For case (a), where no mobile gas is present in the grid block, we find that:

$$A = C_{eff}(A'/A'')$$

$$C_{eff} = -[N_{11}D_P(V_{11})+N_{12}D_P(V_{12})+N_{13}D_P(V_{13})$$

$$+ N_{22}D_P(V_{22}) + N_{33}D_P(V_{33})]$$

$$A' = (N_{13}+N_{33})x_{22} + \eta(N_{12}+N_{22})x_{33} \qquad \text{(III-8a)}$$

$$A'' = V_{13}(N_{13}+N_{33})x_{22} + \eta V_{12}(N_{12}+N_{22})x_{33}$$

$$\delta_2 = [V_{22}A' + \eta N_{13}(V_{12}-V_{13})]/A''$$

$$\delta_3 = [V_{33}A'-N_{12}(V_{12}-V_{13})]/A'' \quad .$$

When a mobile gas phase is present we have case (b) and:

$$A = (C_{eff} + C')/V_{11}$$

$$C' = (V_{11}-V_{12})(N_{12}+N_{22})\theta_{12}/(1-\theta_{12}P)$$

$$+ (V_{11}-V_{13})(N_{13}+N_{33})\theta_{13}/(1-\theta_{13}P) \qquad \text{(III-8b)}$$

$$\delta_2 = [V_{22}(1-\theta_{12}P) + (V_{12}-V_{11})\theta_{12}P]/[V_{11}(1-\theta_{12}P)]$$

$$\delta_3 = [V_{33}(1-\theta_{13}P) + (V_{13}-V_{11})\theta_{13}P]/[V_{11}(1-\theta_{13}P)]$$

Note that in Eqs. (III-8a) and (III-8b) above, C_{eff} is the effective compressibility of the fluid system.

Now, using Darcy's Law (neglecting gravity), we can substitute into Eq. (III-8):

$$q_j = \tau v_j = -\tau\lambda_j \nabla P$$

where:

τ = cross-sectional area perpendicular to flow (τ_F = fracture cross-sectional area, τ_R = reservoir branch cross-sectional area),

λ_j = mobility of phase j,

$\lambda_{jz} = (k_{rj}/\mu_j)K_{frac}$,

$\lambda_{jy} = (k_{rj}/\mu_j)K_{res}$.

For the fracture block adjacent to the well block the total volumetric flow rate across the well-fracture interface is a constant value set by the user of the model. Thus, for k = 2:

$$[AD_t(P)]_k = [W_{1z}+\delta_2 W_{2z}+\delta_3 W_{3z}]_{k-\Delta z/2}$$
$$+ \tau_F[w_{1z}+\delta_2 w_{2z}+\delta_3 w_{3z}]_{k+\Delta z/2} D_z(P)$$
$$+ \tau_R[w_{1y}+\delta_2 w_{2y}+\delta_3 w_{3y}]_{k+h/2} D_y(P) \qquad \text{(III-9)}$$

where:

$w_{1z} = \rho_1\lambda_{1z}+x_{12}\rho_2\lambda_{2z}+x_{13}\rho_3\lambda_{3z}$,

$w_{2z} = x_{22}\rho_2\lambda_{2z}$,

$w_{3z} = x_{33}\rho_3\lambda_{3z}$.

Note that the injection term $[\ ]_{k-\Delta z/2}$ can be written:

$$[\ ] = \rho_1 q_{1z} + (x_{12}+\delta_2 x_{22})\rho_2 q_{2z} + (x_{13}+\delta_3 x_{33})\rho_3 q_{3z}$$

$$= [\rho_1 f_1 + (x_{12}+\delta_2 x_{22})\rho_2 f_2 + (x_{13}+\delta_3 x_{33})\rho_3 f_3]Q$$

where:

f_j = fractional flow of phase j ($f_j = \lambda_j/\Sigma_j\lambda_j$),

Q = total volumetric flow rate.

For the remaining fracture blocks the pressure equation is given by:

$$[AD_t(P)]_k = \Delta z \tau_F [D_z(w_{1z}D_z(P)) + \delta_2 D_z(w_{2z}D_z(P)) + \delta_3 D_z(w_{3z}D_z(P))]_k$$

$$+ \tau_R[(w_{1y}+\delta_2 w_{2y}+\delta_3 w_{3y})D_y(P)]_{k+h/2} \quad . \qquad \text{(III-10)}$$

Finally, the pressure equation for blocks in the reservoir matrix branches is:

$$[AD_t(P)]_k = \Delta y \tau_R [D_y(w_{1y}D_y(P)) + \delta_2 D_y(w_{2y}D_y(P))$$

$$+ \delta_3 D_y(w_{3y}D_y(P))]_k \quad . \qquad \text{(III-11)}$$

The above pressure equations, Eqs. (III-9), (III-10), and (III-11), along with the continuity equations, Eqs. (III-3), (III-4), and (III-5) are now transformed into finite difference equations for a numerical solution for the pressures and compositions in each grid block.

C. Finite Differencing of Equations

The differential equations presented above are differenced according to the following approximations:

$$[D_t(P)]_k \approx (P_k^{\ell+1}-P_k^{\ell})/\Delta t$$

$$[D_z(P)]_{k+\Delta z/2} \approx (P_{k+n_r}^{\ell+1}-P_k^{\ell+1})/\Delta z$$

$$[D_y(P)]_{k+\Delta y/2} \approx (P_{k+1}^{\ell+1}-P_k^{\ell+1})/\Delta y$$

$$[D_z(P)]_{k-\Delta z/2} \approx (P_k^{\ell+1}-P_{k-n_r}^{\ell+1})/\Delta z$$

$$[D_y(P)]_{k-\Delta y/2} \approx (P_k^{\ell+1}-P_{k-1}^{\ell+1})/\Delta y$$

$$D_z(GD_z(P))_k \approx [G_+^{\ell}P_{k+n_r}^{\ell+1}-(G_+^{\ell}+G_-^{\ell})P_k^{\ell+1}+G_-^{\ell}P_{k-n_r}^{\ell+1}]/(\Delta z)^2$$

$$D_y(GD_y(P))_k \approx [G_+^{\ell}P_{k+1}^{\ell+1}-(G_+^{\ell}+G_-^{\ell})P_k^{\ell+1}+G_-^{\ell}P_{k-1}^{\ell+1}]/(\Delta y)^2 \quad .$$

The superscripts ℓ and $\ell+1$ denote, respectively, the current and advanced time levels. The subscripts refer to either grid block or grid interface location (Fig. III-1 and III-2). The G coefficients are evaluated "upstream" so if the flow in the figure is left to right then $G_+ \rightarrow G_k$ and $G_- \rightarrow G_{k-1}$, whereas if the flow is to the left, $G_+ \rightarrow G_{k+1}$ and $G_- \rightarrow G_k$.

We assume that flow into or out of the fracture is governed primarily by the pressure drop in the rock matrix. Therefore the pressure gradient across the fracture-matrix interface is approximated by:

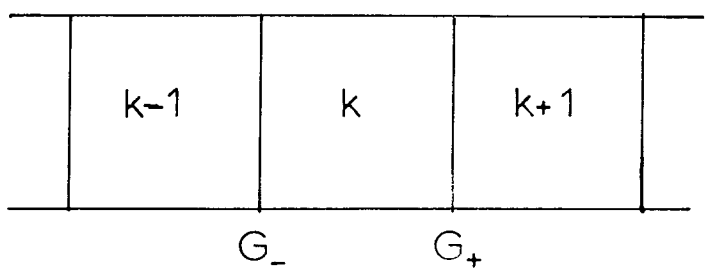

Figure III-2. Grid detail.

$$[D_y(P)]_{k+h/2} = (P_{k+1}^{\ell+1} - P_k^{\ell+1})/(\Delta y/2) \quad .$$

Substitution of the above difference approximations into Eq. (III-10) yields a difference equation in pressure valid for fracture blocks not adjacent to the well block:

$$\alpha_k^\ell P_{k-n_r}^{\ell+1} + \beta_k^\ell P_k^{\ell+1} + \gamma_k^\ell P_{k+n_r}^\ell + \varepsilon_k^\ell P_{k+1}^{\ell+1} = \theta_k^\ell \quad . \qquad \text{(III-12)}$$

where:

$$\alpha_k^\ell = r_z \tau_F a_{k-}^\ell$$

$$\gamma_k^\ell = r_z \tau_F a_{k+}^\ell$$

$$\varepsilon_k^\ell = r_y \tau_R e_{k+}^\ell$$

$$\beta_k^\ell = -\{r_z \tau_F (a_{k+}^\ell + a_{k-}^\ell) + r_y \tau_R e_{k+}^\ell + A_k^\ell\}$$

$$\theta_k^\ell = -A_k^\ell P_k^\ell$$

with:

$$a_{k+}^\ell = w_{1z+}^\ell + \delta_2^\ell w_{2z+}^\ell + \delta_3^\ell w_{3z+}^\ell$$

$$a_{k-}^\ell = w_{1z-}^\ell + \delta_2^\ell w_{2z-}^\ell + \delta_3^\ell w_{3z-}^\ell$$

$$e_{k+}^\ell = w_{1y+}^\ell + \delta_2^\ell w_{2y+}^\ell + \delta_3^\ell w_{3y+}^\ell$$

$$r_z = \Delta t/\Delta z, \quad r_y = 2\Delta t/\Delta y$$

and A, δ_2, δ_3 and the W's were defined earlier.

The difference equation for the fracture block adjacent to the well is obtained by substituting the difference approximations into Eq. (III-9) yielding:

$$\beta_2^{\ell} P_2^{\ell+1} + \gamma_2^{\ell} P_{2+n_r}^{\ell+1} + \varepsilon_2^{\ell} P_3^{\ell+1} = \theta_2^{\ell} \qquad \text{(III-13)}$$

where γ and ε have the same definitions as in the previous case and:

$$\beta_2^{\ell} = -\{r_z \tau_F a_{k+}^{\ell} + r_y \tau_R e_{k+}^{\ell} + A_k^{\ell}\}$$

$$\theta_2^{\ell} = -A_2^{\ell} P_2^{\ell} - \underline{Q}^{\ell}$$

$\underline{Q}^{\ell}$ = number of moles flowing into or out of well block during a time step.

Note that during injecion of a single fluid we have:

$$\underline{Q}^{\ell} = \Delta t [\rho_j]_{k=1}^{\ell} Q_{in}$$

where Q_{in} is the user-defined, total volumetric injection rate.

During the production phase of the huff-n-puff cycle this becomes:

$$\underline{Q}^{\ell} = \Delta t \{\rho_1 f_1 + (x_{12} + \sigma_2 x_{22}) \rho_2 f_2 + (x_{13} + \delta_3 x_{33}) \rho_3 f_3\}_{k=2}^{\ell} Q_{out}$$

where Q_{out} is the volumetric production rate.

Substitution of the appropriate difference approximations into Eq. (III-11) gives the pressure difference equation for matrix grid blocks:

$$\alpha_k^\ell P_{k-1}^{\ell+1} + \beta_k^\ell P_k^{\ell+1} + \varepsilon_k^\ell P_{k+1}^{\ell+1} = \theta_k^\ell$$

where:

$$\alpha_k^\ell = r_y \tau_R e_{k-}^\ell$$

$$\varepsilon_k^\ell = r_y \tau_R e_{k+}^\ell$$

$$\beta_k^\ell = -[r_y \tau_R (e_{k+}^\ell + e_{k-}^\ell) + A_k^\ell]$$

$$\theta_k^\ell = -A_k^\ell P_k^\ell$$

with:

$$e_{k+}^\ell = w_{1y+}^\ell + \delta_2^\ell w_{2y+}^\ell + \delta_3^\ell w_{3y+}^\ell$$

$$e_{k-}^\ell = w_{1y-}^\ell + \delta_2^\ell w_{2y-}^\ell + \delta_3^\ell w_{3y-}^\ell$$

Note that $r_y = \Delta t/\Delta y$ except for the matrix branch grids blocks adjacent to the fracture where $r_y = 2\Delta t/\Delta y$. Consistency with the assumption described above on the pressure gradient governing flow between the fracture and the matrix requires that the w's in the relevant equations contain the absolute permeability of the matrix. This requirement includes the case where fluid is moving from the fracture to the matrix and the relative mobility, λ_j, of the fluid is evaluated "upstream" in the fracture block.

Finally, the continuity or composition equations ((III-3), (III-4), (III-5)) are differenced explicitly. That is, the flow rates q_j in those equations are rewritten as $f_j Q$ where the Q's are calculated from the solutions just obtained to the matrix of pressure equations

discussed above. Following the numbering order discussed in Sec. II, this pressure matrix equation has the form:

$$
\begin{bmatrix}
\beta_2 & \varepsilon_2 & - & - & \gamma_2 & - & \cdots & & \\
\alpha_3 & \beta_3 & \varepsilon_3 & - & - & - & \cdots & & \\
- & \alpha_4 & \beta_4 & \varepsilon_4 & - & - & \cdots & & \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & & \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & & \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdots & & \\
\alpha_{2+n_r} & - & - & - & \beta_{2+n_r} & \varepsilon_{2+n_r} & - & - & \gamma_{2+n_r} \\
- & - & - & - & \alpha_{3+n_r} & \beta_{3+n_r} & \varepsilon_{3+n_r} & - & - \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{bmatrix}
\begin{bmatrix}
P_2 \\ P_3 \\ P_4 \\ \cdot \\ \cdot \\ \cdot \\ P_{2+n_r} \\ P_{3+n_r} \\ \cdot \\ \cdot \\ \cdot
\end{bmatrix}
=
\begin{bmatrix}
\theta_2 \\ \theta_3 \\ \theta_4 \\ \cdot \\ \cdot \\ \cdot \\ \theta_{2+n_r} \\ \theta_{3+n_r} \\ \cdot \\ \cdot \\ \cdot
\end{bmatrix}
$$

Utilizing the tridiagonal nature of segments of this matrix a modified Gaussian elimination algorithm was used to solve this set of equations.

IV. Solution Method

The solution approach employed in this numerical model is sequential. We implicitly solve the matrix of differenced pressure equations and use the solutions to calculate enter-block flow rates. These flow rates are then substituted into the explicitly differenced set of continuity equations to give the overall component composition of each block at the new time level.

To improve the accuracy of the pressure and composition solutions we use an in-step substitution iteration procedure. This consists of using the most recently computed compositions to predict the relative permeabilities and other fluid properties at the advanced time step and then substituting these values back into the coefficients of the pressure matrix and continuity equations and recalculating pressures and compositions. The procedure is repeated until the predicted relative permeabilities satisfy a pre-set convergence criterion.

There are a number of possible sources of error related to the length of the time step that are inherent in sequential solution algorithms. These include time truncation errors, numerical stability of the solutions, overshooting, and continuity of the solution at the bubble point. We have included several time step controls in the model to reduce the effect of these errors.

The time truncation error is most easily controlled by limiting the magnitude of change of some important variable during any time step.[6] We choose to limit the pressure rise allowed during a step. This is readily accomplished by reducing the step size by a factor DPLIM/DPMAX where DPMAX is the maximum pressure change calculated for a step and DPLIM is the maximum change allowed and is defined by the model user.

Overshooting occurs when the flow into a block is less than the flow out of a block and the time step is so long that the block is completely emptied and a negative composition results. This occurs most frequently in the presence of a mobile gas phase. When such a condition is present, the step is halved and the entire calculation for that step is repeated.

The numerical stability limitation is due to the explicit calculation of relative permeabilities. Although we use in-step substitution iterations to improve the relative permeability values, convergence of those values will not occur in such an iterative process for any time step that violates the following condition:[7]

$$2\Delta t D_{sg}(f_g)[|q_y|+|q_z|]/V_p \leq 1$$

where

V_p = pore volume of a block,

q = volumetric flow rate,

$D_{sg}(f_g)$ = the change in the fractional flow of gas with respect to a change in gas saturation during a time step,

Δt = length of time step.

At present we set the well flow rates based upon the above stability constraint. However, this is not sufficient to control the size of $D_{sg}(f_g)$ at the bubble point where this derivative can be relatively large. A subroutine which would incorporate the above stability criterion into the model is included in the listing of the source code (see Appendix), however, it is currently not completely operational and therefore not being implemented.

Finally, if during a time step a mobile gas phase appears in some block k, then to insure a continuous transition from the immobile gas case to the mobile gas case, the step length must be sufficiently short so that only a very small amount of mobile gas is present at the end of the step. This is necessary because the thermodynamic equilibrium conditions that govern the two cases are quite different. Too long a step could result in a solution which was inconsistent with the equilibrium assumption based on conditions at the beginning of the step. We have included in the model a time step control at the bubble point governing the appearance of a mobile gas phase in a block. This control needs to be generalized to include the possibility of the gas phase disappearing during a step as well.

Finally, a volume balance check is performed during each step by comparing the total fluid volume with the pore volume in each grid block. This provides a rough measure of the accuracy of the solutions obtained.

A flow chart of the solution scheme is included in Fig. IV-1.

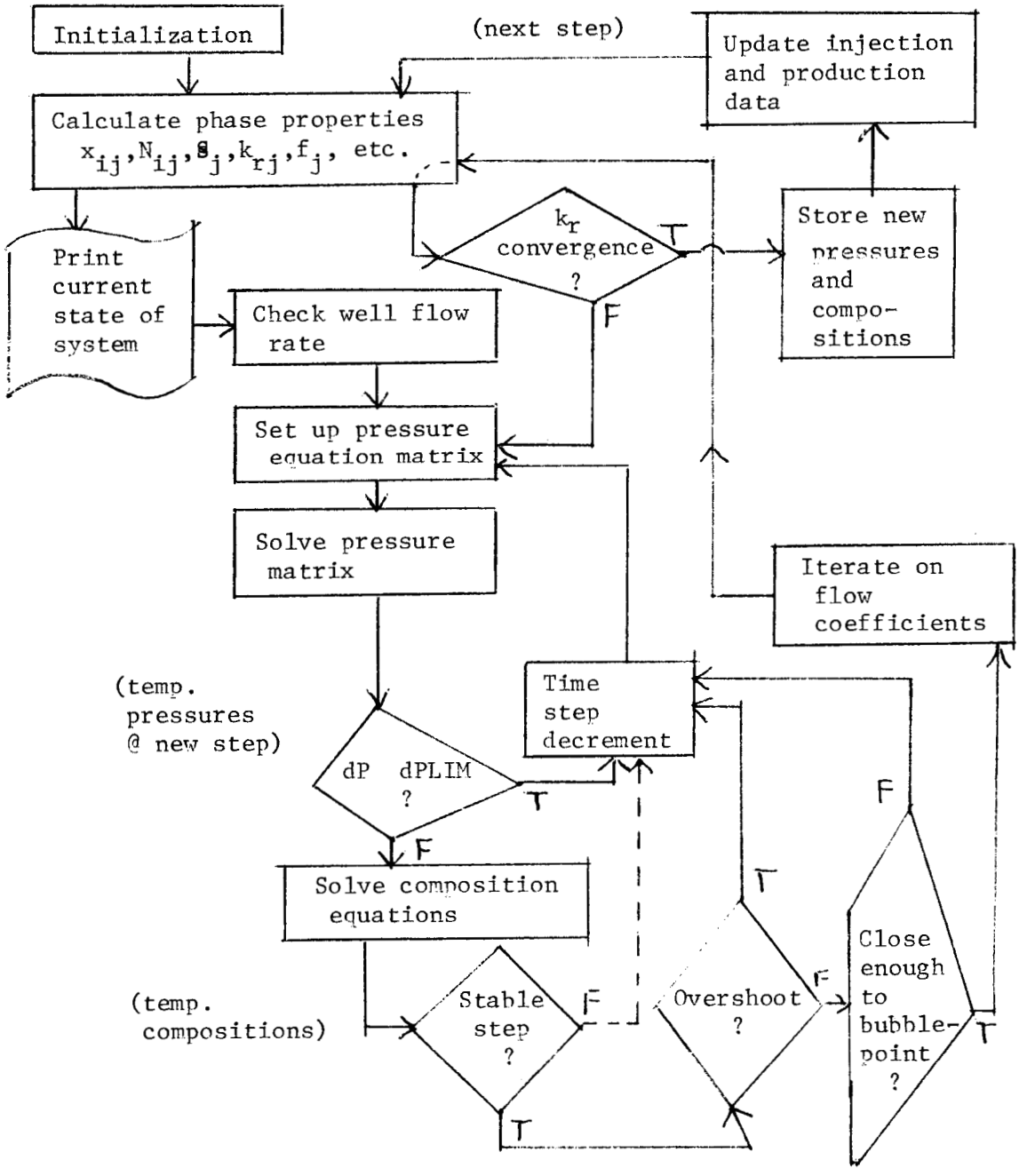

Figure IV-1. Solution flow chart.

V. Phase Properties

A. Thermodynamic Equilibria

The phase behavior of the system modelled by this simulator involves mass transfer between three phases (gas, oil, water) and may be represented on a ternary phase diagram (see Fig. V-1). Phase equilibrium is subject to the following restrictions:

(1) Of the three components present in the system (CO_2, hydrocarbons (HC), and H_2O), only the transfer of CO_2 molecules is assumed to occur between the gas and the two liquid phases.

(2) The carbonated oil and carbonated water solutions are assumed to be "weak".

Assumption (1) implies both immiscibility between the oil and water phases and the absence of any extraction of HC and H_2O molecules into a vapor phase. The latter conclusion is reasonable since we are considering for the purposes of this model only "heavy" oils, that is oils of 20° API gravity and below, which generally contain little or no "light" components. Furthermore, this implies that the gas phase is composed of only CO_2.

Assumption (2) is justified since the primary focus of the model is to investigate the flow behavior of oil and water in shallow reservoirs, where the relatively low formation fracturing pressure limits the useful pressure range to about 500 psia and below. At such low pressures the carbonated oil solutions (and, of course, the carbonated water) can be considered to be weak due to the relatively low solubility of CO_2 (<20 mole % in oil).

