Implicit Black-oil Model in IPARS Framework
IPARSv2, Model I, Keyword BLACKI
Version 1.1

Qin Lu, Malgorzata Peszynska, Xiuli Gai
Texas Institute for Computational and Applied Mathematics
The University of Texas at Austin

October 29, 2001

Contents

1 Introduction

In this document we describe a parallel 3-dimensional implicit black-oil model implemented in the IPARS framework [63, 44, 1]. We discuss the basic assumptions, mathematic model, numerical formulation and results of some numerical experiments, including the comparison with the commercial reservoir simulator Eclipse-100 [56, 57]. The basic concepts and descriptions of petroleum reservoir and fluid properties used in this manual can be found in various books and other references [23, 28, 46, 33, 21, 25, 26, 39, 3]. Numerical methods for reservoir simultion with black-oil and compositional models have been extensively studied [46, 8, 38, 41, 22, 63, 16, 15, 35, 20, 18, 19, 17, 72, 71, 52, 29, 11, 61, 12].

Credits. Several researchers contributed to the IPARS framework [2] which is maintained by the members of Center for Subsurface Modeling. In particular, aside of the authors of this manual, John Wheeler and Joe Eaton contributed to the development of the black-oil model code described in this manual. First version of this manual was written by Qin Lu as part of his thesis [37]. Section 4 was added by Malgorzata Peszynska and Sections 5 and 6 were added by Xiuli Gai who also ran most of the testing and comparative cases presented here and published in [67]. The model has been coupled with a geomechanics model JAS3D [42] and with a poroelastic model under IPARS. In addition, the model is a part of the *multiblock multiphysics* framework under IPARS [50, 51, 68, 36].

2 Mathematical Model

To avoid confusion, we use capital letter subscripts to denote water, oil and gas component (W, O, G, respectively) and lower-case letter subscripts to denote water, oil and gas phases (w, o, g, respectively). The definitions of the variables are listed in Section 2.2.

2.1 Basic Assumptions

The black-oil model is a three phase (water, oil and gas) model describing the flow in a petroleum reservoir [46]. It is assumed that no mass transfer occurs between the water phase and the other two phases. In the hydrocarbon (oil-gas) system, only two components are considered. The *oil* component (also called stock-tank oil) is the residual liquid at atmospheric pressure left after a differential vaporization, while the *gas* component is the remaining fluid.

One should distinguish the difference between *phase* and *component* [46, 23, 33, 28]. For water, these two concepts are the same in the model since no mass transfer occurs between the water phase and the others. As in the above definition, oil and gas components refer to those total hydrocarbon mixture that would be in liquid (usually heavy hydrocarbons) and gaseous (usually light hydrocarbons) states, respectively, after surface separation. The corresponding pressure and temperature are normally called *standard conditions*, that is, usually (but not always) 14.7 *psi* and $60°F$. The concept of *phase* would be meaningful only under a certain pressure and temperature.

The following assumptions have been made in developing the mathematical model:

- The reservoir is isothermal.
- The reservoir is surrounded by an impermeable formation so that no-flow Neumann boundary conditions apply on the external boundary.
- The permeability tensor is diagonal and aligned with the coordinate system.
- Fluid flow is characterized by Darcy's law for multi-phase flow.
- The viscosity of each phase is constant.
- No chemical reactions, precipitation or adsorption are present.
- The well injection and production are treated as source or sink terms.
- The formation is slightly compressible.

2.2 Definitions

The units defined for the following parameters are used as the internal units in the model.

1. Subscript W, O, G refers to water, oil and gas components respectively
2. Subscript w, o, g refers to water, oil and gas phases respectively
3. $B_m \left(P_m\right)$ = Formation volume factor as a function of P_m = Ratio of the volume of phase m (including any dissolved components) under reservoir condition to the stock tank volume of phase m $[bbl/stb]$
4. $B_{do} \left(P_o\right)$ = Formation volume factor of gas free oil (dead-oil) as a function of P_o = Ratio of the volume of gas free oil under reservoir condition to the stock tank volume of oil component $[bbl/stb]$
5. $B^*_{do} = B_{do}$ at the P_{sat} corresponding to a R_o $[bbl/stb]$
6. $B_{so} \left(P_o\right)$ = Formation volume factor of oil saturated with gas = Ratio of the volume of gas saturated oil phase under reservoir condition to the stock tank volume of oil component $[bbl/stb]$
7. $B^*_{so} = B_{so}$ at the P_{sat} corresponding to a R_o $[bbl/stb]$
8. $b_w = 1/B_w$ $[stb/bbl]$
9. c_ϕ = Compressibility constant of rock $[psi^{-1}]$
10. D = Depth $[ft]$
11. g = Gravity magnitude $[psi\text{-}ft^2/lb]$
12. K = Diagonal permeability tensor $[md]$
13. k_{rm} = Relative permeability to phase m $[dimensionless]$
    (a) Two phase, $k_{rw}\left(S_w\right), k_{ro}(S_w)$
    (b) Three phase, $k_{rw}\left(S_w\right), k_{ro}\left(S_w, S_g\right), k_{rg}\left(S_g\right)$, where oil relative permeability $k_{ro}\left(S_w, S_g\right)$ is evaluated using two-phase relative permeabilities.
14. k_{rij} = Relative permeability of phase i flowing with phase j $[dimensionless]$
15. k^o_{rij} = Endpoint relative permeability of phase i flowing with phase j $[dimensionless]$
16. N_M = Stock tank volume of component M per unit pore volume [stb/bbl for water and oil components, and $mscf/cu\text{-}ft$ for gas component]
17. P_m = Pressure in phase m $[psi]$
18. P_{sat} = Saturation pressure = The pressure at which no more gas is able to or is available to be dissolved in oil phase $[psi]$
19. Q_M = Volumetric rate of injection of component M $[cu\text{-}ft/day]$
20. q_M = Mass rate of injection of component M [scf/day for water and oil components, and $mscf/day$ for gas component]

21. R_o = Gas-oil ratio = Stock tank volume of gas dissolved in a stock tank volume of oil $[mscf/scf]$

22. $R_{so}(P_o)$ = solution gas-oil ratio as a function of P_o = Stock tank volume of gas component dissolved in a stock tank volume of oil component when the oil is saturated with gas $[mscf/scf]$

23. S_m = Saturation of phase m = Fraction of the pore volume occupied by phase m $[fraction]$

24. S_{mr} = Residual saturation of phase m $[fraction]$

25. t = Time $[day]$

26. U_M = Mass flow rate of component M through a unit area [scf/ft^2-day for water and oil components, and $mscf/ft^2$-day for gas component]

27. $\phi(P_w)$ = Porosity as a function of P_w $[fraction]$

28. ϕ_z = Porosity at pressure zero $[fraction]$

29. μ_m = Viscosity of phase m $[cp]$

30. ρ_m = Density of phase m $[lb/cu\text{-}ft]$

31. ρ_{MS} = Stock tank density of component M [lb/scf for water and oil components and $lb/mscf$ for gas component]

2.3 Units

All units used by the parameters are listed in Section 2.2. Note that stb/bbl (a barrel under stock tank condition per barrel under reservoir condition) or $scf/cu\text{-}ft$ is used for water component concentration N_W and oil component concentration N_O, while $mscf/cu\text{-}ft$ (a thousand cubic feet under stock tank condition per cubic foot under reservoir condition) is used as the unit for gas component concentration N_G, so that the magnitudes of the residual for gas component conservation equation and gas component flux are compatible with those of water and oil component, which is desirable for the linear solver.

2.4 Mass Conservation Equations

Mass conservation equations for water, oil and gas components are given by

$$\frac{\partial(\phi N_W)}{\partial t} = -\nabla \cdot U_W + q_W \tag{1}$$

$$\frac{\partial(\phi N_O)}{\partial t} = -\nabla \cdot U_O + q_O \tag{2}$$

$$\frac{\partial(\phi N_G)}{\partial t} = -\nabla \cdot (U_G + R_o U_O) + q_G \tag{3}$$

Note that the gas component flux consists of that in gas phase (free gas) and that dissolved in oil phase through gas-oil ratio R_o, as shown in Equation (3). As for water component, since it can never transfer to other phases, it always has the same meaning as water phase.

Darcy's law for multi-phase flow is used to calculate the mass velocity of each component in its corresponding phase

$$\begin{aligned} U_W &= -\frac{Kk_{rw}}{B_w\mu_w}\left(\nabla P_w - \rho_w g \nabla D\right) & (4)\\ U_O &= -\frac{Kk_{ro}}{B_o\mu_o}\left(\nabla P_o - \rho_o g \nabla D\right) & (5)\\ U_G &= -\frac{Kk_{rg}}{B_g\mu_g}\left(\nabla P_g - \rho_g g \nabla D\right) & (6)\end{aligned}$$

Because of volume balance, the saturations must satisfy the constraint

$$S_w + S_o + S_g = 1 \qquad (7)$$

Capillary pressures are defined as functions of saturations in following way

$$\begin{aligned} P_{cow}\left(S_w\right) &= P_o - P_w & (8)\\ P_{cgo}\left(S_g\right) &= P_g - P_o & (9)\end{aligned}$$

We use the following criteria to determine whether the state is three phase or two phase:

$$\begin{aligned} N_G > R_{so}N_O, &\quad \text{three-phase} & (10)\\ N_G \le R_{so}N_O, &\quad \text{two-phase} & (11)\end{aligned}$$

Three phase properties can be evaluated as:

$$\begin{aligned} R_o &= R_{so} & (12)\\ \rho_g &= \frac{\rho_{GS}}{B_g} & (13)\\ \rho_o &= \frac{\rho_{OS} + R_{so}\rho_{GS}}{B_{so}} & (14)\\ \rho_w &= \frac{\rho_{WS}}{B_w} = \rho_{WS}b_w & (15)\\ S_g &= B_g\left(N_G - R_{so}N_O\right) & (16)\\ S_o &= B_{so}N_O & (17)\\ S_w &= B_wN_W = 1 - S_g - S_o & (18)\\ N_W &= \left[1 - \left(B_{so} + B_gR_{so}\right)N_O - B_gN_G\right]b_w & (19)\end{aligned}$$

Two phase properties can be evaluated as:

$$\begin{aligned} R_o &= \frac{N_G}{N_O} & (20)\\ B_o &= B_{do} + \left(B^*_{so} - B^*_{do}\right)\frac{R_o}{R_{so}} & (21)\\ \rho_o &= \frac{\rho_{OS} + R_o\rho_{GS}}{B_o} & (22)\\ \rho_w &= \frac{\rho_{WS}}{B_w} = \rho_{WS}b_w & (23)\\ S_o &= B_oN_O & (24)\\ S_w &= B_wN_W = 1 - S_o & (25)\\ N_W &= \left(1 - B_oN_O\right)b_w & (26)\end{aligned}$$

Formation porosity is taken to be a function of pressure only

$$\phi = \phi_z \left(1 + c_\phi P_w\right) \tag{27}$$

2.5 Relative Permeability

A relative permeability model [24, 9, 34, 59, 60, 30] is necessary for the simultaneous flow of three phases. The two-phase relative permeabilities can be experimental data or they can be estimated by some two-phase models. The water and gas relative permeabilities in three-phase are the same as in two-phase. However, the oil relative permeabilities in three-phase have to be estimated based on two-phase relative permeabilities. Three models: Stone's model I, Stone's model II, a model from Eclipse-100, were implemented to evaluate the three-phase oil relative permeabilities. It is assumed that the water phase is always present and that it is the wetting phase. The choice of the model is made by specifying parameter $MODREL$ which can take values 2,3, and 4. Value $MODREl = 4$ corresponding to the Eclipse model is the default.

