

Appendix G

Compositional Model Theory

The compositional model in IPARS can represent three phases (aqueous, nonaqueous liquid, and nonaqueous vapor) and nc components where nc must not exceed \$MXCOMP+1. Currently, the aqueous component is restricted to the aqueous phase and the nonaqueous components are restricted to the nonaqueous phases. IPARS uses a Peng-Robinson EOS with a volumetric shift parameter to calculate fluid densities. Phase equilibrium requires that the component fugacities be equal in the liquid and vapor phases. Phase split calculations for a single phase can be performed using a stability test, borrowing K-values and doing a negative flash, or borrowing K-values and doing a regular flash. Fluid viscosities are calculated using a Lohrenz-Bray-Clark correlation.

Some Definitions

S_α = Saturation of phase α , i.e. pore volume fraction occupied by phase α .

ρ_α = Molar density of phase α .

Z_α = Z-factor of phase α .

x_{ic}^α = Mole fraction of component ic in phase α .

f_{ic}^α = Fugacity of component ic in phase α .

φ_{ic}^α = Fugacity coefficient of component ic in phase α , $\varphi_{ic}^\alpha = f_{ic}^\alpha / x_{ic}^\alpha P$.

$K_{ic} = x_{ic}^v / x_{ic}^l$ is the K-value for component ic.

$N_{ic} = \sum_\alpha x_{ic}^\alpha \rho_\alpha S_\alpha$ is the concentration of component ic per unit pore volume (all phases).

$\phi(P)$ = Porosity.

q_{ic} = Molar rate of component ic per unit reservoir volume.

K = Diagonal permeability tensor.

$k_{r\alpha}$ = Relative permeability for phase α .

μ_α = Viscosity of phase α .

P_α = Pressure in phase α .

γ_α = Density gradient of phase α .

D = Depth

R = Universal gas constant, 10.732460 psia-ft³/°R-lbm

v = Vapor fraction. Fraction of total nonaqueous moles that are in the vapor phase.

$S_T = \sum_{\alpha} S_{\alpha}$ is the sum of the saturations which does not sum exactly to one.

Equations for Fluid Flow

Partial differential equations representing component mass balances are:

$$\frac{\partial}{\partial t} [\phi N_{ic}] = - \sum_{\alpha} \nabla \cdot x_{ic}^{\alpha} \rho_{\alpha} U_{\alpha} + q_{ic}$$

where the velocity of phase α is given by

$$U_{\alpha} = - \frac{K k_{r\alpha}}{\mu_{\alpha}} \cdot (\nabla P_{\alpha} - \gamma_{\alpha} \nabla D)$$

Phase pressures are defined by

$$P_{\alpha} = P + P_{c\alpha}$$

where $P_{c\alpha}$ is the capillary pressure and P is the reference pressure. The reference pressure is used for all flash calculations, all well calculations, and all geomechanical calculations. The keyword REFPRES determines which phase pressure is used as the reference pressure.

Porosity is a function defined by

$$\phi = \phi_o \left[1 + c_r (P - P_{ref}) \right]$$

where ϕ_o , c_r , and P_{ref} are functions of location, and are prescribed using the keywords POROSITYn, CRn, and PORPRESn, respectively.

Equations for Molar Volumes and Saturations

The molar volumes ($1/\rho_{\alpha}$) of the nonaqueous phases are given by

$$v_{\alpha} = RT Z_{\alpha} / P$$

where Z_{α} is the Z-factor for phase α . The molar volume for the aqueous phase has the form

$$v_w = v_w^{ref} B_w^o / \left[1 + c_w (P - P^o) \right]$$

where v_w^{ref} is the water molar volume at the surface which is calculated from the density entered using the keyword STDENW, and B_w^o is the formation volume factor at pressure P^o and reservoir temperature. The parameters P^o , c_w , and B_w^o are set by the keywords WATERP, WATCMP, and WATFVF, respectively.

The saturations are calculated from the equations

$$S_w = v_w N_w$$

$$S_l = (1 - v) v_l \sum_{ic=2}^{nc} N_{ic}$$

$$S_v = v v_v \sum_{ic=2}^{nc} N_{ic}$$

where v is the vapor fraction. The saturations calculated above may not sum to one so the program applies a saturation constraint of the form $\sum_{\alpha} S_{\alpha} = 1$, where the convergence tolerance for this equation is set by the keyword TOL_SAT.

