FOSSIL ENERGY

DOE/BC/15109-1
(OSTI ID: 761976)

DEVELOPMENT OF AN IMPROVED SIMULATOR FOR CHEMICAL AND MICROBIAL IOR METHODS

Annual Report
September 30, 1998—September 30, 1999

By
Gary A. Pope
Kamy Sepehrnoori
Mojdeh Delshad

Date Published: September 2000

Work Performed Under Contract No. DE-AC26-98BC15109

The University of Texas
Austin, Texas

National Petroleum Technology Office
U.S. DEPARTMENT OF ENERGY
Tulsa, Oklahoma

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, expressed 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.

DOE/BC/15109-1
Distribution Category UC-122

Development of Improved Simulator for Chemical and Microbial IOR Methods

By
Gary A. Pope
Kamy Sepehrnoori
Mojdeh Delshad

September 2000

Work Performed Under Contract No DE-AC26-98BC15109

Prepared for
U.S. Department of Energy
Assistant Secretary for Fossil Energy

Purna Halder, Project Manager
National Petroleum Technology Office
P.O. Box 3628
Tulsa, OK 74101

Prepared by
The University of Texas
Center for Petroleum and Geosystems Engineering
2600 Speedway
Austin, TX 78712

TABLE OF CONTENTS

DEVELOPMENT OF AN IMPROVED SIMULATOR FOR CHEMICAL AND MICROBIAL IOR METHODS

EXECUTIVE SUMMARY

The objective of this research is to extend the capability of an existing simulator (UTCHEM) to improved oil recovery methods that use surfactants, polymers, gels, alkaline chemicals, microorganisms and foam as well as various combinations of these in both conventional and naturally fractured oil reservoirs. Task 1 is the addition of a dual-porosity model for chemical improved oil recovery processes in naturally fractured oil reservoirs. Task 2 is the addition of a foam model. Task 3 addresses several numerical and coding enhancements that will greatly improve the versatility and performance of UTCHEM. Task 4 is the enhancements of physical property models. All of these tasks are on schedule and progressing rapidly. The dual porosity, MEOR and foam models have been completely formulated and partially implemented and tested. Dynamic memory management has been implemented and will make UTCHEM easier to use since the grid size can now be changed without recompiling the code. Relative permeability and capillary pressure options have been expanded. UTCHEM can be run efficiently on Pentium II PCs and work has already started on developing user friendly input and output processing on PCs.

TASK 1: DUAL POROSITY MODEL

Many oil reservoirs in the United States are naturally fractured, and some of the larger ones like Spraberry contains billions of barrels of remaining oil, but relatively little research has been done on the use of advanced oil recovery methods. In addition, very little success has been achieved in increasing the oil production from these complex reservoirs. The use of chemical methods of improved oil recovery from naturally fractured reservoirs has been particularly neglected. Some laboratory experiments have been done to investigate the use of surfactants in fractured chalk (Al-Lawati and Saleh, 1996; Austad, 1994; Keijzer and De Vries, 1990; Schechter *et al.*, 1991). However, the results of these studies are hard to interpret and to apply to field-scale predictions without a model that takes into account both the fluid flow and chemical phenomena in both fractures and rock matrix. The most efficient approach to modeling naturally fractured reservoirs appears to be the dual-porosity model, first proposed by Barenblatt *et al.* (1960) and introduced to the petroleum industry by Warren and Root (1963). The dual-porosity model assumes that two equivalent continuous porous media are superimposed: one for fractures and another for the intervening rock matrix. A mass balance for each of the media yields two continuity equations that are connected by so-called transfer functions that characterize flow between matrix blocks and fractures. Since Kazemi *et al.* (1976) introduced the first multiphase dual-porosity model, almost all subsequent dual-porosity models have been based on modifications of the transfer functions. These transfer functions are what distinguish various dual porosity models in the literature.

We have formulated for multiphase flow, including complex chemical phenomena currently modeled with UTCHEM for both fracture and rock matrix, e.g., the effects of reduced interfacial tension on phase trapping, surfactant adsorption, and so forth. The dual-porosity model handles the flow of tracers in both rock systems as well.

Formulation

Assumptions and formulation of the equations used in UTCHEM are covered in detail in Datta Gupta *et al.* (1986), Saad (1989), and Delshad *et al.* (1996). For consistency, the same assumptions and formulation are used for the mass-conservation equation and the pressure equation in the matrix. The major assumptions are as follows:

1. Slightly compressible fluid and rock properties.
2. Darcy's law applies.
3. Dispersion follows a generalization of Fick's law to multiphase flow in porous media.
4. Ideal mixing.
5. The fluid phase behavior is independent of the reservoir pressure.

The mass conservation equation used in UTCHEM is

$$\frac{\partial}{\partial t}\left(\phi\,\tilde{C}_k\ \rho_k\right)+\bar{\nabla}\cdot\left[\sum_{1=l}^{n_p}\rho_k\left(C_{k\ell}\,\bar{u}_\ell-\phi\,S_\ell\,\bar{\bar{K}}_{\kappa\ell}\cdot\bar{\nabla}C_{\kappa\ell}\right)\right]=R_k \qquad (1)$$

where

$\tilde{C}_\kappa$ = Overall volumetric concentration of component κ, L^3/L^3 PV

$C_{\kappa\ell}$ = Concentration of component κ in phase ℓ, L^3/L^3

$\bar{\bar{K}}_{\kappa\ell}$ = Dispersion coefficient tensor of component κ in phase ℓ, L^2/t

R_κ = Total source/sink for component κ, $m/L^3 t$

$\bar{u}_\ell$ = Volumetric flux of phase ℓ, L / t

ρ_κ = Density of component phase κ, m/L^3

The pressure equation is derived from the mass-conservation equation and is

$$\begin{aligned}&\phi_R\,c_t\frac{\partial p_1}{\partial t}-\bar{\nabla}\cdot\bar{\bar{k}}\cdot\lambda_{rTc}\bar{\nabla}p_1=\\&-\bar{\nabla}\cdot\sum_{\ell=1}^{n_p}\bar{\bar{k}}\cdot\lambda_{r\ell c}\bar{\nabla}D+\bar{\nabla}\cdot\sum_{\ell=2}^{n_p}\bar{\bar{k}}\cdot\lambda_{r\ell c}\bar{\nabla}p_{c\ell 1}+\sum_{k=1}^{n_{cv}}Q_k\end{aligned} \qquad (2)$$

where

c_t = total system compressibility, Lt^2 / m

D = depth, L

$\vec{\vec{k}}$ = permeability tensor, L^2

p_ℓ = pressure of phase ℓ, Lt^2 / m

$p_{c\ell 1}$ = capillary pressure between the given phase ℓ and phase 1, Lt^2 / m

Q_k = source/sink flow for component k per bulk volume, $L^3 / L^3 t$

$\lambda_{r\ell c}$ = relative mobility, $m / L\, t$

λ_{rTc} = total relative mobility, $m / L\, t$

n_p = number of phases

n_{cv} = number of volume - occupying components

A detailed description of the variables used in both the mass-conservation and pressure equations is found in Datta Gupta *et al.* (1986), Saad (1989), and Delshad *et al.* (1996). Equations 1 and 2 may be extended to account for dual-porosity behavior by adding sink/source transfer terms to represent the matrix-fracture transfer. Another set of equations similar to Eqs. 1 and 2 is used to calculate the sink/source transfer terms. No wells are allowed to be completed in the matrix blocks at this time. We have two sets of equations: one set for the fracture system and another for the matrix block. The matrix-block set of equations is used to calculate the sink/source transfer terms used in the fracture-system set of equations. In both sets the pressure equation is solved implicitly while the mass-conservation equations are solved explicitly afterwards. The solution method decouples the matrix-pressure equation from the fracture pressure equation while solving the matrix mass-conservation equations explicitly. The decoupling procedure is discussed below. At each time level, the matrix pressure equation is solved implicitly to calculate the sink/source transfer terms. The sink/source transfer terms are then added to the fracture pressure equation, which in turn is solved implicitly. Next, the matrix mass-conservation equations are solved explicitly to calculate sink/source transfer terms that are added to the fracture mass-conservation equations that are solved explicitly as well. At the end of the timestep, both fracture and matrix variables are updated and a new timestep begins.