2.5.1 (1) Stone's Model I: $MODREL = 2$

This model was proposed by Stone [59, 24, 30] and was normalized with k_{row}^o as

$$k_{ro} = \frac{\bar{S}_o k_{row} k_{rog}}{k_{row}^o (1 - \bar{S}_w)(1 - \bar{S}_g)} \tag{28}$$

where

$$\begin{aligned} \bar{S}_o &= \frac{S_o - S_{or}}{1 - S_{wr} - S_{or} - S_{gr}} \\ \bar{S}_w &= \frac{S_w - S_{wr}}{1 - S_{wr} - S_{or} - S_{gr}} \\ \bar{S}_g &= \frac{S_g - S_{gr}}{1 - S_{wr} - S_{or} - S_{gr}} \end{aligned}$$

2.5.2 (2) Stone's Model II: $MODREL = 3$

The normalized Stone's Model II [60, 24, 30] is described as

$$k_{ro} = k_{row}^o \left[\left(\frac{k_{row}}{k_{row}^o} + k_{rw} \right) \left(\frac{k_{rog}}{k_{row}^o} + k_{rg} \right) - (k_{rw} + k_{rg}) \right] \tag{29}$$

Note that the values of k_{ro} produced by this formula can be negative, which is physically meaningless. So we set any negative k_{ro} values to zero in implementation.

2.5.3 (3) Eclipse's Model: $MODREL = 4$, default

It is a model used by Eclipse-100 [57]. The gas and water saturations are assumed to be completely segregated, the oil saturation is assumed to be constant throughout the cell. Assuming the cell average saturations are S_o, S_w and S_g, the oil relative permeability is then given by

$$k_{ro} = \frac{S_g k_{rog} + (S_w - S_{wr}) k_{row}}{S_g + S_w - S_{wr}} \tag{30}$$

where k_{rog} is the oil relative permeability for a system with oil, gas and residual water, with oil saturation S_o, water saturation S_{wr}, and gas saturation $S_g + S_w - S_{wr}$, and where k_{row} is the oil relative permeability for a system with oil and water only, where oil saturation equals S_o, water saturation equals $S_g + S_w$, and gas saturation equals 0.

3 Numerical Formulation

The solution domain is divided into a tensor product grid. A finite difference method is applied to discretize the non-linear partial differential equations of the problem, with P_w, N_O and N_G chosen as primary unknowns. The equations are discretized using cell-centered difference(CCD) in space and backward Euler scheme [69] in time. Up-winding is applied to cell edge values for numerical stability [46, 8]. CCDF in space is equivalent to the expanded mixed finite element method of the lowest order Raviart Thomas space ($RT0$) on a rectangular grid, with certain quadrature rules [54, 5, 6, 4].

The system of non-linear equations is linearized by Newton's method. A GMRES solver with multi-level preconditioner and an LSOR solver are avalaible to solve the linear system at every Newtonian step.

3.1 Definitions of Discrete Terms

1. Subscript ijk refers to grid cell ijk
2. Superscript n refers to time level n
3. Superscript K refers to the Kth approximation to a quantity evaluated at time level $n+1$
4. $\Delta t^{n+1} = t^{n+1} - t^n =$ time step size at time level $n+1$ [day]
5. $V_{ijk} =$ volume of grid cell ijk [cu-ft]
6. Δx_i, Δy_j, $\Delta z_k =$ edge lengths of a rectangular grid cell ijk [ft]
7. One-point upstream weighting is applied to the mobility λ and the gas-oil ratio R_o

$$\begin{aligned} \lambda^K_{m,i+1/2,jk} &= \Delta t^{n+1} A_{i+1/2,jk} \left(\frac{k_{rm}}{B_m \mu_m} \right)^K_{sjk} \\ R^K_{o,i+1/2,jk} &= R^K_{o,sjk} \end{aligned}$$

   where $s = i$ if $P^K_{m,ijk} \geq P^K_{m,i+1jk}$ and $s = i+1$ if $P^K_{m,ijk} < P^K_{m,i+1jk}$
   Similarly, we have the upstream weighting for y and z directions.

8. Transmissibility constants

$$\begin{aligned} A_{i+1/2,jk} &= 2\Delta y_j \Delta z_k \left(\frac{\Delta x_i}{K_{xx,ijk}} + \frac{\Delta x_{i+1}}{K_{xx,i+1,jk}} \right)^{-1} \\ A_{ij+1/2,k} &= 2\Delta z_k \Delta x_i \left(\frac{\Delta y_j}{K_{yy,ijk}} + \frac{\Delta y_{j+1}}{K_{yy,ij+1,k}} \right)^{-1} \\ A_{ij,k+1/2} &= 2\Delta x_i \Delta y_j \left(\frac{\Delta z_k}{K_{zz,ijk}} + \frac{\Delta z_{k+1}}{K_{zz,ij,k+1}} \right)^{-1} \end{aligned}$$

   where K_{xx}, K_{yy}, K_{zz} are the diagonal terms of the absolute permeability tensor. Actually, harmonic average is used here for the absolute permeability between two adjacent cells to keep mass conservation cell by cell.

9. Average densities are used in gravity force terms

$$\begin{aligned}\rho_{i+1/2,jk} &= \frac{\rho_{ijk}+\rho_{i+1,jk}}{2}\\ \rho_{ij+1/2,k} &= \frac{\rho_{ijk}+\rho_{ij+1,k}}{2}\\ \rho_{ij,k+1/2} &= \frac{\rho_{ijk}+\rho_{ij,k+1}}{2}\end{aligned}$$

However, if a specific phase exists only in one grid cell of two neighboring cells, up-winding is then applied for the density of this phase in gravity force terms.

3.2 Discretization in Time

The backward Euler scheme in time, or the implicit scheme, is applied to the mass conversation equations:

$$\frac{(\phi N_W)^{n+1}-(\phi N_W)^n}{\Delta t^{n+1}} + \nabla\cdot U_W^{n+1} = q_W^{n+1} \tag{31}$$

$$\frac{(\phi N_O)^{n+1}-(\phi N_O)^n}{\Delta t^{n+1}} + \nabla\cdot U_O^{n+1} = q_O^{n+1} \tag{32}$$

$$\frac{(\phi N_G)^{n+1}-(\phi N_G)^n}{\Delta t^{n+1}} + \nabla\cdot (U_G+R_oU_O)^{n+1} = q_G^{n+1} \tag{33}$$

3.3 Residual with Time and Space Discretization

Multiply each discrete-in-time mass conversation equation with Δt^{n+1} and V_{ijk}, and let superscript K denote the Kth approximation to a quantity evaluated at time level $n+1$, then the nonlinear residual equations are given by

$$R^K_{W,ijk} = V_{ijk}\Big\{\Big(\phi N_W\Big)^K_{ijk} - \Big(\phi N_W\Big)^n_{ijk} + \Delta t^{n+1}\Big[\nabla\cdot U_W - q_W\Big]^K_{ijk}\Big\} \tag{34}$$

$$R^K_{O,ijk} = V_{ijk}\Big\{\Big(\phi N_O\Big)^K_{ijk} - \Big(\phi N_O\Big)^n_{ijk} + \Delta t^{n+1}\Big[\nabla\cdot U_O - q_O\Big]^K_{ijk}\Big\} \tag{35}$$

$$R^K_{G,ijk} = V_{ijk}\Big\{\Big(\phi N_G\Big)^K_{ijk} - \Big(\phi N_G\Big)^n_{ijk} + \Delta t^{n+1}\Big[\nabla\cdot \Big(U_G+R_oU_O\Big) - q_O\Big]^K_{ijk}\Big\} \tag{36}$$

Component mass velocities are discretized using CCD in space:

$$\begin{aligned}&\Delta t^{n+1}V_{ijk}\left(\nabla\cdot U_M\right)^K_{ijk} =\\ &\quad -\lambda^K_{m,i+1/2,jk}\left[P^K_{m,i+1,jk} - P^K_{m,ijk} - \rho^K_{m,i+1/2,jk}g\left(D_{i+1,jk}-D_{ijk}\right)\right]\\ &\quad +\lambda^K_{m,i-1/2,jk}\left[P^K_{m,ijk} - P^K_{m,i-1,jk} - \rho^K_{m,i-1/2,jk}g\left(D_{ijk}-D_{i-1,jk}\right)\right]\\ &\quad + \text{similar terms for the } y \text{ and } z \text{ directions.}\end{aligned} \tag{37}$$

$$\begin{aligned}&\Delta t^{n+1}V_{ijk}\left[\nabla\cdot\left(R_oU_O\right)\right]^K_{ijk} =\\ &\quad -R_{o,i+1/2,jk}\lambda^K_{o,i+1/2,jk}\left[P^K_{o,i+1,jk} - P^K_{o,ijk} - \rho^K_{o,i+1/2,jk}g\left(D_{i+1,jk}-D_{ijk}\right)\right]\\ &\quad +R_{o,i-1/2,jk}\lambda^K_{o,i-1/2,jk}\left[P^K_{o,ijk} - P^K_{o,i-1,jk} - \rho^K_{o,i-1/2,jk}g\left(D_{ijk}-D_{i-1,jk}\right)\right]\\ &\quad + \text{similar terms for the } y \text{ and } z \text{ directions.}\end{aligned} \tag{38}$$

The definitions of all variables, subscripts and superscripts are in Section 3.1. Note that in Equation (37), component M (W, O, G, respectively) always corresponds to its own phase m (w, o, g, respectively).

The problem defined by Equations (1)-(3) which have been discreteized as Equations (31)-(33) is then can then be written as

$$R_{W,ijk} = 0 \tag{39}$$

$$R_{O,ijk} = 0 \tag{40}$$

$$R_{G,ijk} = 0 \tag{41}$$

for all grid cells at every time step.

3.4 Newton's Method

The non-linear system described by Equations (39)- (41) is linearized by Newton's method [69, 53] as follows:

$$J(s^K)\delta s^K = -R(s^K) \tag{42}$$

$$s^{K+1} = s^K + \delta s^K \tag{43}$$

where superscript K denotes the Kth Newtonian step, s is the vector of primary unknowns of all grid cells, R is the vector of nonlinear residual equations on all grid cells, and J is the Jacobian matrix:

$$s = \begin{pmatrix} P_w \\ N_O \\ N_G \\ \vdots \end{pmatrix} \tag{44}$$

$$R(s) = \begin{pmatrix} R_W(s) \\ R_O(s) \\ R_G(s) \\ \vdots \end{pmatrix} \tag{45}$$

$$J(s) = \begin{pmatrix} \frac{\partial R_W}{\partial P_w} & \frac{\partial R_W}{\partial N_O} & \frac{\partial R_W}{\partial N_G} & \cdots \\ \frac{\partial R_O}{\partial P_w} & \frac{\partial R_O}{\partial N_O} & \frac{\partial R_O}{\partial N_G} & \cdots \\ \frac{\partial R_G}{\partial P_w} & \frac{\partial R_G}{\partial N_O} & \frac{\partial R_G}{\partial N_G} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{46}$$

3.5 Construction of Jacobian Matrix

Every partial derivative in the Jacobian matrix can be extracted from the complete derivatives of the residuals with respect to the primary unknowns. The residuals can be divided into three terms: accumulation, transport and well. Here we list the complete derivatives of the three terms. Note that the superscript n refers to time level n and superscript K refers to the Kth approximation to a quantity evaluated at time level $n+1$. The complete derivatives (e.g., those of transmissibilities, densities and pressures, etc.) and the partial derivatives that are necessary to calculate the following formulations are listed in the Appendix A.