Equation for Z-Factors

IPARS uses a Peng-Robinson EOS with a volumetric shift parameter to calculate a Z-factor for each nonaqueous phase. The cubic equation has the form

$$\bar{Z}_{\alpha}^3 - (1 - B_{\alpha}) \bar{Z}_{\alpha}^2 + (A_{\alpha} - 3B_{\alpha}^2 - 2B_{\alpha}) \bar{Z}_{\alpha} - (A_{\alpha} B_{\alpha} - B_{\alpha}^2 - B_{\alpha}^3) = 0$$

where $Z_{\alpha} = \bar{Z}_{\alpha} - C_{\alpha}$. The keyword TOL_ZFAC determines the convergence tolerance for the equation above. Iterations are performed until the change in \bar{Z}_{α} is less than TOL_ZFAC. IPARS chooses the root with the lowest Gibbs free energy whenever multiple roots are encountered in the cubic equation.

The parameters A, B, and C are given by

$$A_{\alpha} = \frac{P}{R^2 T^2} \sum_{ic} \sum_{jc} x_{ic}^{\alpha} x_{jc}^{\alpha} (1 - \delta_{icjc}) \sqrt{a_{ic}(T) a_{jc}(T)}$$

$$B_{\alpha} = \frac{P}{RT} \sum_{ic} x_{ic}^{\alpha} b_{ic}$$

$$C_{\alpha} = \frac{P}{RT} \sum_{ic} x_{ic}^{\alpha} c_{ic}$$

where δ_{icjc} is the binary interaction coefficient, b_{ic} is calculated from the input values for critical pressure and critical temperature, and c_{ic} is calculated from input values for volumetric shift, critical pressure, and critical temperature. The parameter a_{ic} is calculated from the current temperature, critical pressure, critical temperature, and acentric factor. The program generates EOS tables for different values of temperature in order to minimize the calculations involving the generation of the parameters A, B, and C above.

Equation for Vapor Fractions

The Rachford-Rice equation for determining vapor fractions is

$$\sum_{ic} \frac{(K_{ic} - 1)z_{ic}}{1 + (K_{ic} - 1)v} = 0$$

where K_{ic} , and z_{ic} are the K-value and overall nonaqueous mole fraction for component ic , respectively. The fluid is a single-phase liquid when v is zero and is a single-phase vapor when v is one. Iterations are performed on the Rachford-Rice equation until the absolute residual is less than the value entered for the keyword TOL_RR. The negative flash algorithm allows v to be less than zero or greater than one in the equation above. Once v is determined from the Rachford-Rice equation, then the mole fractions in the liquid and vapor phases can be calculated from

$$x_{ic}^l = \frac{z_{ic}}{1 + (K_{ic} - 1)v} \quad \text{and} \quad x_{ic}^v = K_{ic} x_{ic}^l$$

Equations for Fugacities

Phase equilibrium at constant temperature and pressure requires that the component fugacities be equal in each phase, i.e.

$$f_{ic}^v = f_{ic}^l$$

This is a necessary, but not sufficient condition for phase equilibrium. Phase equilibrium at constant temperature and pressure also requires that the Gibbs free energy be a minimum. However, for most reservoir applications it is sufficient to work with equality of fugacities and that is the approach currently taken in IPARS. Rather than solving the equations above for fugacities, IPARS solves for $\ln(K_{ic})$ and works with the fugacity coefficients instead of the fugacities ($f_{ic}^\alpha = x_{ic}^\alpha P \phi_{ic}^\alpha$) where we replace the equation above by

$$\ln(\phi_{ic}^l) - \ln(\phi_{ic}^v) - \ln(K_{ic}) = 0$$

The keyword TOL_FLASH determines the convergence tolerance for the equation above. The logarithm of the fugacity coefficient for the Peng-Robinson EOS can be expressed as

$$\ln(\phi_{ic}^\alpha) = -C_{ic} + \frac{B_{ic}}{B_\alpha} (\bar{Z}_\alpha - 1) - \ln(\bar{Z}_\alpha - B_\alpha) - \frac{A_\alpha}{2\sqrt{2} B_\alpha} \left(\frac{2 \sum_{jc} x_{jc}^\alpha A_{icjc}}{A_\alpha} - \frac{B_{ic}}{B_\alpha} \right) \ln \left(\frac{\bar{Z}_\alpha + (1 + \sqrt{2}) B_\alpha}{\bar{Z}_\alpha + (1 - \sqrt{2}) B_\alpha} \right)$$

where $B_{ic} = Pb_{ic}/RT$, $A_{icjc} = P(1 - \delta_{icjc}) \sqrt{a_{ic}(T)a_{jc}(T)}/R^2T^2$, and $C_{ic} = Pc_{ic}/RT$. The term C_{ic} is not actually required in the calculations above because it is the same for both phases.