Assumption (2) then leads to two important conclusions: first, Henry's Law states that the solubility of a weak solution of gas in a

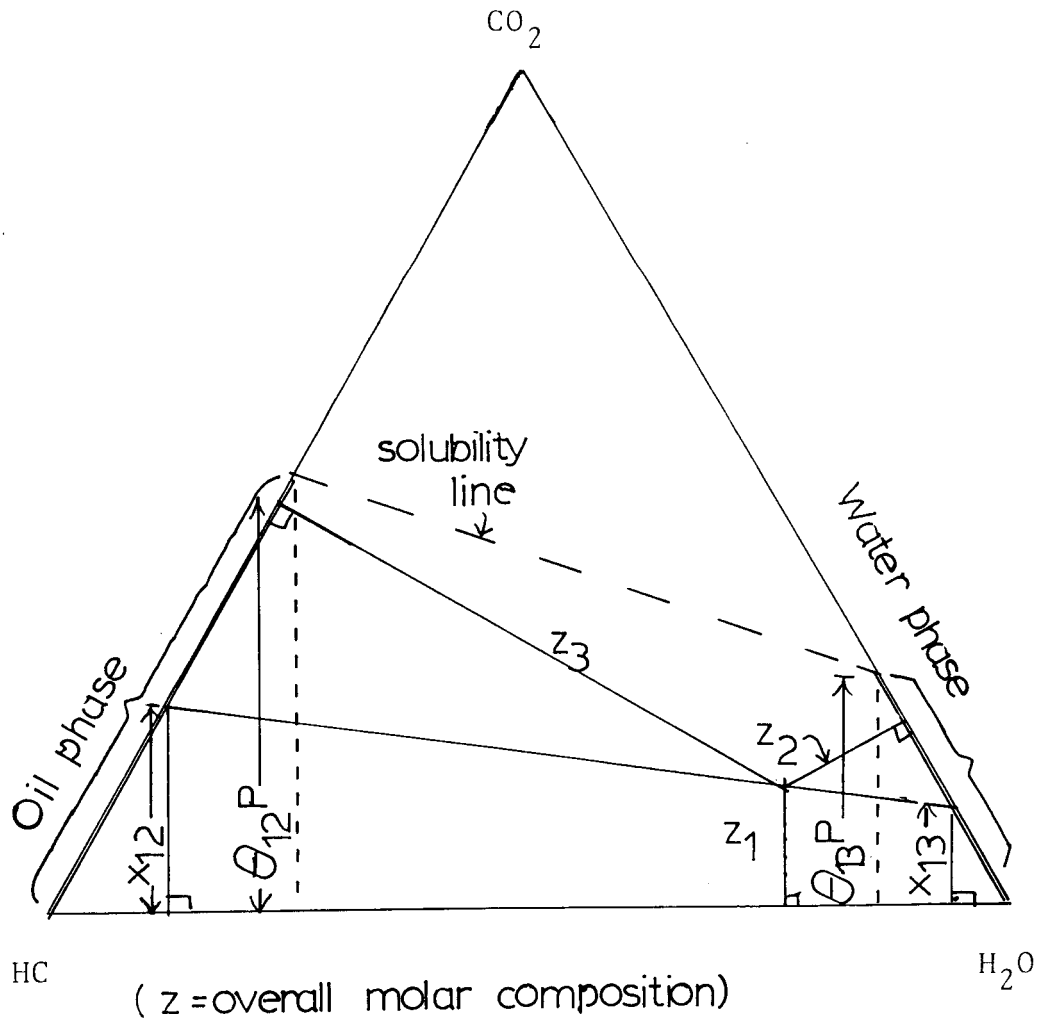

Figure V-1. Fluid phase behavior.

liquid at low pressures is proportional to the gas pressure.[8] Thus, the solubility of CO_2 in the oil and water is approximately linear with pressure. Second, the ratio of the concentrations of a solute in two different solvents (again, assuming "weak" solutions) is constant for a given pressure and temperature (distribution law).[8] This latter conclusion, coupled with the assumptions of solubility linear with pressure and isothermal conditions, leads to the result that the ratio of the concentrations (and the solubilities) is a constant over the pressure range considered above.

Phase equilibrium calculation:

We determine how injected gas is partitioned among the three phases at equilibrium conditions for a given overall composition characterized by a point on the ternary phase diagram (see Fig. V-1). If the composition point, depicted by Z on the diagram, is above the solubility line (which is a function of pressure alone) then three mobile phases - gas, oil, and water - exist. If the composition point is below the line then only the oil and water are mobile and any gas that is present is either in solution or is immobile residual gas. We will henceforth refer to the latter case, where no mobile gas phase is present, as case A and to the case where mobile gas is present as case B.

For case A, the equilibrium condition is given by:

$$x_{12}/x_{13} = \theta_{12}/\theta_{13} = \eta$$

where $\theta_{12}P$ and $\theta_{13}P$ are the solubilities of CO_2 in oil and water, respectively, as functions of the average pressure, P and η is the constant solubility ratio, θ_{12}/θ_{13}.

For case B, the condition is simply:

$$x_{12} = \theta_{12}P$$

$$x_{13} = \theta_{13}P$$

i.e., the mole fractions of CO_2 in oil and water are equal to their saturated values.

If z_i is the overall mole fraction of component i in the system and the index i = 2,3 refers to oil and water, respectively, then it can be shown[16] that the concentration of CO_2 in water with no attendant solubility restriction is:

$$x_{13} = \{[1+\eta z_1+z_2(\eta-1)] - \sqrt{[1+\eta z_1+z_2(\eta-1)]^2-4z_1\eta}\}/2\eta \quad .$$

If $x_{13} < \theta_{13}P$ then the CO_2 is all in solution, no mobile gas exists, and the equilibrium condition of case A can be applied to determine x_{12}. Then, using the constraint that $x_{12}+x_{22} = x_{13}+x_{33} = 1$, x_{22} and x_{33} can be calculated.

If, however, $x_{13} \geq \theta_{13}P$ then the liquid phases are saturated and CO_2 must begin to bubble out of solution. Therefore, the equilibrium condition of case B must be invoked, where then $x_{12} = \theta_{12}P$ and $x_{13} = \theta_{13}P$. Any remaining CO_2 must enter a mobile gas phase.

B. Density

Gas phase density is calculated using the compressibility equation of state:

$$PV = nzRT$$

where:

P = pressure,

V = volume,

n = number of moles,

z = compressibility factor,

R = universal gas constant = 8.3 J/mol°K,

T = temperature (°K).

Assuming that T = 35°C then the molar density of the gas is given by:

$$\rho_1 = P/zRT = 2.7 \times 10^{-6}\ P/z$$

where P is in psia and z is assumed to be a linear function of pressure for $P < 500$ psi.[9]

The calculation of liquid (oil or water) phase densities requires experimental data on the density vs. pressure behavior of the pure liquid and of the CO_2 saturated liquid.[10-15] This information, coupled with the assumption that a molecule of CO_2 dissolved in the liquid occupies the same volume in the unsaturated liquid as it does in the saturated liquid allows us to calculate the molar density of the liquid phases via the following relation:

$$\rho_j = [x_{ij}V_{ij}+x_{jj}V_{jj}]^{-1}$$

where:

j denotes the phase = 2 for oil, = 3 for water,

x_{ij} is the mole fraction of CO_2 in phase j,

x_{jj} is the mole fraction of oil in the oil phase or of water in the water phase,

V_{ij} is the specific molar volume of the component i located in phase j.

The specific volumes of oil and water, V_{22} and V_{33}, are obtained by inverting the mass density of the respective pure component and multiplying by the appropriate molecular weight. Mole fractions are calculated based on the most recent pressure and composition solutions.

The specific volume of CO_2 in oil, V_{12}, is calculated utilizing experimental data on the swelling factor, f, of CO_2-saturated oil. By definition, the volume of a quantity of saturated oil is equal to the product of the swelling factor, the number of moles of dead oil, and the specific volume of the oil at standard conditions. The swelling factor is assumed to be linear with pressure. V_{12} is then found to be given by:

$$V_{12} = (1-\theta_{12}P)[fV_{22}(P=1\ \text{atm})-V_{22}(P)]/(\theta_{12}P) \ .$$

The specific volume of CO_2 in water requires data on the mass densities of pure and CO_2-saturated water at various pressures. We find that:

$$V_{13} = (1-\theta_{13}P)[m_3/\rho_3',\text{sat}-V_{33}]/(\theta_{13}P) + m_1/\rho_3',\text{sat}$$

where:

m_1, m_3 = molecular weights of CO_2 and H_2O,

ρ_3', sat = mass density of CO_2-saturated water.

C. Viscosity

In this model the viscosities of the CO_2 gas phase and the water phase are assumed to be constant. Since the gas is pure and the solubility of CO_2 in water is relatively small this assumption is a reasonable approximation to the actual behavior of the system under consideration. On the other hand, the oil phase viscosity is assumed to be a function of both pressure and composition. Over a limited range of pressure and dissolved gas composition it has been observed that the log of the oil viscosity varies linearly with the pressure and the amount of gas dissolved in the oil.[11,12] Thus, we approximate the oil viscosity by the following relation:

$$\log \mu_2 = AP + Bx_{12} + C$$

where:

μ_2 = oil phase viscosity (cp).

The constants A, B, and C must be determined from experimental data describing the variation of the viscosity of pure and CO_2-saturated oil with pressure. A is the slope of the $\log \mu_2$ vs. P data for pure oil. C is the intercept of that graph with the $\log \mu_2$ axis. B relates the viscosity to the oil composition and is given by $B = (A'-A)/\theta_{12}$, where

A' is the slope of the CO_2-saturated oil viscosity curve and θ_{12} is the molar solubility of CO_2 in the oil.

The user must determine A, A', and C from available experimental data and input these values into the model (see INPUT formats). Note that in general A' < A and B < 0.

D. Saturation

Knowledge of the equilibrium composition of the system and of the specific volumes of the components leads directly to phase saturations.

The total fluid volume is given by:

$$V_f = \sum_{i,j} N_{ij} V_{ij}$$

Phase saturations are then given by:

$$S_j = \sum_i N_{ij} V_{ij} / V_f$$

Thus:

$$\text{gas} - S_1 = N_{11} V_{11} / V_f$$

$$\text{oil} - S_2 = (N_{12} V_{12} + N_{22} V_{22}) / V_f$$

$$\text{water} - S_3 = (N_{13} V_{13} + N_{33} V_{33}) / V_f$$

where:

$$V_f = N_{11} V_{11} + N_{12} V_{12} + N_{13} V_{13} + N_{22} V_{22} + N_{33} V_{33} \cdot$$

E. Compressibility

The effective compressibility of the fluid system is defined by:

$$C = C_{eff}/V_f$$

where:

$$C_{eff} = -[N_{11}D_P(V_{11}) + N_{12}D_P(V_{12}) + N_{13}D_P(V_{13}) + N_{22}D_P(V_{22}) + N_{33}D_P(V_{33})]$$

V_f = the total fluid volume .

Assume that CO_2 behaves like a compressible ideal gas, i.e., PV = nzRT (see Sec. V.B). We expand the volume in a Taylor series:

$$V = V_o + [D_P(V)]_{P_o}\Delta P + \frac{1}{2}[D_P^2(V)]_{P_o}(\Delta P)^2 + \ldots$$

Rearranging:

$$C = -\Delta V/(V_o\Delta P) \cong -\{[D_P(V)]_{P_o} + [D_P^2(V)]_{P_o}\Delta P/2\}/V_o \quad .$$

Applying the first and second derivatives to the above equation of state gives:

$$C = [1-\Delta P/P_o]/(zP_o) \quad .$$

Thus, the first term in the expression for C_{eff} can be written:

$$N_{11}V_{11}[1-\Delta P/P_o]/(zP_o)$$

where

P_o = the most recently computed value for the pressure,

ΔP = the change in the pressure from the last step to the present step.

The second and third terms in C_{eff} involve the compressibility of dissolved gas in oil and water. Referring back to the section on densities it is easily seen that:

$$-N_{12}D_P(V_{12}) = [a_s V_{22}(1)+(V_{22}(1)-V_{22}(P))/(\theta_{12}P^2)]N_{12}$$

where a_s is the slope of the swelling factor f,

$$-N_{13}D_P(V_{13}) = [m_3 a_3 + (m_1-m_3)a_3^{sat}]/(b_3)^2$$

where a_3^{sat} and a_3 are the slopes of mass density of saturated and pure water, respectively, and b_3 is the density of water at atmospheric pressure.

Finally, the last two terms of C_{eff} represent the compressibilities of pure oil and pure water. Noting that:

$$V_{22} = m_2/(a_2P + b_2)$$

$$V_{33} = m_3/(a_3P + b_3)$$

where a_2, a_3 and b_2, b_3 are, respectively, the slopes and intercepts of the oil and water mass density curves, we get:

$$-N_{22}D_P(V_{22}) = a_2N_{22}V_{22}^2/m_2$$

$$-N_{33}D_P(V_{33}) = a_3N_{33}V_{33}^2/m_3 \quad .$$

F. Relative Permeability

The three-phase relative permeabilities of gas, oil and water are calculated from either functional representations of two-phase relative permeabilities or from experimental data obtained on two-phase systems. The general form of the two-phase relative permeability curves is shown in Fig. V-2.

In the figure, k_{rw} and k_{row} are, respectively, the relative permeabilities of water and of oil to water in the oil-water system. In the oil-gas system, k_{rg} and k_{rog} are the relative permeabilities of gas and of oil to gas. When the water saturation S_w is reduced to its connate value S_{wc} its relative permeability k_{rw} is zero. When $S_w = 1-S_{or}$, where S_{or} is the residual oil saturation, k_{rw} reaches its endpoint value E_w. Similarly, the oil relative permeability k_{row} reaches its endpoint value E_{ow} when $S_w = S_{wc}$ and is zero when $S_w = 1-S_{or}$.

In the oil-gas system, S_L is the liquid saturation and equals the sum of the oil and water saturations. When $S_L = S_{Lc} = S_{wc} + S_{og}$, where S_{og} is the residual oil to gas, the relative permeability of the gas k_{rg} is equal to its endpoint value E_g and is zero when the gas saturation S_g equals the critical gas saturation S_{gc}. The oil endpoint of the oil-gas curve is E_{og} and for consistency should equal E_{ow}.

The relative permeability of a phase can be considered to be a function of the effective saturation of that phase with the following general form:[16,17]

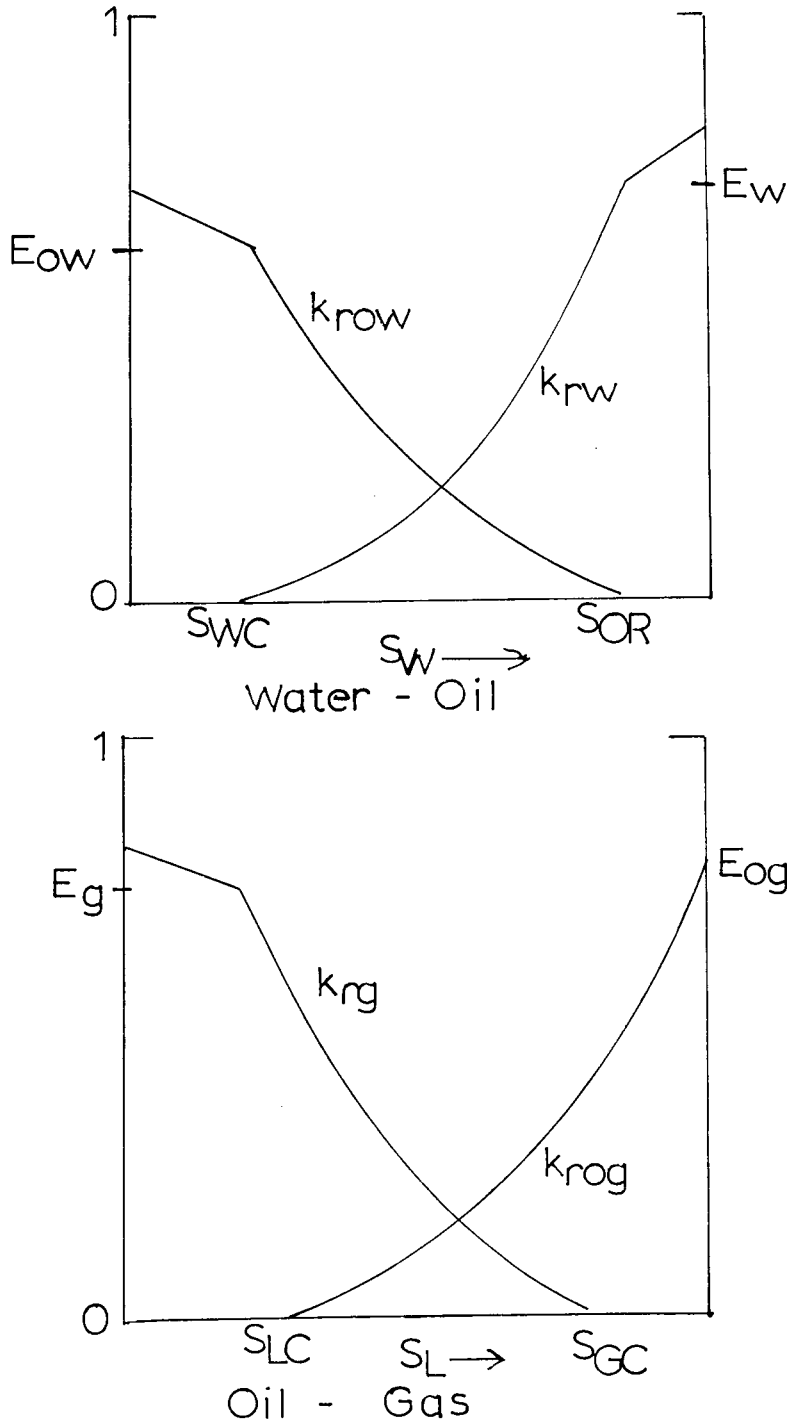

Figure V-2. Relative permeability.

$$k_{rj} = E_j (S_{eff})^{e_j}$$

There are four such functions needed in this model. The endpoint values E_j and the exponents e_j are set by the user. In the oil-water system, for j = w the effective saturation S_{eff} is given by $(S_w-S_{wc})/(1-S_{or}-S_{wc})$ and for j = ow it is $(1-S_w-S_{or})/(1-S_{or}-S_{wc})$. In the oil-gas system, for j = og, $S_{eff} = (S_L-S_{Lc})/(1-S_{Lc})$ and for j = g, $S_{eff} = (S_g-S_{gc})/(1-S_{Lc}-S_{gc})$, where $S_L = 1-S_g$ and $S_{Lc} = S_{wc}+S_{og}$.

Three-phase relative permeabilities are then calculated from the two-phase permeabilities using Stone's second model[18] with a slight modification when $EK_{row} < 1$. The water and gas permeabilities are equal to their respective two-phase values and the oil permeability is given by:

$$k_{ro} = (k_{row}+k_{rw})(k_{rog}+k_{rg})/E_{og}-(k_{rw}+k_{rg}) \quad .$$

Note that for $S_o < S_{or}$, $k_{row} = 0$ and k_{rw} is obtained by linearly interpolating between E_w and 1. Similarly, for $S_w < S_{wc}$, $k_{rw} = 0$ and k_{row} is obtained by interpolation and for $S_L < S_{LC}$, $k_{rog} = 0$ and k_{rg} is evaluated by interpolation.

Finally, the relative mobilities λ_j of the three phases are given by:

$$\lambda_j = k_{rj}/\mu_j$$

and the fractional flow rates f_j by

$$f_j = \lambda_j/\Sigma_j\lambda_j \quad .$$

VI. Sample Runs

Three sample problems were run using this numerical model. In each of these problems the following input parameters were held constant.

Number of grid blocks in fracture (NBZ) = 4
Number of grid blocks in matrix branch (NBY) = 3
Length of fracture (FRLTH) = 90.0 cm
Length of matrix branch (RSLTH) = 2.0 cm
Fracture width (HFRAC) = 0.1 cm
Vertical thickness of system (DEPTH) = 10.0 cm
Fracture porosity (PORF) = 1.0
Matrix porosity (PORR) = 0.3
Absolute permeability of fracture (PRMF) = 10^7 md
Absolute permeability of matrix (PRMR) = 10^2 md
Initial reservoir pressure (PRSI) = 15Ø
Initial time step length (DT) = 3.0 min
Maximum pressure change allowed per step (DPLIM) = 5Ø
Injection flow rate (QINJ1) = 2.Ø cm^3/min
Production flow rate (QPROD) = Ø.1 cm^3/min
Initial matrix composition (:vol%), 5% CO_2, 45% HC, 50% H_2O
Solubility of CO_2 in oil = (OILSOL*P) = 4.54×10^{-4} P
Oil density: 3.53×10^{-6} P + 0.9943
Saturated H_2O density: 1.68×10^{-4} P + .9943
CO_2 viscosity = .015 c.p.
H_2O viscosity = 1.0 c.p.
Oil viscosity = $10^{(1.09 \times 10^{-4} P - 3.985 x_{12} + 2.699)}$ c.p.

Note: oil densities, solubilities, viscosity data taken from Miller.[12]

For the following runs, well cycling control is pressure limited. That is, the well is switched to injection mode when the pressure drops to some minimum value (in this case 150 psi) and to production mode when the pressure rises to some maximum value (250 psi for these runs). A brief soak period occurs between the injection and production periods. Table VI-1 contains the accumulated values of moles of CO_2 input and moles of CO_2, oil, and water output at the end of each injection/production cycle, for each of the three cases examined. Table VI-2 shows the location of the gas front at the end of each cycle for the three runs made.

Table VI-1. Results of Sample Runs*

	Run 1	Run 2	Run 3
Cycle 1			
CO_2 injected	.0586	.0547	.0546
CO_2 produced	.0248	.0214	.0194
Oil produced	1.03×10^{-4}	2.11×10^{-4}	6.38×10^{-4}
Water produced	.163	.587	.424
Cycle 2			
CO_2 injected	.143	.106	.140
CO_2 produced	.0774	.0578	.072
Oil produced	2.88×10^{-4}	3.36×10^{-4}	9.47×10^{-4}
Water produced	.455	.898	.632
Cycle 3			
CO_2 injected	.215	.166	
CO_2 produced	.133	.104	
Oil produced	4.65×10^{-4}	4.5×10^{-4}	
Water produced	.735	1.167	
Cycle 4			
CO_2 injected		.229	
CO_2 produced		.161	
Oil produced		6.5×10^{-4}	
Water produced		1.686	

*See text for details of run parameters. Quantities are in units of accumulated moles.

Table VI-2. Gas Front Location During Sample Runs

	Run 1	Run 2	Run 3
Cycle 1	2 6 10 14 3 7 11 15 4 8 12 16 5 9 13 17	2 6 10 14 3 7 11 15 4 8 12 16 5 9 13 17	2 6 10 14 3 7 11 15 4 8 12 16 5 9 13 17
Cycle 2	2 6 10 14 3 7 11 15 4 8 12 16 5 9 13 17	2 6 10 14 3 7 11 15 4 8 12 16 5 9 13 17	2 6 10 14 3 7 11 15 4 8 12 16 5 9 13 17
Cycle 3	2 6 10 14 3 7 11 15 4 8 12 16 5 9 13 17	2 6 10 14 3 7 11 15 4 8 12 16 5 9 13 17	
Cycle 4		2 6 10 14 3 7 11 15 4 8 12 16 5 9 13 17	

Results are given describing the efficiency of the recovery process and the fraction of CO_2 trapped in the reservoir for each of the three sample cases. Recovery efficiency is calculated in thousands of standard cubic feet (MSCF) of CO_2 injected and not recovered per barrel (bbl) of oil recovered. The fraction of CO_2 trapped is a dimensionless quantity calculated from the moles of CO_2 injected less the moles of CO_2 recovered divided by the moles of CO_2 injected.