3.5.1 (1) Accumulation Term

$$\begin{aligned}
V_{ijk}\Delta\left(\phi N_W\right)_{ijk}^{n+1} &= V_{ijk}\left[\left(\phi\frac{\partial N_W}{\partial N_O}\right)_{ijk}^{K}\Delta N_{O,ijk}^{n+1} + \left(\phi\frac{\partial N_W}{\partial N_G}\right)_{ijk}^{K}\Delta N_{G,ijk}^{n+1}\right. \\
&\qquad \left. + \left(c_\phi\phi_z N_W + \phi\frac{\partial N_W}{\partial P_w}\right)_{ijk}^{K}\Delta P_{w,ijk}^{n+1}\right] \qquad (47)\\
V_{ijk}\Delta\left(\phi N_G\right)_{ijk}^{n+1} &= V_{ijk}\left(\phi_{ijk}\Delta N_{G,ijk}^{n+1} + c_\phi\phi_{z,ijk}N_{G,ijk}^{K}\Delta P_{w,ijk}^{n+1}\right) \qquad (48)\\
V_{ijk}\Delta\left(\phi_{ijk} N_O\right)_{ijk}^{n+1} &= V_{ijk}\left(\phi_{ijk}\Delta N_{O,ijk}^{n+1} + c_\phi\phi_{z,ijk}N_{O,ijk}^{K}\Delta P_{w,ijk}^{n+1}\right) \qquad (49)
\end{aligned}$$

3.5.2 (2) Transport Terms

$$\begin{aligned}
&\Delta t^{n+1}V_{ijk}\Delta\left(\nabla\cdot U_M\right)_{ijk}^{n+1} = \\
&\quad -\Delta\lambda_{m,i+1/2,jk}^{n+1}\left[P_{m,i+1,jk}^{K} - P_{m,ijk}^{K} - \rho_{m,i+1/2,jk}^{K}g\left(D_{i+1,jk} - D_{ijk}\right)\right] \\
&\quad +\Delta\lambda_{m,i-1/2,jk}^{n+1}\left[P_{m,ijk}^{K} - P_{m,i-1,jk}^{K} - \rho_{m,i-1/2,jk}^{K}g\left(D_{ijk} - D_{i-1,jk}\right)\right] \\
&\quad -\lambda_{m,i+1/2,jk}^{K}\left[\Delta P_{m,i+1,jk}^{n+1} - \Delta P_{m,ijk}^{n+1} - \Delta\rho_{m,i+1/2,jk}^{n+1}g\left(D_{i+1,jk} - D_{ijk}\right)\right] \\
&\quad +\lambda_{m,i-1/2,jk}^{K}\left[\Delta P_{m,ijk}^{n+1} - \Delta P_{m,i-1,jk}^{n+1} - \Delta\rho_{m,i-1/2,jk}^{n+1}g\left(D_{ijk} - D_{i-1,jk}\right)\right] \\
&\quad + \text{similar terms for the } y \text{ and } z \text{ directions.} \qquad (50)
\end{aligned}$$

Note that component M (W, O, G, respectively) always corresponds to its own phase m (w, o, g, respectively) in the above equation.

3.6 Linear Solver

The Jacobian matrix is sparse, highly non-symmetric and poorly conditioned. A parallel GMRES solver [31, 55, 7] with multi-level preconditioner [27] is used to solve the linear system described by Equation (42).

An LSOR (line-SOR) solver [70, 55, 7] is also available in the IPARS framework. It may have better performance than the GMRES solver for problems of small grid size.

3.7 Flash Calculation

The saturations must be evaluated to get the relative permeabilities and to construct residuals and the Jacobian matrix. The saturations must satisfy the constraint

$$S_w + S_o + S_g = 1 \qquad (51)$$

where S_o and S_g can be expressed as functions of S_w according to the phase properties described in Section (2.4). We can view it as a nonliear equation with S_w as unknown

$$S_w + S_o(P_w + P_{cow}(S_w)) + S_g(P_w + P_{cow}(S_w) + P_{cgo}(1 - S_w - S_o)) = 1 \qquad (52)$$

The nonlinearity is introduced because of capillary pressures. With given primary unknowns P_w, N_O and N_G, we can solve for S_w, and S_o and S_g follow. For convenience, we call this procedure the *flash* calculation. This term is introduced from the phase equilibrium calculations in the compositional model [33, 49, 48, 40, 43, 62, 10].

Four methods, Ridder's method [53], bisection method, a combination of bisection-Newton method and the interior point method [13], have been implemented to solve this equation. Since the *flash* calculation needs to be done in every permeable grid cell, the speeds of the methods are important. Figure 1 is a comparison of the four methods for 2-phase and 3-phase cases respectively.

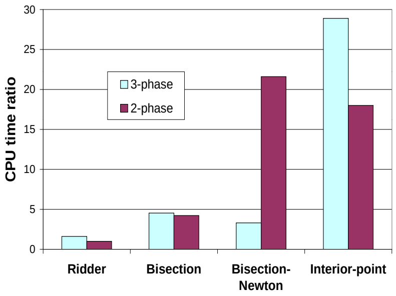

Figure 1: Comparison of the speed of flash calculations

The Ridder's method is usually the fastest one, and the interior point method is normally the slowest. However, the latter method can be extended for the flash calculation in the compositional model, which will be implemented in IPARS in the future. With Ridder's method, the CPU time for the *flash* calculation counts 5% - 15% of the total CPU time for moderate or large grid size problems, while most of the remaining CPU time is spent in the linear solver.

4 Initialization

The problem (1)-(3) is transient and therefore it is necessary to define an initial condition of the reservoir. The black-oil model needs to initialize its primary unknowns as well as its auxiliary variables. In general, there are two approaches to initialization: the first one assumes that the reservoir is in hydrostatic equilibrium under the given pressure and fluid content conditions from which the values of primary unknowns are computed; this is the default option. The second option allows for ad-hoc values of the primary unknowns. While the latter is easier to implement and while it makes comparison and validation studies of different models and simulators easier, it assigns the responsibility of defining a physically meaningful problem to the user. On the other hand, implementation of hydrostatic equilibrium may be tedious but its advantage is that it initiates the model in a manner consistent with the physics and therefore the problem is consequently easier to solve.

4.1 Initial Hydrostatic Equilibrium

4.1.1 Input

The following data is input:

- DINIT (depth D^{ref}, default=0.0, unit=[ft], optional)

- PWINIT (water pressure P_w^{ref} at DINIT, unit [psi], no defaults, required)
- SWINIT (water saturation S_w^{ref} at DINIT, no defaults, required)
- GOC (depth of gas oil contact D^{GOC}, no defaults, unit=[ft], optional)
- SGINIT (gas saturation S_g^{ref} at Dinit, default=0.0, optional)
- SATALL (default=true)
- SATTOP (default=false)
- GSUB, DGFSUB (default=unspecified)
- ROD (table of Ro vs depth, no defaults, optional)

4.1.2 Algorithm

It is assumed that the initial saturation and pressure distributions of the reservoir are obtained from hydrostatic equilibrium (capillary-gravity equilibrium) [23, 28]. This equilibrium is uniquely determined by the values of several parameters at a reference depth D^{ref}. These parameters are P_w^{ref}, S_o^{ref}. Additionally S_g^{ref} or possibly also some additional information about the depth at which gas-oil contact GOC (where S_g changes from zero to positive) occurs are specified.

Dead-oil case. In order to briefly describe the initialization algorithm, we consider first the case of *dead-oil* or only the two-phase case with water phase and oil phase present and no gas dissolved or in other words with $R_o = 0$ everywhere. In this case, the water phase pressure at any depth D can be calculated from

$$P_w(D) \;=\; P_w^{ref} + \int_{D^{ref}}^{D} \rho_w G \, dh \tag{53}$$

where h increases with depth. Practically, this is done using a finite difference approximation

$$P_w(D) \approx P_w^{ref} + (D - D^{ref})\rho_w^* G$$

where water density ρ_w^* is evaluated at some intermediate depth between D and D^{ref}. Then, a local Newtonian iteration is used to resolve the resulting nonlinear problem (if water compressibility is zero, then the problem becomes linear).

Next, we compute P_o^{ref} using P_w^{ref} and using capillary pressure values at S_w^{ref}. An analogous procedure to the one above applies then to oil pressure: since we assume now that the oil phase does not have any gas dissolved, the oil density and P_o can be easily computed and any depth.

Finally, from values of P_w and P_o at any depth D we invert the capillary pressure relationship with $P_{cow}(S_w) = P_o - P_w$ to get S_w at D. Furthermore, $S_g = 0$ from which it follows that $N_g = 0$ at every depth D. This completes initialization of primary unknowns of the model.

General case. The general conditions include cases with gas cap and saturated as well as unsaturated cases and they are far more complicated. Finding the equilibrium involves several nested Newtonian iterations. We describe the logic of the initialization algorithm below.

The input routines allow for specifications of conditions with non-zero gas amount in several mutually exclusive ways. First, GOC can be specified in two ways: explicitly, as value of the depth at which it occurs D^{GOC} or implicitly, as the value of S_g^{ref} at the reference depth. The latter has a default zero value. Next, values of R_o below GOC can be specified with at most one of the keywords listed below. Default option is called SATALL and it is active if none of the other keywords are specified:

- if SATALL is true, then R_o is set to the value of R_{so} at given depth (pressure).
- if SATTOP is specified, then R_o is set to the value of R_{so} at GOC.
- if GSUB, DGFSUB are specified, they they deliver the value of R_o and the depth at which it is specified, respectively. These values are used for linear interpolation between GOC and DGFSUB.
- if table ROD is specified, it gives value of R_o as a function of depth below GOC. Here, if $R_o > R_{so}$ at given pressure, then R_o is set to the value of R_{so}.

Determination of GOC. If GOC is not specified, then we need to compute the depth at which it occurs. If GOC is specified, then S_g^{ref} cannot be specified. Note that if $D^{GOC} \leq D^{ref}$ then this implies $S_g^{ref} = 0$. On the other hand, if S_g^{ref} is specified and $S_g^{ref} > 0$, then GOC cannot be specified and D^{GOC} is computed which satisfies $D^{GOC} \leq D^{ref}$. Finally, if neither GOC nor S_g^{ref} are specified or if $S_g^{ref} = 0$, then GOC is assumed to be just above the reservoir. We then set

$$D^{GOC} = \min(D^{MIN}, D^{ref}) - \varepsilon$$

where D^{MIN} is the minimum depth in the reservoir and where $\varepsilon = .1ft$.

Computation of conditions at D^{ref} and at D^{GOC}. At D^{ref} both P_w^{ref} and S_w^{ref} must be specified. P_o^{ref} is computed from capillary pressure data. For S_g^{ref} see above. After GOC is known, then P_w^{GOC} can be computed. Further, oil pressure P_o^{GOC} and density at GOC are computed assuming R_o at GOC equals R_{so}. An additional difficulty arises when the option SATTOP is assumed and if D^{ref} occurs in the two phase region because then both R_o and, in consequence, oil density at D^{ref} are not known. Newton iteration is used to solve these nonlinear relationships.

Pressure of gas P_g^{GOC} is computed from (gas) capillary pressure using $S_g^{GOC} = 0$. Similary, we compute S_w^{GOC}. Pressure of gas P_g^{ref} at D^{ref} can be computed if D^{ref} is in the three phase region. Otherwise its value is irrelevant.

Evaluation of conditions above GOC. At depths $D \leq D^{GOC}$ the code assumes three phase conditions. Phase pressures and densities are computed by iteration with respect to conditions at GOC. Oil phase density is computed for saturated conditions. Water and gas phase saturations are computed from capillary pressure data. Oil saturation is computed from S_w and S_g.

Evaluation of conditions below GOC. At depths $D > D^{GOC}$, the code assumes two phase saturated or undersaturated conditions. The default choice is SATALL which means that the dissolved gas concentration everywhere below GOC is set equal to the saturated value at given pressure. The default can be altered in one of the three ways as explained above. Once R_o is known at given depth, oil phase density and pressure can be obtained by iteration respective to their values at D^{ref}. Water phase pressure and density are computed similarly. Gas pressure and density are irrelevant below GOC. Water and oil saturation values follow from capillary pressure data.

Remarks. Code checks for consistency of the initial data. If initialization fails at any stage then user is informed in the output file regardless of output level.

One can initialize a dead oil case by using constant $R_o = 0$.

4.2 User Specified Initial Values

4.2.1 Input Keywords for User Specified Initialization

- BREADIN (flag): must be present (default value=.false.)
- Water pressure at every cell: use keyword WATERPRES
- Oil saturation at every cell: use keyword SOIL
- Gas saturation at every cell: use keyword SGAS
- RO_ALL = ratio of gas to oil in initialized cells R_o^{all}.

The input follows usual IPARS keyword input rules. For example, to input water pressure in faultblock 1 and layer i=2, use

```
WATERPRES1(2,,)=3020.   $ layer i=2
```

Note: these values MUST be specified otherwise they will have default value zero.

4.2.2 Algorithm

This option is active if the flag $BREADIN$ is specified. The user must provide values of P_w, S_o, S_g for every cell. From these values it is straightforward to compute S_w and concentration and density of water phase / component. Capillary pressure values and consequently P_o and P_g are calculated.

To get oil phase densities as well as gas concentration and densities, we need to know the amount of gas or the gas/oil ratio R_o. In this input option we assume for simplicity that this ratio is constant for all cells and it is called R_o^{all} and input as RO_ALL. If $S_g = 0$ in a cell, then the code will initialize $N_g = R_o^{all} * N_o$. For consistency, if $R_o^{all} \geq R_{so}$ at the given cell pressure, of if $S_g > 0$ at a cell, then the value of R_{so} is used.