Flash Iterations for Phase Equilibrium

All two-phase cells must undergo flash iterations to satisfy the fugacity equations. The flash algorithm assumes that the pressure, temperature and overall compositions are constant, starting values of the K_{ic} are available, and then follows the basic steps below:

- (1) Determine the vapor fraction v using the Rachford-Rice equation
- (2) Calculate new mole fractions for the liquid and vapor phases
- (3) Calculate new residuals for equations involving the fugacity coefficients
- (4) Check new residuals for convergence. Stop if converged
- (5) Calculate the Jacobian of the fugacity coefficient equations and the Rachford-Rice equation where the Jacobian is expanded in terms of $\ln(K_{ic})$ and v
- (6) Eliminate the vapor fraction v from the fugacity coefficient equations using the Rachford-Rice equation
- (7) Factor the Jacobian and solve for new values of $\ln(K_{ic})$
- (8) Stop if the trivial solution is encountered, i.e. all K -values are approximately one
- (9) Return to step (1)

Iterations are performed until the fugacity residuals satisfy

$$\sum_{ic} [\ln(\phi_{ic}^l) - \ln(\phi_{ic}^v) - \ln K_{ic}]^2 / \sum_{jc} [\ln K_{jc}]^2 < TOL_FLASH$$

or

$$\sum_{ic} (\ln K_{ic})^2 < TOL_TRIV.$$

Determining Phase Splits for Single-Phase Fluids

IPARS must check single-phase cells throughout the reservoir every timestep to see if each single-phase cell will split into two phases. This is a computationally intensive process and the program has several options to reduce the computational costs associated with this process. The program tests for phase splits for only selected cells within the reservoir unless the keyword TESTALL is true. When TESTALL is false, the only cells that are tested for phase splits are those single-phase cells that are beside two-phase cells, single-phase cells that contain well completions, single-phase vapor cells that are beside single-phase liquid cells, single-phase liquid cells that are beside single-phase vapor cells, and single-phase cells that are on processor boundaries (when program run in parallel mode).

IPARS has three different techniques for determining when a single-phase fluid will split into two separate phases. The first technique is a stability test presented by M. L. Michelsen in the paper "The Isothermal Flash Problem. Part I. Stability," Fluid Phase Equilibria, 9 (1982), 1-19. This technique uses a tangent plane criterion that involves iterations for an incipient liquid phase and iterations for an incipient vapor phase. Initial K -values for the calculations are generated using the Wilson equation. The stability test can be time consuming, so IPARS contains two other techniques that can be used for calculating phase splits when there are two-phase neighbors adjoining a single-phase cell.

Another technique for determining phase splits borrows K -values values from neighboring two-phase cells and performs normal flash iterations on the fugacity equations to determine a solution. If the vapor fraction becomes one or zero during the iterations the program assumes that the fluid remains a single phase and terminates the iterations. This technique is used when the keyword BORROW_K is true and there is an adjoining two-phase neighbor.

The default technique for determining phase splits in IPARS uses a negative flash described by C. H. Whitson and M. L. Michelsen in "The Negative Flash," Fluid Phase Equilibria, 53 (1989), 51-71. The first time that a cell is tested for a phase split, the program borrows K -values from a neighboring cell and performs conventional flash iterations where the vapor fraction is allowed to be larger than one or less than zero. If the iterations converge to a solution where the vapor fraction is larger than one or less than zero, then the cell is single-phase for that timestep. However, on subsequent timesteps the calculations do not borrow K -values from neighboring cells, but instead begin at the K -values that were calculated during the previous timestep. Normally the solution changes very slowly between timesteps and a small number of iterations are required for convergence during subsequent timesteps. This technique is used when the keyword NEG_FLASH is true and there is an adjoining two-phase neighbor. This technique also requires that BORROW_K be true.

Surface Separation of Phases

All fluids must be flashed through surface separators to convert component rates into volumetric rates. Each surface separator can have multiple stages. Separator flash calculations assume that the liquid phase from stage i feeds into stage $i+1$ in a separator train. The program assumes ideal gas behavior at surface temperature and surface pressure to determine the volumes of the vapor streams from all stages. The liquid density for the final liquid stream for a separator is based upon the temperature and pressure of the last stage in that separator.