Discretized matrix equations

The spatial domain will be divided into nested grids in the horizontal direction and stacked grids in the vertical direction utilizing a modified MINC style (Wu and Pruess, 1986) as shown in Fig 1. The advantage of this approach is it reduces the problem to one dimension in the horizontal direction; the whole problem is reduced from three-dimensional problem to a two-dimensional one. Keeping this in mind, Eq. 1 can be expanded for each component as

$$\begin{aligned}&\frac{\partial}{\partial t}\left\{\phi\,\tilde{C}_\kappa\left[1+c_\kappa^o\left(p_\ell-p_R\right)\right]\right\}\\&=-\frac{\partial}{\partial h}\left\{\sum_{\ell=1}^{n_p}\left[1+c_\kappa^o\left(p_\ell-p_R\right)\right]\left(C_{\kappa\ell}u_{h\ell}-\phi S_\ell K_{hh\kappa\ell}\frac{\partial C_{\kappa\ell}}{\partial h}\right)\right\}\\&-\frac{\partial}{\partial z}\left\{\sum_{\ell=1}^{n_p}\left[1+c_\kappa^o\left(p_\ell-p_R\right)\right]\left(C_{\kappa\ell}u_{z\ell}-\phi S_\ell K_{zz\kappa\ell}\frac{\partial C_{\kappa\ell}}{\partial z}\right)\right\}\\&+Q_\kappa\end{aligned} \qquad (3)$$

where the overall concentration of each component κ is given by

$$\tilde{C}_\kappa=\left(1-\sum_{\kappa=1}^{n_c}\hat{C}_\kappa\right)\sum_{\ell=1}^{n_p}C_{\kappa\ell}S_\ell+\hat{C}_\kappa \qquad (4)$$

where

$\hat{C}_\kappa$ = adsorbed concentration of component κ, L^3/L^3 PV

c_κ^o = compressibility of component κ, Lt^2/m

h = horizontal direction

$\vec{\vec{K}}$ = dispersion tensor, L^2/t

S_ℓ = satuaration of phase ℓ, L^3/L^3 PV

u = Darcy's flux, L/t

z = vertical direction

The porosity in the accumulation term is approximated as

$$\phi=\phi_R\left[1+c_r\left(p_l-p_R\right)\right] \qquad (5)$$

where ϕ_R is the porosity at a specific pressure p_R, p_l is the aqueous phase pressure, and c_r is the rock compressibility at p_R. Substituting the expression for rock compressibility and neglecting the terms containing the product $c_k{}^o c_r$ because they are small, we have

$$\begin{aligned} \mathrm{LHS} &= \phi_R\left(c_r + c_\kappa^o\right)\frac{\partial p_l}{\partial t}\tilde{C}_\kappa^n \\ &\quad + \phi_R\left[1 + c_\kappa^o(p_l - p_R)^{n+1} + c_r(p_l - p_R)^{n+1}\right]\frac{\partial \tilde{C}_\kappa}{\partial t} \end{aligned} \tag{6}$$

Keeping in mind that a modified MINC-style subgridding scheme is used, the spatial derivatives are evaluated as follows:

$$\begin{aligned} \mathrm{RHS} = -\frac{2\Delta z_k}{V_{bik}}\sum_{\ell=1}^{n_p}\Big\langle & u_{x\ell,i+1/2k}\,y_{ik}\left[1 + c_\kappa^o(p_l - p_R)\right]_{ik}C_{\kappa\ell,i+1/2k} \\ & + u_{y\ell,i+1/2k}\,x_i\left[1 + c_\kappa^o(p_l - p_R)\right]_{ik}C_{\kappa\ell,i+1/2k} \\ & - u_{x\ell,i-1/2k}\,y_{i-1}\left[1 + c_\kappa^o(p_l - p_R)\right]_{i-1k}C_{\kappa\ell,i-1/2k} \\ & - u_{y\ell,i-1/2k}\,x_{i-1}\left[1 + c_\kappa^o(p_l - p_R)\right]_{i-1k}C_{\kappa\ell,i-1/2k} \\ & - K_{xx}\left\{\phi_R S_\ell\left[1 + c_\kappa^o(p_l - p_R)\right]_{ik}y_i\frac{C_{\kappa\ell,i+1k} - C_{\kappa\ell,ik}}{x_{nod,i+1} - x_{nod,i}}\right. \\ & \left. - \phi_R S_\ell\left[1 + c_\kappa^o(p_l - p_R)\right]_{i-1k}y_{i-1}\frac{C_{\kappa\ell,ik} - C_{\kappa\ell,i-1k}}{x_{nod,i} - x_{nod,i-1}}\right\} \\ & - K_{yy}\left\{\phi_R S_\ell\left[1 + c_\kappa^o(p_l - p_R)\right]_{ik}x_i\frac{C_{\kappa\ell,i+1k} - C_{\kappa\ell,ik}}{y_{nod,i+1} - y_{nod,i}}\right. \\ & \left. - \phi_R S_\ell\left[1 + c_\kappa^o(p_l - p_R)\right]_{i-1k}x_{i-1}\frac{C_{\kappa\ell,ik} - C_{\kappa\ell,i-1k}}{y_{nod,i} - y_{nod,i-1}}\right\}\Big\rangle \\ - \frac{1}{\Delta z_k}\sum_{\ell=1}^{n_p}\Big\langle & u_{z\ell,ik+1/2}\left[1 + c_\kappa^o(p_l - p_R)\right]_{ik}C_{\kappa\ell,ik+1/2} \\ & - u_{z\ell,ik-1/2}\left[1 + c_\kappa^o(p_l - p_R)\right]_{ik-1}C_{\kappa\ell,ik-1/2} \\ & - 2K_{zz}\left\{\phi_R S_\ell\left[1 + c_\kappa^o(p_l - p_R)\right]_{ik}\frac{C_{\kappa\ell,ik+1} - C_{\kappa\ell,ik}}{\Delta z_{k+1} + \Delta z_k}\right. \\ & \left. - \phi_R S_\ell\left[1 + c_\kappa^o(p_l - p_R)\right]_{ik-1}\frac{C_{\kappa\ell,ik} - C_{\kappa\ell,ik-1}}{\Delta z_k + \Delta z_{k-1}}\right\}\Big\rangle \end{aligned} \tag{7}$$

In Eq. 7 the convection terms are evaluated using one point upstream weighting, shown in Eq. 7 for the case when the potential $\Phi_{i-1} > \Phi_i > \Phi_{i+1}$. Physical dispersion is modeled in the matrix blocks using a diagonal dispersion tensor. The elements of this tensor are given by

$$K_{xxk\ell} = \frac{D_{\kappa\ell}}{\tau} + \frac{\alpha_{L\ell} u_{x\ell}^2}{\phi S_\ell |u_\ell|} + \frac{\alpha_{T\ell} u_{y\ell}^2}{\phi S_\ell |u_\ell|} + \frac{\alpha_{T\ell} u_{z\ell}^2}{\phi S_\ell |u_\ell|} \qquad (8)$$

$$K_{yyk\ell} = \frac{D_{\kappa\ell}}{\tau} + \frac{\alpha_{L\ell} u_{y\ell}^2}{\phi S_\ell |u_\ell|} + \frac{\alpha_{T\ell} u_{x\ell}^2}{\phi S_\ell |u_\ell|} + \frac{\alpha_{T\ell} u_{z\ell}^2}{\phi S_\ell |u_\ell|} \qquad (9)$$

$$K_{zzk\ell} = \frac{D_{\kappa\ell}}{\tau} + \frac{\alpha_{L\ell} u_{z\ell}^2}{\phi S_\ell |u_\ell|} + \frac{\alpha_{T\ell} u_{x\ell}^2}{\phi S_\ell |u_\ell|} + \frac{\alpha_{T\ell} u_{y\ell}^2}{\phi S_\ell |u_\ell|} \qquad (10)$$

where

$$|u_\ell| = \sqrt{u_{x\ell}^2 + u_{y\ell}^2 + u_{z\ell}^2} \qquad (11)$$

and

$D_{\kappa\ell}$ = diffusion coefficient of component κ in in phase ℓ, L^2 / t

α_L, α_T = longitudinal and transverse dispersivity, L

The fluxes $u_{x\ell}$, $u_{y\ell}$ and $u_{z\ell}$ are modeled through the use of Darcy's law for multiphase flow through permeable media, which is given by

$$\vec{u}_\ell = -\bar{\bar{k}}\lambda_{r\ell}\left(\vec{\nabla} p_\ell - \gamma_\ell \vec{\nabla} D\right) \qquad (12)$$

where

$\lambda_{r\ell}$ = relative mobility of phase ℓ, L^2 / t

γ_ℓ = specific weight of phase ℓ, $m / L^2 t$

The pressure equation, Eq. 2, can be rewritten as

$$\phi_R\, c_t \frac{\partial p_1}{\partial t} - \vec{\nabla}\cdot \sum_{\ell=1}^{n_p} \bar{\bar{k}} \cdot \lambda_{r\ell c} \vec{\nabla} p_1 = -\vec{\nabla}\cdot \sum_{\ell=1}^{n_p} \bar{\bar{k}} \cdot \lambda_{r\ell c} \vec{\nabla} D + \vec{\nabla}\cdot \sum_{\ell=2}^{n_p} \bar{\bar{k}} \cdot \lambda_{r\ell c} \vec{\nabla} p_{c\ell 1} \quad (13)$$