Run 1

For this case, it was assumed that CO_2 was soluble in water, i.e., the molar solubility of CO_2 in H_2O at pressure P was given by WATSOL*P→ 2.1×10^{-5} P. Furthermore, the fracture was assumed to initially contain 50% oil and 50% water.

Three cycles were run for this case. The recovery factor, R, at the completion of the first cycle, which lasted 76.8 minutes, was calculated from the data in Table VI-1 to be 217 MSCF/bbl. The fraction of CO_2 trapped, F, was found to be 58%. The second cycle lasted 124.7 minutes, with R = 150 MSCF/bbl and T = 46% at the end of the cycle. The third cycle continued for 116.3 minutes, at the end of which R = 116 MSCF/bbl and T = 38%.

Run 2

Here, CO_2 was not allowed to dissolve into the water. This was done by lowering the solubility factor four orders of magnitude from the value used in Run 1. The initial composition of the fracture was the same as in Run 1. Four cycles were completed.

The first cycle lasted 77.2 minutes, at the end of which R = 104 MSCF/bbl and T = 61%. The second cycle lasted 79.8 minutes, with

R = 90.9 MSCF/bbl and T = 37% at its conclusion. The fourth and final cycle lasted 109.3 minutes. The final recovery factor and fraction of trapped gas were, respectively, R = 69.Ø MSCF/bbl and T = 3Ø%.

Run 3

In this last run, the solubility of CO_2 in water was increased to the level used in Run 1 and the initial composition of the fracture was changed to 70% volume saturation of oil and 30% volume saturation of water. Only two cycles were run for this case.

The first cycle lasted 66.6 minutes. R was then found to equal 36.4 MSCF/bbl and T to be 65%. At the end of the second cycle, which lasted 128.8 minutes, R = 47.4 MSCF/bbl and T = 49%.

Analysis and Conclusions

Reduction of the solubility of CO_2 in pure water had the expected effect of allowing the CO_2 gas to migrate further into the fracture/matrix system than for the case of CO_2 being allowed to dissolve normally into pure water. Since reservoir brines generally have varying levels of dissolved salts which act to reduce the CO_2 solubility of water the actual CO_2 gas migration will be intermediate to that indicated by Runs 1 and 2.

If the reservoir is assumed to be water-wet then the initial composition of a fracture could well lie between the oil-water volume ratio of 50/50 used in Run 2 and 70/30 used in Run 3. The latter case has clearly the most optimistic projection of MCF/bbl of the three runs made.

In the first two runs each succeeding injection/production cycle resulted in an improved recovery factor and a lower amount of

trapped gas. However, in the final run the opposite effect was observed. It would be interesting to investigate the effect of a larger number of cycles per run on the recovery factor and trapped gas fraction.

Optimally, the injection cycle should continue until the reservoir pressure rises close to the fracturing pressure of the rock in order to obtain the maximum stimulation of the oil possible.[20] Furthermore, the investigation of the effects of cycles of different duration upon the production is clearly an area where more work is needed.

VII. Discussion and Conclusions

Previous work has demonstrated the utility of carbon dioxide as an agent in the recovery of heavy crude oil.[20] Similarly, hydraulic fracturing has been used to increase the injectivity of a reservoir and thereby improve recovery. It is not well understood, however, what effect introducing a fracture into a reservoir would have on a CO_2 stimulation process.

We have developed a numerical model for use in the investigation of the compressible flow behavior of injected and reservoir fluids in the vicinity of a high-conductivity fracture. The two-dimensional model includes a rock matrix that is bounded on one side by a linear fracture. Flow into and out of the system is only through a well at one end of the fracture. We have neglected gravity and have assumed that there are no variations in the vertical direction. The combination of a high conductivity fracture with a horizontal areal model significantly reduces the computation necessary to describe the evolution of the system because a direct matrix solution of the problem is possible.

We are interested in modeling the behavior of fluids in fractured shallow reservoirs where pressures must be kept small enough (<500 psi) to preserve the competency of the rock. This pressure restriction thus allows us to assume a linear behavior for most fluid properties, thereby making complex equation-of-state calculations unnecessary and further reducing the computational effort.

The model allows up to three fluid phases to exist - gas, oil, and water - and furthermore allows dissolved gas to exist in both the oil and water phases. Well behavior can be either injection, production, or soak, and at the user's option, can be either pressure or time limited.

This arrangement facilitates the modeling of a huff-n-puff process. The model could, if desired, be readily converted to investigate drive processes as well by placing well block(s) on the other boundaries of the rock matrix.

In our runs we have observed, as expected, that the existence of a fracture increases the portion of the reservoir contacted by the injected CO_2. In general we find that the water, being much more mobile than the oil, is displaced away from the fracture while the oil remains near the fracture and is stimulated by the CO_2. The degree of stimulation and consequent increase in oil mobility is very clearly dependent on the amount of mobile gas phase present in the system.

A large free gas saturation reduces the efficiency of the recovery process by blocking the pores and inhibiting oil and water flow. Several runs indicate that limiting the presence of free gas in the system is an important factor in improving recovery efficiencies.

One possible way to accomplish this is by alternately injecting water into the system (WAG process) thereby increasing the pressure and forcing free gas into solution with the oil. A few runs were made to investigate the efficacy of such a process. Preliminary indications are that such a process would indeed result in higher oil recoveries than a pure CO_2-stimulation treatment. Furthermore, this prediction is supported by various field studies such as the Lick Creek project.[20]

We have also looked at how stimulation is affected by the degree of solubility of CO_2 in the water. If the reservoir brine has a high dissolved salt content (>200,000 ppm) then the solubility of CO_2 in water falls to 20% that of the pure water case. The model user should

estimate roughly the salt content of the reservoir brine and adjust the input solubility of CO_2 in water accordingly.

We now turn to the numerical features of this model. Although the basic formulation of the model is of the form of an implicit pressure-explicit composition calculation, the inclusion of a scheme for in-step iterations on the coefficients of the pressure-composition terms makes the model more implicit and therefore less subject to stability problems. Recently, however, we have found that limiting the iterations to one or two passes seems to be sufficient to keep the accuracy of the model within reasonable bounds. The automatic volume balance check that is included in the model predicts less than a 5% volume mismatch for the case of limited iterations.

Time truncation error is controlled by restricting the pressure change allowed during a step. This method of time step control works quite well and could be easily modified to investigate the use of other possible control parameters, such as saturation or fractional flow rates.

We have found that time step control at the bubble-point of the system is an important factor in maintaining the desired level of accuracy in the model's predictions. This is due to the existence of different fluid phase equilibrium conditions on the saturated and unsaturated sides of the bubble-point. Different equilibrium conditions imply that slightly different systems of equations govern the flow behavior in the two cases. Therefore, in order to make the transition from unsaturated case to the saturated case as smooth as possible, the time step at the bubble-point must be sufficiently small so that only a very small amount of gas phase is present at the end of the time step in which the gas first appears. We implemented an algorithm which

appears to adequately satisfy this criterion. The algorithm has been generalized to include the effect of gas disappearing in the system.

The formats of the input data required and the results output from the model are detailed in the Appendices. Note that the source code, with the exception of the PROGRAM statement, is ANSI standard FORTRAN 77.

VIII. Nomenclature

C_{eff} - effective compressibility

f_j - fractional flow of phase j

g - gravitational acceleration

h - fracture width (cm)

i - subscript index denoting component

j - subscript index denoting phase

k - grid block index

k_{rj} - relative permeability of phase j

$K_F;K_R$ - absolute permeability of (fracture;rock matrix) (millidarcies)

$\ell_F;\ell_R$ - length of (fracture;matrix branch) (cm)

$n_z;n_y$ - number of grid blocks along (fracture:z; matrix branch:y)

N_{ij} - number of moles of component i in phase j

P_k - average pressure in grid block k (psi)

q_z,q_y - volumetric flow rate along (fracture:z; matrix branch:y) (cm^3/sec)

r_z,r_y - mesh ratio ($\Delta t/\Delta z$; $\Delta t/\Delta y$)

S_j - saturation of phase j

Δt - length of time step (sec)

V_{ij} - specific molar volume of component i in phase j

W_{iz},W_{iz} - pressure equation coefficients containing mobility terms

x_{ij} - mole fraction of component i in phase j

$\Delta y, \Delta z$ - grid block dimension in direction (y:along matrix branch; z:along fracture)

$\alpha, \beta, \gamma, \theta, \delta_2, \delta_3$ - coefficients in pressure equation

θ_{12} - molar solubility coefficient for CO_2 in oil

θ_{13} - molar solubility coefficient for CO_2 in water

η - θ_{12}/θ_{13}

λ_j - mobility of phase j (= k_{rj}/μ_j)

μ_j - viscosity of phase j (c.p.)

ρ_j - molar density of phase j (moles/cm^3)

τ - area open to flow (cm^2)

ϕ - porosity

$D_t(F(y,z,t,P,N_{ij}))$ - operator denoting partial derivative of F with respect to t

IX. References

1. U.S. Bureau of Mines: Heavy Crude Oil Resource, Reserve, and Potential Production in the United States, IC 8352, 1967.

2. Ball Associates, Ltd.: Surface and Shallow Oil-Impregnated Rocks and Shallow Oil Fields in the United States, Bu-Mines Monograph 12 (1965).

3. Petroleum Training and Technical Services: Technical Review of Enhanced Oil Recovery Literature - Final Report, DOE/ER/10086-T1 (April 1980).

4. Smith, C. R.: Mechanics of Secondary Oil Recovery, Kreiger Publishing Co., Huntington, New York (1975).

5. Lamb, H.: Hydrodynamics, Cambridge University Press (1932).

6. Goodrich, J. H. and Watts, R.: Target Reservoirs for CO_2 Miscible Flooding - Final Report, DOE/MC/08341-17 (October 1980).

7. Todd, N. R., O'Dell, P. M., and Hirasaki, G. J.: "Methods for Increased Accuracy in Numerical Reservoir Simulators," Soc. Pet. Eng. J. (December 1972) 515-530.

8. Peaceman, D. W.: Fundamentals of Numerical Reservoir Simulation, Elsevier Scientific Publishing Co., Amsterdam (1977).

9. Landau, L. D. and Lifshitz, E. M.: Statistical Physics, Pergamon Press, Ltd., Oxford, England (1980).

10. Crawford, H. R., Neill, G. H., Bucy, B. J., and Crawford, P. B.: "Carbon Dioxide - A Multipurpose Additive for Effective Well Stimulation," J. Pet. Tech. (March 1963) 237-242.

11. Walker, J. R. and Dunlop, D. D.: "Physical Properties of Carbonated Oil," J. Pet. Tech. (August 1963) 873-876.

12. Miller, J. S. and Jones, R. A.: "A Laboratory Study to Determine Physical Characteristics of Heavy Oil After CO_2 Saturation," SPE/DOE 9789 (1981).

13. Simon, R. and Graue, D. J.: "Generalized Correlations for Predicting Solubility, Swelling, and Viscosity Behavior of CO_2-Crude Oil Systems," J. Pet. Tech. (January 1965) 102-106.

14. Dodds, W. S., Stutzman, L. F., and Sollami, B. J.: "Carbon Dioxide Solubility in Water," Ind. and Eng. Chem. (1956) 1 92-95.

15. Parkinson, W. J. and DeNevers, N.: "Partial Molar Volume of Carbon Dioxide in Water Solutions," I and EC Fundamentals (November 1969) 709-713.

16. Orr, F. M.: "Simulation of the One-Dimensional Convection of Four Phase, Four Component Mixtures," DOE/ET/12082-8 (June 1980).

17. Crichlow, H. B.: Modern Reservoir Engineering - A Simulation Approach, "Prentice-Hall, Inc., Englewood Cliffs, New Jersey (1977).

18. Stone, H. L.: "Estimation of Three-Phase Relative Permeability and Residual Oil Data," J. Can. Pet. Tech. (1973) 12 53-61 .

19. Patton, J. T., Coats, K. H., and Spence, K.: "Carbon Dioxide Well Stimulation: Part 1 - A Parametric Study," J. Pet. Tech. (August 1982) 1798-1804.

20. Reid, T. B. and Robinson, H. J.: "Lick Creek Meakin Sand Unit Immiscible CO_2/Waterflood Project," J. Pet. Tech. (September 1981) 1723-1729.

X. Appendix

A. Sample Input Format

```
- - - C O 2 C Y C L - - -

(HUF N' PUF) CO2 DISPLACING 17 DEG.WILMINGTON OIL IN FRAC'D. RES. .

NUMBER OF GRID BLOCKS
         IN FRACTURE =      4
         IN RESERVOIR BRANCH =      3

TIME STEP LENGTH      = 3.000   SECONDS
NUMBER OF TIME STEPS =  200

 PRESSURE CHANGE LIMIT PER TIME STEP =    50.0000PSI

 BUBBLE-EDGE TOLERANCE ON MOLE FRACTION OF CO2 IN OIL =      .0010

 MAXIMUM NUMBER OF CONVERGENCE ITERATIONS ALLOWED         5


 RESERVOIR GRID SCHEME
         -----> Z
         .
         .
         . Y
         V


   1     2     6    10    14
         3     7    11    15
         4     8    12    16
         5     9    13    17


 LENGTH OF FRACTURE (CM)=    90.0000
 LENGTH OF RESERVOIR BRANCH (CM)=     2.0000
 WIDTH OF FRACTURE (CM)=      .1000
 DEPTH OF SYSTEM (CM) =    10.0000

 FRACTURE X-SECTIONAL AREA (SQ.CM.)=     1.0000
 RES. BRANCH X-SECTIONAL AREA (SQ.CM.)=   225.0000

 ASSUMED POROSITY IN RESERVOIR=      .3000
                    IN FRACTURE=     1.0000

 PERMEABILITY IN FRACTURE (MD) =  .1000E+08
 PERMEABILITY IN RESERVOIR (MD) =   100.0000


 INITIAL RESERVOIR PRESSURE (PSI)=   150.0000
```

```
CO2 INJECTION FLOW RATE (CC/SEC)=      2.0000
PRODUCTION FLOW RATE            =       .1000


FUNCTIONAL FORM OF FLUID DENSITIES

       BLACK OIL :            .3530E-05 X P +  .9450E+00
       SAT-D. OIL SWELLING FACTOR:     .1000E-03 X P + 1.
       PURE H2O :             .3000E-05 X P +  .9943E+00
       SAT-D. H2O:            .1680E-04 X P +  .9943E+00

SLOPE OF GAS COMPRESSIBILITY FACTOR = -.3500E-03


  SOLUBILITY  COEFFCIENTS

  OIL -            .4540E-03
  WATER -          .2100E-04

VISCOSITY OF OIL =
           10 ** (.1090E-03*P -.3985E+01*X12 + .2699E+01 ) C.P.
                  (WHERE X12 IS THE MOLE FRACTION OF CO2 IN OIL)


VIS. CO2 =      .0150     VIS. H2O =     1.0000    (CP)

MOLEC. WGTS. (GMS/MOL) - CO2: 44.0000   OIL: 200.0000   H2O: 18.0000

RELATIVE PERMEABILITY DATA

    SWC= .100          EKRO =1.000          EO =  2.00
    SOR= .200          EKRW =1.000          EW =  2.00
    SOG= .200          EKROG=1.000          EOG=  2.00
    SGC= .050          EKRG =1.000          EG =  2.00


MESH RATIO FOR PRESSURE EQUATION
         IN FRACTURE =  .1333E+00
         IN RESERVOIR =  .4500E+01
```

```
INITIAL COMPOSITION IN MATRIX (VOLUME FRACTIONS)
         .0500      .4500      .5000

                     IN FRACTURE
         .0000      .5000      .5000

INJECTION COMPOSITION (VOLUME FRACTIONS)

        1.0000      .0000      .0000


HUF-N-PUF WELL MODE SEQUENCE
         MODE: 1 = CO2 INJECTION
               2 = WATER INJECTION
              -1 = PRODUCTION
             +10 = POST-INJECTION SOAK
             -10 = POST-PRODUCTION SOAK

    1  10  -1 -10   1  10  -1 -10   1  10  -1 -10   1  10  -1 -10
    1  10  -1 -10   1  10  -1 -10   1  10  -1 -10   1  10  -1 -10
    1  10  -1 -10   1  10  -1 -10


 MAX. PRESSURE BUILD-UP ALLOWED DURING INJECTION CYCLE:   250.0000

 MIN. PRESSURE ALLOWED:   150.0000

 PRESSURE DIFFERENCE REQUIRED TO END SOAK CYCLE:      .0500


PRINT KEYS

    IPRNT= 5   IP(1)=1   IP(2)=1   IP(3)=0   IP(4)=
```

B. Sample Output Format

```
IT=    5          TIME=   7.48274      STEP =      .60642. SEC
================================

CUMI (PV)                .023755   .000000   .000000
CMOLIN (MOLES)           .006473   .000000   .000000
CMOLPR (MOLES)           .000000   .000000   .000000

WELL MODE:  CO2 INJECTION
    CYCLE:    1


        TSTRT - INITIAL STEP (SEC) =  .3000E+01
        TDPCHG(I) - TIME STEP AFTER TRUNCATION ADJUST.

        TOVERF - TIME STEP AFTER OVERSHOOT ADJUST.(FRAC.)

        TOVERR - TIME STEP AFTER OVERSHOOT ADJUST.(RES.)

        TBUBBL - TIME STEP AFTER BUBBLE EDGE ADJUST.
           .6064E+00
        IMOB - PASSES THROUGH SUBR. MOBITR =   9


        TIME SPENT IN SUBR. IMPEM =  .2358E-01 SECONDS.
        TIME SPENT IN SUBR. PHASE =  .1343E-02 SECONDS.


PRESSURES
           (FRACTURE ------>)

K =   2       6      10      14
   169.809 169.747 169.712 169.694
   169.674 169.631 169.596 169.578
   169.443 169.399 169.363 169.346
   169.327 169.283 169.247 169.229
```

K	QZ	QY	N11	N12	N22
1	.2000E+01				
2	.1111E+01	.3678E+00	.0000E+00	.4242E-02	.5312E-01
3	.0000E+00	.1778E+00	.9617E-03	.1176E-03	.9575E-01
4	.0000E+00	.8916E-01	.9617E-03	.1771E-06	.9574E-01
5	.0000E+00	.0000E+00	.9617E-03	.1016E-09	.9574E-01
6	.5347E+00	.2664E+00	.0000E+00	.3649E-03	.5321E-01
7	.0000E+00	.1785E+00	.9617E-03	.1482E-05	.9574E-01
8	.0000E+00	.8950E-01	.9617E-03	.1648E-08	.9574E-01
9	.0000E+00	.0000E+00	.9617E-03	.1294E-11	.9574E-01
10	.2675E+00	.2668E+00	.0000E+00	.3086E-05	.5319E-01
11	.0000E+00	.1787E+00	.9617E-03	.9423E-08	.9574E-01
12	.0000E+00	.8961E-01	.9617E-03	.1412E-10	.9574E-01
13	.0000E+00	.0000E+00	.9617E-03	.1360E-14	.9574E-01
14	.0000E+00	.2670E+00	.0000E+00	.9821E-08	.5319E-01
15	.0000E+00	.1788E+00	.9617E-03	.3994E-10	.9574E-01
16	.0000E+00	.8966E-01	.9617E-03	.3673E-13	.9574E-01
17	.0000E+00	.0000E+00	.9617E-03	.1360E-14	.9574E-01

K	N13	N33	CMPEFF	VTOT	VOLERR
1					
2	.1473E-02	.4292E+00	.1017E-04	.1924E+02	-.1450E+00
3	.7147E-04	.1259E+01	.2714E-03	.4502E+02	.4324E-03
4	.1078E-06	.1260E+01	.2720E-03	.4502E+02	.4126E-03
5	.6181E-10	.1260E+01	.2723E-03	.4502E+02	.4025E-03
6	.1955E-03	.6203E+00	.3871E-05	.2250E+02	-.2697E-05
7	.9019E-06	.1260E+01	.2714E-03	.4502E+02	.4286E-03
8	.1003E-08	.1260E+01	.2721E-03	.4502E+02	.4084E-03
9	.7874E-12	.1260E+01	.2724E-03	.4502E+02	.3984E-03
10	.1669E-05	.6218E+00	.3379E-05	.2250E+02	-.9086E-08
11	.5735E-08	.1260E+01	.2715E-03	.4502E+02	.4256E-03
12	.8592E-11	.1260E+01	.2722E-03	.4502E+02	.4055E-03
13	.8279E-15	.1259E+01	.2725E-03	.4502E+02	.3955E-03
14	.5311E-08	.6218E+00	.3374E-05	.2250E+02	.2050E-09
15	.2431E-10	.1260E+01	.2716E-03	.4502E+02	.4241E-03
16	.2235E-13	.1260E+01	.2722E-03	.4502E+02	.4040E-03
17	.8279E-15	.1259E+01	.2726E-03	.4502E+02	.3941E-03

K	V11	V12	V22	V13	V33
2	.2051E+04	.4449E+02	.2115E+03	.3224E+02	.1809E+02
3	.2053E+04	.4449E+02	.2115E+03	.3224E+02	.1809E+02
4	.2056E+04	.4450E+02	.2115E+03	.3224E+02	.1809E+02
5	.2058E+04	.4450E+02	.2115E+03	.3224E+02	.1809E+02
6	.2052E+04	.4449E+02	.2115E+03	.3224E+02	.1809E+02
7	.2054E+04	.4449E+02	.2115E+03	.3224E+02	.1809E+02
8	.2057E+04	.4450E+02	.2115E+03	.3224E+02	.1809E+02
9	.2058E+04	.4450E+02	.2115E+03	.3224E+02	.1809E+02
10	.2053E+04	.4449E+02	.2115E+03	.3224E+02	.1809E+02
11	.2054E+04	.4449E+02	.2115E+03	.3224E+02	.1809E+02
12	.2057E+04	.4450E+02	.2115E+03	.3224E+02	.1809E+02
13	.2059E+04	.4450E+02	.2115E+03	.3224E+02	.1809E+02
14	.2053E+04	.4449E+02	.2115E+03	.3224E+02	.1809E+02
15	.2054E+04	.4449E+02	.2115E+03	.3224E+02	.1809E+02
16	.2057E+04	.4450E+02	.2115E+03	.3224E+02	.1809E+02
17	.2059E+04	.4450E+02	.2115E+03	.3224E+02	.1809E+02