5 Wells

5.1 Peaceman's Well Model

The volumetric rate of flow of fluid phase m (w, o, g, respectively) from the wellbore to a grid cell can be expressed as

$$Q_m = \frac{GLKk_{rm}}{\mu_m}\left(P_{wb} - P_m\right) \tag{54}$$

where P_{wb} is the wellbore pressure (hydrostatic pressure), P_m is the pressure of phase m in the grid cell, K is the absolute permeability of the grid cell, L is the length of open wellbore penetrating the grid cell, G is the dimensionless geometric factor given by

$$G = \frac{2\pi}{ln\left(\frac{r_{eq}}{r_w}\right) + s} \tag{55}$$

where s is the skin factor, r_w is wellbore radius, r_{eq} is the equivalent radius of the grid cell center. Assuming that the well penetrates the full thickness of the grid cell, through its center and in

$KINDWELL$	$Type$
1	Water injection well, pressure specified
2	Water injection well, mass rate specified
3	Production well, pressure specified
4	Production well, total mass rate specified
5	Production well, oil mass rate specified
11	Gas injection well, pressure specified
12	Gas injection well, rate specified

Table 1: Well types implemented in IPARS black-oil model.

direction perpendicular to x-y plane, with Peaceman's model [47], r_{eq} can be evaluated by

$$r_{eq} = 0.28 \frac{\left[(\Delta x)^2 \left(\frac{k_y}{k_x}\right)^{1/2} + (\Delta y)^2 \left(\frac{k_x}{k_y}\right)^{1/2} \right]^{1/2}}{\left(\frac{k_y}{k_x}\right)^{1/4} + \left(\frac{k_x}{k_y}\right)^{1/4}} \tag{56}$$

where Δx, Δy are the x and y dimensions of the grid cell, and k_x, k_y are the x and y direction permeabilities.

We can rewrite Equation 54 in mass rate of component M (equals to W, O, G, respectively) in its corresponding phase m (equals to w, o, g, respectively),

5.2 Numerical Treatment of Well Terms

The well terms (source and sink terms) are treated semi-implicitly. That is, the pressure and mobilities in the grid cells containing the well affect in the Jacobian matrix, but the bottom hole pressure and wellbore fluid density are estimated explicitly using values from the previous time step or Newtonian step.

According to John Wheeler's experiences in a 2-phase hydrology model [65], the bottom hole pressure and wellbore fluid densities for a rate specified well should be evaluated only at the first two or three Newtonian steps and then kept constant. Using this approach the required flow rate can be achieved without unnecessary oscillations in Newtonian iterations. The same strategy is used in the black-oil model for stability reasons.

5.3 Well Types

Five kinds of well types have been fully tested in the black oil model. They are pressure and rate specified water injection wells and pressure and rate specified production wells. Well types are specified using keywork KINDWELL() in input files. Table 1 lists all the types and the corresponding KINDWELL values.

6 Input Summary

In this section, we just list some input keywords which are specific only for the black-oil model. Please refer to some relevant IPARS documentation for other information on data input, such as title, simulation time, linear solver parameters, grid definition, reservoir description and transient data input.

A few types of keywords used in black oil model are listed as follows.

1. Flag type: Presence or absence of the variable will be noted. It may not be followed by any data.

2. Numeric type: An integer or real number must be assigned to the variable.

3. Character string: The variable name must be followed by character string entry in quote marks.

4. Array of numeric or character string type: The variable name must be followed by one or more data entries. See the file IPARSv2/doc/reader.txt for the detail.

5. Block input type: It is used for function input. Within a block, x, y data is read along with information on how the x, y data is to be used to identify and specify information. The special keywords, which are valid only inside of the block, include **Interpolation, Extrapolation, Constraint, Derivative, Nodes, Reverse, Data and Pole**. See the file IPARSv2/doc/reader.txt for more detials about each special keyword.

Remarks: In the following keyword description, all pressures are in [psi] unit. we do not provide any examples under each keyword. See Section A.3 for examples.

6.1 Model Specification

- BLACK_OIL_IMPLICIT (flag): Specify black oil model as the physical model. It should not be used together with any other physical model keyword in the same grid block.

6.2 Black Oil Model Control Parameters

The control parameters listed here are mainly nonlinear iteration related. See the relevant linear solver documentation for input of linear solver control parameters.

- CVTOL (real): Convergence tolerance for nonlinear iterations. (fraction, default=1.0e-9, optional)

- MAXNEWT(integer): Maximum number of nonlinear iterations. (no units, default=50, optional)

- FLASH (integer): Choose a flash calculation method. (no units, default=0, optional)

```
FLASH=0 (Ridder's method)
FLASH=1 (bisection method)
FLASH=2 (combination of bisection-newton method)
FLASH=3 (interior point)
```

- FACTOR_RATE (real): Specifies the extrapolation rate for pressure and concentration variables at the beginning of each time step. (fraction, default=1.0, optional)

6.3 Fluid Properties

- STDENO(real): Specify stock tank oil density. (default unit=[lb/ft^3], default value=56, optional)
- STDENW(real): Specify stock tank water density. (default unit=[lb/ft^3], default value=62.3, optional)
- STDENG(real): Specify stock tank gas density. (default unit=[lb/ft^3], default value=0.04228, optional)
- OILVIS(real): Specify oil viscosity. (default unit=[cp], default value=2, optional)
- WATERVIS(real): Specify water viscosity. (default unit=[cp], default value=0.5, optional)
- GASVIS(real): Specify gas viscosity. (default unit=[cp], default value=0.012, optional)

6.4 Rock Properties

- ROCKCMP: Specify rock compressibility. (default unit=[psi^{-1}], default value=5.0e-7, optional)

Please see IPARv2/doc/reader.txt for the input of porosity and permeability.

6.5 Relative Permeability and Capillary Pressure

Relative permeability and capillary pressure curves are read in by framework routines. But for convenience, they are also described here.

- KOSW(i) (block data input): Table of k_{ro} vs S_w for rock type i. (0 $\leq k_{ro}$,$S_w \leq$ 1, dimensionless, no default values, required).
- KWSW(i) (block data input): Table of k_{rw} vs S_w for rock type i. (0 $\leq k_{rw}$,$S_w \leq$ 1, dimensionless, no default values, required).
- KOSG(i) (block data input): Table of k_{ro} vs S_g for rock type i. (0 $\leq k_{ro}$,$S_g \leq$ 1, dimensionless, no default values, required).
- KGSG(i) (block data input): Table of k_{rg} vs S_g for rock type i. (0 $\leq k_{rg}$,$S_g \leq$ 1, dimensionless, no default values, required).
- PCOW(i) (block data input): Table of oil-water capillary pressure P_{cow} vs S_w for rock type i. (0 $\leq S_w \leq$ 1, default unit for P_{cow}=[psi], no default values, required)
- PCGO(i) (block data input): Table of oil-gas capillary pressure P_{cgo} vs S_o for rock type i. (0 $\leq S_o \leq$ 1, default unit for P_{cow}=[psi], no default values, required)

6.6 PVT Data

- BSO(i) (block data input): Table of formation volume factor for gas saturated oil(B_{so}) vs pressure. (default B_{so} unit=$[cf/scf]$, no default values, required)
- BOD(i) (block data input): Table of formation volume factor for dead oil(B_{od}) vs pressure. (default B_{od} unit=$[cf/scf]$, no default values, required)
- RBW(i) (block data input): Table of reciprocal water formation volume factor(RB_w) vs pressure. (default RB_w unit=$[scf/cf]$, no default values, required)
- BG(i) (block data input): Table of gas formation volume factor(B_g) vs pressure. (default B_g unit=$[mscf/scf]$, no default values, required)
- RSO(b) (block data input): Table of dissolved gas/oil ratio for gas saturated oil(R_{so}) vs pressure. (default R_{so} unit=$[mscf-gas/scf-oil]$, no default values, required)
- PSAT(i) (block data input): Table of saturation pressure(P_{sat}) vs dissolved gas/oil ratio(R_o). (default P_{sat} unit=$[psi]$, default R_o unit=$[mscf-gas/scf-oil]$

6.7 Initialization Data

The initialization data is described in Sections 4.1.1 and 4.2.1.

6.8 Well Data

Well data input is taken care of by the framework. Again they are described here just for convenience.

6.8.1 Well Definition

- WELLNAME(t)(character string): Input well name for well t. It is followed by a character string in quote marks. (no units, no defaults, required)
- KINDWELL(t) (integer): Specify the current well type for well t. (no units, no defaults, required)

```
KINDWELL(1)=1 (pressure specified water injection well)
KINDWELL(1)=2 (rate specified water injection well)
KINDWELL(1)=3 (pressure specified production well)
KINDWELL(1)=4 (total mass rate specified production well)
KINDWELL(1)=5 (oil mass rate specified production well)
KINDWELL(1)=11 (pressure specified gas injection well)
KINDWELL(1)=12 (rate specified gas injection well)
```

- WELLTOP(,s,t) (array of real number): Specify the x, y and z location of the top of the interval s in well t. Three values have to be specified for each interval in each well. (default unit=$[ft]$, no default values, required)
- WELLBOTTOM(,s,t) (array of real number): Specify the x, y, and z location of the bottom of the interval s in well t. Three values have to be specified for each interval in each well. (default unit=$[ft]$, no default values, required)

- WELLPQ(t) (block data input): Table of well rate vs time. (default rate unit= $[stb/day]$ for rate specified wells, no default values, required)

6.8.2 Well Output

- WELLOUTKEY (integer): Well history data output key. (no defaults, required)

```
WELLOUTKEY=0 (discard, do not collect)
WELLOUTKEY=1 (print in standard output)
WELLOUTKEY=2 (output to a seperate well file)
WELLOUTKEY=3 (both print in standard output and output to a seperate file)
```

- WELLFILE (character string): Specify the file name for well history output. It is only needed when WELLOUTKEY is equal to 2 or 3. (no defaults, optional)
- WELLBHPOUT (flag): Indicate the output of well bottom hole pressure together with well history data for rate specified wells. (default=false, optional)

6.9 Visualization Data

Variables which can be visualized in black oil model are listed as follows. Please refer to IPARSv2/doc/vis.txt and vis1.txt for more information on visualization input.

VIS_SCL_NAME	Description
POIL	Oil phase pressure
PWAT	Water phase pressure
PGAS	Gas phase pressure
COIL	Oil concentration
CGAS	Gas concentration
CWAT	Water concentration
SOIL	Oil saturation
SGAS	Gas saturation

VIS_VEC_NAME	Description
VEL_WAT	Water flux
VEL_OIL	Oil flux
VEL_GAS	Gas flux

7 Numerical Experiments

The black-oil model discussed here has been validated by comparing its results with the results from the commercial simulator Eclipse-100 [56, 57] and with the analytical results of Buckley-Leverett flow. Furthermore, the parallel efficiency was tested on a parallel PC cluster and superlinear speedup was achieved for many cases [67].

P_{sat} (psi)	R_{so} (scf/stb)	B_{so} (bbl/stb)
4000	1500	1.1200
3600	1390	1.1100
3200	1270	1.0985
2800	1130	1.0870
2400	985	1.0750
2000	828	1.0630
1600	665	1.0510
1200	500	1.0380
800	335	1.0255
400	165	1.0120
14.7	0	1.0000

S_g	k_{rg}	k_{rog}
0	0	1.0
0.04	0.0	0.6
0.1	0.022	0.33
0.2	0.1	0.1
0.3	0.24	0.02
0.4	0.34	0.0
0.5	0.42	0.0
0.6	0.5	0.0
0.7	0.8125	0.0
0.88491	1.0	0.0

Table 2: Left: Reservoir fluid properties from the ninth SPE comparative solution project. Right: Solution gas-oil ratio and formation volume factor of oil phase.

7.1 The Ninth SPE Comparative Solution Project

The input data was from the ninth SPE comparative solution project [32] with grid size of 9000, high degree of permeability heterogeneity and 26 wells. Table 2 contains some of the input data (see the definition of the parameters in Section 2.2). The water and the oil relative permeabilities are the same as those in Figure 5. Other necessary data can be found in the paper [32].