Note that the sink/source term has been removed, since no wells are completed in the matrix blocks in this formulation. The finite-difference form of the left side of Eq. 13, using a MINC style approach, can be written as

$$\begin{aligned}
\mathrm{LHS} = {} & \phi_R c_t \frac{p_{1,ik}^{n+1} - p_{1,ik}^{n}}{\Delta t} \\
& - \frac{2\Delta z_k}{V_{b,ik}} \sum_{\ell=1}^{n_p} \Bigg[k_x \Bigg(\lambda_{r\ell c,i+1/2k} y_i \frac{p_{1,i+1k}^{n+1} - p_{1,ik}^{n+1}}{x_{nod,i+1} - x_{nod,i}} \\
& - \lambda_{r\ell c,i-1/2k} y_{i-1} \frac{p_{1,ik}^{n+1} - p_{1,i-1k}^{n+1}}{x_{nod,i} - x_{nod,i-1}} \Bigg) \\
& + k_y \Bigg(\lambda_{r\ell c,i+1/2k} x_i \frac{p_{1,i+1k}^{n+1} - p_{1,ik}^{n+1}}{y_{nod,i+1} - y_{nod,i}} - \lambda_{r\ell c,i-1/2k} x_{i-1} \frac{p_{1,ik}^{n+1} - p_{1,i-1k}^{n+1}}{y_{nod,i} - y_{nod,i-1}} \Bigg) \Bigg] \\
& - \frac{2}{\Delta z_k} \sum_{\ell=1}^{n_p} k_z \Bigg(\lambda_{r\ell c,ik+1/2} \frac{p_{1,ik+1}^{n+1} - p_{1,ik}^{n+1}}{\Delta z_{k+1} + \Delta z_k} - \lambda_{r\ell c,ik-1/2} \frac{p_{1,ik}^{n+1} - p_{1,ik-1}^{n+1}}{\Delta z_k + \Delta z_{k-1}} \Bigg) \qquad (14)
\end{aligned}$$

The right side of Eq. 13 can be separated into a gravity term and a capillary-pressure term. The gravity term in the right side of Eq. 13 can be expanded as

$$
\begin{aligned}
G = -\frac{2\Delta z_k}{V_{b,ik}} \sum_{\ell=1}^{n_p} \Bigg[k_x \Bigg(& \lambda_{r\ell c,i+1/2k} \gamma_{\ell,i+1/2k} y_i \frac{D_{i+1} - D_i}{x_{nod,i+1} - x_{nod,i}} \\
& - \lambda_{r\ell c,i-1/2k} \gamma_{\ell,i-1/2k} y_{i-1} \frac{D_i - D_{i-1}}{x_{nod,i} - x_{nod,i-1}} \\
& + k_y \Bigg(\lambda_{r\ell c,i+1/2k} \gamma_{\ell,i+1/2k} x_i \frac{D_{i+1} - D_i}{y_{nod,i+1} - y_{nod,i}} \\
& - \lambda_{r\ell c,i-1/2k} \gamma_{\ell,i-1/2k} x_{i-1} \frac{D_i - D_{i-1}}{y_{nod,i} - y_{nod,i-1}} \Bigg) \Bigg] \\
& - \frac{2}{\Delta z_k} \sum_{\ell=1}^{n_p} k_z \Bigg(\lambda_{r\ell c,ik+1/2} \gamma_{\ell,ik+1/2} \frac{D_{k+1} - D_k}{\Delta z_{k+1} + \Delta z_k} \\
& - \lambda_{r\ell c,ik-1/2} \gamma_{\ell,ik-1/2} \frac{D_k - D_{k-1}}{\Delta z_k + \Delta z_{k-1}} \Bigg)
\end{aligned}
\tag{15}
$$

Equation 15 can be simplified by realizing that the matrix blocks are modeled as horizontal matrix blocks. Rewriting Eq. 15, we obtain

$$
G = -\frac{1}{\Delta z_k} \sum_{\ell=1}^{n_p} k_z \left(\lambda_{r\ell c,ik+1/2} \gamma_{\ell,ik+1/2} - \lambda_{r\ell c,ik-1/2} \gamma_{\ell,ik-1/2} \right)
\tag{16}
$$

Similarly, the capillary-pressure term in the right side of Eq. 13 can be expanded as

$$
PC = \frac{2\Delta z_k}{V_{b,ik}} \sum_{\ell=2}^{n_p} \Bigg[k_x \Bigg(\lambda_{r\ell c,i+1/2} y_i \frac{p_{c\ell 1,i+1k} - p_{c\ell 1,ik}}{x_{nod,i+1} - x_{nod,i}}
$$

$$
\begin{aligned}
&\left. -\lambda_{r\ell c,i-1/2k} y_{i-1} \frac{p_{c\ell 1,ik} - p_{c\ell 1,i-1k}}{x_{nod,i} - x_{nod,i-1}} \right) \\
&+ k_y \left(\lambda_{r\ell c,i+1/2k} x_i \frac{p_{c\ell 1,i+1k} - p_{c\ell 1,ik}}{y_{nod,i+1} - y_{nod,i}} \right. \\
&\left. \left. -\lambda_{r\ell c,i-1/2k} x_{i-1} \frac{p_{c\ell 1,ik} - p_{c\ell 1,i-1k}}{y_{nod,i} - y_{nod,i-1}} \right) \right] \\
&+ \frac{2}{\Delta z_k} \sum_{\ell=2}^{n_p} k_z \left(\lambda_{r\ell c,ik+1/2} \frac{p_{c\ell 1,ik+1} - p_{c\ell 1,ik}}{\Delta z_{k+1} + \Delta z_k} \right. \\
&\left. -\lambda_{r\ell c,ik-1/2} \frac{p_{c\ell 1,ik} - p_{c\ell 1,ik-1}}{\Delta z_k + \Delta z_{k-1}} \right)
\end{aligned}
\qquad (17)
$$

$\lambda_{r\ell c}$ in the above equations is evaluated using one-point upstream weighing and is given by

$$
\lambda_{r\ell c} = \lambda_{r\ell} \sum_{\kappa=1}^{n_c} \left(1 + c_\kappa^o \Delta p\right) C_{\kappa\ell} \qquad (18)
$$

For the case when the potential $\Phi_{i-1} > \Phi_i > \Phi_{i+1}$, $\lambda_{r\ell c,i-1/2}$ is evaluated at *i*-1 and $\lambda_{r\ell c,i+1/2}$ is evaluated at *I*. $\lambda_{r\ell}$ in Eq. 18 is given by

$$
\lambda_{r\ell} = \frac{k_{r\ell}}{\mu_\ell} \qquad (19)
$$

Decoupled equations

The matrix-block pressure equation will be decoupled from the fracture pressure equation to minimize coding effort (Chen, 1993). The transfer functions added to the fracture pressure equation have the form

$$
\begin{aligned}
\tau_{mf} = {} & N_m \sum_{\ell=1}^{n_p} \sum_{i=1}^{N_{subh}} k_z \frac{2V_{b,iN_{subv}} \lambda_{r\ell c,iN_{subv}+1/2}}{\Delta z_{N_{subv}}} \\
& \left[\frac{p_{1m,iN_{subv}}^{n+1} - p_{1f}^{n+1} - \gamma_{\ell,iN_{subv}+1/2}\left(D_{iN_{subv}} - D_f\right)}{\Delta z_{N_{subv}}} \right] \\
& + N_m \sum_{\ell=1}^{n_p} \sum_{i=1}^{N_{subh}} k_z \frac{2V_{b,i1}\lambda_{r\ell c,i1/2}}{\Delta z_1} \left[\frac{p_{1m,i1}^{n+1} - p_{1f}^{n+1} - \gamma_{\ell,i1/2}\left(D_{i1} - D_f\right)}{\Delta z_1} \right] \\
& + N_m \sum_{\ell=2}^{n_p} \sum_{i=1}^{N_{subh}} k_z \frac{2V_{b,iN_{subv}} \lambda_{r\ell c,iN_{subv}+1/2}}{\Delta z_{N_{subv}}} \left[\frac{p_{c\ell 1,iN_{subv}} - p_{c\ell 1,f}}{\Delta z_{N_{subv}}} \right] \\
& + N_m \sum_{\ell=2}^{n_p} \sum_{i=1}^{N_{subh}} k_z \frac{2V_{b,i1}\lambda_{r\ell c,i1/2}}{\Delta z_1} \left[\frac{p_{c\ell 1,i1} - p_{c\ell 1,f}}{\Delta z_1} \right] \\
& + N_m \sum_{\ell=1}^{n_p} \sum_{k=1}^{N_{subv}} 4\Delta z_k \lambda_{r\ell c,N_{subh}+1/2k} \Big[p_{1m,N_{subh}k}^{n+1} - p_{1f}^{n+1} \\
& - \gamma_{\ell,N_{subh}+1/2,k}\left(D_{N_{subh}k} - D_f\right) \Big] \left(\frac{k_x y_{N_{subh}}}{x_{N_{subh}} - 2x_{nod,N_{subh}}} + \frac{k_y x_{N_{subh}}}{y_{N_{subh}} - 2y_{nod,N_{subh}}} \right) \\
& + N_m \sum_{\ell=2}^{n_p} \sum_{k=1}^{N_{subv}} 4\Delta z_k \lambda_{r\ell c,N_{subh}+1/2k} \left(p_{c\ell 1,N_{subh}k} - p_{c\ell 1,f} \right) \\
& \left(\frac{k_x y_{N_{subh}}}{x_{N_{subh}} - 2x_{nod,N_{subh}}} + \frac{k_y x_{N_{subh}}}{y_{N_{subh}} - 2y_{nod,N_{subh}}} \right)
\end{aligned}
\qquad (20)
$$

where

N_m = number of matrix blocks per gridblock

N_{subh} = number of lateral matrix subgrids

N_{subv} = number of vertical matrix subgrids

The decoupling method substitutes the unknown matrix pressure in Eq. 20 with a function that is dependent on the fracture pressure at the next time level. The decoupling procedure is provided next.