K	RHOMOL(1)	RHOMOL(2)	RHOMOL(3)	F1	F2	F3	TMOB
2	.4875E-03	.5021E-02	.5512E-01	.00000	.00549	.99451	.40840
3	.4870E-03	.4733E-02	.5526E-01	.00000	.00066	.99934	.33692
4	.4863E-03	.4728E-02	.5527E-01	.00000	.00065	.99935	.33709
5	.4860E-03	.4728E-02	.5527E-01	.00000	.00065	.99935	.33703
6	.4873E-03	.4754E-02	.5525E-01	.00000	.00204	.99796	.50011
7	.4869E-03	.4728E-02	.5527E-01	.00000	.00065	.99935	.33715
8	.4862E-03	.4728E-02	.5527E-01	.00000	.00065	.99935	.33707
9	.4858E-03	.4728E-02	.5527E-01	.00000	.00065	.99935	.33701
10	.4872E-03	.4728E-02	.5527E-01	.00000	.00191	.99809	.50099
11	.4868E-03	.4728E-02	.5527E-01	.00000	.00065	.99935	.33713
12	.4861E-03	.4728E-02	.5527E-01	.00000	.00065	.99935	.33705
13	.4857E-03	.4728E-02	.5527E-01	.00000	.00065	.99935	.33699
14	.4871E-03	.4728E-02	.5527E-01	.00000	.00191	.99809	.50099
15	.4868E-03	.4728E-02	.5527E-01	.00000	.00065	.99935	.33712
16	.4860E-03	.4728E-02	.5527E-01	.00000	.00065	.99935	.33704
17	.4857E-03	.4728E-02	.5527E-01	.00000	.00065	.99935	.33698

	Z1	Z2	Z3	S1	S2	S3	X(1,2)	X(1,3)
2	.01171	.10884	.87945	.00000	.59384	.40616	.073956	.003421
3	.00014	.07065	.92921	.04386	.44996	.50618	.001227	.000057
4	.00000	.07063	.92937	.04393	.44979	.50629	.000002	.000000
5	.00000	.07064	.92936	.04396	.44979	.50625	.000000	.000000
6	.00083	.07894	.92023	.00000	.50091	.49909	.006811	.000315
7	.00000	.07063	.92937	.04387	.44981	.50632	.000015	.000001
8	.00000	.07063	.92937	.04394	.44979	.50627	.000000	.000000
9	.00000	.07064	.92936	.04397	.44979	.50624	.000000	.000000
10	.00001	.07880	.92120	.00000	.49997	.50003	.000058	.000003
11	.00000	.07063	.92937	.04388	.44980	.50631	.000000	.000000
12	.00000	.07064	.92936	.04395	.44979	.50626	.000000	.000000
13	.00000	.07064	.92936	.04398	.44979	.50622	.000000	.000000
14	.00000	.07880	.92120	.00000	.49996	.50004	.000000	.000000
15	.00000	.07063	.92937	.04389	.44981	.50631	.000000	.000000
16	.00000	.07064	.92936	.04395	.44979	.50626	.000000	.000000
17	.00000	.07064	.92936	.04399	.44979	.50622	.000000	.000000

	S2	X1	X2	X3	RHO	VIS	KR
2	.5938	.0740	.9260	.0000	.9463	264.7446	.5938
3	.4500	.0012	.9988	.0000	.9456	515.9482	.1139
4	.4498	.0000	1.0000	.0000	.9456	521.7494	.1137
5	.4498	.0000	1.0000	.0000	.9456	521.7430	.1137
6	.5009	.0068	.9932	.0000	.9457	490.1874	.5009
7	.4498	.0000	1.0000	.0000	.9456	521.7088	.1137
8	.4498	.0000	1.0000	.0000	.9456	521.7524	.1137
9	.4498	.0000	1.0000	.0000	.9456	521.7372	.1137
10	.5000	.0001	.9999	.0000	.9456	521.5157	.5000
11	.4498	.0000	1.0000	.0000	.9456	521.7777	.1137
12	.4498	.0000	1.0000	.0000	.9456	521.7478	.1137
13	.4498	.0000	1.0000	.0000	.9456	521.7325	.1137
14	.5000	.0000	1.0000	.0000	.9456	521.7902	.5000
15	.4498	.0000	1.0000	.0000	.9456	521.7759	.1137
16	.4498	.0000	1.0000	.0000	.9456	521.7454	.1137
17	.4498	.0000	1.0000	.0000	.9456	521.7302	.1137