The volumetric rate at the surface is expressed as

$$Q_\alpha = \sum_{ic} C_{ic}^\alpha Q_{ic}$$

where the coefficients C_{ic}^α are calculated by flashing the molar rates Q_{ic} sequentially through the stages in a separator.

Each phase rate is a homogenous function of first order in the component rates so the coefficients that are used in the above equation are the partial derivatives of the phase rates with respect to the component rates, i.e.

$C_{ic}^\alpha = \partial Q_\alpha / \partial Q_{ic}$. This produces a simple expression for the Jacobian when looking at rate changes in the equation above.

IPARS minimizes the number of flash calculations for wells by determining the K-values for each stage of a separator at the beginning of a timestep and keeping the K-values constant throughout the timestep. This requires that a full flash be done at the beginning of each timestep, but only the Rachford-Rice and cubic equations need to be solved during subsequent nonlinear iterations.

Well controls normally involve a prescribed phase rate or a prescribed bottomhole pressure. If a phase rate is prescribed, then IPARS solves the equation above for the appropriate phase using an iterative technique. The well routines are supplied the cell pressures, cell mole fractions, cell mobilities, and cell densities. The well routines then return values for the well coefficients, bottomhole pressure, the wellbore gradient, and the well rates. The solution of the well equation involves an iterative process and the tolerances that control the iterations within the well routines are TOL_PW which sets the convergence tolerance for the bottomhole pressure and TOL_RATE which sets the relative convergence tolerance for the volumetric rate constraint.

IMPEC Implementation

The IMPEC calculations in IPARS assume that all terms on the right hand side of the mass balance equations defined earlier are evaluated at the old timestep except for the fluid pressures. The accumulation term is expanded as

$$Accum_{ic}^{k+1} = \frac{1}{\Delta t} \left(\phi^k N_{ic}^k + N_{ic}^k \delta \phi + \phi^k \delta N_{ic} - \phi^n N_{ic}^n \right)$$

where k is the iteration number for timestep n . Once the term $\delta \phi$ is expanded in terms of the change in pressure, the mass conservation equations can then be rearranged to express the change in N_{ic} in terms of the change in pressures.

The saturation constraint is written as

$$\sum_{\alpha} S_{\alpha}^{k+1} = 1$$

IPARS assumes that the saturations can be expressed in terms of the P , N_{ic} , and $\ln K_{ic}$, and the saturation constraint is expanded as

$$\frac{\partial S_T}{\partial P} \delta P + \sum_{ic} \frac{\partial S_T}{\partial N_{ic}} \delta N_{ic} + \sum_{ic} \frac{\partial S_T}{\partial \ln K_{ic}} \delta \ln K_{ic} = 1 - S_T^k$$

where $S_T = \sum_{\alpha} S_{\alpha}$. The fugacity equations can be expanded in terms of changes in P, N_{ic} , and $\ln K_{ic}$, and can then be

rearranged to express the change in $\ln K_{ic}$ in terms of changes in P, and N_{ic} . This expression for $\ln K_{ic}$ can then be substituted into the saturation constraint leaving only changes in P and N_{ic} . The mass conservations can then be substituted into the saturation constraint to eliminate the changes in N_{ic} which then leaves a single equation in terms of the changes in cell pressures. This equation in pressures can be solved using a standard linear solver. Once the pressures are determined, the new porosities can be calculated and then new concentrations can be determined using the equation

$$N_{ic}^{k+1} = (\Delta t Accum_{ic}^{k+1} + \phi^n N_{ic}^n) / \phi^{k+1}$$

where the accumulation term $Accum_{ic}^{k+1}$ has been calculated from the mass conservation equations using the new pressures. This technique for updating N_{ic} avoids material balance errors that arise due to the product of N_{ic}^{k+1} and ϕ^{k+1} .

IMPEC calculations can become unstable if timesteps are too large during a simulation. IPARS currently uses a saturation-type control to limit timestep sizes for the IMPEC implementation. The saturation change for a component is defined as

$$(\Delta S_T)_{ic} = \frac{\partial S_T}{\partial N_{ic}} \Delta N_{ic}$$

where ΔN_{ic} is the change in concentration for the timestep. The program requires that

$$|(\Delta S_T)_{ic}| \leq DS_{MAX}$$

for all components ic during a timestep. One could calculate a saturation change for each phase and use that as the timestep control rather than using the expression above, but this causes difficulties for simulations where a single phase fluid oscillates between the liquid and vapor phases during a simulation.