If we solve the matrix pressure equation, Eq. 13, with boundary condition p_{lf}^{n}, the solution would represent how the matrix pressure changes if the boundary condition were kept constant. Let this solution be represented by

$$\phi_R\, c_t \frac{\partial \overline{p}_{1m}}{\partial t} - \vec{\nabla} \cdot \sum_{\ell=1}^{n_p} \vec{\vec{k}} \cdot \lambda_{r\ell c} \vec{\nabla} \overline{p}_{1m} = -\vec{\nabla} \cdot \sum_{\ell=1}^{n_p} \vec{\vec{k}} \cdot \lambda_{r\ell c} \vec{\nabla} D + \vec{\nabla} \cdot \sum_{\ell=2}^{n_p} \vec{\vec{k}} \cdot \lambda_{r\ell c} \vec{\nabla} p_{c\ell 1} \tag{21}$$

Now let

$$\tilde{p}_{1m}^{n+1} = \frac{p_{1m}^{n+1} - \overline{p}_{1m}^{n+1}}{p_{1f}^{n+1} - p_{1f}^{n}} \tag{22}$$

If we substitute Eq. 22 in the original pressure equation, Eq. 13, we obtain

$$\phi_R\, c_t \frac{\partial \tilde{p}_{1m}}{\partial t} - \vec{\nabla} \cdot \sum_{l=1}^{n_p} \vec{\vec{k}} \cdot \lambda_{rlc} \vec{\nabla} \tilde{p}_{1m} = 0 \tag{23}$$

The appropriate boundary condition for Eq. 23 is $\tilde{p}_1 = 1$ by evaluating Eq. 22 at the boundary. Note that neither Eq. 21 nor Eq. 23 have any fracture unknowns. Furthermore, p_1^{n+1} can easily be evaluated from

$$p_{1m}^{n+1} = \overline{p}_{1m}^{n+1} + \left(p_{1f}^{n+1} - p_{1f}^{n}\right)\tilde{p}_{1m}^{n+1} \tag{24}$$

Substituting the expression for p_{1m}^{n+1} into Eq. 20 and rearranging, we obtain

$$\begin{aligned}
\tau_{mf} = N_m \Bigg\{ p_{1f}^{n+1} \sum_{\ell=1}^{n_p} \Bigg[& \sum_{i=1}^{N_{subh}} \frac{2V_{b,iN_{subv}} k_z \lambda_{r\ell c,iN_{subv}+1/2} \left(\tilde{p}_{1m,iN_{subv}}^{n+1} - 1\right)}{\Delta z_{N_{subv}}^2} \\
& + \sum_{i=1}^{N_{subh}} \frac{2V_{b,i1} k_z \lambda_{r\ell c,i1/2} \left(\tilde{p}_{1m,i1}^{n+1} - 1\right)}{\Delta z_1^2} \\
& + \sum_{k=1}^{N_{subv}} 4\Delta z_k \lambda_{r\ell c,N_{subv}+1/2k} \left(\tilde{p}_{1m,N_{subh}k}^{n+1} - 1\right) \Bigg(\frac{k_x y_{N_{subh}}}{x_{N_{subh}} - 2x_{nod,N_{subh}}} \\
& + \frac{k_y x_{N_{subh}}}{y_{N_{subh}} - 2y_{nod,N_{subh}}} \Bigg) \Bigg]
\end{aligned}$$

$$
\begin{aligned}
&-p_{1f}^{n}\sum_{\ell=1}^{n_p}\left[\sum_{i=1}^{N_{subh}}k_z\frac{2V_{b,iN_{subv}}\lambda_{r\ell c,iN_{subv}+1/2}\tilde{p}_{1m,iN_{subv}}^{n+1}}{\Delta z_{N_{subv}}^{2}}\right.\\
&+\sum_{i=1}^{N_{subh}}k_z\frac{2V_{b,i1}\lambda_{r\ell c,i1/2}\tilde{p}_{1m,i1}^{n+1}}{\Delta z_{1}^{2}}\\
&+\sum_{k=1}^{N_{subv}}4\Delta z_k\lambda_{r\ell c,N_{subh}+1/2k}\tilde{p}_{1m,N_{subh}k}^{n+1}\left(\frac{k_x y_{N_{subh}}}{x_{N_{subh}}-2x_{nod,N_{subh}}}\right.\\
&\left.\left.+\frac{k_y x_{N_{subh}}}{y_{N_{subh}}-2y_{nod,N_{subh}}}\right)\right]
\end{aligned}
$$

$$
\begin{aligned}
&+\sum_{\ell=1}^{n_p}\left[\sum_{i=1}^{N_{subh}}k_z\frac{2V_{b,iNsubv}\lambda_{r\ell c,iNsubv+1/2}\bar{p}_{1m,iN_{subv}}^{n+1}}{\Delta z_{N_{subv}}^{2}}\right.\\
&+\sum_{i=1}^{N_{subh}}k_z\frac{2V_{b,i1}\lambda_{r\ell c,i1/2}\bar{p}_{1m,i1}^{n+1}}{\Delta z_{1}^{2}}+\sum_{k=1}^{Nsubv}4\Delta z_k\bar{p}_{1m,Nsubhk}^{n+1}\lambda_{r\ell c,N_{subv}+1/2k}\\
&\left.\left(\frac{k_x y_{N_{subh}}}{x_{N_{subh}}-2x_{nod,N_{subh}}}+\frac{k_y x_{N_{subh}}}{y_{N_{subh}}-2y_{nod,N_{subh}}}\right)\right]
\end{aligned}
$$

$$
\begin{aligned}
&-\sum_{\ell=1}^{n_p}\left[\sum_{i=1}^{N_{subh}}k_z\frac{2V_{biN_{subv}}\lambda_{r\ell c,N_{subv}+1/2}\gamma_{\ell,N_{subv}+1/2}\left(D_{iN_{subv}}-D_f\right)}{\Delta z_{N_{subv}}^{2}}\right.\\
&+\sum_{i=1}^{N_{subh}}k_z\frac{2V_{bi}\lambda_{r\ell c,i1/2}\gamma_{\ell,i1/2}\left(D_{i1}-D_f\right)}{\Delta z_{1}^{2}}\\
&+\sum_{k=1}^{N_{subv}}4\Delta z_k\lambda_{r\ell c,N_{subv}+1/2k}\gamma_{\ell,N_{subv}+1/2k}\left(D_{N_{subh}k}-D_f\right)\\
&\left.\left(\frac{k_x y_{N_{subh}}}{x_{N_{subh}}-2x_{nod,N_{subh}}}+\frac{k_y x_{N_{subh}}}{y_{N_{subh}}-2y_{nod,N_{subh}}}\right)\right]
\end{aligned}
$$

$$
\begin{aligned}
&+\sum_{\ell=2}^{n_p}\left[\sum_{i=1}^{N_{subh}}k_z\frac{2V_{biN_{subv}}\lambda_{r\ell c,iN_{subv}+1/2}\left(p_{c\ell 1,iN_{subv}}-p_{c\ell 1,f}\right)}{\Delta z_{Nsubv}^{2}}\right.\\
&+\sum_{i=1}^{N_{subh}}k_z\frac{2V_{bi}\lambda_{r\ell c,i1/2}\left(p_{c\ell 1,i1}-p_{c\ell 1,f}\right)}{\Delta z_{1}^{2}}\\
&+\sum_{k=1}^{N_{subv}}4\Delta z_k\lambda_{r\ell c,N_{subv}+1/2k}\left(p_{c\ell 1,N_{subh}k}-p_{c\ell 1,f}\right)\\
&\left.\left.\left(\frac{k_x y_{N_{subh}}}{x_{N_{subh}}-2x_{nodN_{subh}}}+\frac{k_y x_{N_{subh}}}{y_{N_{subh}}-2y_{nodN_{sub}h}}\right)\right]\right\}
\end{aligned}
\tag{25}
$$

The transfer function given in Eq. 25 can be evaluated by first solving Eq. 21, then solving Eq. 23. This procedure effectively eliminates any matrix unknowns at the n+1 time level from the transfer function, which facilitates solving the fracture and matrix pressure equations separately. Once the fracture pressures are known, the matrix pressures at the n+1 time level can be evaluated using Eq. 24.