C. Source Code for CO2CYCL

```
      PROGRAM CO2 ( TAPE1=INPUT,TAPE4=OUTPUT)
C
C                          TWO DIMENSIONAL SIMULATION
C                                    OF
C                        THREE PHASE, THREE COMPONENT
C                                   FLOW
C                                   IN A
C                          FRACTURED POROUS MEDIUM
C
C
C                 WRITTEN BY: ROLAND BEGIN
C                             DEPARTMENT OF PHYSICS
C                             COLORADO STATE UNIVERSITY
C                             FORT COLLINS, COLORADO  80521
C                 LAST REVISION: MAY 15, 1983
C
      PARAMETER (ISZ=52)
      COMMON / COUNT /  IT,ITMAX,ITCNT,ICNVRG,IBUBBL,IDPCHG,IOVERF,
     .                  IOVERR,ISTABL,IMOB,IMOB1,IPRNT,NOUNIT,MXCNVRG
      COMMON / TIMES /  TPHASE,TIMPEM,TDPCHG(100),TSTRT,TOVERF(100),
     .                  TOVERR(100),TBUBBL(100),XJUMP
      DATA  NOUNIT / 4 /
C
C
C                         VARIABLE DEFINITIONS.
C
C
C     VBLK(K) - VOLUME OF GRID BLOCK
C     C(I,K)  - VOLUME FRACTION OF COMPONENT I IN BLOCK K.
C     CN(I,K) - MOLES OF COMPONENT I IN BLOCK K
C     CNGC(K) - MOLES OF RESIDUAL GAS IN IN BLOCK K
C               (IMMOBILE AND INSOLUBLE IN OIL AND WATER PHASES)
C     X(I,J,K)- MOLE FRACTION OF COMPONENT I IN PHASE J IN
C               BLOCK K.
C     S(J,K)  - SATURATION OF PHASE J IN BLOCK K.
C     F(J,K)  - FRACTIONAL FLOW OF PHASE J IN BLOCK K.
C     WT(I)   - MOLECULAR WGT. OF COMPONENT I  (GM / MOLE)
C     RHO(J,K)- DENSITY OF PHASE J IN BLOCK K  (GM / CC)
C     RHOMOL(J,K) - MOLAR DENSITY OF PHASE J IN BLOCK K (MOL/CC)
C     VISO(K)- VISCOSITY OF OIL PHASE  IN BLOCK K  (CP)
C     VISG - VISCOSITY OF GAS PHASE
C     VISW - VISCOSITY OF WATER PHASE
C     AKR(J,K) - RELATIVE PERMEABILITY OF PHASE J IN BLOCK K.
C     PRSAV(K) - AVERAGE PRESSURE IN BLOCK K (PSI)
C     QZ(K)   - VOLUMETRIC FLOW RATE IN FRACTURE (CC/SEC)
C     QY(K)   - VOLUMETRIC FLOW RATE IN ROCK MATRIX
C                     (PERPENDICULAR TO FRACTURE)
```

```
C
C     I - COMPONENT INDEX.
C
C       1 - CO2
C       2 -  HYDROCARBON.
C       3 - WATER.
C
C     J - PHASE INDEX
C
C       1 - GAS PHASE.
C       2 - OIL PHASE.
C       3 - WATER PHASE.
C
C
C     INITIALIZE COMPOSITIONS, RELATIVE PERMEABILITY AND
C     PHASE BEHAVIOR DATA.
C
      T1 = SECOND()
      CALL INITL
      T2 = SECOND()
      TINIT = T2 - T1
      WRITE (NOUNIT,50) TINIT
 50   FORMAT ('1',' TIME SPENT IN SUBR. INITL = ',E10.4,' SECONDS.'//)
C
C     BEGIN MAIN LOOP. SET PHASE COMPOSITIONS AND SATURATIONS.
C
 100  T1 = SECOND()
      CALL PHASE
      T2 = SECOND()
      TPHASE = T2 - T1
C
C     PRINT RESULTS AND WRITE OUT PLOT DATA.
C
      CALL PRNTS
C
C     CALCULATE OVERALL COMPOSITIONS AT THE NEW TIME LEVEL.
C     PERFORM ACCURACY CHECKS AND IN-STEP ITERATIONS ON
C             PHASE PROPERTIES.
C
      T1 = SECOND()
      CALL IMPEM
      T2 = SECOND()
      TIMPEM = T2 -T1
C
C     REPEAT LOOP UNTIL MAX TIME REACHED.
C
      IF (IT .LE. ITMAX) GO TO 100
      STOP
      END
```

```
      SUBROUTINE INITL
C
C     INITIALIZES RESERVOIR GEOMETRY, FLUID PROPERTIES AND FINITE
C                 DIFFERENCE GRID SCHEME
C
      PARAMETER (ISZ=52)
      COMMON / GRID /    DT,DZ,DY,DTDZ,DTDY,DTT,DTDZT,DTDYT,TIME,DTNEW,
     .                   DTSTBL,NBZ,NBY,NR,NBFIN,FRLTH,RSLTH,HFRAC,
     .                   DEPTH,AFRAC,ARES,VBLK(ISZ),VFRBLK,VRSBLK,VPORE
      COMMON / TIMES /   TPHASE,TIMPEM,TDPCHG(100),TSTRT,TOVERF(100),
     .                   TOVERR(100),TBUBBL(100),XJUMP
      COMMON / COUNT /   IT,ITMAX,ITCNT,ICNVRG,IBUBBL,IDPCHG,IOVERF,
     .                   IOVERR,ISTABL,IMOB,IMOB1,IPRNT,NOUNIT,MXCNVRG
      COMMON / MOLES /   CN(3,ISZ),CNG(3,ISZ),CNGC(ISZ),X(3,3,ISZ),
     .                   OILSOL,WATSOL,RATSOL,OILRAT,WATRAT,OIPMOL(3)
      COMMON / PRESS /   PRSAV(ISZ),PRSOLD(ISZ),PDSOAK,DPLIM,PRSI,
     .                   PRSMAX,PRSMIN
      COMMON / PRSMAT /  ALPHA(ISZ),BETA(ISZ),GAMMA(ISZ),EPSLON(ISZ),
     .                   THETA(ISZ),DELTA2(ISZ),DELTA3(ISZ),A(ISZ),
     .                   W1Z(ISZ),W2Z(ISZ),W3Z(ISZ),W1Y(ISZ),W2Y(ISZ),
     .                   W3Y(ISZ)
      COMMON / FLOW /    QZ(ISZ),QY(ISZ),QFLOW,QINJ1,QINJ2,QPROD,
     .                   IWELL(50),JWELL,TWELL,TLIMIN,TLIMPR,TLIMSK
      COMMON / XDENSE /   RHOMOL(3,ISZ),RHO(3,ISZ),VN(3,3,ISZ),
     .                   CMPEFF(ISZ),ZSLOPE,A22DEN,B22DEN,ASWELL,
     .                   A33DEN,B33DEN,A3SDEN,B3SDEN,V22ATM,WT(3)
      COMMON / VISC /    VISO(ISZ),VISG,VISW,AVIS1,AVIS2,BVIS,CVIS
      COMMON / XPERM /    PRMF,PRMR,PORF,PORR,AKR(3,ISZ),F(3,ISZ),
     .                   TMOB(ISZ),SWC,SOR,SOG,SGC,EKRO,EKRW,EKROG,
     .                   EKRG,EO,EW,EOG,EG,S(3,ISZ),SGCX(ISZ)
      COMMON / PRM /     IPERM,NSW,NSL,TSW(3,20),TSL(3,20),ITPT
      COMMON / TMPVAL /  PRM1(3,ISZ),PRMT(3,ISZ),XCN(3,ISZ),XPRSAV(ISZ),
     .                   RHOT(3,ISZ),XT(3,3,ISZ),FT(3,ISZ),TMOBT(ISZ)
      COMMON / UPDATE /  CUMI(3),CUMP(3),CMOLIN(3),CMOLPR(3),PROD(3)
      COMMON / MISC /    C(3),TITLE(20),ILIMIT,IP(3),ISTEP(4),C22DUM,
     .                   C13DUM,C33DUM
C
      INTEGER KK(20)
      DATA NOUNIT/ 4 /
C
C
C              ***********    READ  INPUT DATA    *************
C
      READ (1,26) TITLE
 800  FORMAT (2I5)
      WRITE(NOUNIT,32) TITLE
      READ (1,26) RECORD
C     READ NO. OF TIME STEPS, BLOCKS, TIME STEP LENGTH .
      READ (1,1) ITMAX,NBZ,NBY,DT
      WRITE(NOUNIT,2) NBZ,NBY,DT,ITMAX
```

```
C
C                    READ LIMITING PRESSURE CHANGE
      READ (1,26) RECORD
      READ (1,809) DPLIM
      READ (1,26) RECORD
      READ (1,809) XJUMP
      WRITE (NOUNIT,802) DPLIM
 802  FORMAT (2X,'PRESSURE CHANGE LIMIT PER TIME STEP = ',F10.4,
     1          'PSI')
      WRITE (NOUNIT,801) XJUMP
 801  FORMAT (/2X,'BUBBLE-EDGE TOLERANCE ON MOLE FRACTION OF CO2 ',
     .        'IN OIL = ',F10.4/)
      READ (1,26) RECORD
      READ (1,9) MXCNVRG
      WRITE (NOUNIT,803) MXCNVRG
 803  FORMAT (2X,'MAXIMUM NUMBER OF CONVERGENCE ITERATIONS ALLOWED',
     .        5X,I5/)
C
      DZ=1./FLOAT(NBZ)
      DTDZ=DT/DZ
      IF (NBY.EQ.0) GO TO  888
      DY = 1./FLOAT(NBY)
      GO TO 889
 888  DY = 1.
 889  CONTINUE
      DTDY = DT/DY
      TIME=0.
      TWELL=0
      DTNEW = 1.
      NR = NBY + 1
      NBFIN = NR * NBZ + 1
C
      WRITE (NOUNIT,50)
 50      FORMAT (/2X,'RESERVOIR GRID SCHEME',/,10X,'-----> Z',/,10X,
     1          '.',/,10X,'.',/,10X,'. Y',/,10X,'V')
      NBZP1 = NBZ + 1
      DO 51 KF=1,NBZP1
 51       KK(KF) = 2 + (KF-1) * NR
      WRITE (NOUNIT,52)  (KK(I),I=1,NBZ)
 52      FORMAT (4X,'1',20I5)
        IF (NBY.EQ.0) GO TO 56
      DO 53 J=1,NBY
         DO 54 I=1,NBZ
 54           KK(I) = KK(I) + 1
 53       WRITE (NOUNIT,55)  (KK(I),I=1,NBZ)
 55         FORMAT (5X,20I5)
C
C
 56    IT = 0
       ITCNT = 0
       ISTABL = 0
       IMOB = 0
       ICNVRG = 0
       IBUBBL = 0
```

```
C
C
C        **************  READ DATA FOR PRESSURE EQUATION **************
C
       READ (1,26) RECORD
       READ (1,26) RECORD
      READ (1,809) FRLTH,RSLTH,HFRAC,DEPTH,PORR,PORF
 809      FORMAT (6F10.4)
       AFRAC = HFRAC * DEPTH
       ARES = DZ * FRLTH * DEPTH
       READ (1,26) RECORD
      READ (1,811) A22DEN,B22DEN,ASWELL,A33DEN,B33DEN,A3SDEN,
     1                 B3SDEN
 811   FORMAT (8E10.4)
        READ (1,26) RECORD
        READ (1,26) RECORD
      READ (1,810) PRSI,PRMF,PRMR
 810   FORMAT (4E10.4)
C
       VFRBLK = FRLTH * DZ * AFRAC * PORF
       VRSBLK = RSLTH * DY * ARES * PORR
      WRITE (NOUNIT,820) FRLTH,RSLTH,HFRAC,DEPTH,AFRAC,ARES,PORR,PORF,
     1                   PRMF,PRMR,PRSI
 820   FORMAT (/// 2X,'LENGTH OF FRACTURE (CM)= ',F10.4,
     1          /2X,'LENGTH OF RESERVOIR BRANCH (CM)= ',F10.4,
     1         /2X,'WIDTH OF FRACTURE (CM)= ',F10.4,
     .         /2X,'DEPTH OF SYSTEM (CM) = ',F10.4,
     2        // 2X,'FRACTURE X-SECTIONAL AREA (SQ.CM.)= ',F10.4,
     3          /2X,'RES. BRANCH X-SECTIONAL AREA (SQ.CM.)= ',F10.4,
     4          // 2X,'ASSUMED POROSITY IN RESERVOIR= ',F10.4,
     4          /2X,'                    IN FRACTURE= ',F10.4,
     5          // 2X,'PERMEABILITY IN FRACTURE (MD) = ',E10.4,
     6          /2X,'PERMEABILITY IN RESERVOIR (MD) = ',F10.4,
     7       /// 2X,'INITIAL RESERVOIR PRESSURE (PSI)= ',F10.4)
C
       READ (1,26) RECORD
       READ (1,810) QINJ1,QINJ2,QPROD
       WRITE (NOUNIT,823) QINJ1,QINJ2,QPROD
 823    FORMAT (//2X,'CO2 INJECTION FLOW RATE (CC/SEC)= ',F10.4,
     1           /2X,'WATER INJECTION FLOW RATE       = ',F10.4,
     2           /2X,'PRODUCTION FLOW RATE            = ',F10.4)
C
      WRITE (NOUNIT,825)  A22DEN,B22DEN,ASWELL,A33DEN,B33DEN,
     1                              A3SDEN,B3SDEN
 825   FORMAT (///2X,'FUNCTIONAL FORM OF FLUID DENSITIES',//,
     1         T10,'BLACK OIL :',T30,E10.4,' X P + ',E10.4,/,
     2       T10,'SAT-D. OIL SWELLING FACTOR: ',T40,E10.4,' X P + 1.'
     3       ,/,T10,'PURE H2O :',T30,E10.4,' X P + ',E10.4,/,
     4         T10,'SAT-D. H2O:',T30,E10.4,' X P + ',E10.4/)
C
```

```
      READ (1,26) RECORD
      READ (1,811) ZSLOPE
      WRITE (NOUNIT,826) ZSLOPE
 826    FORMAT (2X,'SLOPE OF GAS COMPRESSIBILITY FACTOR = ',E10.4/)
C
C
C       ************  END PRESSURE DATA  ***************
C
C                     READ SOLUBILITY COEFFICIENTS
C
            READ (1,26) RECORD
            READ (1,811) OILSOL,WATSOL
            WRITE (NOUNIT,830) OILSOL,WATSOL
 830       FORMAT (///T5,'SOLUBILITY  COEFFCIENTS:',//,T5,'OIL -',
     1                T20,E10.4,/,T5,'WATER -',T20,E10.4)
      RATSOL = OILSOL / WATSOL
      SUMSOL = OILSOL + WATSOL
      OILRAT = OILSOL / SUMSOL
      WATRAT = WATSOL / SUMSOL
C
C
C     READ COMPONENT NAMES AND PROPERTIES
C
      READ (1,26) RECORD
      READ (1,6) VISG,VISW,(WT(I),I=1,3)
C                         READ OIL VISCOSITY PARAMETERS
        READ (1,26) RECORD
        READ (1,811) AVIS1,AVIS2,CVIS
 6      FORMAT (5F10.0)
C
C       CALCULATE VOLUME OF OIL MOLE AT P = 1 ATM.
      V22ATM = WT(2) / (A22DEN * 14.7 + B22DEN)
C
C                     CALCULATE OTHER COEFFICIENTS NEEDED LATER IN CODE
C
C           FOR SUBR. PHASE
      C22DUM = A22DEN / WT(2)
      C33DUM = A33DEN / WT(3)
      C13DUM = (WT(3) * A33DEN + (WT(1)-WT(3)) * A3SDEN) / B33DEN**2
C           FOR SUBROUTINE VISOIL
      BVIS = (AVIS2-AVIS1) / OILSOL
C
        WRITE (NOUNIT,815) AVIS1,BVIS,CVIS
        WRITE (NOUNIT,27) VISG,VISW,(WT(I),I=1,3)
 815     FORMAT (/1X,'VISCOSITY OF OIL = 10 ** (',E10.4,' * P + ',
     1          E10.4,' * X12 + ',E10.4,' )  C.P.',/,20X,
     2          '(WHERE X12 IS THE MOLE FRACTION OF CO2 IN OIL)')
```

```
C
C     READ RELATIVE PERMEABILITY DATA.
C
      READ (1,9) IPERM,NSW,NSL
      IF (IPERM .EQ. 0) GO TO 45
      DO 30 I=1,NSW
         READ (1,3) (TSW(L,I),L=1,3)
   30 CONTINUE
      DO 35 I=1,NSL
         READ (1,3) (TSL(L,I),L=1,3)
   35 CONTINUE
      EKRW=TSW(2,NSW)
      WRITE(NOUNIT,17)
      WRITE(NOUNIT,18) (L,TSW(1,L),TSW(2,L),TSW(3,L),L=1,NSW)
      WRITE(NOUNIT,19)
      WRITE(NOUNIT,18) (L,TSL(1,L),TSL(2,L),TSL(3,L),L=1,NSL)
      GO TO 48
 45    READ (1,26) RECORD
       READ (1,26) RECORD
      READ (1,3)SWC,SOR,SOG,SGC,EKRO,EKRW,EKROG,EKRG,EO,EW,
     1EOG,EG
      WRITE(NOUNIT,5)SWC,EKRO,EO,SOR,EKRW,EW,SOG,EKROG,EOG,SGC,EKRG,EG
      IF (EKRO .NE. EKROG) WRITE(NOUNIT,36)
   48 CONTINUE
C
C
C
C          ***************************************************
C              INITIALIZE PRESSURES, DENSITIES, RELATIVE FLOW RATES,
C                       AND MESH RATIO
C
       DO 110 K=2,NBFIN
          QZ(K)=0
 110       QY(K)=0
C
C        INITIALIZE MESH RATIO
              DTDZ = DTDZ / FRLTH
              DTDY = DTDY / RSLTH
              DTT = DT
              DTDZT = DTDZ
              DTDYT = DTDY
C
            WRITE (NOUNIT,816) DTDZ,DTDY
 816      FORMAT (//1X,'MESH RATIO FOR PRESSURE EQUATION:',/,
     1           10X,'IN FRACTURE = ',E10.4,/,
     2           10X,'IN RESERVOIR = ',E10.4)
C
C        INITIALIZE PRESSURES AND PERMEABILITIES
            DO 130 K=2,NBFIN
                PRSAV(K) = PRSI
                PRSOLD(K) = PRSI
                DO 130 J=1,3
 130              AKR(J,K) = 0
```

```
C
C
C
C
      RHOMOL(1,2) = 2.7E-6 * PRSI /(1.+ ZSLOPE * PRSI)
      RHOMOL(2,2) = (A22DEN * PRSI + B22DEN) / WT(2)
      RHOMOL(2,1) = RHOMOL(2,2)
      RHOMOL(3,2) = (A33DEN * PRSI + B33DEN) / WT(3)
C
C     READ INITIAL COMPOSITIONS (VOLUME FRACTIONS).
C
      READ (1,26) RECORD
C                        READ RESERVOIR MATRIX COPOSITIONS
      READ (1,14) (C(I),I=1,3)
      CN1RS = C(1) * RHOMOL(1,2) * VRSBLK
      CN2RS = C(2) * RHOMOL(2,2) * VRSBLK
      CN3RS = C(3) * RHOMOL(3,2) * VRSBLK
      WRITE (NOUNIT,13) (C(I),I=1,3)
C
C                        READ FRACTURE COMPOSITIONS
      READ (1,26) RECORD
      READ (1,14) (C(I),I=1,3)
      CN1FR = C(1) * RHOMOL(1,2) * VFRBLK
      CN2FR = C(2) * RHOMOL(2,2) * VFRBLK
      CN3FR = C(3) * RHOMOL(3,2) * VFRBLK
      WRITE (NOUNIT,15) (C(I),I=1,3)
C       CALCULATE MOLES OF RESIDUAL GAS PRESENT
      CNGCFR = 0
      CNGCRS = SGC * RHOMOL(1,2) * VRSBLK
      KB = 2
      DO 128 KF=1,NBZ
         KBDUM = KB + NBY
         DO 125 K=KB,KBDUM
            DO 122 I=1,3
               DO 122 J=1,3
122            X(I,J,K) = 0
               X(1,1,K) = 1.
            CNG(2,K) = 0
            CNG(3,K) = 0
            IF (K.GT.KB) GO TO 124
               CN(1,K) = CN1FR
               CN(2,K) = CN2FR
               CN(3,K) = CN3FR
               CNGC(K) = CNGCFR
               VBLK(K) = VFRBLK
               GO TO 125
124         CN(1,K) = CN1RS
            CN(2,K) = CN2RS
            CN(3,K) = CN3RS
            CNGC(K) = CNGCRS
            VBLK(K) = VRSBLK
125     CONTINUE
```

```
 128      KB = 2 + KF * NR
          VPORE = 0
          DO 135 K=2,NBFIN
 135         VPORE = VPORE + VBLK(K)
      DO 150 I=1,3
          CMOLIN(I)=0
          CMOLPR(I)=0
          OIPMOL(1) = (CN1FR + NBY * CN1RS) * NBZ
          OIPMOL(2) = (CN2FR + NBY * CN2RS) * NBZ
          OIPMOL(3) = (CN3FR + NBY * CN3RS) * NBZ
          CUMI(I)=0
          CUMP(I)=0
          PROD(I)=0
  150 CONTINUE
C
C     READ INITIAL INJECTION COMPOSITIONS.
C
       READ (1,26) RECORD
      READ (1,14) (C(I),I=1,3)
      F(1,1)=1.
      F(2,1) = 0
      F(3,1) = 0
      DO 200 J=1,3
        DO 200 I =1,3
 200        X(I,J,1) = 0
      X(1,1,1) = 1
      WRITE(NOUNIT,11) (C(I),I=1,3)
C
C                 READ INSTRUCTIONS FOR HUF-N-PUF CYCLE CONTROL
C
      READ (1,26) RECORD
C
        READ (1,40) (IWELL(J),J=1,40)
 40       FORMAT (16I4)
C
C                        NOTE: IWELL=1 => HUF (INJECT CO2)
C                              IWELL=2 => HUF (INJECT WATER)
C                              IWELL=-1 => PUF (PRODUCTION)
C                              IWELL=+10 => SOAK (POST INJECT)
C                              IWELL=-10 => SOAK (POST PROD.)
C
         WRITE (NOUNIT,43) (IWELL(J),J=1,40)
 43      FORMAT (//1X,'HUF-N-PUF WELL MODE SEQUENCE',/10X,
     1   'MODE: 1 = CO2 INJECTION',/16X,'2 = WATER INJECTION',/15X,
     2   '-1 = PRODUCTION',/14X,'+10 = POST-INJECTION SOAK',/14X,
     3   '-10 = POST-PRODUCTION SOAK',//2X,16I4,/,2X,16I4,/,2X,8I4///)
C
C
C              INITIALIZE WELL MODE COUNTER
        JWELL = 1
```

```
C
C       READ PARAMETERS DEFINING WELL MODE CHANGES
        READ (1,26) RECORD
        READ (1,9) ILIMIT
        READ (1,26) RECORD
          IF (ILIMIT.EQ.1) THEN
             READ (1,41) PRSMAX,PRSMIN,PDSOAK
             WRITE (NOUNIT,46) PRSMAX,PRSMIN,PDSOAK
          ELSE IF (ILIMIT.EQ.2) THEN
             READ (1,41) TLIMIN,TLIMPR,TLIMSK
             WRITE (NOUNIT,47) TLIMIN,TLIMPR,TLIMSK
          END IF
 41        FORMAT (3F10.4)
 46        FORMAT (1X,'MAX. PRESSURE BUILD-UP ALLOWED DURING ',
     1             'INJECTION CYCLE: ',F10.4,//,2X,'MIN. PRESSURE',
     2             ' ALLOWED: ',F10.4,//,2X,'PRESSURE DIFFERENCE ',
     3             'REQUIRED TO END SOAK CYCLE: ',F10.4//)
 47        FORMAT (2X,'INJECTION CYCLE LIMITED TO ',F10.4,'SEC.',/,
     1             2X,'PRODUCTION CYCLE LIMITED TO ',F10.4,'SEC.',
     2             /,2X,'SOAK CYCLE LIMITED TO ',F10.4,'SEC.'//)
C
          IF (IWELL(1).EQ.1) QZ(1) = QINJ1
          IF (IWELL(1).EQ.2) QZ(1) = QINJ2
C
C
C       READ PRINT KEYS.
C
        READ (1,26) RECORD
        READ (1,9) IPRNT,(IP(J),J=1,3)
        WRITE(NOUNIT,12) IPRNT,(IP(J),J=1,3)
        RETURN
      1 FORMAT(3I5,2F10.0)
      2 FORMAT(' NUMBER OF GRID BLOCKS:',/,10X,'IN FRACTURE = ',I5,/,
     1                        10X,'IN RESERVOIR BRANCH = ',I5,//,
     2        ' TIME STEP LENGTH     =',F6.3,'   SECONDS '/
     3        ' NUMBER OF TIME STEPS =',I5/)
      3 FORMAT(4F10.0/4F 10.0/4F10.0)
      5 FORMAT(//' RELATIVE PERMEABILITY DATA:'//5X,
     1'SWC=',F5.3,10X,'EKRO =',F5.3,10X,'EO =',F6.2/5X,
     2'SOR=',F5.3,10X,'EKRW =',F5.3,10X,'EW =',F6.2/5X,
     3'SOG=',F5.3,10X,'EKROG=',F5.3,10X,'EOG=',F6.2/5X,
     4'SGC=',F5.3,10X,'EKRG =',F5.3,10X,'EG =',F6.2/)
      9 FORMAT (15I5)
     11 FORMAT(/' INJECTION COMPOSITION (VOLUME FRACTIONS):'//5X,3F10.4/)
     12 FORMAT (/' PRINT KEYS:'//5X,'IPRNT=',I2,3X,'IP(1)=',I1,3X,
     1          'IP(2)=',I1,3X,'IP(3)=',I1,3X,'IP(4)=',I1/'1')
     13 FORMAT (/' INITIAL COMPOSITION IN MATRIX (VOLUME FRACTIONS):',
     1         /,5X,3F10.4/)
     14 FORMAT (7F10.0)
     15   FORMAT ('                          IN FRACTURE :',/,5X,3F10.4)
```

```
17 FORMAT (//1X,'OIL-WATER RELATIVE PERMEABILITIES:'//15X,
  1          'SW',8X,'KRW',7X,'KROW')
18 FORMAT (20(/5X,I3,3F10.3))
19 FORMAT (//1X,'GAS-OIL RELATIVE PERMEABILITIES:'//15X,
  1          'SL',7X,'KROG',7X,'KRG')