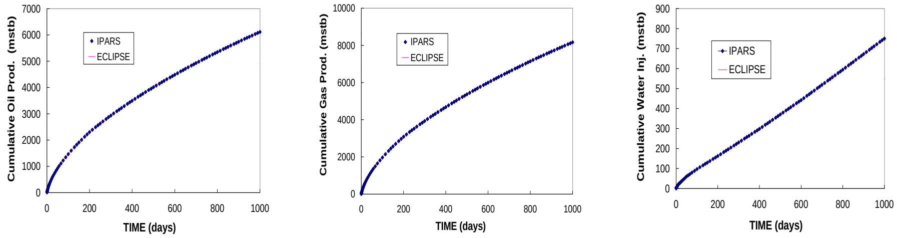

Figure 2: Field cumulative oil and gas produced and water injected (from left to right).

Between IPARS black-oil model and Eclipse, we obtained an almost exact match for cumulative oil and gas production and water injection rate (Figure 2). Excellent match has also been achieved for a single well. Here we show the results for well #21 in Figure 3 with bottom hole pressure specified. The oil and gas component production rates calculated by these two simulators matched nearly perfectly.

However, there was a small discrepancy of water production rate between these two simulators. The reason is that our model uses water pressure to calculate water production rate, while Eclipse uses oil pressure. This discrepancy is detectable only when the difference between oil pressure and water pressure is relatively big; in other words, when capillary pressures are very high or water saturations are very low. Therefore, it happens only when water production rates are very low, note that the water rates shown in Figure 3 are much lower than oil rates, and therefore this effect does not affect the production in the whole reservoir.

We also compared the results of the case with oil rate specified production wells, as shown from

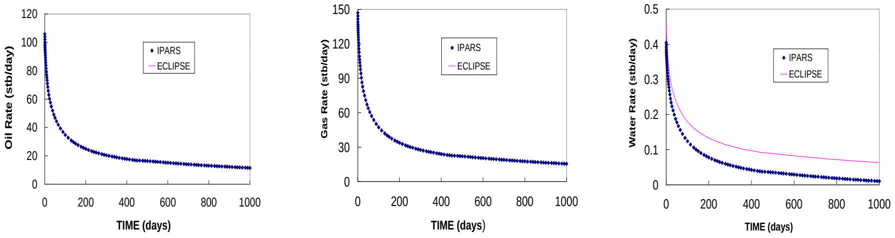

Figure 3: Oil, gas and water production rate in well #21 (bottom hole pressure specified).

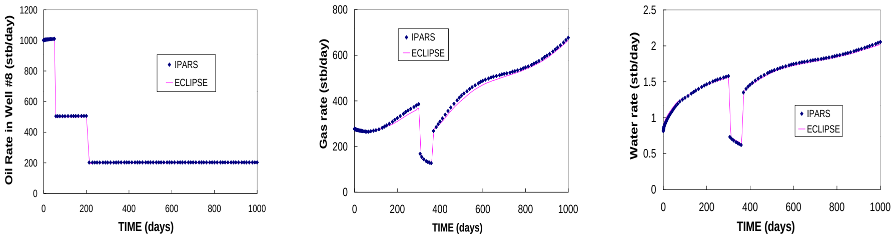

Figure 4: Oil, gas and water production rate in well #8 (oil rate specified).

Figure 4. Still, the comparisons show overall excellent agreement.

7.2 Buckley-Leverett Flow

Buckley-Leverett flow [14, 64, 21, 23] is a standard 1-dimensional example of water-flooding (water phase replacing oil phase) in a core. The flow is assumed to be incompressible, immiscible, and without capillary pressure. Also, the gravity force is neglected. The water front is piston-like and analytical results are available.

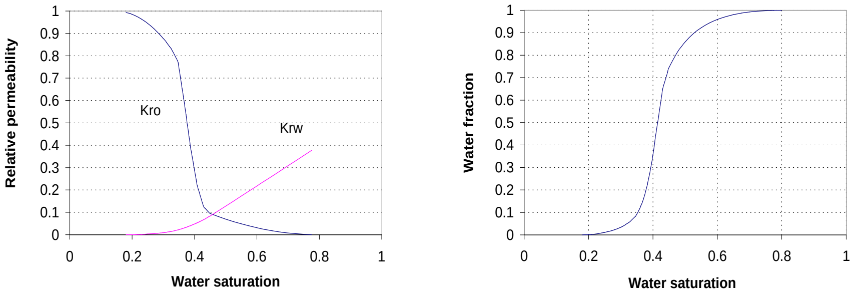

Figure 5: Relative permeability curves used for Buckley-Leverett flow and the associated fractional flow curve.

Given relative permeability curves and the corresponding fractional flow curve shown in Figure 5, we compare the numerical results with the analytic results for the water saturations at the outlet of

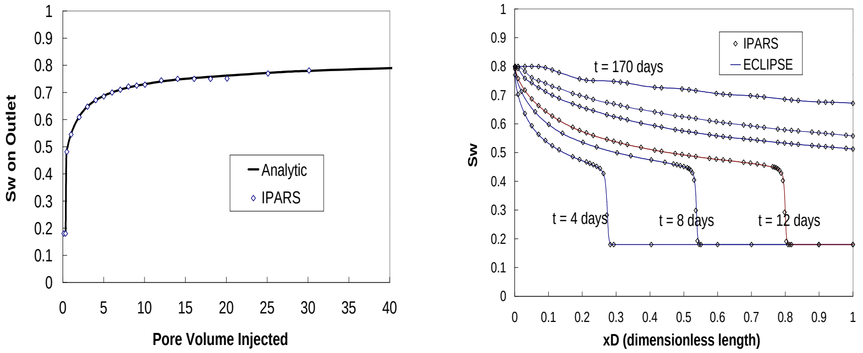

Figure 6: Water saturation at outlet and water saturation distribution for Buckley-Leverett example.

the core, and in the core, as shown in Figure 6, with the results from Eclipse. Excellent agreement has been achieved.

7.3 Parallel Efficiency

The computer platform is a 64-node PC cluster connected by a 1.28 GB/sec Myrinet network, with total 32 GB RAM and over 200 GB collective storage; each node is a 300 MHz Intel Pentium II processor.

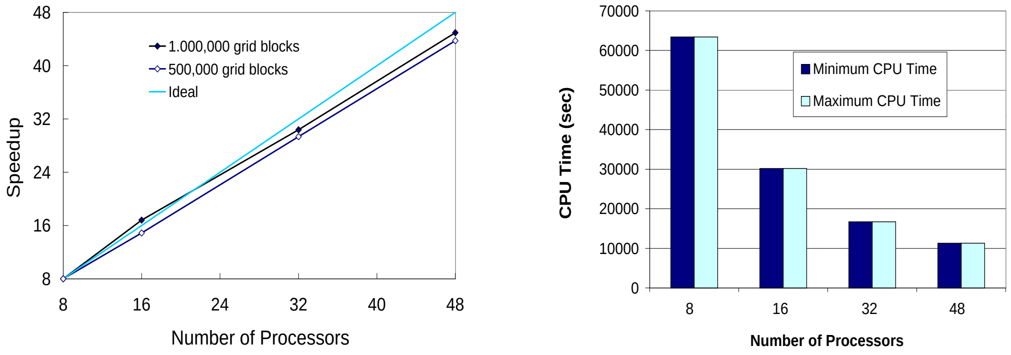

Figure 7: Speedup normalized to 8 processors and load balancing for the case with 1 million grid cells.

As shown in Figure 7, the parallel speedup [45] is pretty close to linear speedup, which is ideal. At 16 nodes, we observed superlinear speedup, this is because of the cache effect (cache is a kind of computer memory whose access rate is much faster than the main memory [45, 58]). For a problem with a fixed grid size, the memory size required for each processor decreases as the number of processors increases. At a certain number of processors (it was 16 in this example), the size of cache is big enough to fit the required memory size, so that no main memory is necessary, then we may get superlinear speedup. As the number of processors further increases, the parallel communication overhead will overcome the cache effects, hence, we can not always maintain superlinear results.

Figure 7 also demonstrates the load balancing of the IPARS implementation. As we can see, the maximum CPU time of a processor is virtually equal to the minimum CPU time of a processor, which means all processors have virtually the same load. This is the key for high parallel efficiency [66].

A Jacobian Formulation for the Implicit Black-Oil Model

All partial derivatives used to construct Jacobian matrix can be extracted from corresponding complete derivatives. Therefore we list all necessary complete derivatives, and all partial derivatives that are needed to calculate them. The definitions of all the denotations are listed in section 2.2.

A.1 Expansions of Complete Derivatives

A.1.1 $\Delta\lambda$ Expansions, Two Phases (Oil and Water)

$$\Delta\lambda^{n+1}_{o,i+1/2.jk} = \frac{\Delta t^{n+1} A_{i+1/2,jk}}{\mu_o B^K_{o,i+1/2,jk}} \left(\begin{array}{l} \left[\frac{dk_{ro}}{dS_w}\frac{\partial S_w}{\partial N_O} - \frac{k_{ro}}{B_o}\frac{\partial B_o}{\partial N_O} \right]^K_{i+1/2,jk} \Delta N^{n+1}_{O,i+1/2,jk} \\ + \left[\frac{dk_{ro}}{dS_w}\frac{\partial S_w}{\partial N_G} - \frac{k_{ro}}{B_o}\frac{\partial B_o}{\partial N_G} \right]^K_{i+1/2,jk} \Delta N^{n+1}_{G,i+1/2,jk} \\ + \left[\frac{dk_{ro}}{dS_w}\frac{\partial S_w}{\partial P_w} - \frac{k_{ro}}{B_o}\frac{\partial B_o}{\partial P_w} \right]^K_{i+1/2,jk} \Delta P^{n+1}_{w,i+1/2,jk} \end{array} \right)$$

$$\Delta\lambda^{n+1}_{w,i+1/2.jk} = \frac{\Delta t^{n+1} A_{i+1/2,jk}}{\mu_w} \left(\begin{array}{l} \left[b_w \frac{dk_{rw}}{dS_w}\frac{\partial S_w}{\partial N_O} \right]^K_{i+1/2,jk} \Delta N^{n+1}_{O,i+1/2,jk} \\ + \left[b_w \frac{dk_{rw}}{dS_w}\frac{\partial S_w}{\partial N_G} \right]^K_{i+1/2,jk} \Delta N^{n+1}_{G,i+1/2,jk} \\ + \left[b_w \frac{dk_{rw}}{dS_w}\frac{\partial S_w}{\partial P_w} + k_{rw}\frac{db_w}{dP_w} \right]^K_{i+1/2,jk} \Delta P^{n+1}_{w,i+1/2,jk} \end{array} \right)$$

Note that up-stream weighting is applied to the items with subscript $i + 1/2$.

Similarly, we can get the expansions on y and z directions.