We are now in the process of implementing the dual porosity model in UTCHEM.

TASK 2: FOAM MODEL

We have developed a semi-empirical foam model that describes gas mobility reduction and accounts for the effects of shear thinning due to gas flow rate, and water and oil saturations. This model accounts for these effects in a way consistent with the capillary-pressure mechanism that has been successful in representing foam displacement to date (Rossen *et al.*, 1994). It makes use of the observation that foam in a porous medium breaks catastrophically when capillary pressure exceeds a critical value P_C^* (Khatib *et al.*, 1988) which depends mainly on the surfactant used, its concentration, and the ionic strength of the surfactant solution.

The gas mobility reduction parameter R is calculated as a function of water saturation (S_w) as follows:

$$\begin{cases} R=1 \quad for \quad S_o > S_o^* \\ R=1 \quad for \quad C_s < C_s^* \\ R=1 \quad for \quad S_w < S_w^* - \varepsilon \\ R=1+(R_{\max}-1)\left(\dfrac{S_w - S_w^* + \varepsilon}{2\varepsilon}\right) \quad for \quad (S_w^* - \varepsilon) \le S_w \le (S_w^* + \varepsilon) \\ R = R_{\max} \quad otherwise \end{cases} \tag{26}$$

where R_{max}, ε, S_o^*, and C_s^* are input parameters. The known water saturation (S_w^*) corresponding to critical capillary pressure is for a given rock permeability. We are in the process of developing an algorithm to compute S_w^* as a function of rock permeability using laboratory coreflood data with a known permeability.

Effect of gas velocity (u_g) and foam shear thinning is accounted for as follows:

$$R' = R\left(\frac{u_g}{u_{g,ref}}\right)^{(\sigma-1)} \tag{27}$$

where $u_{g,ref}$ and σ are input parameters.

Gas relative permeability (k_{rg}) is then reduced by the factor R' to represent the foam flow as

$$k_{rf} = \frac{k_{rg}}{R'} \tag{28}$$

We implemented the proposed foam model in UTCHEM. To test and validate the model and its implementation in the code, we compared the simulation results with the coreflood results of Kibodeaux (1997). The core was an unfired Berea sandstone 24.2 cm long with a cross section of 15.6 cm^2 positioned vertically. The core had a permeability of 720 md and porosity of 0.22. The core was first saturated with water and nitrogen to a water saturation of 0.9. First the surfactant solution was injected followed by slugs of co-injection of gas and surfactant solution starting with 20% aqueous solution (f_w = 0.2). The injected water cut was then decreased incrementally to zero corresponding to 100% nitrogen gas injection. The injected surfactant solution was a mixture of 1 wt% sodium chloride, 0.01 wt% calcium chloride and 1 wt% surfactant. The coreflood results included the pressure drop measured across three sections of the core and along the length of the core and the average water saturation at different times. Several UTCHEM simulations were performed to simulate the experiment and match the pressure drops and average water saturations. The input foam parameters were varied to find the sensitivity of the model results to these parameters.

A comparison of the measured and simulated pressure drop as a function of pore volumes injected is shown in Fig. 2. The measured average water saturation is compared with the simulation results in Fig. 3. These preliminary results seem encouraging since the model can qualitatively mimic the behavior of foam. However, we need to simulate other experimental foam floods and enhance and fine tune the foam model as needed.

TASK 3. NUMERICAL AND CODING ENHANCEMENTS

In this task, we present our progress on the numerical and code enhancements of UTCHEM.

3.1 Dynamic Memory Management

The UTCHEM simulator is coded in FORTRAN 77, which does not allow the allocation of the memory during execution time of the simulator. We have implemented FORTRAN 90 features in the code to add dynamic memory capability. This feature allows users to run multiple data sets with different numbers of gridblocks using the same executable code. We first collected all the common blocks and the variables within these common blocks used in all subroutines and declared them in a module. The arrays are then defined as allocatable arrays. Each subroutine can then share the variables of the same module. The storage for the array is allocated when the ALLOCATE statement is executed; that is, the array becomes dynamically allocated. The allocatable array is available until it is automatically deallocated or until a DEALLOCATE statement is executed.

3.3 Graphical User Interface for Windows

We are making good progress on the design of the UTCHEM graphical user interface (GUI) to set up the input, make the run, and post process the output files of the simulation. The interface is being developed in Windows (NT, 95 or higher) operating systems using Visual Basic.

The GUI includes an utility called UTHIST for post-processing the UTCHEM output files that contain the output data as a function of time (called history files). The GUI is also linked to Microsoft Excel. Once the data are processed, Excel files are generated for each well for further data manipulation or graphical enhancements.

To generate 2-D maps (contour plots) of output data on a PC, a utility called UTSURF is used to convert the UTCHEM output files to a form compatible with Surfer software licensed from Golden Software, Inc. Surfer is a grid-based contouring and three-dimensional surface-plotting graphics program that runs under Microsoft Windows, Windows 95, and Windows NT. UTSURF reads the output files of UTCHEM that contain data such as pressure, saturation, and species concentration and the output file that contains the grid and well data. Once the Surfer script file is executed, 2-D images of UTCHEM output files are generated. The maps automatically include the grid, the well locations, and well names. An example of an areal cross section of surfactant concentration generated using UTSURF is given in Fig. 4.

3.4 Solvers

The discretization of the pressure equation in the UTCHEM simulator leads to a linear system of algebraic equations of the form Ax = b, where A is an n x n positive definite banded matrix, x is the vector of discrete pressures, and b is the boundary data vector. Efficient and accurate numerical solution of this system of equations is an important part of the overall solution methods for the UTCHEM simulator. Since we use large number of gridblocks in our field-scale simulations, the computational efficiency of the code requires a fast solver. The solver currently used for the solution of the pressure equation uses a Jacobi Conjugate Gradient (JCG) method. This solver has been vectorized and is very efficient on vector computers, especially on large, three-dimensional problems involving thousands of gridblocks even when simulating reservoirs with moderate to severe heterogeneities. To compare the efficiency of the existing solver in UTCHEM against other solvers on scaler computers such as PCs, we tested two solver packages coded for solving large sparse linear systems of equations that contain an implementation of various iterative methods in conjunction with the simulator. The packages that we tested are as follows:

1- PETSc, developed at Argonne National Lab. (Balay *et al.*, 1995). This solver package is available at the public domain: http//www-fp.mcs.anl.gov/petsc/.

2- NETLIB developed at the Lawrence Livermore National Laboratory. This solver package is available at the public domain: http//www.netlib.org.

We implemented the solvers in UTCHEM simulator and made several 3-D surfactant/polymer simulations with different numbers of gridblocks. The particular new solvers implemented in UTCHEM, for comparison, were JCG and Diagonally Scaled Conjugate Gradient (DSCG) from PETSc and NETLIB libraries, respectively. Table 1 gives a comparison of the total CPU time. The simulations were performed on Intel Pentium II 300 MHz chips. The results show that for this particular example, the existing JCG solver is the most efficient among the solvers tested.

3.5 Enhancements in the Geochemical Option

UTCHEM presently handles an arbitrary number of chemical reactions involving aqueous species, including precipitation/dissolution, adsorption, ion exchange and speciation. All geochemical reactions assume local thermodynamic equilibrium. The existing geochemistry option in UTCHEM has not been optimized in terms of computational efficiency. Thus 3D simulations involving geochemical species may require hours of workstation CPU time. For example, 57% to 77% of total simulation CPU time is spent in the reaction routines for the two runs discussed below. Therefore, the practical utility of UTCHEM for problems involving geochemistry would clearly benefit from a focused effort at reducing the computational overhead in the geochemistry routines. The underlying algorithm for the geochemistry module is derived from a batch module for which run time was not a concern. In order to investigate the possibility of reduction of the computational time for the geochemical option in UTCHEM, we compared the existing routines in the simulator that solve the linear and nonlinear equations for the equilibrium calculation of species concentrations with routines from two public domain library packages. The non-linear set of equations that calculate the equilibrium concentrations of fluid chemical species, solid species, matrix adsorbed cations, and cations adsorbed on micellar surfaces are solved using the Newton method. We selected a nonlinear system solver called SNES from the PETSc library to compare with the existing nonlinear solver in UTCHEM. We also compared our solver for the systems of linear equations with a routine that uses a Gaussian elimination LU decomposition method from NETLIB library. Modifications were also made to reduce the computational efforts in the procedure for calculating the Jacobian matrix used in conjunction with the existing routine that implements the Newton method.

A summary of the two data sets used in the comparison study is given in Table 2. The results of this study indicated that there was not a significant improvement in computational time by using the linear and nonlinear solvers from the public domain libraries. However as shown in Tables 3 and 4, there is about 7-20% CPU reduction due to the modification of the procedure for evaluating the Jacobian matrix.