21 FORMAT (//1X,'PLOT KEYS:'//6X,'IPLT=',I2,3X,'ISAT=',I2,3X,
  1         'IBLK=',I2,3X,'IFLO=',I2,3X,'IPRO=',I2,3X,'IPRESS = ',I2)
22 FORMAT (2A4)
23 FORMAT (//1X,'SATURATION AND COMPOSITION PROFILES STORED IN FILE
  1         ',2A4,' AT TIME STEPS:'//5X,10I5)
24 FORMAT (//1X,'BLOCK OVERALL COMPOSITIONS STORED IN FILE ',2A4,
  1         ' FOR BLOCKS:'//5X,10I5)
25 FORMAT (//1X,'FLOWING COMPOSITIONS STORED IN FILE ',2A4,
  +         ' FOR BLOCKS:'//5X,10I5)
26 FORMAT (20A4)
27 FORMAT (///1X,'VIS. CO2 = ',F10.4,5X,'VIS. H2O = ',F10.4,4X,
  1    '(CP)',//,1X,'MOLEC. WGTS. (GMS/MOL) - CO2: ',F10.4,3X,
  2       'OIL: ',F10.4,3X,'H2O: ',F10.4)
31 FORMAT (1X,I3,3F10.4)
32 FORMAT ('1'/26X,'- - - C O 2 C Y C L - - -'///1X,20A4/)
36 FORMAT (/1X,'-------- WARNING: EKROG NOT EQUAL TO EKRO ------')
37 FORMAT (//1X,' COMPONENT PRODUCTION DATA STORED IN FILE ',2A4)
   END
```

```
      SUBROUTINE PHASE
C
C
C                THIS SUBROUTINE WILL CALCULATE MOLE FRACTIONS OF
C           COMPONENTS PRESENT IN THE VARIOUS PHASES AND THE VOLUME
C           SATURATIONS OF THOSE PHASES.
C
      PARAMETER (ISZ=52)
      COMMON / GRID /    DT,DZ,DY,DTDZ,DTDY,DTT,DTDZT,DTDYT,TIME,DTNEW,
     .                   DTSTBL,NBZ,NBY,NR,NBFIN,FRLTH,RSLTH,HFRAC,
     .                   DEPTH,AFRAC,ARES,VBLK(ISZ),VFRBLK,VRSBLK,VPORE
      COMMON / MOLES /   CN(3,ISZ),CNG(3,ISZ),CNGC(ISZ),X(3,3,ISZ),
     .                   OILSOL,WATSOL,RATSOL,OILRAT,WATRAT,OIPMOL(3)
      COMMON / PRESS /   PRSAV(ISZ),PRSOLD(ISZ),PDSOAK,DPLIM,PRSI,
     .                   PRSMAX,PRSMIN
      COMMON / XDENSE /   RHOMOL(3,ISZ),RHO(3,ISZ),VN(3,3,ISZ),
     .                   CMPEFF(ISZ),ZSLOPE,A22DEN,B22DEN,ASWELL,
     .                   A33DEN,B33DEN,A3SDEN,B3SDEN,V22ATM,WT(3)
      COMMON / VISC /    VISO(ISZ),VISG,VISW,AVIS1,AVIS2,BVIS,CVIS
      COMMON / XPERM /    PRMF,PRMR,PORF,PORR,AKR(3,ISZ),F(3,ISZ),
     .                   TMOB(ISZ),SWC,SOR,SOG,SGC,EKRO,EKRW,EKROG,
     .                   EKRG,EO,EW,EOG,EG,S(3,ISZ),SGCX(ISZ)
      DIMENSION X1O(ISZ)
C
C
C              CALCULATE MOLE FRACTIONS OF CO2 IN OIL AND WATER
      CALL EQULIB (CNGC,CN,PRSAV,CNG,X,X1O)
C
C              CALCULATE PHASE DENSITIES
      CALL DENSE (PRSAV,X,RHO,RHOMOL,VN)
C
C              CALCULATE OIL PHASE VISCOSITY
      DO 50 K=2,NBFIN
 50   VISO(K) = VISOIL (PRSAV(K),X(1,2,K))
C
C              CALCULATE EFFECTIVE COMPRESSIBILITY
C                (NOTE: EFF. COMP.= CMPEFF/P.V.)
      CALL CMPRES (CNG,CN,PRSAV,CMPEFF,VN)
C
C              CALCULATE PHASE SATURATIONS
      CALL SCALC (CNG,CN,VN,SGCX,S,CNGC)
C
C              CALCULATE RELATIVE PERMEABILITIES AND
C                         FRACTIONAL FLOW FUNCTIONS
      CALL PERM (S,VISO,SGCX,AKR,F,TMOB)
C
C              SET DENSITY OF GAS AND WATER IN WELL
      RHOMOL(1,1) = RHOMOL(1,2)
      RHOMOL(3,1) = (A33DEN * PRSAV(2) + B33DEN) / WT(3)
      RETURN
      END
```

```
      SUBROUTINE EQULIB (CNGCX,CNX,P,CNGX,XX,X1O)
C
      PARAMETER (ISZ=52)
      COMMON / GRID /   DT,DZ,DY,DTDZ,DTDY,DTT,DTDZT,DTDYT,TIME,DTNEW,
     .                  DTSTBL,NBZ,NBY,NR,NBFIN,FRLTH,RSLTH,HFRAC,
     .                  DEPTH,AFRAC,ARES,VBLK(ISZ),VFRBLK,VRSBLK,VPORE
      COMMON / MOLES /  CN(3,ISZ),CNG(3,ISZ),CNGC(ISZ),X(3,3,ISZ),
     .                  OILSOL,WATSOL,RATSOL,OILRAT,WATRAT,OIPMOL(3)
      DIMENSION CNGCX(ISZ),CNX(3,ISZ),P(ISZ),CNGX(3,ISZ),XX(3,3,ISZ),
     .          X1O(ISZ)
C
C
C                 PHASE EQUILIBRIUM CALCULATION
C
C                PARTITION COMPONENTS INTO PHASES
      DO 100 K=2,NBFIN
         DENOM = CNX(1,K) - CNGCX(K) + CNX(2,K) + CNX(3,K)
         Z2 = CNX(2,K) / DENOM
         Z3 = CNX(3,K) / DENOM
         Z1 = 1.- Z2 - Z3
         IF (Z1.LT.1.E-20) Z1 = 0
         DUM = 1. + RATSOL * Z1 + Z2 * (RATSOL-1.)
         DUM1 = SQRT( DUM**2 - 4. * Z1 * RATSOL)
         X1W = (DUM - DUM1)/(2. * RATSOL)
         X1O(K) = RATSOL * X1W
         DUMSOL = OILSOL * P(K)
         IF (X1O(K) .LT. DUMSOL) THEN
C             UNSATURATED LIQUID / IMMOBILE GAS
            XX(1,3,K) = X1W
            XX(1,2,K) = X1O(K)
            XX(2,2,K) = 1. - X1O(K)
            XX(3,3,K) = 1. - X1W
         ELSE
C             SATURATED LIQUID / MOBILE GAS
            XX(1,2,K) = OILSOL * P(K)
            XX(1,3,K) = WATSOL * P(K)
            XX(2,2,K) = 1. - XX(1,2,K)
            XX(3,3,K) = 1. - XX(1,3,K)
         END IF
      CNGX(2,K) = XX(1,2,K) * CNX(2,K) / XX(2,2,K)
      CNGX(3,K) = XX(1,3,K) * CNX(3,K) / XX(3,3,K)
      CNGX(1,K) = CNX(1,K) - CNGX(2,K) - CNGX(3,K)
      IF (CNGX(1,K).LT.1.E-10) CNGX(1,K) = 0
 100  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DENSE (P,XX,DENGM,DENMOL,VNX)
C
      PARAMETER (ISZ=52)
      COMMON / GRID /   DT,DZ,DY,DTDZ,DTDY,DTT,DTDZT,DTDYT,TIME,DTNEW,
     .                  DTSTBL,NBZ,NBY,NR,NBFIN,FRLTH,RSLTH,HFRAC,
     .                  DEPTH,AFRAC,ARES,VBLK(ISZ),VFRBLK,VRSBLK,VPORE
      COMMON / MOLES /  CN(3,ISZ),CNG(3,ISZ),CNGC(ISZ),X(3,3,ISZ),
     .                  OILSOL,WATSOL,RATSOL,OILRAT,WATRAT,OIPMOL(3)
      COMMON / XDENSE /  RHOMOL(3,ISZ),RHO(3,ISZ),VN(3,3,ISZ),
     .                  CMPEFF(ISZ),ZSLOPE,A22DEN,B22DEN,ASWELL,
     .                  A33DEN,B33DEN,A3SDEN,B3SDEN,V22ATM,WT(3)
      DIMENSION P(ISZ),XX(3,3,ISZ),DENGM(3,ISZ),DENMOL(3,ISZ),
     .          VNX(3,3,ISZ)
C
C
C           CALCULATES PHASE DENSITIES
C
      DO 100 K=2,NBFIN
C
C           GAS PHASE DENSITY
         ZCMP = 1.+ ZSLOPE * P(K)
         DENMOL(1,K) = 2.7E-6 * P(K) / ZCMP
         VNX(1,1,K) = 1. / DENMOL(1,K)
         DENGM(1,K) = WT(1) * DENMOL(1,K)
C
C           OIL PHASE DENSITY
C           DENSITY DATA FROM MILLER'S WILMINGTON 17 DEG. API DATA
C                              (T=95 DEG. F.)
         RODEAD = A22DEN * P(K) + B22DEN
         VNX(2,2,K) = WT(2) / RODEAD
         WT2 = XX(1,2,K) * WT(1) + XX(2,2,K) * WT(2)
         SOLOIL = P(K) * OILSOL
         FWELL = ASWELL * P(K) + 1.
         VNX(1,2,K)=(1.- SOLOIL)*(FWELL*V22ATM-VNX(2,2,K))/SOLOIL
         DENGM(2,K)=WT2/(WT(2)*XX(2,2,K)/RODEAD+VNX(1,2,K)*XX(1,2,K))
         DENMOL(2,K) = DENGM(2,K) / WT2
C
C           WATER PHASE DENSITY
C           DENSITY DATA FROM PARKINSON & DE NEVERS SAT'D. WATER
C                               DATA @ T=95 DEG.F. (SEE FIG. 7)
         RHOSAT = B3SDEN + A3SDEN * P(K)
         RHOH2O = B33DEN + A33DEN * P(K)
         VNX(3,3,K) = WT(3) / RHOH2O
         WT3 = XX(1,3,K) * WT(1) + XX(3,3,K) * WT(3)
         SOLWAT = P(K) * WATSOL
         VNX(1,3,K) = WT(3) * (1.-SOLWAT) * (1/RHOSAT - 1/RHOH2O) /
     1                SOLWAT + WT(1) / RHOSAT
         IF (VNX(1,3,K).LT.0) VNX(1,3,K)=1.E-20
         DENGM(3,K)=WT3/(WT(3)*XX(3,3,K)/RHOH2O+VNX(1,3,K)*XX(1,3,K))
         DENMOL(3,K) = DENGM(3,K) / WT3
 100  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CMPRES (CNGX,CNX,P,CMP,VNX)
C
      PARAMETER (ISZ=52)
      COMMON / GRID /    DT,DZ,DY,DTDZ,DTDY,DTT,DTDZT,DTDYT,TIME,DTNEW,
     .                   DTSTBL,NBZ,NBY,NR,NBFIN,FRLTH,RSLTH,HFRAC,
     .                   DEPTH,AFRAC,ARES,VBLK(ISZ),VFRBLK,VRSBLK,VPORE
      COMMON / MOLES /   CN(3,ISZ),CNG(3,ISZ),CNGC(ISZ),X(3,3,ISZ),
     .                   OILSOL,WATSOL,RATSOL,OILRAT,WATRAT,OIPMOL(3)
      COMMON / PRESS /   PRSAV(ISZ),PRSOLD(ISZ),PDSOAK,DPLIM,PRSI,
     .                   PRSMAX,PRSMIN
      COMMON / XDENSE /   RHOMOL(3,ISZ),RHO(3,ISZ),VN(3,3,ISZ),
     .                   CMPEFF(ISZ),ZSLOPE,A22DEN,B22DEN,ASWELL,
     .                   A33DEN,B33DEN,A3SDEN,B3SDEN,V22ATM,WT(3)
      COMMON / MISC /    C(3),TITLE(20),ILIMIT,IP(3),ISTEP(4),C22DUM,
     .                   C13DUM,C33DUM
      DIMENSION CNGX(3,ISZ),CNX(3,ISZ),P(ISZ),CMP(ISZ),VNX(3,3,ISZ)
C
C            CALCULATES AVERAGE COMPRESSIBILITY OF FLUID SYSTEM
      DO 100 K=2,NBFIN
         PSQDEN = 1./(P(K)*P(K))
         V22SQ = VNX(2,2,K)*VNX(2,2,K)
         V33SQ = VNX(3,3,K)*VNX(3,3,K)
         ZCMP = 1. + ZSLOPE * P(K)
C
         C11 = CNGX(1,K)*VNX(1,1,K)/(ZCMP*P(K))*(1-(P(K)-PRSOLD(K))/
     1                                                   PRSOLD(K))
         C22 = C22DUM * CNX(2,K) * V22SQ
         C33 = C33DUM * CNX(3,K) * V33SQ
         C12 = CNGX(2,K) * (ASWELL * V22ATM + (V22ATM-VNX(2,2,K))
     1                        * PSQDEN /   OILSOL)
         C13 = CNGX(3,K) * C13DUM
C
         CMP(K) = C11 + C22 + C33 + C12 + C13
 100  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SCALC (CNGX,CNX,VNX,XSGC,SX,CNGCX)
C
      PARAMETER (ISZ=52)
      COMMON / GRID /   DT,DZ,DY,DTDZ,DTDY,DTT,DTDZT,DTDYT,TIME,DTNEW,
     .                  DTSTBL,NBZ,NBY,NR,NBFIN,FRLTH,RSLTH,HFRAC,
     .                  DEPTH,AFRAC,ARES,VBLK(ISZ),VFRBLK,VRSBLK,VPORE
      DIMENSION CNGX(3,ISZ),CNX(3,ISZ),VNX(3,3,ISZ),XSGC(ISZ),
     .          SX(3,ISZ),CNGCX(ISZ)
C
C
C             CALCULATE PHASE SATURATONS
      DO 100 K=2,NBFIN
         V11BLK = CNGX(1,K) * VNX(1,1,K)
         V12BLK = CNGX(2,K) * VNX(1,2,K)
         V13BLK = CNGX(3,K) * VNX(1,3,K)
         V22BLK = CNX(2,K) * VNX(2,2,K)
         V33BLK = CNX(3,K) * VNX(3,3,K)
         VTOT   = V11BLK + V12BLK + V13BLK + V22BLK + V33BLK
         SX(2,K) = (V12BLK + V22BLK) / VTOT
         SX(3,K) = (V13BLK + V33BLK) / VTOT
         SX(1,K) = 1. - SX(2,K) - SX(3,K)
         IF (SX(1,K).LT.1.E-10) SX(1,K) = 0
         VGC = CNGCX(K) * VNX(1,1,K)
         XSGC(K) = VGC / VTOT
 100  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PERM (SX,VISX,XSGC,AKRX,FX,XTMOB)
C
      PARAMETER (ISZ=52)
      COMMON / GRID /    DT,DZ,DY,DTDZ,DTDY,DTT,DTDZT,DTDYT,TIME,DTNEW,
     .                   DTSTBL,NBZ,NBY,NR,NBFIN,FRLTH,RSLTH,HFRAC,
     .                   DEPTH,AFRAC,ARES,VBLK(ISZ),VFRBLK,VRSBLK,VPORE
      COMMON / XPERM /    PRMF,PRMR,PORF,PORR,AKR(3,ISZ),F(3,ISZ),
     .                   TMOB(ISZ),SWC,SOR,SOG,SGC,EKRO,EKRW,EKROG,
     .                   EKRG,EO,EW,EOG,EG,S(3,ISZ),SGCX(ISZ)
      COMMON / PRM /     IPERM,NSW,NSL,TSW(3,20),TSL(3,20),ITPT
      COMMON / VISC /    VISO(ISZ),VISG,VISW,AVIS1,AVIS2,BVIS,CVIS
      DIMENSION SX(3,ISZ),VISX(ISZ),XSGC(ISZ),AKRX(3,ISZ),FX(3,ISZ),
     .          XTMOB(ISZ)
C
C            CALCULATE RELATIVE PERMEABILITIES OF FLUID PHASES
C
      KB = 2
      DO 1000 K=2,NBFIN
      IF (K.EQ.KB) THEN
C             SET REL. PERMS. IN FRACTURE EQUAL TO SATN'S.
         AKRX(1,K) = SX(1,K)
         AKRX(2,K) = SX(2,K)
         AKRX(3,K) = SX(3,K)
         KB=KB+NR
      ELSE
C
      DSW=1.-SOR-SWC
      SW=SX(3,K)
      SG=SX(1,K)
      IF (IPERM .EQ. 0) GO TO 245
C
C               TABLE LOOK-UP AND INTERPOLATION OF REL. PERMS.
      EKRO=TSW(3,1)
      IF (SW .GT. TSW(1,1)) GO TO 205
      AKRW=0.
      IF (TSW(1,1) .EQ. 0.) GO TO 202
      AKROW=1.-SW*(1.-TSW(3,1))/TSW(1,1)
      GO TO 225
  202 AKROW=1.
      GO TO 225
  205 IF (SW .LT. TSW(1,NSW)) GO TO 210
      AKROW=0.
      AKRW=1.-(1.-SW)*(1.-TSW(2,NSW))/(1.-TSW(1,NSW))
      GO TO 225
  210 DO 215 L=2,NSW
         IF (SW .LE. TSW(1,L)) GO TO 220
  215 CONTINUE
```

```
  220 LM1=L-1
      DTSW=TSW(1,L)-TSW(1,LM1)
      DSW=SW-TSW(1,LM1)
      AKRW=TSW(2,LM1)+DSW*(TSW(2,L)-TSW(2,LM1))/DTSW
      AKROW=TSW(3,LM1)+DSW*(TSW(3,L)-TSW(3,LM1))/DTSW
  225 SL=1.-SG
      IF (SL .GT. TSL(1,1)) GO TO 230
      AKROG=0.
      AKRG=1.-SL*(1.-TSL(3,1))/TSL(1,1)
      GO TO 600
  230 IF (SL .LT. TSL(1,NSL)) GO TO 235
      AKRG=0.
      AKROG=TSL(2,NSL)+(SL-TSL(1,NSL))*(1.-TSL(2,NSL))/(1.-TSL(1,NSL))
      GO TO 600
  235 DO 240 L=2,NSL
         IF (SL .LE. TSL(1,L)) GO TO 242
  240 CONTINUE
  242 LM1=L-1
      DTSL=TSL(1,L)-TSL(1,LM1)
      DSL=SL-TSL(1,LM1)
      AKROG=TSL(2,LM1)+DSL*(TSL(2,L)-TSL(2,LM1))/DTSL
      AKRG=TSL(3,LM1)+DSL*(TSL(3,L)-TSL(3,LM1))/DTSL
      GO TO 600
C
C                    COREY RELATIVE PERMEABILITY CURVES
C
C                          CALCULATE OIL-WATER RELATIVE PERMEABILITIES
  245 IF (SW .GE. SWC) GO TO 250
      AKRW=0.
      AKROW=1.-SW*(1.-EKRO)/SWC
      GO TO 400
  250 IF (SW .LE. 1.-SOR) GO TO 300
      AKROW=0.
      AKRW=1.-(1.-SW)*(1.-EKRW)/SOR
      GO TO 400
  300 AKRW=EKRW*((SW-SWC)/DSW)**EW
      AKROW=EKRO*((1.-SW-SOR)/DSW)**EO
```

```
C                                CALCULATE OIL-GAS RELATIVE PERMEABILITIES
  400 SL=1.-SG
      IF (SL .GE. SWC+SOG) GO TO 450
      AKROG=0.
      AKRG=1.-SL*(1.-EKRG)/(SWC+SOG)
      GO TO 600
 450  IF (SG-XSGC(K).GT.1.E-3) GO TO 500
      AKRG=0.
      AKROG=EKRO*((SL-SWC-SOG)/(1.-SWC-SOG))**EOG
      GO TO 600
 500  AKRG=EKRG*((SG-XSGC(K))/(1.-XSGC(K)-SOG-SWC))**EG
      AKROG = EKRO * ((SL-SWC-SOG)/(1.-SWC-SOG))**EOG
C
 600  AKRO=(AKROW+AKRW)*(AKROG+AKRG)/EKRO-(AKRW+AKRG)
      IF (AKRO .LT. 0.) AKRO=0.
      AKRX(1,K) = AKRG
      AKRX(2,K) = AKRO
      AKRX(3,K) = AKRW
      END IF
C
C         CALCULATE TOTAL FLUID MOBILITY AND FRACTIONAL FLOW FUNCTIONS
      UMOB1 = AKRX(1,K) / VISG
      UMOB2 = AKRX(2,K) / VISX(K)
      UMOB3 = AKRX(3,K) / VISW
      XTMOB(K) = UMOB1+UMOB2+UMOB3
      FX(1,K) = UMOB1 / XTMOB(K)
      FX(2,K) = UMOB2 / XTMOB(K)
      FX(3,K) = UMOB3 / XTMOB(K)
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRNTS
C
      PARAMETER (ISZ=52)
      COMMON / GRID /    DT,DZ,DY,DTDZ,DTDY,DTT,DTDZT,DTDYT,TIME,DTNEW,
     .                   DTSTBL,NBZ,NBY,NR,NBFIN,FRLTH,RSLTH,HFRAC,
     .                   DEPTH,AFRAC,ARES,VBLK(ISZ),VFRBLK,VRSBLK,VPORE
      COMMON / COUNT /   IT,ITMAX,ITCNT,ICNVRG,IBUBBL,IDPCHG,IOVERF,
     .                   IOVERR,ISTABL,IMOB,IMOB1,IPRNT,NOUNIT,MXCNVRG
      COMMON / TIMES /   TPHASE,TIMPEM,TDPCHG(100),TSTRT,TOVERF(100),
     .                   TOVERR(100),TBUBBL(100),XJUMP
      COMMON / MOLES /   CN(3,ISZ),CNG(3,ISZ),CNGC(ISZ),X(3,3,ISZ),
     .                   OILSOL,WATSOL,RATSOL,OILRAT,WATRAT,OIPMOL(3)
      COMMON / PRESS /   PRSAV(ISZ),PRSOLD(ISZ),PDSOAK,DPLIM,PRSI,
     .                   PRSMAX,PRSMIN
      COMMON / PRSMAT /  ALPHA(ISZ),BETA(ISZ),GAMMA(ISZ),EPSLON(ISZ),
     .                   THETA(ISZ),DELTA2(ISZ),DELTA3(ISZ),A(ISZ),
     .                   W1Z(ISZ),W2Z(ISZ),W3Z(ISZ),W1Y(ISZ),W2Y(ISZ),
     .                   W3Y(ISZ)
      COMMON / FLOW /    QZ(ISZ),QY(ISZ),QFLOW,QINJ1,QINJ2,QPROD,
     .                   IWELL(50),JWELL,TWELL,TLIMIN,TLIMPR,TLIMSK
      COMMON / XDENSE /   RHOMOL(3,ISZ),RHO(3,ISZ),VN(3,3,ISZ),
     .                   CMPEFF(ISZ),ZSLOPE,A22DEN,B22DEN,ASWELL,
     .                   A33DEN,B33DEN,A3SDEN,B3SDEN,V22ATM,WT(3)
      COMMON / VISC /    VISO(ISZ),VISG,VISW,AVIS1,AVIS2,BVIS,CVIS
      COMMON / XPERM /    PRMF,PRMR,PORF,PORR,AKR(3,ISZ),F(3,ISZ),
     .                   TMOB(ISZ),SWC,SOR,SOG,SGC,EKRO,EKRW,EKROG,
     .                   EKRG,EO,EW,EOG,EG,S(3,ISZ),SGCX(ISZ)
      COMMON / PRM /     IPERM,NSW,NSL,TSW(3,20),TSL(3,20),ITPT
      COMMON / TMPVAL /  PRM1(3,ISZ),PRMT(3,ISZ),XCN(3,ISZ),XPRSAV(ISZ),
     .                   RHOT(3,ISZ),XT(3,3,ISZ),FT(3,ISZ),TMOBT(ISZ)
      COMMON / UPDATE /  CUMI(3),CUMP(3),CMOLIN(3),CMOLPR(3),PROD(3)
      COMMON / MISC /    C(3),TITLE(20),ILIMIT,IP(3),ISTEP(4),C22DUM,
     .                   C13DUM,C33DUM
C
C
      DIMENSION KK(ISZ)
C ************************************************************
C
      IWRIT1 = 8
C                NOTE: IWRIT1=7 => WRITE MATRIX COEFFS.
C                      IWRIT1=8 => WRITE NO. MOLES OF I IN J
      IFLAG=0
      ITT=IT/IPRNT
      DIT=FLOAT(IT)/FLOAT(IPRNT)
      DDIT=DIT-FLOAT(ITT)
      IF (ABS(DDIT) .LT. 1.E-7) IFLAG=1
C
```

```
      WRITE(NOUNIT,2) IT,TIME,DTT,(CUMI(I),I=1,3),
     1  (CMOLIN(I),I=1,3),(CMOLPR(I),I=1,3)
    2 FORMAT (//' IT=',I5,10X,'TIME=',F10.5,5X,'STEP = ',F10.5,' SEC',
     1       /1X,33('=')//' CUMI (PV)'
     1 ,6X,3F10.6/' CMOLIN (MOLES)',6X,3F10.6/' CMOLPR (MOLES)',
     2                 6X,3F10.6/)
C
       IF (IWELL(JWELL).EQ.1) WRITE (NOUNIT,30)
       IF (IWELL(JWELL).EQ.2) WRITE (NOUNIT,31)
       IF (IWELL(JWELL).EQ.-1) WRITE (NOUNIT,32)
       IF (IABS(IWELL(JWELL)).EQ.10) WRITE (NOUNIT,33)
 30     FORMAT (/' WELL MODE:  CO2 INJECTION')
 31     FORMAT (/' WELL MODE:  H2O INJECTION')
 32     FORMAT (/' WELL MODE:  PRODUCTION')
 33     FORMAT (/' WELL MODE:  SOAK')
C
      NCYCLE = 1 + (JWELL-1) / 4
      WRITE (NOUNIT,42) NCYCLE
 42    FORMAT (5X,'CYCLE:',I5,//)
C
C                    WRITE TIMES SPENT IN SUBROUTINES
C                          AND ADJUSTED TIME STEPS
      WRITE (NOUNIT,39) TSTRT
 39    FORMAT (//10X,'TSTRT - INITIAL STEP (SEC) = ',E10.4)
      WRITE (NOUNIT,34)
 34   FORMAT (10X,'TDPCHG(I) - TIME STEP AFTER TRUNCATION ADJUST.')
      WRITE (NOUNIT,35) (TDPCHG(I),I=1,IDPCHG)
 35   FORMAT (10X,10(2X,E10.4))
      WRITE (NOUNIT,36)
 36   FORMAT (10X,'TOVERF - TIME STEP AFTER OVERSHOOT ADJUST.(FRAC.)')
      WRITE (NOUNIT,35) (TOVERF(I),I=1,IOVERF)
      WRITE (NOUNIT,37)
 37   FORMAT (10X,'TOVERR - TIME STEP AFTER OVERSHOOT ADJUST.(RES.)')
      WRITE (NOUNIT,35) (TOVERR(I),I=1,IOVERR)
      WRITE (NOUNIT,38)
 38   FORMAT (10X,'TBUBBL - TIME STEP AFTER BUBBLE EDGE ADJUST.')
      WRITE (NOUNIT,35) (TBUBBL(I),I=1,IBUBBL)
      WRITE (NOUNIT,40) IMOB
 40   FORMAT (10X,'IMOB - PASSES THROUGH SUBR. MOBITR = ',I3)
      WRITE (NOUNIT,29) TIMPEM
 29   FORMAT (//10X,'TIME SPENT IN SUBR. IMPEM = ',E10.4,' SECONDS.')
      WRITE (NOUNIT,27) TPHASE
 27   FORMAT (10X,'TIME SPENT IN SUBR. PHASE = ',E10.4,' SECONDS.'//)
C
C
      IF (IFLAG .NE. 1) RETURN
```

```
C
C                 PRINT PRESSURES, RELATIVE FLOW RATES, PHASE DENSITIES,
C                       AND OVERALL MOLES OF COMPONENT I PER VOL.
C
      WRITE (NOUNIT,20)
 20     FORMAT (/1X,'PRESSURES',/,12X,'(FRACTURE ------>)')
      DO 21 KF=1,NBZ
 21         KK(KF) = 2+(KF-1)*NR
      WRITE (NOUNIT,24)  (KK(I),I=1,NBZ)