A.1.2 $\Delta\lambda$ Expansions, Three Phases

$$\Delta\lambda^{n+1}_{o,i+1/2.jk} = \frac{\Delta t^{n+1} A_{i+1/2,jk}}{\mu_o B^K_{o,i+1/2,jk}}$$
$$\left(\begin{array}{l} \left[\frac{\partial k_{ro}}{\partial S_w}\frac{\partial S_w}{\partial N_O} + \frac{\partial k_{ro}}{\partial S_g}\frac{\partial S_g}{\partial N_O} - \frac{k_{ro}}{B_{so}}\frac{\partial B_{so}}{\partial N_O} \right]^K_{i+1/2,jk} \Delta N^{n+1}_{O,i+1/2,jk} \\ + \left[\frac{\partial k_{ro}}{\partial S_w}\frac{\partial S_w}{\partial N_G} + \frac{\partial k_{ro}}{\partial S_g}\frac{\partial S_g}{\partial N_G} - \frac{k_{ro}}{B_{so}}\frac{\partial B_{so}}{\partial N_G} \right]^K_{i+1/2,jk} \Delta N^{n+1}_{G,i+1/2,jk} \\ + \left[\frac{\partial k_{ro}}{\partial S_w}\frac{\partial S_w}{\partial P_w} + \frac{\partial k_{ro}}{\partial S_g}\frac{\partial S_g}{\partial P_w} - \frac{k_{ro}}{B_{so}}\frac{\partial B_{so}}{\partial P_w} \right]^K_{i+1/2,jk} \Delta P^{n+1}_{w,i+1/2,jk} \end{array} \right)$$

$$\Delta\lambda_{w,i+1/2.jk}^{n+1} = \frac{\Delta t^{n+1} A_{i+1/2,jk}}{\mu_w} \begin{pmatrix} \left[b_w \frac{dk_{rw}}{dS_w}\frac{\partial S_w}{\partial N_O}\right]_{i+1/2,jk}^{K} \Delta N_{O,i+1/2,jk}^{n+1} \\ + \left[b_w \frac{dk_{rw}}{dS_w}\frac{\partial S_w}{\partial N_G}\right]_{i+1/2,jk}^{K} \Delta N_{G,i+1/2,jk}^{n+1} \\ + \left[b_w \frac{dk_{rw}}{dS_w}\frac{\partial S_w}{\partial P_w} + k_{rw}\frac{db_w}{dP_w}\right]_{i+1/2,jk}^{K} \Delta P_{w,i+1/2,jk}^{n+1} \end{pmatrix}$$

$$\Delta\lambda_{g,i+1/2.jk}^{n+1} = \frac{\Delta t^{n+1} A_{i+1/2,jk}}{\mu_g B_{g,i+1/2,jk}^{K}} \begin{pmatrix} \left[\frac{dk_{rg}}{dS_g}\frac{\partial S_g}{\partial N_O} - \frac{k_{rg}}{B_g}\frac{\partial B_g}{\partial N_O}\right]_{i+1/2,jk}^{K} \Delta N_{O,i+1/2,jk}^{n+1} \\ + \left[\frac{dk_{rg}}{dS_g}\frac{\partial S_g}{\partial N_G} - \frac{k_{rg}}{B_g}\frac{\partial B_g}{\partial N_G}\right]_{i+1/2,jk}^{K} \Delta N_{O,i+1/2,jk}^{n+1} \\ + \left[\frac{dk_{rg}}{dS_g}\frac{\partial S_g}{\partial P_w} - \frac{k_{rg}}{B_g}\frac{\partial B_g}{\partial P_w}\right]_{i+1/2,jk}^{K} \Delta P_{w,i+1/2,jk}^{n+1} \end{pmatrix}$$

Note that up-stream weighting is applied to the items with subscript $i + 1/2$.

Similarly, we can get the expansions on y and z directions.

A.1.3 ΔP_o Expansions, Two or Three Phases

$$\Delta P_{o,ijk}^{n+1} = \begin{pmatrix} \left[\frac{dP_{cow}}{dS_w}\frac{\partial S_w}{\partial N_O}\right]_{ijk}^{K} \Delta N_{O,ijk}^{n+1} \\ + \left[\frac{dP_{cow}}{dS_w}\frac{\partial S_w}{\partial N_G}\right]_{ijk}^{K} \Delta N_{G,ijk}^{n+1} \\ + \left[1 + \frac{dP_{cow}}{dS_w}\frac{\partial S_w}{\partial P_w}\right]_{ijk}^{K} \Delta P_{w,ijk}^{n+1} \end{pmatrix}$$

A.1.4 ΔP_g Expansions, Three Phases

$$\Delta P_{g,ijk}^{n+1} = \begin{pmatrix} \left[\frac{dP_{cow}}{dS_w}\frac{\partial S_w}{\partial N_O} + \frac{dP_{cgo}}{dS_g}\frac{\partial S_g}{\partial N_O}\right]_{ijk}^{K} \Delta N_{O,ijk}^{n+1} \\ + \left[\frac{dP_{cow}}{dS_w}\frac{\partial S_w}{\partial N_G} + \frac{dP_{cgo}}{dS_g}\frac{\partial S_g}{\partial N_G}\right]_{ijk}^{K} \Delta N_{G,ijk}^{n+1} \\ + \left[1 + \frac{dP_{cow}}{dS_w}\frac{\partial S_w}{\partial P_w} + \frac{dP_{cgo}}{dS_g}\frac{\partial S_g}{\partial P_w}\right]_{ijk}^{K} \Delta P_{w,ijk}^{n+1} \end{pmatrix}$$

A.1.5 $\Delta\rho_w$ Expansions, Two or Three Phases

$$\Delta\rho_{w,ijk}^{n+1} = \left[\rho_{WS}\frac{db_w}{dP_w}\right]_{ijk}^{K} \Delta P_{w,ijk}^{n+1}$$

A.1.6 $\Delta\rho_o$ Expansions, Two Phases

$$\Delta\rho^{n+1}_{o,ijk} = \left(\begin{array}{l} \left[-\frac{\rho_{GS} R_o}{B_o N_O} - \frac{\rho_o}{B_o}\frac{\partial B_o}{\partial N_O} \right]^K_{ijk} \Delta N^{n+1}_{O,ijk} \\ + \left[\frac{\rho_{GS}}{B_o N_O} - \frac{\rho_o}{B_o}\frac{\partial B_o}{\partial N_G} \right]^K_{ijk} \Delta N^{n+1}_{G,ijk} \\ - \left[\frac{\rho_o}{B_o}\frac{\partial B_o}{\partial P_w} \right]^K_{ijk} \Delta P^{n+1}_{w,ijk} \end{array} \right)$$

A.1.7 $\Delta\rho_o$ Expansions, Three Phases

$$\Delta\rho^{n+1}_{o,ijk} = \left(\begin{array}{l} \left[\left(\rho_{GS}\frac{\partial R_{so}}{\partial P_o} - \rho_o \frac{dB_{so}}{dP_o} \right) \frac{1}{B_{so}} \frac{dP_{cow}}{dS_w} \frac{\partial S_w}{\partial N_O} \right]^K_{ijk} \Delta N^{n+1}_{O,ijk} \\ + \left[\left(\rho_{GS}\frac{\partial R_{so}}{\partial P_o} - \rho_o \frac{dB_{so}}{dP_o} \right) \frac{1}{B_{so}} \frac{dP_{cow}}{dS_w} \frac{\partial S_w}{\partial N_O} \right]^K_{ijk} \Delta N^{n+1}_{G,ijk} \\ + \left[\left(\rho_{GS}\frac{\partial R_{so}}{\partial P_w} - \rho_o \frac{\partial B_{so}}{\partial P_w} \right) \frac{1}{B_{so}} \right]^K_{ijk} \Delta P^{n+1}_{w,ijk} \end{array} \right)$$

A.1.8 $\Delta\rho_g$ Expansions, Three Phases

$$\Delta\rho^{n+1}_{g,ijk} = - \left[\frac{\rho_g}{B_g}\frac{\partial B_g}{\partial N_O} \right]^K_{ijk} \Delta N^{n+1}_{O,ijk} - \left[\frac{\rho_g}{B_g}\frac{\partial B_g}{\partial N_G} \right]^K_{ijk} \Delta N^{n+1}_{G,ijk} - \left[\frac{\rho_g}{B_g}\frac{\partial B_g}{\partial P_w} \right]^K_{ijk} \Delta P^{n+1}_{w,ijk}$$

A.2 Partial Derivatives

The following formulae present the partial derivatives that are needed to calculate the complete derivatives.

A.2.1 Two Phases

$$\begin{aligned}
\frac{\partial S_w}{\partial P_w} &= -\frac{C_1 N_O}{C_3} \\
\frac{\partial S_w}{\partial N_O} &= \frac{-B_o + (C_0 + C_2)\, R_o}{C_3} \\
\frac{\partial S_w}{\partial N_G} &= -\frac{C_0 + C_2}{C_3} \\
\frac{\partial B_o}{\partial P_w} &= -\frac{1}{N_O}\frac{\partial S_w}{\partial P_w} \\
\frac{\partial B_o}{\partial N_O} &= -\frac{1}{N_O}\left(B_o + \frac{\partial S_w}{\partial N_O}\right) \\
\frac{\partial B_o}{\partial N_G} &= -\frac{1}{N_O}\frac{\partial S_w}{\partial N_G} \\
\frac{\partial N_W}{\partial P_w} &= b_w \frac{\partial S_w}{\partial P_w} + S_w \frac{db_w}{dP_w} \\
\frac{\partial N_W}{\partial N_O} &= b_w \frac{\partial S_w}{\partial N_O} \\
\frac{\partial N_W}{\partial N_G} &= b_w \frac{\partial S_w}{\partial N_G}
\end{aligned}$$

where

$$\begin{aligned}
C_0 &= \frac{B^*_{so} - B^*_{do}}{R_{so}} \\
C_1 &= \frac{dB_{do}}{dP_o} - C_0 \frac{R_o}{R_{so}}\frac{dR_{so}}{dP_o} \\
C_2 &= \frac{R_o}{R_{so}}\left(\left.\frac{dB_{so}}{dP_o}\right|_{P_o=P_{sat}} - \left.\frac{dB_{do}}{dP_o}\right|_{P_o=P_{sat}}\right)\frac{dP_{sat}}{dR_o} \\
C_3 &= 1 + C_1 N_O \frac{dP_{cow}}{dS_w}
\end{aligned}$$

A.2.2 Three Phases

$$\begin{aligned}
\frac{\partial S_o}{\partial P_w} &= \frac{N_O}{D_4}\frac{dB_{so}}{dP_o}\left(1 - D_3\frac{dP_{cgo}}{dS_g}\right) \\
\frac{\partial S_o}{\partial N_O} &= \frac{1}{D_4}\left\{B_{so}\left[1 + D_3\left(\frac{dP_{cow}}{dS_w} - \frac{dP_{cgo}}{dS_g}\right)\right] - B_g N_O\left(D_2 B_{so} - D_0 R_{so}\right)\right\} \\
\frac{\partial S_o}{\partial N_G} &= -B_g N_O \frac{D_0}{D_4} \\
\frac{\partial S_g}{\partial P_w} &= \frac{1}{D_4}\left(D_3 - B_g N_O \frac{dR_{so}}{dP_o}\right) \\
\frac{\partial S_g}{\partial N_O} &= \frac{1}{D_4}\left[B_g\left(D_2 B_{so} N_O - D_1 R_{so}\right) - D_3 B_{so}\frac{dP_{cow}}{dS_w}\right] \\
\frac{\partial S_g}{\partial N_G} &= \frac{B_g}{D_4}(1 + D_0 N_O) \\
\frac{\partial S_w}{\partial P_w} &= -\frac{\partial S_o}{\partial P_w} - \frac{\partial S_g}{\partial P_w} \\
\frac{\partial S_w}{\partial N_O} &= -\frac{\partial S_o}{\partial N_O} - \frac{\partial S_g}{\partial N_O} \\
\frac{\partial S_w}{\partial N_G} &= -\frac{\partial S_o}{\partial N_G} - \frac{\partial S_g}{\partial N_G} \\
\frac{\partial P_o}{\partial P_w} &= 1 + \frac{dP_{cow}}{dS_w}\frac{\partial S_w}{\partial P_w} \\
\frac{\partial P_g}{\partial P_w} &= 1 + \frac{dP_{cow}}{dS_w}\frac{\partial S_w}{\partial P_w} + \frac{dP_{cgo}}{dS_g}\frac{\partial S_g}{\partial P_w} \\
\frac{\partial B_{so}}{\partial P_w} &= \frac{dB_{so}}{dP_o}\frac{\partial P_o}{\partial P_w} \\
\frac{\partial R_{so}}{\partial P_w} &= \frac{dR_{so}}{dP_o}\frac{\partial P_o}{\partial P_w} \\
\frac{\partial B_g}{\partial P_w} &= \frac{dB_g}{dP_g}\frac{\partial P_g}{\partial P_w} \\
\frac{\partial N_W}{\partial P_w} &= b_w\frac{\partial S_w}{\partial P_w} + S_w\frac{db_w}{dP_w} \\
\frac{\partial N_W}{\partial N_O} &= b_w\frac{\partial S_w}{\partial N_O} \\
\frac{\partial N_W}{\partial N_O} &= b_w\frac{\partial S_w}{\partial N_G}
\end{aligned}$$

where

$$
\begin{aligned}
D_0 &= \frac{dB_{so}}{dP_o}\frac{dP_{cow}}{dS_w} \\
D_1 &= 1 + D_0 N_O \\
D_2 &= \frac{dR_{so}}{dP_o}\frac{dP_{cow}}{dS_w} \\
D_3 &= \frac{dB_g}{dP_g}\left(N_G - R_{so}N_O\right) \\
D_4 &= 1 + D_3\left(\frac{dP_{cow}}{dS_w} - D_1\frac{dP_{cgo}}{dS_g}\right) - D_2 B_g N_O + D_0 N_O
\end{aligned}
$$