TASK 4. PHYSICAL PROPERTY ENHANCEMENTS

4.1 Relative Permeability and Capillary Pressure Models

In addition to several functional forms such as Brooks-Corey, van Genuchten, Parker, and Lenhard available for relative-permeability and capillary-pressure curves, we have added the capability of table look-up for these properties. This option currently works only for two-phase oil/water flow and up to two tables can be included. These tables may represent two different wetting conditions i.e. water-wet and mixed-wet, or different rock types.

We have also incorporated a hysteretic gas relative permeability model in our existing Corey function. Here we give a brief description of the model. The three-phase oil/water/gas relative permeabilities in UTCHEM are calculated using Corey-type functions as below:

$$k_{ro} = k_{ro}^{o} \left(\frac{S_o - S_{or}}{1 - S_{or} - S_{wr} - S_{gr}}\right)^{e_o} \qquad (29)$$

$$k_{rw} = k_{rw}^{o} \left(\frac{S_w - S_{wr}}{1 - S_{orw} - S_{wr} - S_{gr}}\right)^{e_w} \tag{30}$$

$$k_{rg} = k_{rg}^{o} \left(\frac{S_g - S_{gr}}{1 - S_{org} - S_{wr} - S_{gr}}\right)^{e_g} \tag{31}$$

where

$k_{r\ell}$ = Relative permeability of phase ℓ

$k_{r\ell}^{o}$ = Endpoint relative permeability of phase ℓ

S_{gc} = Critical gas saturation of phase

S_{ℓ} = Saturation of phase ℓ

$S_{\ell r}$ = Residual saturation of phase ℓ

S_{or} = Residual oil saturation in three-phase flow

S_{org} = Residual oil saturation to gas

S_{orw} = Residual oil saturation to water

ℓ = Phase number w: water, g: gas, o: oil

The residual oil saturation in three-phase flow (S_{or}) is calculated based on the model developed by Fayers and Matthews (1982). Both the residual water and gas saturations are assumed to be constant and input to the simulator in the original model.

$$S_{or} = f(S_g, S_w) = b * S_{orw} + (1 - b) * S_{org} \tag{32}$$

$$b = 1 - \frac{S_g}{1 - S_{wr} - S_{org}} \tag{33}$$

To incorporate the gas trapping and hysteresis in gas relative permeability, the residual gas saturation is first modified using the following expression:

$$S_{gr}^{*} = \frac{S_{g\max}}{1 - S_{org} - S_{wr}} S_{gr} \tag{34}$$

$$\text{if} S_{gr}^{*} \le S_{gc}, \text{ then } S_{gr}^{*} = S_{gc} \tag{35}$$

This accounts for the fact that the endpoint of the gas imbibition curve will not be the residual gas saturation, S_{gr}, unless the drainage process has been carried all the way to the endpoint of the curve. Reversal of the process prior to this will cause the gas relative permeability to decline along a drainage curve that terminates somewhere between the critical gas saturation (S_{gc}) and the residual gas saturation. That value is empirically estimated here as S^*_{gr}. A second adjusted residual gas saturation, S^{**}_{gr}, is calculated as below:

$$S^{**}_{gr} = f\, S^*_{gr} + (1-f)\, S_{gc} \qquad (36)$$

where $f = \dfrac{S_{g\,max} - S_g}{S_{g\,max} - S^*_{gr}}$

This is used to smoothly interpolate from the drainage curve to the imbibition curve. As the gas saturation varies from the historical maximum in any simulation gridblock (S_{gmax}) to the adjusted residual gas saturation (S^*_{gr}), the value of S^{**}_{gr} varies from the critical saturation S_{gc}, to the adjusted residual value. This causes the imbibition curve to merge smoothly into the drainage curve. Once the above saturations are calculated, the adjusted gas saturation is then calculated as below:

$$S^*_g = S_{gc} + \frac{(S_g - S^{**}_{gr})(1 - S_{wr} - S_{org} - S_{gc})}{(1 - S_{wr} - S_{org} - S^*_{gr})} \qquad (37)$$

If $S_g = S_{gmax}$ then $S^*_g = Sg.$

If $S_g = S^*_{gr}$ then $S^*_g = Sgc$

The gas relative permeability is then calculated from the gas drainage curve using the adjusted gas saturation. Thus the only input data for the simulator are gas drainage relative permeability curve and residual gas saturation. The imbibition curves are generated as described above. We have successfully incorporated and tested the model and its implementation in UTCHEM.

4.2 Microbial Enhanced Oil Recovery Model

UTCHEM can simulate the biodegradation of chemical compounds that can serve as substrates (carbon and/or energy source) for microorganisms. The model simulates the destruction of substrates, the consumption of electron acceptors (e.g. oxygen, nitrates, etc.), and the growth of biomass. Substrates can be biodegraded by free-floating microorganisms in the bulk liquid phase or by attached biomass. Multiple substrates, electron acceptors, and biological species are accommodated by the model. Biodegradation reactions are modeled through operator splitting, in which the solution to the flow equations is used as the initial conditions for the biodegradation reactions. The details of the model can be found in de Blanc (1999). The model includes the following features:

- Inhibition of biodegradation by electron acceptors.
- Inhibition of biodegradation by substrates toxic to microorganisms.
- First-order abiotic decay reactions.
- Enzyme competition between multiple substrates.
- Monod, first-order or instantaneous biodegradation kinetics.
- Formation of biodegradation by-products.
- Mass transfer resistances to biodegradation.

The objective of this subtask was to extend the existing biodegradation model to microbial enhanced oil recovery (MEOR). A literature survey was completed to identify the microbial enhanced oil recovery mechanisms that can be incorporated into UTCHEM. The important mechanisms that are being considered are gas generation and trapping, surfactant production (with concomitant reductions in oil/water capillary pressures), pore volume reduction, and filtration from bacterial flocs. The last two mechanisms reduce the formation permeability, so that a major part of the modeling will be a permeability-reduction model. Permeability reduction is intricately related to transport and accumulation of bacteria, and several methods of describing bacterial transport have been investigated. Some important issues related to MEOR are addressed below.

Bacterial transport: Bacteria transport is modeled as an convective-diffusion process with a first-order, reversible process for attachment and detachment of bacteria. This is essentially a filtration process for bacterial transport. The exiting biodegradation code is being used to solve the kinetically limited attachment and detachment of bacteria. Two general options are being incorporated: 1) a user-dependent option in which the user inputs the filtration coefficient based on literature or experimental data and 2) an automatic option in which the filtration coefficient is calculated and changes with time and space in the simulation domain. Biomass filtration allows UTCHEM to calculate the biomass adsorption rate constant automatically based on filtration theory. This option is useful when experimental data are unavailable and equilibrium partitioning is judged to be an inadequate representation of biomass transport. The biomass rate constant is calculated as follows (Rajagopalan and Tien, 1976; Logan *et al.*, 1995):

$$k_f = \frac{3}{2}\frac{(1-\phi)\eta q}{d_c} \tag{38}$$

$$\eta = 4A_s^{1/3}N_{Pe}^{-2/3} + A_s N_{Lo}^{1/8} N_R^{15/8} + 0.00338 A_s N_G^{1.2} N_R^{-0.4} \tag{39}$$

$$A_s = \frac{2(1-\gamma^5)}{2-3\gamma+3\gamma^5-2\gamma^6} \tag{40}$$

$$\gamma = (1-\phi)^{1/3} \tag{41}$$

$$N_{Pe} = \frac{3\pi\mu d_c d_p q}{\kappa T} \tag{42}$$

$$N_{Lo} = \frac{4H}{9\pi\mu d_p^{\,2} U} \tag{43}$$

$$U = \frac{g(\rho_{bio}-\rho)d_p^{\,2}}{18\mu} \tag{44}$$

$$N_R = \frac{d_p}{d_c} \tag{45}$$

$$N_G = \frac{U}{q} \tag{46}$$

where

U = particle settling velocity
T = time
g = acceleration of gravity
ρ_{bio} = density of attached biomass
ρ = density of water
μ = viscosity of water
η = single particle collector efficiency
d_c = collector diameter
d_p = biomass particle diameter
q = approach (Darcy) velocity
H = Hamaker constant
κ = Boltzman's constant

Permeability reduction: The permeability reduction from biomass growth is very complex and is a very active field of investigation. Current models are either very simple, such as the Carman-Kozeny equation relating permeability reduction to reductions in porosity, or very complex. The more complex models relate permeability reduction to pore-size distributions or other characteristics of the porous medium. In general, the more complex models predict permeability reductions better than do the simple models. However, the complex models require a large number of medium-specific parameters and are very computationally intensive. Therefore, for this work, the Carman-Kozeny equation is currently being used to describe permeability reductions resulting from biomass growth. Other options considered required excessive computational effort and were not practical for field-scale applications. The Carmen-Kozeny equation incorporated in the model is

$$k = \frac{d_p^2 \, \phi^3}{300\,(1-\phi)^2} \tag{47}$$

where

k = initial permeability

ϕ = initial porosity

d_p = average particle size diameter

Dead Biomass: Dead biomass can accumulate in the aquifer, but eventually decays. Dead biomass will become substrate for bacteria still alive in the formation.