 24       FORMAT (/1X,'K =',2X,15(I2,6X))
      DO 25 J=1,NR
 25     WRITE (NOUNIT,23)  (PRSAV(KK(I)+J-1),I=1,NBZ)
 23       FORMAT (3X,15F8.3)
      IF (IWRIT1.EQ.7) WRITE (NOUNIT,7)
      IF (IWRIT1.EQ.8) WRITE (NOUNIT,8)
 7          FORMAT (//T3,'K',T12,'QZ',T23,'QY',T30,'ALPHA',T45,
     1            'BETA',T60,'GAMMA',T75,'THETA',T90,'EPSLON',
     2               T105,'CMPEFF',T120,'VTOT'/)
 8          FORMAT (//T3,'K',T12,'QZ',T23,'QY',T33,'N11',T45,'N12',
     1      T58,'N22',T70,'N13',T80,'N33',T93,'CMPEFF',T106,'VTOT',
     2      T120,'VOLERR'/)
      WRITE (NOUNIT,26) QZ(1)
 26          FORMAT (1X,' 1',2X,E10.4)
      DO 700 K=2,NBFIN
        VTOT = CNG(1,K)*VN(1,1,K) + CNG(2,K)*VN(1,2,K) +
     1         CN(2,K)*VN(2,2,K) + CNG(3,K)*VN(1,3,K) +
     2         CN(3,K)*VN(3,3,K)
        VOLERR = (VTOT-VBLK(K))/VBLK(K)
        CMPDUM = CMPEFF(K) / VTOT
        IF (IWRIT1.EQ.7) WRITE (NOUNIT,601) K,QZ(K),QY(K),
     1        ALPHA(K),BETA(K),GAMMA(K),THETA(K),EPSLON(K),CMPEFF(K)
        IF(IWRIT1.EQ.8) WRITE (NOUNIT,601) K,QZ(K),QY(K),CNG(1,K),
     1         CNG(2,K),CN(2,K),CNG(3,K),CN(3,K),CMPDUM,VTOT,VOLERR
 601     FORMAT (1X,I2,2(2X,E10.4),8(2X,E10.4))
 700      CONTINUE
C
C
C                 WRITE VIJ QUANTITIES
      WRITE (NOUNIT,51)
 51   FORMAT (//T3,'K',T9,'V11',T24,'V12',T39,'V22',T54,'V13',T69,
     1            'V33',T95,'RHOMOL(J)'/)
      DO 18 K=2,NBFIN
 18   WRITE (NOUNIT,12) K,VN(1,1,K),VN(1,2,K),VN(2,2,K),
     1         VN(1,3,K),VN(3,3,K),RHOMOL(1,K),RHOMOL(2,K),RHOMOL(3,K)
 12   FORMAT (1X,I2,3X,8(E10.4,5X))
```

```
C
C                 PRINT TABLE HEADING FOR OVERALL MOLE AND SATURATION
C                           FRACTION DATA
C
      WRITE(NOUNIT,3)
    3 FORMAT (/8X,'Z1',6X,'Z2',6X,'Z3',6X,'S1',6X,
     1  'S2',6X,'S3',6X,'F1',6X,'F2',6X,'F3',5X,'TMOB',10X,'X(1,2)',
     2                                          10X,'X(1,3)')
    4 FORMAT (1X,I2,10F8.5,2(6X,F10.6))
C
      KB=2
      DO 200 KF=1,NBZ
        KBDUM=KB + NBY
        DO 190 K=KB,KBDUM
C
       DENOM = CN(1,K) - CNGC(K) + CN(2,K) + CN(3,K)
       Z2 = CN(2,K) / DENOM
       Z3 = CN(3,K) / DENOM
       Z1 = 1.- Z2 - Z3
       IF (Z1.LT.1.E-6) Z1=0
C
C
         WRITE(NOUNIT,4)K,Z1,Z2,Z3,S(1,K),S(2,K),S(3,K),
     +      F(1,K),F(2,K),F(3,K),TMOB(K),X(1,2,K),X(1,3,K)
  190 CONTINUE
  200  KB = 2 + KF * NR
C
      DO 300 J=1,3
         IF (IP(J) .NE. 1) GO TO 300
         IHD=0
         DO 275 K=2,NBFIN
            IF (J.EQ.1 .AND. S(1,K)-SGCX(K) .LT.1.E-3) GO TO 275
            IF (S(J,K) .LT. 1.E-6) GO TO 275
            IF (IHD .EQ. 1) GO TO 225
            IHD=1
            WRITE(NOUNIT,5) J
 225        IF (J.EQ.1)
     +      WRITE(NOUNIT,15)K,S(J,K),X(1,J,K),X(2,J,K),X(3,J,K),RHO(J,K)
     +            ,VISG,AKR(J,K)
            IF (J.EQ.2)
     +      WRITE(NOUNIT,15)K,S(J,K),X(1,J,K),X(2,J,K),X(3,J,K),RHO(J,K)
     +            ,VISO(K),AKR(J,K)
            IF (J.EQ.3)
     +      WRITE(NOUNIT,15)K,S(J,K),X(1,J,K),X(2,J,K),X(3,J,K),RHO(J,K)
     +            ,VISW,AKR(J,K)
  275    CONTINUE
  300 CONTINUE
    5 FORMAT (//7X,'S',I1,6X,'X1',6X,'X2',6X,'X3',5X,'RHO',6X,'VIS',
     +          6X,'KR'/)
 15   FORMAT (1X,I2,5F8.4,F9.4,4F8.4)
      RETURN
      END
```

```
      SUBROUTINE IMPEM
C
      PARAMETER (ISZ=52)
      COMMON / GRID /    DT,DZ,DY,DTDZ,DTDY,DTT,DTDZT,DTDYT,TIME,DTNEW,
     .                   DTSTBL,NBZ,NBY,NR,NBFIN,FRLTH,RSLTH,HFRAC,
     .                   DEPTH,AFRAC,ARES,VBLK(ISZ),VFRBLK,VRSBLK,VPORE
      COMMON / COUNT /   IT,ITMAX,ITCNT,ICNVRG,IBUBBL,IDPCHG,IOVERF,
     .                   IOVERR,ISTABL,IMOB,IMOB1,IPRNT,NOUNIT,MXCNVRG
      COMMON / TIMES /   TPHASE,TIMPEM,TDPCHG(100),TSTRT,TOVERF(100),
     .                   TOVERR(100),TBUBBL(100),XJUMP
      COMMON / MOLES /   CN(3,ISZ),CNG(3,ISZ),CNGC(ISZ),X(3,3,ISZ),
     .                   OILSOL,WATSOL,RATSOL,OILRAT,WATRAT,OIPMOL(3)
      COMMON / PRESS /   PRSAV(ISZ),PRSOLD(ISZ),PDSOAK,DPLIM,PRSI,
     .                   PRSMAX,PRSMIN
      COMMON / PRSMAT /  ALPHA(ISZ),BETA(ISZ),GAMMA(ISZ),EPSLON(ISZ),
     .                   THETA(ISZ),DELTA2(ISZ),DELTA3(ISZ),A(ISZ),
     .                   W1Z(ISZ),W2Z(ISZ),W3Z(ISZ),W1Y(ISZ),W2Y(ISZ),
     .                   W3Y(ISZ)
      COMMON / FLOW /    QZ(ISZ),QY(ISZ),QFLOW,QINJ1,QINJ2,QPROD,
     .                   IWELL(50),JWELL,TWELL,TLIMIN,TLIMPR,TLIMSK
      COMMON / XDENSE /   RHOMOL(3,ISZ),RHO(3,ISZ),VN(3,3,ISZ),
     .                   CMPEFF(ISZ),ZSLOPE,A22DEN,B22DEN,ASWELL,
     .                   A33DEN,B33DEN,A3SDEN,B3SDEN,V22ATM,WT(3)
      COMMON / VISC /    VISO(ISZ),VISG,VISW,AVIS1,AVIS2,BVIS,CVIS
      COMMON / XPERM /    PRMF,PRMR,PORF,PORR,AKR(3,ISZ),F(3,ISZ),
     .                   TMOB(ISZ),SWC,SOR,SOG,SGC,EKRO,EKRW,EKROG,
     .                   EKRG,EO,EW,EOG,EG,S(3,ISZ),SGCX(ISZ)
      COMMON / PRM /     IPERM,NSW,NSL,TSW(3,20),TSL(3,20),ITPT
      COMMON / TMPVAL /  PRM1(3,ISZ),PRMT(3,ISZ),XCN(3,ISZ),XPRSAV(ISZ),
     .                   RHOT(3,ISZ),XT(3,3,ISZ),FT(3,ISZ),TMOBT(ISZ)
      COMMON / UPDATE /  CUMI(3),CUMP(3),CMOLIN(3),CMOLPR(3),PROD(3)
      COMMON / MISC /    C(3),TITLE(20),ILIMIT,IP(3),ISTEP(4),C22DUM,
     .                   C13DUM,C33DUM
C
      REAL CTMOL(4)
      DATA TDPCHG/100*0.0/ TOVERF/100*0.0/ TOVERR/100*0.0/
      DATA TBUBBL/100*0.0/
C
C
C                 INCREMENT TIME STEP COUNTER AND CHANGE INJECTION
C                     RATE IF TIME MATCHES INPUT PARAMETERS
C
      IT=IT+1
      ITCNT = ITCNT + 1
C
      DTT = DT
      DTDZT = DTDZ
      DTDYT = DTDY
      DTSTBL = DT
```

```
C                    STORE OLD PRESSURES
      DO 50 K=2,NBFIN
 50       PRSOLD(K) = PRSAV(K)
      KPY = 1
      KNY = 1
      KPZ = 1
      KPY = 1
      CALL FLOCHG
      IF (ABS(QZ(1)).LT.1.E-4)   THEN
         IF (ITCNT.EQ.1) CALL STPCHG (.1,DTT,DTDZT,DTDYT)
         IF (ITCNT.GE.3) CALL STPCHG (10.,DTT,DTDZT,DTDYT)
         GO TO 90
      END IF
C                                          ELSE ENTER RAMP
      IF (ITCNT.EQ.1) CALL STPCHG (.1,DTT,DTDZT,DTDYT)
      IF (ITCNT.EQ.2) CALL STPCHG (.5,DTT,DTDZT,DTDYT)
C
C        ******************************************************
C
C              P R E S S U R E       E Q U A T I O N
C
C        SET UP PRESSURE MATRIX EQUATION ( SEE NOTES FOR DERIVATION
C                          OF FINITE DIFFERENCED PREESURE SCHEME )
 90   IDPCHG = 0
      IOVERF = 0
      IOVERR = 0
      TSTRT = DTT
      IBUBBL = 0
      IMOB = 0
C                         RESET VARIABLES AND PRESS. EQN. COEFFS.
 100  ICNVRG = 0
      IMOB1 = 0
      DO 105 K=1,NBFIN
         DO 104 J =1,3
         DO 103 I=1,3
 103     XT(I,J,K) = 0
         XT(1,J,K) = X(1,J,K)
         RHOT(J,K) = RHOMOL(J,K)
         PRM1(J,K) =AKR(J,K)
 104     FT(J,K) = F(J,K)
         XT(2,2,K) = X(2,2,K)
         XT(3,3,K) = X(3,3,K)
         TMOBT(K) = TMOB(K)
         W1Y(K) = 0
         W2Y(K) = 0
         W3Y(K) = 0
         ALPHA(K) =0
         BETA(K)  =0
         GAMMA(K) =0
         THETA(K) =0
 105     EPSLON(K)=0
```

```
C
C                    CALCULATE RELATIVE MOBILITIES AND FLOW TERMS
C
      CALL FLOFN1 (PRM1,VISO,RHOMOL,X)
      CALL FLOFN2 (CNG,CN,VN,X,PRSAV,CMPEFF,S,SGCX)
C
 140  QFLOW =QFLOFN(QZ(1),DTT,RHOMOL(1,1),RHOMOL(3,1),IWELL(JWELL),
     1              RHOMOL(1,2),RHOMOL(2,2),RHOMOL(3,2),F(1,2),
     2              F(2,2),F(3,2),X(2,2,2),X(3,3,2),DELTA2(2),
     3              DELTA3(2),X(1,2,2),X(1,3,2))
C
C              ENTER PRESSURE/COMPOSITION ITERATIVE CALCULATION LOOP
 150    CONTINUE
       KB=2
       RFRAC=2.* DTDYT
C
       DO 250 KF=1,NBZ
         KBDUM=KB+NBY
         DO 220 K=KB,KBDUM
C
C                        DEFINE UPSTREAM INDICES BASED ON
C                         MOST RECENT FLOW RATES
C
            IF (K.EQ.2) THEN
C                                          FIRST FRACTURE BLOCK
               IF (QZ(K).GE.0) KPZ=K
               IF (QZ(K).LT.0) KPZ=K+NR
               IF (QZ(1).GE.0) KNZ=1
               IF (QZ(1).LT.0) KNZ=K
               IF (NBY.NE.0) THEN
                  IF (QY(K).GE.0) KPY=K
                  IF (QY(K).LT.0) KPY=K+1
               END IF
            ELSE IF (K.EQ.KB .AND. K.GT.2) THEN
C                                         REMAINING FRACTURE BLOCKS
               IF (QZ(K).GE.0) KPZ=K
               IF (QZ(K).LT.0) KPZ=K+NR
               IF (QZ(K-NR).GE.0) KNZ=K-NR
               IF (QZ(K-NR).LT.0) KNZ=K
               IF (NBY.NE.0) THEN
                  IF (QY(K).GE.0) KPY=K
                  IF (QY(K).LT.0) KPY=K+1
               END IF
               GO TO 180
            ELSE
C                                          RESERVOIR MATRIX BLOCKS
               IF (QY(K).GE.0) KPY=K
               IF (QY(K).LT.0) KPY=K+1
               IF (QY(K-1).GE.0) KNY=K-1
               IF (QY(K-1).LT.0) KNY=K
               GO TO 200
            END IF
```

```
C                                   BRANCH TO APPROPRIATE SYSTEM EQN.
C                                    AND BUILD COEFFICIENTS
C
C                       INJECTION B.C. PRESSURE EQN.
C
            ALPHA(2) = 0
            BETA(2) = -( DTDZT * AFRAC * (W1Z(KPZ)
     1                + DELTA2(K)*W2Z(KPZ) + DELTA3(K)*W3Z(KPZ))
     1                + A(K) + RFRAC * ARES *(W1Y(KPY)
     2                + DELTA2(K)*W2Y(KPY) + DELTA3(K)*W3Y(KPY)))
            GAMMA(2) = DTDZT*(W1Z(KPZ) + DELTA2(K)*W2Z(KPZ) +
     1                DELTA3(K)*W3Z(KPZ)) * AFRAC
            EPSLON(2) = RFRAC*(W1Y(KPY) + DELTA2(K)*W2Y(KPY) +
     1                DELTA3(K)*W3Y(KPY)) * ARES
            IF (NBZ.EQ.1) THEN
               BETA(2) = BETA(2) + GAMMA(2)
               GAMMA(2) = 0
            END IF
            THETA(2) = - A(K) * PRSAV(2) - QFLOW
            GO TO 220
C
C                 SET FRACTURE COEFFICIENTS
C
 180        ALPHA(K) = DTDZT * (W1Z(KNZ) + DELTA2(K)*W2Z(KNZ) +
     1                DELTA3(K)*W3Z(KNZ)) * AFRAC
            BETA(K) = -( DTDZT * AFRAC * (W1Z(KPZ)+W1Z(KNZ)
     1                + DELTA2(K) * (W2Z(KPZ)+W2Z(KNZ)) + DELTA3(K)*
     1                (W3Z(KPZ) + W3Z(KNZ))) + A(K) +
     2                ARES *  RFRAC * (W1Y(KPY) +
     3                DELTA2(K)*W2Y(KPY) + DELTA3(K)*W3Y(KPY)))
            GAMMA(K) = DTDZT * (W1Z(KPZ) + DELTA2(K)*W2Z(KPZ) +
     1                DELTA3(K)*W3Z(KPZ)) * AFRAC
C                           END OF FRACTURE B.C.
            IF (K.EQ.2+(NBZ-1)*NR) THEN
               BETA(K) = BETA(K) + GAMMA(K)
               GAMMA(K) = 0
            END IF
            EPSLON(K) = RFRAC * (W1Y(KPY) + DELTA2(K)*W2Y(KPY) +
     1                DELTA3(K)*W3Y(KPY)) * ARES
            THETA(K) = - A(K) * PRSAV(K)
            GO TO 220
```

```
C
C                    SET RESERVOIR COEFFICIENTS
C
 200        ALPHA(K) = DTDYT * (W1Y(KNY) + DELTA2(K)*W2Y(KNY) +
     1                 DELTA3(K)*W3Y(KNY)) * ARES
            IF (K.EQ.KB+1) ALPHA(K) = 2 * ALPHA(K)
            BETAP = DTDYT * (W1Y(KPY) + DELTA2(K) * W2Y(KPY) +
     1              DELTA3(K) * W3Y(KPY))
            BETAN = DTDYT * (W1Y(KNY) + DELTA2(K) * W2Y(KNY) +
     1              DELTA3(K) * W3Y(KNY))
            IF (K.EQ.KB+1) BETAN = 2 * BETAN
            BETA(K) = -((BETAN + BETAP) * ARES + A(K))
            EPSLON(K) = DTDYT * (W1Y(KPY) + DELTA2(K)*W2Y(KPY) +
     1                  DELTA3(K)*W3Y(KPY)) * ARES
            IF (K.EQ.KB+NBY) BETA(K) = -ALPHA(K)-A(K)
            IF (K.EQ.KB+NBY) EPSLON(K) = 0
            THETA(K) = - A(K) * PRSAV(K)
C
 220      CONTINUE
       KB=2+KF*NR
 250   CONTINUE
C
C                    SOLVE PRESSURE MATRIX
C
       CALL GAUSS
C
C              CHECK TRUNCATION ERROR (MAX. PRESSURE DIFFERENCE
C                                      FROM LAST STEP)
         DPMAX = 0
         DO 260 K=2,NBFIN
            PDIF = ABS(XPRSAV(K)-PRSOLD(K))
 260        DPMAX = AMAX1 (DPMAX,PDIF)
         IF (DPMAX-DPLIM.GT.1.E-1) THEN
            CALL STPCHG (DPLIM/DPMAX,DTT,DTDZT,DTDYT)
            IDPCHG = IDPCHG + 1
            TDPCHG(IDPCHG) = DTT
            GO TO 140
         END IF
```

```
C
C     ******************************************************************
C
C            C O M P O S I T I O N     E Q U A T I O N
C
C      CALCULATE RELATIVE FLOW RATES IN GRID BLOCK K
C
       KB=2
       DO 300 KF=1,NBZ
          IF (KB.GE.NBFIN-NBY) GO TO 270
          PRSDIF = XPRSAV(KB) - XPRSAV(KB+NR)
          IF (PRSDIF.GE.0) KBUPZ = KB
          IF (PRSDIF.LT.0) KBUPZ = KB + NR
          QZ(KB) = TMOBT(KBUPZ) * PRMF * 6.8E-5 * AFRAC * PRSDIF /
     1                                          (FRLTH * DZ)
 270      IF (NBY.EQ.0) GO TO 300
          PRSDIF =(XPRSAV(KB) - XPRSAV(KB+1))
          IF (PRSDIF.GE.0) KBUPY = KB
          IF (PRSDIF.LT.0) KBUPY = KB + 1
          QY(KB) = TMOBT(KBUPY) * PRMR * 6.8E-5 * ARES * 2 * PRSDIF /
     1                                 (DY * RSLTH)
          DO 290 K=KB+1,KB+NBY-1
             PRSDIF = XPRSAV(K) - XPRSAV(K+1)
             IF (PRSDIF.GE.0) KUP = K
             IF (PRSDIF.LT.0) KUP = K + 1
 290         QY(K) = TMOBT(KUP) * PRMR * 6.8E-5 * ARES * PRSDIF /
     1                                        (RSLTH * DY)
C                          END OF RESERVOIR B.C.
           QY(KB+NBY) = 0
 300       KB = 2 + KF * NR
C                          END OF FRACTURE B.C.
          QZ(NBFIN-NBY) = 0
C
C      CALCULATE COMPOSITIONS (MOLES I) AT NEW TIME LEVEL
C
C                 SOLVE COMPOSITION EQUATION IN FRACTURE
       DO 400 KF=1,NBZ
           K=2+(KF-1)*NR
           KM1=K-NR
           IF (K.EQ.2) KM1=1
            IF (QZ(KM1).GE.0) K1 = KM1
            IF (QZ(KM1).LT.0) K1 = K
            IF (QZ(K).GE.0) K2Z = K
            IF (QZ(K).LT.0) K2Z = K + NR
            IF (QY(K).GE.0) K2Y = K
            IF (QY(K).LT.0) K2Y = K + 1
```

```
      DO 330 I=1,3
330        CTMOL(I) = 0
          DO 350 J=1,3
             DO 350 I=1,3
                IF (J.EQ.1 .AND. I.NE.1) GO TO 350
                IF (J.EQ.2 .AND. I.EQ.3) GO TO 350
                IF (J.EQ.3 .AND. I.EQ.2) GO TO 350
                CHG = DTT * (QZ(KM1) * RHOT(J,K1) * XT(I,J,K1) *
     1                FT(J,K1) - QZ(K) * RHOT(J,K2Z) * XT(I,J,K2Z) *
     2                FT(J,K2Z) -  QY(K) *
     3                RHOT(J,K2Y) * XT(I,J,K2Y) * FT(J,K2Y))
                CTMOL(I) = CTMOL(I) + CHG
350             CONTINUE
          DO 370 I=1,3
C                    OVERSHOOT CHECK
               IF (CN(I,K)+CTMOL(I).LT.-1.E-6) GO TO 410
370            XCN(I,K) = CN(I,K) + CTMOL(I)
400    CONTINUE
       GO TO 415
410         CALL STPCHG (0.5,DTT,DTDZT,DTDYT)
            IOVERF = IOVERF + 1
            TOVERF(IOVERF) = DTT
                  IF (DTT.LT.1.E-5) THEN
                      PRINT 430,DTT
                      STOP
430            FORMAT ('STOP: OVERSHOOT. (FRAC.) DTT = ',E10.4)
                  END IF
            GO TO 150
415   IF (NBY.EQ.0) GO TO 550
C                  SOLVE COMPOSITION EQN. IN RESERVOIR
      KB=2
      DO 500 KF=1,NBZ
         KBI = KB+1
         KBF = KB+NBY
         DO 490 K=KBI,KBF
            KM1=K-1
             IF (QY(KM1).GE.0) K1 = KM1
             IF (QY(KM1).LT.0) K1 = K
             IF (QY(K).GE.0) K2 = K
             IF (QY(K).LT.0) K2 = K + 1
      DO 425 I=1,3
425        CTMOL(I)=0
         DO 450 J=1,3
            DO 450 I=1,3
                IF (J.EQ.1 .AND. I.NE.1) GO TO 450
                IF (J.EQ.2 .AND. I.EQ.3) GO TO 450
                IF (J.EQ.3 .AND. I.EQ.2) GO TO 450
                CHG = DTT * (QY(KM1) * RHOT(J,K1) * XT(I,J,K1) *
     1                FT(J,K1) - QY(K) * RHOT(J,K2) * XT(I,J,K2) *
     2                FT(J,K2))
                CTMOL(I) = CTMOL(I) + CHG
450             CONTINUE
```

```
C
C                        OVERSHOOT CHECK
          DO 470 I=1,3
              IF (CN(I,K)+CTMOL(I).LT.-1.E-6) GO TO 520
C
C                        TEMPORARY NEW COMPOSITION
 470          XCN(I,K) = CN(I,K) + CTMOL(I)
 490       CONTINUE
          KB = 2 + KF * NR
 500     CONTINUE
        GO TO 550
 520      CALL STPCHG (0.5,DTT,DTDZT,DTDYT)
          IOVERR = IOVERR + 1
          TOVERR(IOVERR) = DTT
                 IF (DTT.LT.1.E-5) THEN
                         PRINT 431,DTT
                         STOP
 431          FORMAT ('STOP: OVERSHOOT (RES) DTT = ',E10.4)
               END IF
          GO TO 150
C
C                    CALCULATE DF/DS AND EVALUATE NUMERICAL STABILITY.
C                         ADJUST TIME STEP IF NECESSARY.
C550   IF (ISTABL.EQ.0) CALL STABL (*150,*560)
 550                 CONTINUE
C
C                       ITERATE ON MOBILITIES UNTIL CONVERGENCE
C                        OF RELATIVE PERMEABILITIES
          CALL MOBITR (*100,*150)
C
C                         NEW PRESSURES AND COMPOSITIONS
          DO 570 K=2,NBFIN
            PRSAV(K) = XPRSAV(K)
            DO 570 I=1,3
 570          CN(I,K) = XCN(I,K)
C
C              UPDATE PRODUCTION AND INJECTION DATA (MOLAR)
C
             IF (QZ(1) .GT. 0) GO TO 600
C
       DO 580 J=1,3
             DO 580 I=1,3
 580             CMOLPR(I) = CMOLPR(I) - DTDZT * QZ(1) * RHOMOL(J,2) *
     1                                   X(I,J,2) * F(J,2) * FRLTH * DZ
```

```
C
C
C
C                    CALCULATE APPROXIMATE PORE VOLUMES PRODUCED
C
                RHOOIL = (3.53E-6 * PRSAV(2) + .945) / WT(2)
C
C
                CUMP(1) = CMOLPR(1) /(RHOMOL(1,2) * VFRBLK)
                CUMP(2) = CMOLPR(2) /(RHOOIL * VFRBLK)
                CUMP(3) = CMOLPR(3) * 18. / VFRBLK
C
C                    CALCULATE FRACTION OF OIP PRODUCED
C
                DO 590 I=1,3
                   PROD(I) = CMOLPR(I) / OIPMOL(I)
 590            CONTINUE
C
          GO TO 700
C
C
C                  MOLES INJECTED (ACCUMULATED)