A.3 Example Input File

The following file was created for SPE9 data case. For brevity, we omit specification of wells number 3..26. The complete file is in IPARSv2/data/SPE9.dat.

```
$ This data set is modified from the initial SPE9 so that only bottom
$ hole pressure injection wells and production wells exist
$ The permeability data are in file SPE9.xyperms and SPE9.zperms

TITLE(1)="SPE Comparative Solution Case 9"

DESCRIPTION()=
"GRID ELEMENTS : 5000 ACTIVE ELEMENTS (15 x 24 x 25)"
"DIMENSIONS : 359 ft x 7200 x 7500 ft"
"DATE : 1/OCT/98"

BLACK_OIL_IMPLICIT

TIMEEND = 900.0

$ CONVGERGENCE AND TOLERANCE FOR LINEAR SOLVER

MAXNEWT = 20
CVTOL = .1E-6
LSOL_TOL = 1.0E-4
LSOL_ITMAX = 500
$SORTOL = 1E-4
FLASH = 1

$ the parameter below is for GMRES : controls the number of Gauss-Seidel
$ block pressure solves
N_GS_STEP = 10

$ I/O OPTIONS

OUTLEVEL = 3
WELLOUTKEY = 3  WELLFILE = "SPE9.WEL"

$ FAULT BLOCK AND MESH DATA
```

```
DOWN() = 1 0 0

NX(1) = 15  NY(1) = 24  NZ(1) = 25

DX() = 20 15 26 15 16 14 8 8 18 12 19 18 20 50 100

DY() = 300.

DZ() = 300.

$ FLUID PROPERTIES

STDENW = 62.96305
STDENO = 44.9264   OILVIS = 1.0   WATERVIS = .3  GASVIS = 0.015
STDENG = 0.0702   $ lb/SCF

$ INITIAL CONDITIONS

XYZ111() = 9000.  0.  0.

$ 3-phase
DINIT = 9010.  PWINIT = 3550  SWINIT = 0.1763  SGINIT = 0.3

$ POROSITY

POROSITY1(1,,)  = .087
POROSITY1(2,,)  = .097
POROSITY1(3,,)  = .111
POROSITY1(4,,)  = .16
POROSITY1(5,,)  = .13
POROSITY1(6,,)  = .17
POROSITY1(7,,)  = .17
POROSITY1(8,,)  = .08
POROSITY1(9,,)  = .14
POROSITY1(10,,) = .13
POROSITY1(11,,) = .12
POROSITY1(12,,) = .105
POROSITY1(13,,) = .12
POROSITY1(14,,) = .116
POROSITY1(15,,) = .157


$ PERMEABILITIES

Include SPE9.zperms
Include SPE9.xyperms

$ Rock Properties translated from VIP file by J Wheeler 1 Oct 98

KOSW(1) Block        $ OIL RELATIVE PERMEABILITY VS Sw - SPE #9
  Interpolation Spline2
  Extrapolation Constant
  Constraint  0 At .8
  Derivative  0 At .8
  Constraint  1 At .15
  Derivative  0 At .15
  Nodes  .3473  .3633  .4385
  Data .15      1.0      , 0.178350 0.991590 , 0.203350 0.978830
  0.253350 0.943730
  0.360000 0.700470 , 0.364395 0.642258 , 0.368790 0.584046
  0.370000 0.568020 , 0.380000 0.434980
  0.433450 0.125310 , 0.461390 0.094980 , 0.489320 0.070530
  0.517250 0.051130 , 0.573120 0.024640 , 0.601060 0.016190
  0.656930 0.005940 , 0.712800 0.001590 , 0.80     0.0
EndBlock

KWSW(1) Block        $ WATER RELATIVE PERMEABILITY VS Sw - SPE #9
  Interpolation Spline2
  Extrapolation Constant
  Constraint  0 At .18
  Derivative  0 At .18
  Constraint  1 At   1
  Nodes  .2981  .4617  .8406
  Data 0.18  0.0 , 0.253350 0.000030 , 0.360000 0.010340
  0.364395 0.015548 , 0.368790 0.020756 , 0.370000 0.022190
  0.380000 0.035890 , 0.400000 0.069530 , 0.433450 0.087900
  0.461390 0.104910 , 0.489320 0.123290 , 0.517250 0.143030
  0.573120 0.186590 , 0.601060 0.210380 , 0.656930 0.261900
  0.712800 0.318650 , 0.811110 0.430920 , 0.881490 0.490000 , 1. 1.
EndBlock

KOSG(1) Block        $ OIL RELATIVE PERMEABILITY VS Sg - SPE #9
  Interpolation Spline2
  Extrapolation Constant
  Constraint  1 At  0
  Constraint  0 At .4
  Derivative  0 At .4
  Nodes  .0717  .2565
  Data 0   1. , .04  .6 , .1  .33 , .2  .1 , .3  .02 , .4   0
EndBlock

KGSG(1) Block        $ GAS RELATIVE PERMEABILITY VS Sg - SPE #9
  Interpolation Spline2
  Extrapolation Constant
  Constraint  0 At .04
  Derivative  0 At .04
  Constraint  1 At .85
  Derivative  0 At .85
  Nodes  .2315  .5454  .6349
  Data .04 0 , .1 .022 , .2 .1 , .3 .24 , .4 .34 , .5 .42 , .6 .5
  .7 .8125 , 0.85  1.
EndBlock

PCOW(1) Block        $ WATER-OIL CAPILLARY PRESSURE - SPE #9
  Interpolation Spline3
  Extrapolation Same
  Nodes  .3306  .5015
```

```
$  Derivative  0 At .881490
  Pole   .1232
  Data 0.156470 79.060 , 0.165850 40.010 , 0.178350 27.930
0.203350 20.400 , 0.253350 15.550 , 0.360000  0.463
0.400000 -1.604 , 0.433450 -1.710 , 0.461390 -1.780
0.489320 -1.860 , 0.517250 -1.930 , 0.573120 -2.070
0.601060 -2.130 , 0.656930 -2.260 , 0.712800 -2.380
0.811110 -2.600 , 0.881490 -2.750
EndBlock


PCGO(1) Block        $ GAS-OIL CAPILLARY PRESSURE - SPE #9
  Interpolation Spline3
  Extrapolation Same
  Derivative  0 At .15
  Nodes  .37
  Data .15  3.9 , .3  3.5 , .4  3 , .5  2.5 , .6  2 , .7  1.5 , .8  1
  .9  .5 , .96 .2 , 1.  0
EndBlock

$ NB IPARS units on Rso are Stock Tank Volume Gas/Stock Tank Volume Oil
$ whereas conventional units are SCF/STB
$ **********************************************************************
$ *********  must divide conventional Rs data by 5.615 ft3/bbl   *********
$ **********************************************************************
RSO(1) Block         $ DISSOLVED GAS-OIL RATIO - SPE #9
  Interpolation  Linear
  Extrapolation Same
  Data
$ Unit: (mscf of gas)/(scf of oil)
     14.7     0.0
      400     0.0294
      800     0.0597
     1200     0.089
     1600     0.1184
     2000     0.1475
     2400     0.1754
     2800     0.2012
     3200     0.2262
     3600     0.2476
     4000     0.2671
     4400     0.2871

EndBlock

PSAT(1) Block         $ SATURATION PRESSURE - SPE #9
  Interpolation  Linear
  Extrapolation Same
  Reverse        $ x and y will be reversed in the data block
  Data
$ Unit: (mscf of gas)/(scf of oil)
     14.7     0.0
      400     0.0294
      800     0.0597
     1200     0.089
     1600     0.1184
     2000     0.1475
     2400     0.1754
     2800     0.2012
     3200     0.2262
     3600     0.2476
     4000     0.2671
     4400     0.2871
EndBlock


RBW(1) Block $Reciprocal water formation volume factor
  Interpolation Linear
  Extrapolation Same
  Data
  14.7                 1.0
  500.                 1.001603
  1000.                1.003257
  1500.                1.004914
  2000.                1.006573
  2500.                1.008235
  3000.                1.009900
  3500.                1.01157
  4000.                1.01324
EndBlock

BG(1) Block  $Gas formation volume factor, RCF/SCF
  Interpolation Linear
  Extrapolation Same
  Data
$ Unit: Reser-cf/mscf
      14.7    899.3
       400    33.0
       800    16.5
      1200    11.0
      1600    8.3
      2000    6.6
      2400    5.5
      2800    4.7
      3200    4.1
      3600    3.7
      4000    3.3

EndBlock


$ Units of Bso are RB/STB
BSO(1) Block  $Formation volume factor for saturated oil - Rock type 1
  Interpolation Linear
  Extrapolation Same
  Data
 14.7  1.000
 400   1.012
 800   1.0255
 1200  1.038
 1600  1.051
 2000  1.063
 2400  1.075
 2800  1.087
 3200  1.0985
 3600  1.110
 4000  1.12

EndBlock

$ ****************************************************************
$ NB Following properties have NOT been taken from the VIP file...
$ ****************************************************************

BOD(1) Block $Formation volume factor for dead oil- rock type 1
  Interpolation Linear
  Extrapolation Same
  Data
$ if OILCOMP = .4D-4
 14.7                       1.000
 500.0000000                0.9807751775
 1000.000000                0.9613545537
 1500.000000                0.9423184395
 2000.000000                0.9236593246
 2500.000000                0.9053696394
 3000.000000                0.8874421120
 3500.000000                0.8698695898
 4000.000000                0.8526449800
EndBlock


NUMWELL=26

$ WELL 1  AT J = 24 , K = 25
WELLNAME(1) = "Well #1"
KINDWELL(1) = 1
WELLTOP(1 TO 3,1,1) =  9152  7050  7350
WELLBOTTOM(1 TO 3,1,1) =  9359  7050  7350
WELLPQ(1) Block     $ pressure limited for entire simulation
  Interpolation Linear
  Extrapolation Constant
  Data  0.  4000.
EndBlock

$ pressure limited 0-300 days, rate lim to 360,
$ then pressure limited again

$ WELL 2  AT J = 5 , K = 1
WELLNAME(2) = "Well #2"
KINDWELL(2) = 3
WELLTOP(1 TO 3,1,2) =  9020  1350  150
WELLBOTTOM(1 TO 3,1,2) =  9076  1350  150
```

```
WELLPQ(2) Block
  Interpolation Linear
  Extrapolation Constant
  Data  0. 3350., 200 2000, 420 1200, 500 1000
EndBlock

$$$$$$ the wells numbered 3..26 are not included here for brevity

EndInitial

$ TRANSIENT DATA INPUT BLOCKS

BeginTime    0.
DELTIM = .1 DTIMMUL = 1.2  DTIMMAX = 20.0  TIMOUT = 1. DTIMOUT = 100.

EndTime
```

References

[1] *IPARS: Integrated Parallel Accurate Reservoir Simulator.* http://www.ticam.utexas.edu/CSM/ACTI/ipars.html.

[2] *IPARS: Integrated Parallel Accurate Reservoir Simulator, CVS distribution of IPARSv2.* Center for Subsurface Modeling, TICAM, University of Texas at Austin.

[3] J. W. Amyx, D. M. Bass, and R. L. Whiting. *Petroleum reservoir engineering - physical properties.* McGraw-Hill, 1960.

[4] T. Arbogast, C. N. Dawson, P. T. Keenan, M. F. Wheeler, and I. Yotov. Enhanced cell-centered finite differences for elliptic equations on general geometry. *SIAM J. Sci. Comp.*, 19:404–425, 1998.

[5] T. Arbogast, M. F. Wheeler, and I. Yotov. Logically rectangular mixed methods for flow in irregular, heterogeneous domains. In A. A. Aldama et al., editors, *Computational Methods in Water Resources XI*, pages 621–628, Southampton, 1996. SIAM, Computational Mech. Publ.

[6] T. Arbogast, M. F. Wheeler, and I. Yotov. Mixed finite elements for elliptic problems with tensor coefficients as cell-centered finite differences. *SIAM J. Numer. Anal.*, 34:828–852, 1997.