Gas Production: Gas production from active bacteria can be a significant MEOR mechanisms and will be simulated allowing bacteria to generate gas. No significant changes in the code are anticipated to facilitate gas production.

The modified UTCHEM model was tested through simple simulations to identify and correct coding errors and to compare simulated results to those reported in the literature. Concurrent with code testing, realistic MEOR simulations are being designed to demonstrate the ability of the model to simulate field-scale MEOR processes.

REFERENCES

Al-Lawati, S. and S. Saleh: "Oil Recovery in Fractured Oil Reservoirs by Low IFT Imbibition Process," paper SPE 36688 presented at the 1996 Society of Petroleum Engineers Annual Technical Conference and Exhibition, Denver, CO, Oct. 6-9.

Austad, T.: "An Evaluation of Using Surfactants in Order to Improve the Oil Recovery from the Ekofisk Chalk Formation by Water Imbibition," paper presented at the RUTH Seminar, Stavanger, Norway, September 5-6, 1994.

Balay, S., W. Gropp, L.C. McINnes, and B. Smith: PETsc 2.0 User's Manual, Math. And Computer Sci., Argone National Lab., IL (Nov. 1995).

Barenblatt, G.E., I.P. Zheltov, and I.N. Kochina: "Basic Concepts in the Theory of Homogeneous Liquids in Fissured Rocks," *J. Appl. Math. Mech.* (USSR) (1960) **24** (5), 852.

Chen, J.: "New Approachs to Dual Prosity Modeling of water Flooding in Naturally Fractured Reservoirs," PhD dissertation, The U. Of Texas, Austin (1993).

de Blanc, P.C.: "Development and Demonstration of a Biodegradation Model for Non-Aqueous Phase Liquids," Ph.D. dissertation, The University of Texas at Austin, 1999.

Datta Gupta, A. G.A. Pope, K. Sepehrnoori, and R.L. Thrasher: " A Symmetric, Positive Definie Formulation of A Three Dimensional Micellar/Polymer Simulator," *SPE Reservoir Eng.* (1986) **1**(6), 622-632.

Delshad, M. G.A. Pope and K. Sepehrnoori: "A Compositional Simulator for Modeling Surfactant Enhanced Aquifer Remediation 1. Formulation," *J. of Contaminant Hydrology* (1996) **23**, 303-327.

Logan, B.E., D.G. Jewett, R.G. Arnold, E.J. Bouwer, and C.R. O'Melia: "Clarification of Clean-Bed Filtration Models," *Journal of Environmental Engineering* (December 1995) **121**(12), 869-873.

Fayers, F.J. and J.P. Mattews: "Evaluation of Normalized Stone's Methods for Estimating Three-Phase Relative Permeabilities," *SPEJ* **24** (1982), 224-232.

Kazemi, H., L.S. Merrill, Jr., K.L. Porterfield, and P.R. Zeman: "Numerical Simulation of Water-Oil in Naturally Fractured Reservoirs," *Soc. Pet. Eng. J.* (December 1976), 317-326

Keijzer, P.P.M., and A.S. de Vries: "Imbibition of Surfactant Solutions," Paper SPE/DOE 20222 presented at the 7th Symposium on EOR, Tulsa, OK, April 22-25, 1990.

Rajagopalan, R. and C. Tien: "Trajectory Analysis of Deep-Bed Filtration with the Sphere-in-cell Porous Media Model," AIChE *Journal* (May 1976) **22**(3), 523-533.

Saad, N.: "Field Scale Simulation of Chemical Flooding," Ph.D. dissertation, The University of Texas at Austin (1989).

Schechter, D.S., D. Zhou, and F.M. Orr: "Capillary Imbibition and Gravity Segregation in Low IFT Systems," paper SPE 22594 presented at the 66th Society of Petroleum Engineers Annual Technical Conference, Dallas, TX, Oct. 6-9, 1991.

Warren, J.E., and P.J. Root: "The Behavior of Naturally Fractured Reservoirs," *Soc. Pet. Eng. J.* (Sept. 1963), 245-255.

Wu, Y.S. and K. Pruess: "A Multiple-Porosity Method for Simulation of Naturally Fractured Reservoir," paper SPE 15129 presented at the 1986 Soc. Pet. Eng. California Regional Meeting, Oakland, CA, April 2-4.

Khatib, Z.I., G.J. Hirasaki, and A.H. Falls: "Effects of Capillary Pressure on Coalescence and Phase Mobilities in Foams Flowing Through Porous Media," *SPE Reser. Eng.*, **3**(3), 919-926 (August 1988).

Kibodeaux, K.R: "Experimental and Theoretical Studies of Foam Mechanisms in Enhanced Oil Recovery and Matrix Acidization Applications," Ph.D. dissertation, The University of Texas at Austin, Aug. 1997.

Rossen, W.R., S.C. Zeilinger, J.-X. Shi, and M.T. Lim: "Mechanistic Simulation of Foam Processes in Porous Media," paper 28940 presented at the 1994 Annual Technical Conference and Exhibition, New Orleans, LA, September 26-28.

Table 1: Summary of UTCHEM execution times using several solver packages

	Execution Times, Seconds		
Solver	**1000 gridblocks (10x10x10)**	**5000 gridblocks (25x20x10)**	**10,000 gridblocks (40x25x10)**
UTCHEM – JCG	946	43,280	231,816
PETSc - JCG	938	47,235	276,057
NETLIB-DSCG	1,270	43,346	245,987

Table 2: Summary of input for geochemical simulations

Description	**Simulation No.**	
	EX07	**EX45**
No. of gridblocks	19x19x3	100
No. of components	11	15
No. of reacting elements	7	12
No. of fluid species	18	51
No. of solid species	4	7
No. of adsorbed species	4	0
Simulation time, days	251	150

Table 3: Summary of UTCHEM execution times (seconds) for geochemical simulations on DEC-alpha 550 workstation

	Execution Times on DEC-alpha (Seconds)	
Method	**Run EX07**	**Run EX45**
Newton (original UTCHEM)	3435	45
Newton (modified Jacobian)	3202	37

Table 4: Summary of UTCHEM execution times (seconds) for geochemical simulations on 300 MHz Intel Pentium

	Execution Times on DEC-alpha (Seconds)	
Method	**Run EX07**	**Run EX45**
Newton (original UTCHEM)	4711	91
Newton (modified Jacobian)	4366	73

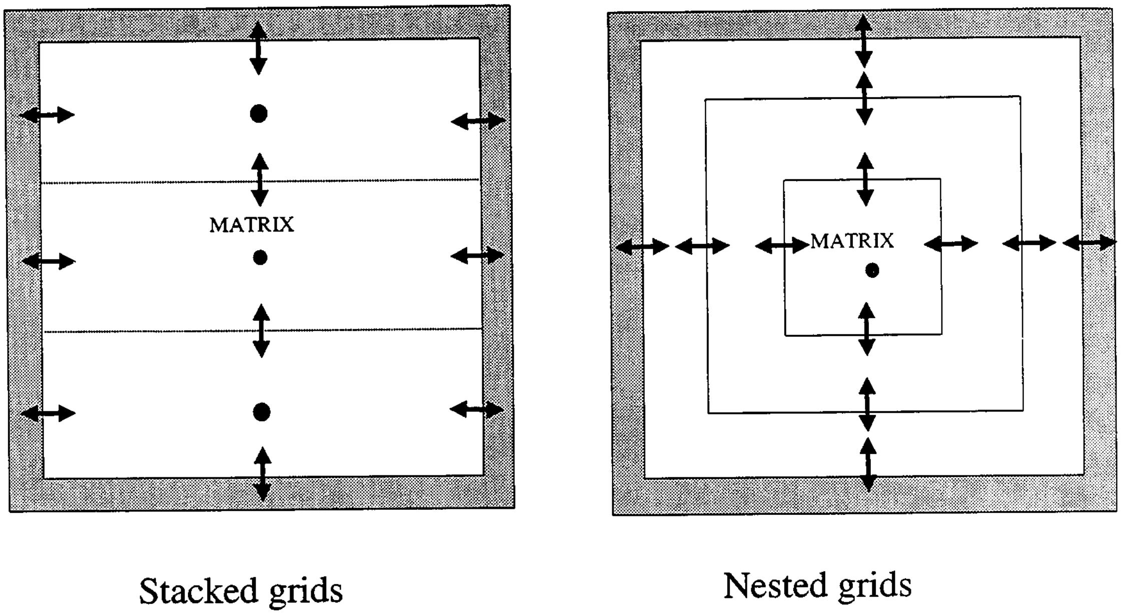


Fig. 1: Subgrids of a single matrix block.