C
 600    IF (IWELL(JWELL).EQ.1) CMOLIN(1) = CMOLIN(1) + QZ(1) * DTT *
     1                                                   RHOMOL(1,1)
        IF (IWELL(JWELL) ..2) CMOLIN(3) = CMOLIN(3) + QZ(1) * DTT *
     3                                                   RHOMOL(3,1)
C                 PORE VOLUMES OF CO2 INJECTED
          CUMDUM = QZ(1) * DTT/((VFRBLK+VRSBLK*NBY)*NBZ )
           IF (IWELL(JWELL).EQ.1) CUMI(1) = CUMI(1) + CUMDUM
           IF (IWELL(JWELL).EQ.2) CUMI(3) = CUMI(3) + CUMDUM
C
C
 700   TIME=TIME+DTT
       TWELL=TWELL+DTT
C
       RETURN
       END


       SUBROUTINE STPCHG (X,DT,DTDZ,DTDY)
C
C
C          THIS SUBR. CHANGES THE TIME STEP ISZE PER THE VALUE OF X.
C
          REAL X,DT,DTDZ,DTDY
C
          DT = X * DT
          DTDZ = X * DTDZ
          DTDY = X * DTDY
C
          RETURN
          END
```

```
      SUBROUTINE FLOCHG
C
C           THIS SUBROUTINE DETERMINES THE CORRECT WELL FLOW RATE
C                DEPENDING ON THE VALUES INPUT FOR NUMBER OF STEPS
C                PER INJECTION,SOAK, AND PRODUCTION CYCLES.
C
      PARAMETER (ISZ=52)
      COMMON / GRID /    DT,DZ,DY,DTDZ,DTDY,DTT,DTDZT,DTDYT,TIME,DTNEW,
     .                   DTSTBL,NBZ,NBY,NR,NBFIN,FRLTH,RSLTH,HFRAC,
     .                   DEPTH,AFRAC,ARES,VBLK(ISZ),VFRBLK,VRSBLK,VPORE
      COMMON / COUNT /   IT,ITMAX,ITCNT,ICNVRG,IBUBBL,IDPCHG,IOVERF,
     .                   IOVERR,ISTABL,IMOB,IMOB1,IPRNT,NOUNIT,MXCNVRG
      COMMON / MOLES /   CN(3,ISZ),CNG(3,ISZ),CNGC(ISZ),X(3,3,ISZ),
     .                   OILSOL,WATSOL,RATSOL,OILRAT,WATRAT,OIPMOL(3)
      COMMON / PRESS /   PRSAV(ISZ),PRSOLD(ISZ),PDSOAK,DPLIM,PRSI,
     .                   PRSMAX,PRSMIN
      COMMON / FLOW /    QZ(ISZ),QY(ISZ),QFLOW,QINJ1,QINJ2,QPROD,
     .                   IWELL(50),JWELL,TWELL,TLIMIN,TLIMPR,TLIMSK
      COMMON / XPERM /    PRMF,PRMR,PORF,PORR,AKR(3,ISZ),F(3,ISZ),
     .                   TMOB(ISZ),SWC,SOR,SOG,SGC,EKRO,EKRW,EKROG,
     .                   EKRG,EO,EW,EOG,EG,S(3,ISZ),SGCX(ISZ)
      COMMON / MISC /    C(3),TITLE(20),ILIMIT,IP(3),ISTEP(4),C22DUM,
     .                   C13DUM,C33DUM
C
C
C
C             HUFF N' PUFF WELL-BLOCK FLOW RATE CHANGE
C
C
         IFLO = IWELL(JWELL)
      IF (ILIMIT.EQ.1) THEN
C                              PRESSURE DEPENDENT WELL CONTROL
         PMAX=0
         PMIN=1.E30
         DO 10 K=2,NBFIN
            PMAX = AMAX1(PMAX,PRSAV(K))
 10         PMIN = AMIN1(PMIN,PRSAV(K))
         IF (IFLO.EQ.1 .OR. IFLO.EQ.2) THEN
C                                          INJECTION CHECK
            IF (PMAX.GE.PRSMAX) GO TO 50
            RETURN
         ELSE IF (IFLO.EQ.-1) THEN
C                                 PRODUCTION CHECK
            IF (PMIN.LE.PRSMIN) GO TO 50
            RETURN
         ELSE
C                            SOAK CHECK
            IF (PMAX-PMIN.LE.PDSOAK) GO TO 50
            RETURN
         END IF
```

```
      ELSE IF (ILIMIT.EQ.2) THEN
C                                   TIME DEPENDENT WELL CONTROL
         IF (IFLO.EQ.1 .OR. IFLO.EQ.2) THEN
            IF (TWELL.GE.TLIMIN) THEN
               TWELL=0
               GO TO 50
            ELSE
               RETURN
            END IF
         ELSE IF (IFLO.EQ.-1) THEN
            IF (TWELL.GE.TLIMPR) THEN
               TWELL=0
               GO TO 50
            ELSE
               RETURN
            END IF
         ELSE
            IF (TWELL.GE.TLIMSK) THEN
               TWELL=0
               GO TO 50
            ELSE
               RETURN
            END IF
         END IF
      END IF
C
 50    JWELL=JWELL+1
      IF (IWELL(JWELL).EQ.1) GO TO 100
      IF (IWELL(JWELL).EQ.2) GO TO 200
      IF (IWELL(JWELL).EQ.10) GO TO 300
      IF (IWELL(JWELL).EQ.-10) GO TO 350
      IF (IWELL(JWELL).EQ.-1) GO TO 400
C
C                 CHANGE TO CO2 INJECTION MODE
 100  QZ(1) = QINJ1
      X(3,3,1) = 0
      X(1,1,1) = 1.0
      F(1,1) = 1.0
      F(3,1) = 0
      GO TO 500
C
C                 CHANGE TO WATER INJECTION MODE
 200  QZ(1) = QINJ2
      X(1,1,1) = 0
      X(3,3,1) = 1.0
      F(3,1) = 1.0
      F(1,1) = 0
      GO TO 500
```

```
C
C                    CHANGE TO POST-INJECTION SOAK MODE
 300  QZ(1) = 1.E-20
      GO TO 500
C
C                     CHANGE TO POST-PRODUCTION SOAK MODE
 350  QZ(1) = -1.E-20
      GO TO 500
C
C                          CHANGE TO PRODUCTION MODE
 400  QZ(1) = QPROD
C
 500  ITCNT = 1
      RETURN
       END
```

```
      SUBROUTINE FLOFN1 (PRM,XVISO,XRHO,XX)
C
      PARAMETER (ISZ=52)
      COMMON / GRID /    DT,DZ,DY,DTDZ,DTDY,DTT,DTDZT,DTDYT,TIME,DTNEW,
     .                   DTSTBL,NBZ,NBY,NR,NBFIN,FRLTH,RSLTH,HFRAC,
     .                   DEPTH,AFRAC,ARES,VBLK(ISZ),VFRBLK,VRSBLK,VPORE
      COMMON / PRSMAT / ALPHA(ISZ),BETA(ISZ),GAMMA(ISZ),EPSLON(ISZ),
     .                   THETA(ISZ),DELTA2(ISZ),DELTA3(ISZ),A(ISZ),
     .                   W1Z(ISZ),W2Z(ISZ),W3Z(ISZ),W1Y(ISZ),W2Y(ISZ),
     .                   W3Y(ISZ)
      COMMON / VISC /    VISO(ISZ),VISG,VISW,AVIS1,AVIS2,BVIS,CVIS
      COMMON / XPERM /    PRMF,PRMR,PORF,PORR,AKR(3,ISZ),F(3,ISZ),
     .                   TMOB(ISZ),SWC,SOR,SOG,SGC,EKRO,EKRW,EKROG,
     .                   EKRG,EO,EW,EOG,EG,S(3,ISZ),SGCX(ISZ)
      DIMENSION RMOB(3,ISZ),PRM(3,ISZ),XVISO(ISZ),XRHO(3,ISZ),
     .          XX(3,3,ISZ)
C
C             CALCULATE FLOW TERMS FOR PRESSURE EQN.
C
      CR = PRMR * 6.8E-5
      CF = PRMF * 6.8E-5
C                                   (6.85E-5 IS FOR UNIT CONVERSION)
      CFG = CF / VISG
      CFW = CF / VISW
      CRG = CR / VISG
      CRW = CR / VISW
      KB = 2
      DO 1000 K=2,NBFIN
         IF (K.EQ.KB) THEN
            KB = KB + NR
C
C                              MOBILITY TERMS ALONG FRACTURE
            RMOB(1,K) = CFG * PRM(1,K)
            RMOB(2,K) = PRM(2,K) * CF / XVISO(K)
            RMOB(3,K) = CFW * PRM(3,K)
            W1Z(K) = XRHO(1,K) * RMOB(1,K) + XX(1,2,K) * XRHO(2,K) *
     .               RMOB(2,K) + XX(1,3,K) * XRHO(3,K) * RMOB(3,K)
            W2Z(K) = XX(2,2,K) * XRHO(2,K) * RMOB(2,K)
            W3Z(K) = XX(3,3,K) * XRHO(3,K) * RMOB(3,K)
         END IF


C
C                         FLOW TERMS PERPENDICULAR TO FRACTURE
         RMOB(1,K) = CRG * PRM(1,K)
         RMOB(2,K) = PRM(2,K) * CR / XVISO(K)
         RMOB(3,K) = CRW * PRM(3,K)
         W1Y(K) = XRHO(1,K) * RMOB(1,K) + XX(1,2,K) * XRHO(2,K) *
     .            RMOB(2,K) + XX(1,3,K) * XRHO(3,K) * RMOB(3,K)
         W2Y(K) = XX(2,2,K) * XRHO(2,K) * RMOB(2,K)
         W3Y(K) = XX(3,3,K) * XRHO(3,K) * RMOB(3,K)
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FLOFN2 (XCNG,XCN,XV,XX,XP,XCMP,XS,XSGC)
C
      PARAMETER (ISZ=52)
      COMMON / GRID /    DT,DZ,DY,DTDZ,DTDY,DTT,DTDZT,DTDYT,TIME,DTNEW,
     .                   DTSTBL,NBZ,NBY,NR,NBFIN,FRLTH,RSLTH,HFRAC,
     .                   DEPTH,AFRAC,ARES,VBLK(ISZ),VFRBLK,VRSBLK,VPORE
      COMMON / MOLES /   CN(3,ISZ),CNG(3,ISZ),CNGC(ISZ),X(3,3,ISZ),
     .                   OILSOL,WATSOL,RATSOL,OILRAT,WATRAT,OIPMOL(3)
      COMMON / PRSMAT /  ALPHA(ISZ),BETA(ISZ),GAMMA(ISZ),EPSLON(ISZ),
     .                   THETA(ISZ),DELTA2(ISZ),DELTA3(ISZ),A(ISZ),
     .                   W1Z(ISZ),W2Z(ISZ),W3Z(ISZ),W1Y(ISZ),W2Y(ISZ),
     .                   W3Y(ISZ)
      DIMENSION XCNG(3,ISZ),XCN(3,ISZ),XV(3,3,ISZ),XX(3,3,ISZ),
     .          XP(ISZ),XCMP(ISZ),XS(3,ISZ),XSGC(ISZ)
C
      DO 1000 K=2,NBFIN
C
C           CALCULATE DELTA COEFFICIENTS AND A(K)
C
         DUM3 = XCN(3,K) + XCNG(3,K)
         DUM2 = XCN(2,K) + XCNG(2,K)
         IF (XS(1,K)-XSGC(K).LE.1.E-4) THEN
C                                          IMMOBILE GAS
            DUM3 = DUM3 * XX(2,2,K)
            DUM2 = DUM2 * XX(3,3,K) * RATSOL
            DUMV = XV(1,2,K) - XV(1,3,K)
            DNUM = DUM3 + DUM2
            DENOM = XV(1,3,K) * DUM3 + XV(1,2,K) * DUM2
            DELTA2(K) = (XV(2,2,K) * DNUM + RATSOL * XCNG(3,K) *
     .                   DUMV)/DENOM
            DELTA3(K) = (XV(3,3,K) * DNUM - XCNG(2,K) * DUMV)/DENOM
            A(K) = XCMP(K) * DNUM / DENOM


         ELSE
C                                       MOBILE GAS
            DUMV2 = XV(1,1,K) - XV(1,2,K)
            DUMV3 = XV(1,1,K) - XV(1,3,K)
            SOL2  = OILSOL * XP(K)
            SOL3  = WATSOL * XP(K)
            SOL2M = 1.- SOL2
            SOL3M = 1.- SOL3
            DELTA2(K) = (XV(2,2,K) * SOL2M - DUMV2 * SOL2) /
     .                  (XV(1,1,K) * SOL2M)
            DELTA3(K) = (XV(3,3,K) * SOL3M - DUMV3 * SOL3) /
     .                  (XV(1,1,K) * SOL3M)
            CMPMOB = DUMV2 * DUM2 * OILSOL / SOL2M + DUMV3 * DUM3 *
     .               WATSOL / SOL3M
            A(K) = (XCMP(K) + CMPMOB) / XV(1,1,K)
         END IF
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GAUSS
C
      PARAMETER (ISZ=52)
      COMMON / GRID /    DT,DZ,DY,DTDZ,DTDY,DTT,DTDZT,DTDYT,TIME,DTNEW,
     .                   DTSTBL,NBZ,NBY,NR,NBFIN,FRLTH,RSLTH,HFRAC,
     .                   DEPTH,AFRAC,ARES,VBLK(ISZ),VFRBLK,VRSBLK,VPORE
      COMMON / PRSMAT / ALPHA(ISZ),BETA(ISZ),GAMMA(ISZ),EPSLON(ISZ),
     .                   THETA(ISZ),DELTA2(ISZ),DELTA3(ISZ),A(ISZ),
     .                   W1Z(ISZ),W2Z(ISZ),W3Z(ISZ),W1Y(ISZ),W2Y(ISZ),
     .                   W3Y(ISZ)
      COMMON / TMPVAL / PRM1(3,ISZ),PRMT(3,ISZ),XCN(3,ISZ),XPRSAV(ISZ),
     .                   RHOT(3,ISZ),XT(3,3,ISZ),FT(3,ISZ),TMOBT(ISZ)
C
C
C                     THIS SUBROUTINE WILL SOLVE THE PRESSURE MATRIX
C                        ARISING OUT OF THE QUASI-TWO DIMENSIONAL
C                        CHARACTERIZATION OF A FRACTURED RESERVOIR.
C
C
      REAL XALPHA(ISZ),XBETA(ISZ),XTHETA(ISZ),XGAMMA(ISZ),XEPSLN(ISZ)
C
      XTHETA(1) = 0
      XPRSAV(NBFIN+1) = 0
C
      DO 100 K=2,NBFIN
         XPRSAV(K) = 0
         XALPHA(K) = ALPHA(K)
         XBETA(K) = BETA(K)
         XGAMMA(K) = GAMMA(K)
         XTHETA(K) = THETA(K)
100       XEPSLN(K) = EPSLON(K)
```

```
C
      KB=2
      K=KB
C
C                                BEGIN FORWARD SUBSTITUTION
 120   XGAMMA(K) = XGAMMA(K) / XBETA(K)
 150   XEPSLN(K) = XEPSLN(K) / XBETA(K)
      XTHETA(K) = (XTHETA(K) - XALPHA(K) * XTHETA(K-1)) / XBETA(K)
      IF (K.GE.NBFIN) GO TO 400
      IF (K.GE.KB+NBY) GO TO 300
      K = K + 1
      XBETA(K) = XBETA(K) - XALPHA(K) * XEPSLN(K-1)
      XGAMMA(K) = -XALPHA(K) * XGAMMA(K-1) / XBETA(K)
      GO TO 150
C
 300   KB = KB + NR
      I = NR
 350  XBETA(KB) = XBETA(KB) - XALPHA(KB) * XGAMMA(KB-I)
      IF (I.LE.1) GO TO 360
      XTHETA(KB) = XTHETA(KB) - XALPHA(KB) * XTHETA(KB-I)
      XALPHA(KB) = -XALPHA(KB) * XEPSLN(KB-I)
      I = I - 1
      GO TO 350
C
 360  K = KB
      GO TO 120
C
C                            BEGIN BACKWARD SUBSTITUTION
 400   KB = NBZ * NR + 2
      K = NBFIN
C
 450   XPRSAV(K) = XTHETA(K) - XEPSLN(K) * XPRSAV(K+1) - XGAMMA(K) *
     1                                                     XPRSAV(KB)
      IF (K.LE.2) RETURN
      IF (KB-K.GE.NR) KB=KB-NR
      K = K - 1
      GO TO 450
      END
```

```
      SUBROUTINE MOBITR (*,*)
C
      PARAMETER (ISZ=52)
      COMMON / GRID /    DT,DZ,DY,DTDZ,DTDY,DTT,DTDZT,DTDYT,TIME,DTNEW,
     .                   DTSTBL,NBZ,NBY,NR,NBFIN,FRLTH,RSLTH,HFRAC,
     .                   DEPTH,AFRAC,ARES,VBLK(ISZ),VFRBLK,VRSBLK,VPORE
      COMMON / COUNT /   IT,ITMAX,ITCNT,ICNVRG,IBUBBL,IDPCHG,IOVERF,
     .                   IOVERR,ISTABL,IMOB,IMOB1,IPRNT,NOUNIT,MXCNVRG
      COMMON / TIMES /   TPHASE,TIMPEM,TDPCHG(100),TSTRT,TOVERF(100),
     .                   TOVERR(100),TBUBBL(100),XJUMP
      COMMON / MOLES /   CN(3,ISZ),CNG(3,ISZ),CNGC(ISZ),X(3,3,ISZ),
     .                   OILSOL,WATSOL,RATSOL,OILRAT,WATRAT,OIPMOL(3)
      COMMON / PRESS /   PRSAV(ISZ),PRSOLD(ISZ),PDSOAK,DPLIM,PRSI,
     .                   PRSMAX,PRSMIN
      COMMON / PRSMAT /  ALPHA(ISZ),BETA(ISZ),GAMMA(ISZ),EPSLON(ISZ),
     .                   THETA(ISZ),DELTA2(ISZ),DELTA3(ISZ),A(ISZ),
     .                   W1Z(ISZ),W2Z(ISZ),W3Z(ISZ),W1Y(ISZ),W2Y(ISZ),
     .                   W3Y(ISZ)
      COMMON / FLOW /    QZ(ISZ),QY(ISZ),QFLOW,QINJ1,QINJ2,QPROD,
     .                   IWELL(50),JWELL,TWELL,TLIMIN,TLIMPR,TLIMSK
      COMMON / XDENSE /   RHOMOL(3,ISZ),RHO(3,ISZ),VN(3,3,ISZ),
     .                   CMPEFF(ISZ),ZSLOPE,A22DEN,B22DEN,ASWELL,
     .                   A33DEN,B33DEN,A3SDEN,B3SDEN,V22ATM,WT(3)
      COMMON / VISC /    VISO(ISZ),VISG,VISW,AVIS1,AVIS2,BVIS,CVIS
      COMMON / XPERM /    PRMF,PRMR,PORF,PORR,AKR(3,ISZ),F(3,ISZ),
     .                   TMOB(ISZ),SWC,SOR,SOG,SGC,EKRO,EKRW,EKROG,
     .                   EKRG,EO,EW,EOG,EG,S(3,ISZ),SGCX(ISZ)
      COMMON / PRM /     IPERM,NSW,NSL,TSW(3,20),TSL(3,20),ITPT
      COMMON / TMPVAL /  PRM1(3,ISZ),PRMT(3,ISZ),XCN(3,ISZ),XPRSAV(ISZ),
     .                   RHOT(3,ISZ),XT(3,3,ISZ),FT(3,ISZ),TMOBT(ISZ)
      COMMON / UPDATE /  CUMI(3),CUMP(3),CMOLIN(3),CMOLPR(3),PROD(3)
      COMMON / MISC /    C(3),TITLE(20),ILIMIT,IP(3),ISTEP(4),C22DUM,
     .                   C13DUM,C33DUM
      DIMENSION XCNG(3,ISZ),RHOGMT(3,ISZ),VNT(3,3,ISZ),XVISO(ISZ),
     .          XSGC(ISZ),CMPEFT(ISZ),ST(3,ISZ),X10(ISZ)
      LOGICAL X1,X2,X3,X4,XX1,XX2,XX3
C
C
       IMOB = IMOB + 1
       IMOB1 = IMOB1 + 1
       IF (IMOB1.LE.1) THEN
       DO 40 K=2,NBFIN
         DO 40 J=1,3
 40         PRM1(J,K) = AKR(J,K)
       ELSE
       DO 45 K=2,NBFIN
          DO 45 J=1,3
 45          PRM1(J,K) = PRMT(J,K)
       END IF
```

```
C
C                              CALCULATE NEW PHASE COMPOSITIONS
      CALL EQULIB (CNGC,XCN,XPRSAV,XCNG,XT,X1O)
C*****************************************************************
C                 CHECK FOR APPEARANCE/DISAPPEARANCE OF GAS IN A BLOCK
C                        AND ADJUST TIME STEP FOR MINIMUM CROSS-OVER
C                        AT BUBBLE POINT
      KMAX = 0
      TCHGMX = 1.E6
      DO 80 K=2,NBFIN
        C0 = OILSOL * PRSOLD(K)
        C1 = X(1,2,K) - C0
        C2 = X1O(K) - OILSOL * XPRSAV(K)
        X1 = ABS(C1) .LE. 1.E-05
        X2 = C2 .LE. -XJUMP
        X3 = C1 .LE. -1.E-5
        X4 = C2 .GE. XJUMP
        XX1 = X1 .AND. X2
        XX2 = X3 .AND. X4
        XX3 = XX1 .OR. XX2
        IF (XX3) THEN
          IF (XX1) THEN
C                              SATURATED TO UNSATURATED CASE
            TCHG = ( OILSOL * (XPRSAV(K)-PRSOLD(K)) )  /
     .                 (X1O(K) - X(1,2,K))
            TCHGMX = AMIN1 (TCHGMX,ABS(TCHG))
            IF (ABS(TCHGMX-TCHG).LE.1.E-6) KMAX=K
          ELSE IF (XX2) THEN
C                              UNSATURATED TO SATURATED CASE
            DUM = (X1O(K)-X(1,2,K))/(XPRSAV(K)-PRSOLD(K))
            PX = (X1O(K)-DUM*XPRSAV(K))/(OILSOL-DUM)
            TCHG = (PX-PRSOLD(K))/(XPRSAV(K)-PRSOLD(K))
C           TCHG = (OILSOL * XPRSAV(K) - X(1,2,K)) /
C    .                                 (X1O(K) - X(1,2,K))
            TCHGMX = AMIN1 (TCHGMX,ABS(TCHG))
            IF (ABS(TCHGMX-TCHG).LE.1.E-6) KMAX=K
          END IF
         END IF
 80     CONTINUE
       IF (KMAX.GT.0) THEN
            CALL STPCHG (TCHGMX,DTT,DTDZT,DTDYT)
           IBUBBL = IBUBBL + 1
           TBUBBL(IBUBBL) = DTT
           RETURN 1
       END IF
```

```
C                                CALCULATE PHASE DENSITIES
      CALL DENSE (XPRSAV,XT,RHOGMT,RHOT,VNT)
C
C                                CALCULATE OIL VISCOSITY
      DO 60 K=2,NBFIN
 60   XVISO(K) = VISOIL (XPRSAV(K),XT(1,2,K))
C
C                                CALCULATE AVERAGE COMPRESSIBILITY
      CALL CMPRES (XCNG,XCN,XPRSAV,CMPEFT,VNT)
C
C                                CALCULATE PHASE SATURATIONS
      CALL SCALC (XCNG,XCN,VNT,XSGC,ST,CNGC)
C
C                               CALCULATE RELATIVE PERMEABILITIES
      CALL PERM (ST,XVISO,XSGC,PRMT,FT,TMOBT)
C
C                            CALCULATE FLOW COEFFICIENTS FOR PRES. EQN.
      CALL FLOFN1 (PRMT,XVISO,RHOT,XT)
      CALL FLOFN2 (XCNG,XCN,VNT,XT,XPRSAV,CMPEFT,ST,XSGC)
C
C
      QFLOW = QFLOFN (QZ(1),DTT,RHOT(1,1),RHOT(3,1),
     1                IWELL(JWELL),RHOT(1,2),RHOT(2,2),RHOT(3,2),
     2                FT(1,2),FT(2,2),FT(3,2),XT(2,2,2),XT(3,3,2),
     3                DELTA2(2),DELTA3(2),XT(1,2,2),XT(1,3,2))
C
      IF (IMOB1.LE.1) RETURN 2
C                            CHECK PERMEABILITY CONVERGENCE
      DO 140 K=2,NBFIN
         DO 140 J=1,3
            PRMDIF = PRM1(J,K) - PRMT(J,K)
            IF (PRMDIF.EQ.0..OR. PRM1(J,K).EQ.0.) GO TO 140
            PERDIF = ABS(PRMDIF)/PRM1(J,K)
            IF (PERDIF.GT..01) GO TO 150
 140        CONTINUE
      RETURN
C                        NON-CONVERGENCE - REPEAT ITERATION
 150  IF (ICNVRG.LE. MXCNVRG) THEN
         ICNVRG = ICNVRG + 1
C        PRINT 200, K,J,AKR(J,K),PRMT(J,K),PRM1(J,K),PERDIF,ICNVRG
 200     FORMAT (20X,'K, J, AKR, PRMT, PRM1, PERDIF, ICNVRG : ',2I3,
     .            4E12.4,I3)
         RETURN 2
      ELSE
         RETURN
      END IF
      END
```

```
      FUNCTION VISOIL (P,X12)
C
      PARAMETER (ISZ=52)
      COMMON / VISC /   VISO(ISZ),VISG,VISW,AVIS1,AVIS2,BVIS,CVIS
C
C
        VISOIL = 10 ** (AVIS1 * P + BVIS * X12 + CVIS)
           RETURN
           END
C*********************************************************************
      FUNCTION QFLOFN (Q,DT,RO11,RO31,IW,RO12,RO22,RO32,F1,F2,F3,X22,
     1                 X33,DEL2,DEL3,X12,X13)
C
        QDUM = 1. + DEL3
        IF (IW.EQ.1) QFLOFN = Q * DT * RO11
        IF (IW.EQ.2) QFLOFN = Q * DT * RO31 * QDUM
        IF (IABS(IW).EQ.10) QFLOFN = 0
        IF (IW.EQ.-1)    QFLOFN = Q * DT * (F1 * RO12 + F2 * RO22 *
     1          (X12 + DEL2 * X22) + F3 * RO32 * (X13 + DEL3 * X33))
C
       RETURN
       END
```


★U.S. GOVERNMENT PRINTING OFFICE: 1983- 769-029/73