[7] O. Axelsson. *Iterative solution methods.* Cambridge University Press, 1994.

[8] K. Aziz and A. Settari. *Petroleum reservoir simulation.* Applied Science Publishers, Ltd, London, 1979.

[9] L. E. Baker. Three-phase oil relative permeability correlations. In *the SPE/DOE Enchanced Oil Recovery Symposium*, pages 539–554, Oklahoma, Tulsa, April 1988. SPE 17369.

[10] L. E. Baker, A. C. Pierce, and K. D. Luks. Gibbs energy analysis of phase equilibria. *Soc. Pet. Eng. J.*, 22(5):731–742, 1982.

[11] R. P. Batycky, M. J. Blunt, and M. R. Thiele. A 3d multi–phase streamline simulator with gravity and changing well conditions. In *17th Intl. Energy Agency Coll. Project on Enhanced oil Recovery*, Sydney, Australia, Sep 29 – Oct 2 1996.

[12] L. Bergamaschi, S. Mantica, and G. Manzini. A mixed finite element-finite volume formulation of the black-oil model. *SIAM J. Sci. Comp.*, 20(3):970–997, 1998.

[13] S. Bryant, W. Lee, and M. F. Wheeler. Mathematical optimization approach using an interior-point method for finding phase equlibrium in resevoir simulation. Technical report, Texas instit. for comp. and appl. math., Austin, Texas, 2000 (to appear).

[14] S. E. Buckley and M. C. Leverett. Mechanisms of fluid displacement in sands. *Transactions American Institute of Mining and Metallurgical Engineers*, 146:107–116, 1942.

[15] Y. B. Chang. *Development and application of an equation of state compositional simulator.* PhD thesis, The University of Texas at Austin, Austin, Texas, Aug. 1990.

[16] Y. B. Chang, G. B. Pope, and K. Sepehrnoori. A higher-order finite difference compositional reservoir simulator. *J. Pet. Sci. Eng.*, 5:35–50, 1990.

[17] M. C. H. Chien, S. T. Lee, and W. H. Chen. A new fully implicit sompositional simulator. In *the SPE Reservoir Simulation Symposium*, Dallas, Texas, 1985. SPE 13385.

[18] K. H. Coats. An equation of state compositional model. *SPEJ*, pages 363–376, Oct. 1980.

[19] K. H. Coats. Reservoir simulation: state of the arts. *J. Pet. Tech*, pages 1633–1642, Aug. 1982.

[20] K. H. Coats, L. K. Thomas, and R. G. Pierson. Compositional and black oil reservoir simulator. In *the 13th SPE symposium on reservoir simulation*, San Antonio, Texas, Feb 12-15 1995. SPE 29111.

[21] R. E. Collin. *Flow of fluids through porous media.* Petroleum Publishing, Tulsa, Okla, 1976.

[22] D. A. Collins, L. X. Nghiem, and Grabenstetter. An efficient approach to adaptive-implicit compositional simulation with an equation of state. *SPERE*, pages 259–264, May 1992.

[23] L. P. Dake. *Fundamentals of reservoir engineering.* Elsevier Scientific Publishing Company, Oxford-New York, 1978.

[24] M. Delshad and G. A. Pope. Comparison of the three-phase oil relative permeability models. *Transport in Porous Media*, 4:59–83, 1989.

[25] E. C. Donaldson, G. V. Chilingarian, and T. Y. Yen, editors. *Enhanced oil recovery*, volume I and II. Elsevier Science Publishers B. V., 1985.

[26] M. J. Economides and A. D. Hill. *Petroleum production systems.* Prentice Hall, Englewood Cliffs, New Jersey, 1994.

[27] H. C. Edwards. A parallel multilevel-preconditioned gmres solver for multiphase flow models in the implicit parallel accurate reservoir simulator. Technical report, Texas instit. for comp. and appl. math., Austin, Texas, Feb. 1998.

[28] H. B. Bradley et al., editor. *Petroleum engineering handbook.* Soc. Petrol. Eng., Richardson, Texas, 1987.

[29] L. T. Fussel and D. D. Fussel. A iterative technique for compositional reservoir models. *SPEJ*, page 211, Aug. 1979.

[30] M. Honarpour, L. Kedritz, and A. H. Harvey. *Relative permeability of petroleum reservoir.* CRC Press, Inc., Boca Raton, Florida, 1986.

[31] C. T. Kelly. *Iterative methods for linear and nonlinear equations.* SIAM, Philadelphia, 1995.

[32] J. E. Killough. Ninth SPE comparative solution projection: a reexamination of black-oil simulation. In *the 13th Symposium on Reservoir Simulation*, pages 135–147, San Antonio, Texas, Feb. 1995. SPE 29110.

[33] L. W. Lake. *Enhanced oil recovery.* Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[34] A. S. Lawal. The legacy of Stone: three-phase oil relative permeability resolved for reservoir simulation. In *the SPE Permian Basin Oil and Gas Recovery Conference*, Midland, Texas, March 1992. SPE 23969.

[35] J. Liu. *High-resolution methods for enchanced oil recovery simultion.* PhD thesis, The University of Texas at Austin, Austin, Texas, Aug. 1993.

[36] Q. Lu, M. Peszynska, and M. F. Wheeler. A parallel multi-block black-oil model in multi-model implementation. In *2001 SPE Reservoir Simulation Symposium*, Houston, Texas, 2001. SPE 66359.

[37] Qin Lu. *A Parallel Multi-Block / Multi-Physics Approach for Multi-Phase Flow in Porous Media.* PhD thesis, University of Texas at Austin, Austin, Texas, 2000.

[38] C. C. Mattax and R. L. Dalton. Reservoir simulation. In *SPE Monograph Series, volume 13*, Richardson, Texas, 1990.

[39] W. D. McCain. *The properties of petroleum fluids.* Petroleum Publishing Co., Tulsa, 1973.

[40] J. L. Michelsen. The isothermal flash problem. part I – stability. *Fluid phase Equil.*, 9:1–19, 1982.

[41] R. T. Miffin, J. W. Watts, and A. Weiser. A fully coupled, fully implicit reservoir simulator for thermal and other complex reservoir processes. In *the 11th SPE Reservoir Simulation Symposium*, Anaheim, CA, Feb. 1991. SPE 21252.

[42] S. Minkoff, C. M. Stone, J. G. Arguello, S. Bryant, J. Eaton, M. Peszynska, and M. F. Wheeler. Staggered in time coupling of reservoir flow simulation and geomechanical deformation: Step 1 - one-way coupling. In *1999 SPE Symposium on Reservoir Simulation*, Houston, Texas, 1999. SPE 51920.

[43] L. X. Nghiem, K. Aziz, and Y. K. Li. A robust iterative method for flash calculations using Soave-Redlich-Kwong or the Peng-Robinson equation of state. *Soc. Pet. Eng. J.*, pages 521–530, June 1983.

[44] M. Parashar, J. A. Wheeler, J. C. Browne, G. Pope, K. Wang, and P. Wang. A new generation EOS compositional reservoir simulator: Part II – framework and multiprocessing. In *1997 SPE Reservoir Simulation Symposium*, Houston, Texas, 1997. SPE 37977.

[45] D. A. Patterson and J. L. Hennessy. *Computer architecture : a quantitative approach.* Morgan Kaufmann Publisher. Inc., San Mateo, CA, 1990.

[46] D. W. Peaceman. *Fundamentals of numerical reservoir simulation.* Elsevier Scientfic Publishing Company, Amsterdam-Oxford-New York, first edition, 1977.

[47] D. W. Peaceman. Interpretation of well-block pressure in numerical reservior simulation with non-square grid blocks and anisotropic permeability. *Soc. Pet. Eng. J.*, 23(3):531–543, 1983.

[48] D. Y. Peng and D. B. Robinson. A new two-constant equation of state. *AIChеJ*, 23:137–144, 1977.

[49] D. Y. Peng and D. B. Robinson. A rigorous method for predicting the critical properties of multicomponent systems from an equation of state. *AIChеJ*, 23:137–144, 1977.

[50] M. Peszynska, Q. Lu, and M. F. Wheeler. Coupling different numerical algorithms for two phase fluid flow. In J. R. Whiteman, editor, *MAFELAP Proceedings of Mathematics of Finite Elements and Applications*, pages 205–214, Uxbridge, U.K., 1999. Brunel University.

[51] M. Peszynska, Q. Lu, and M. F. Wheeler. Multiphysics coupling of codes. In L. R. Bentley, J. F. Sykes, C. A. Brebbia, W. G. Gray, and G. F. Pinder, editors, *Computational Methods in Water Resources*, pages 175–182. A. A. Balkema, 2000.

[52] D. K. Ponting, B. A. Foster, P. F. Naccache, M. O. Nicholas, R. K. Pollard, J. Rae, D. Banks, and Walsh S. K. An efficient fully implicit simulator. In *European Offshore Petroleum Conference and Exihibition*, 1980.

[53] W. H. Press, S. A Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipe in C.* Cambridge University Press, second edition, 1996.

[54] T. F. Russell and M. F. Wheeler. Finite element and finite difference method methods for continuous flows in porous media. In R. E. Ewing, editor, *The Mathematics of Reservoir Simulation*, pages 35–106, Philadelphia, 1983. SIAM.

[55] Y. Saad. *Iterative methods for sparse linear systems.* PWS Publishing Company, Boston, MA, 1996.

[56] Schlumberger Technology Corporation. *Eclipse-100, Reference Manual*, 1998.

[57] Schlumberger Technology Corporation. *Eclipse-100, Technical Description*, 1998.

[58] A. Silberschatz and P. B. Galvin. *Operating system concepts.* Addison-Wesley Publishing Company, fourth edition, 1994.

[59] H. L. Stone. Probility model for estimating three-phase relative permeability. *J. Petrol. Technol.*, 20:214–218, 1970.

[60] H. L. Stone. Estimation of three-phase relative permeability and residual data. *J. Canad. Petrol. Technol.*, 12:53–61, 1973.

[61] J. A. Trangenstein and J. B. Bell. Mathematical structure of the black-oil model for petroleum reservoir simulation. *SIAM J. Appl. Math.*, 49(3):749–783, 1989.

[62] P. Wang and J. Baker. A comparison of flash calculations in compositional reservoir simulation. Dallas, Texas, Oct. 1995. the 70th SPE Annual Technical Conference and Exhibition. SPE 30787.

[63] P. Wang, I. Yotov, M. Wheeler, T. Arbogast, C. Dawson, M. Parashar, and K. Sephernoori. A new generation EOS compositional reservoir simulator: Part I – formulation and discretization. In *1997 SPE Reservoir Simulation Symposium*, Houston, Texas, 1997. SPE 37979.

[64] H. J. Welge. A simplified method for computing recovery by gas or water drive. *Transactions American Institute of Mining and Metallurgical Engineers*, 195:91–98, 1952.

[65] J. Wheeler. Personal communication, 1998.

[66] M. Wheeler, M. Peszynska, X. Gai, and O. El-Domeiri. Modeling subsurface flow on PC cluster. In *High Performance Computing 2000*, April 2000.

[67] M. F. Wheeler, M. Peszynska, X. Gai, and O. El-Domeiri. Modeling subsurface flow on pc cluster. In A. Tentner, editor, *High Performance Computing*, pages 318–323. SCS, 2000.

[68] M. F. Wheeler, J. A. Wheeler, and M. Peszynska. A distributed computing portal for coupling multi-physics and multiple domains in porous media. In L. R. Bentley, J. F. Sykes, C. A. Brebbia, W. G. Gray, and G. F. Pinder, editors, *Computational Methods in Water Resources*, pages 167–174. A. A. Balkema, 2000.

[69] D. M. Young and R. T. Gregory. *A survey of numerical mathematics*, volume I. General Publishing Company, Toronto, Canada, 1973.

[70] D. M. Young and R. T. Gregory. *A survey of numerical mathematics*, volume II. General Publishing Company, Toronto, Canada, 1973.

[71] L. C. Young. A generalized compositional approach for reservoir simulation. *SPEJ*, page 727, Oct. 1983.

[72] L. C. Young and K. Hemanth-Kumar. High-performance black oil computation. In *the 11th SPE symposium on reservoir simulation*, Anaheim, CA, Feb 17-20 1991. SPE 21215.