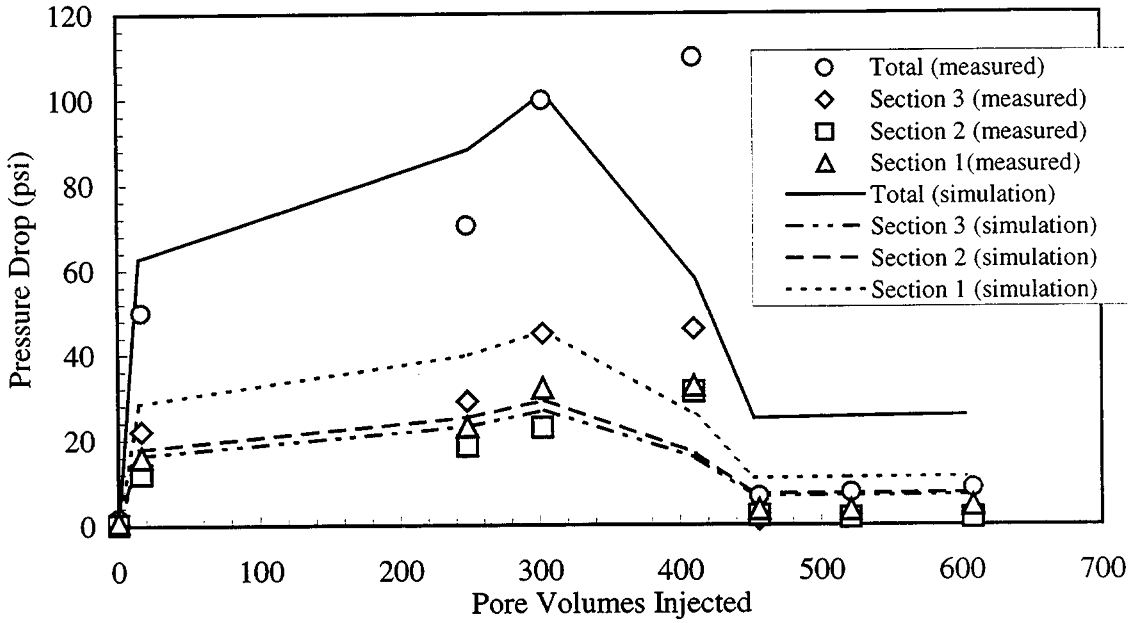


Fig. 2: Measured and simulated pressure drop across different sections of the core during the foam experiment (Kibodeaux, 1997).

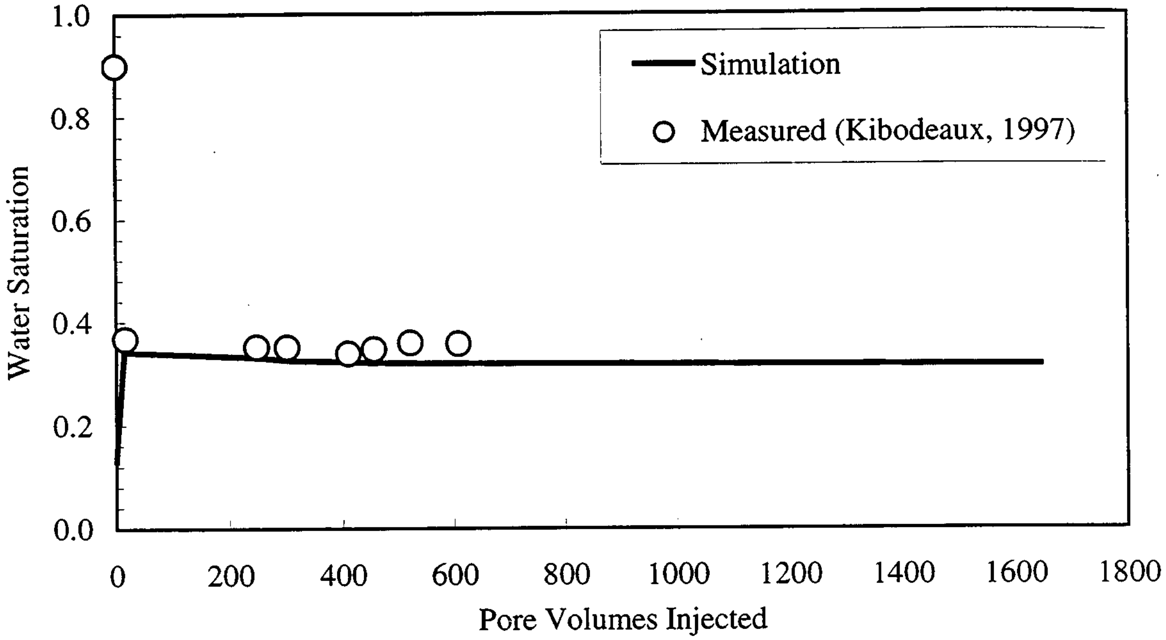
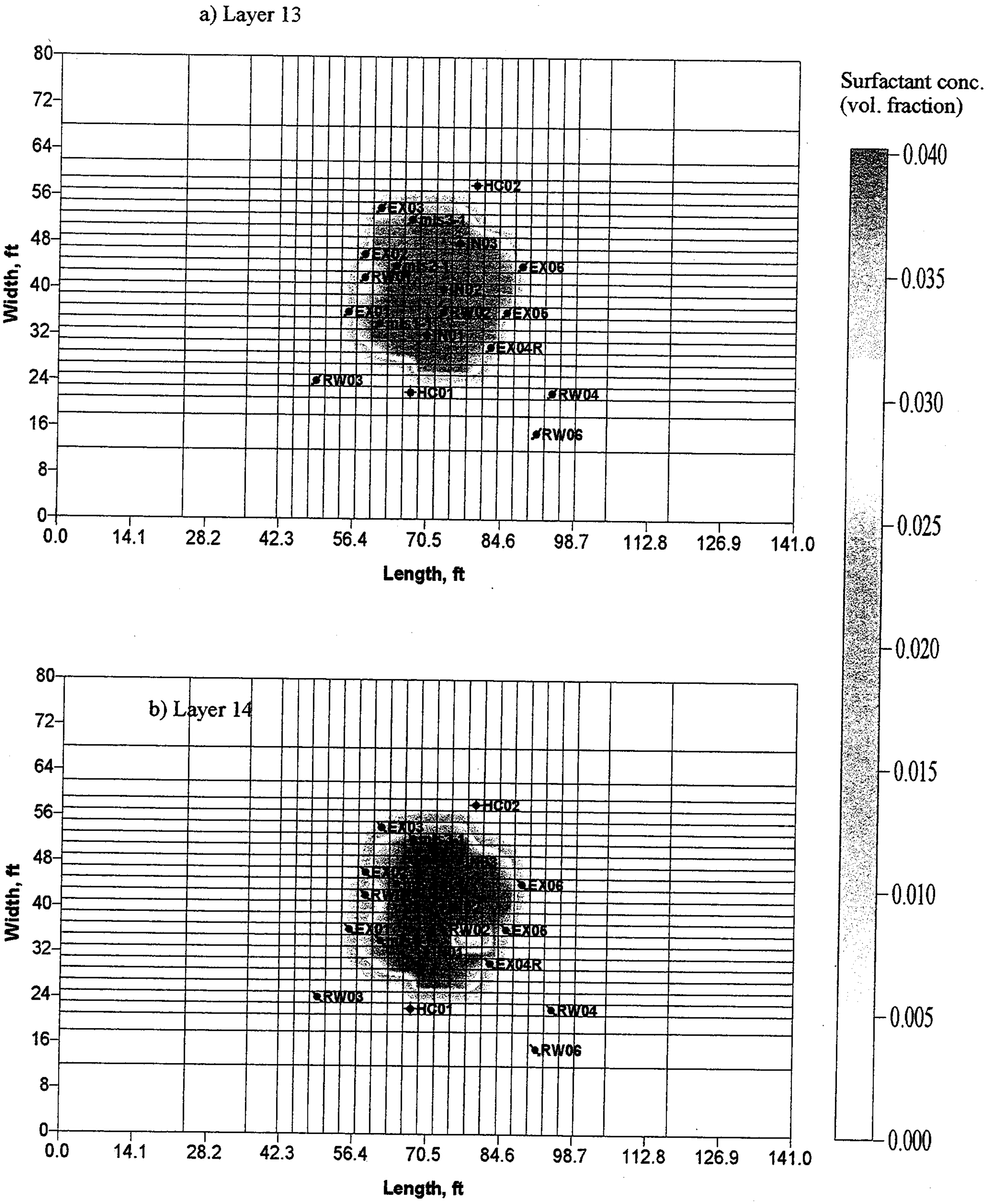


Fig. 3: Measured and simulated average water saturation during the foam experiment.



a) Layer 13

b) Layer 14

Fig. 4: Examples of surfactant concentration distributions produced using UTSURF postprocessing utility