Linearization Techniques of Reservoir Simulation Equations - IMPES/IMPEM cases

FARKAS, Éva, MOL Plc Hungary


ABSTRACT

The main object of this paper is to examine and compare models based on the numerical background of linearization techniques. A new transformation procedure is introduced that helps to separate the nonlinearity of the flow equations from the nonlinearity of fluid calculations. This transformation procedure can help to decide an optimal technique when developing a new model. In addition, it clears the real and the apparent differences among the models, and it gives a possibility to arrive at a concise survey of the solution techniques. The detailed examination of the differences of IMPES and IMPEM techniques has revealed the origin of material balance error of the IMPES technique. The object-oriented C++ programming language was applied to the comparison studies. The basic C++ classes and member functions are presented in the Appendix at the end of the paper.

INTRODUCTION

The reservoir simulators and numerical solution techniques of the model equations have developed greatly over the last four decades. In the early years, from the mid-fifties to the end of sixties, black oil modeling techniques were worked out. The direct sequential IMPES technique was developed in this period, and it has been widely used since then. All the coefficient functions of the governing equations of the IMPES model depend on pressure and saturations; thus the primary variables of this solution technique are the pressures and the saturations. The fluid characterization of black oil models depending only on pressure is simple, but suitable to simulate appropriately the conventional displacement technologies. Some problems, however, have arisen when applying the original IMPES method. Such problems are the correct description of the appearance or disappearance of phases, and modeling undersaturated systems when using only saturations as primary variables beside the pressure. To solve these problems the variable substitution is applied, e.g., \( R_s \) is used as primary variable instead of \( S_g \). Another problem is the material balance error. Many papers have dealt with the correct solution of these problems up to now.

The two pseudo hydrocarbon components of black oil models are unsuitable for describing processes in which the compositions of phases change greatly during displacement. Thus, from the early seventies, there has been a demand for the accurate simulation of enhanced oil recovery technologies using compositional and thermal models. In compositional cases the number of equations, and thus the unknown functions can be numerous, and the nonlinearity of governing equations is more severe than in black oil models. The mass (energy) conservation equations are often coupled with highly nonlinear algebraic systems. The coefficient functions of the governing differential equations, beside the pressure and saturations depend also on phase composition. In this case the number of conservation equations i.e. the number of components is greater than the number of phases. The well-proven practice of using black oil primary sets does not work. The saturations can easily be eliminated from the accumulation terms of the mass conservation equation of black oil models as the number of phases is equal to the number of components. In compositional cases, however, the elimination of the saturation and concentration terms from the accumulation term of the conservation equations is not a simple task. Many compositional and thermal models were developed in the seventies, but all models used an iteration technique, even in the IMPES-type case.

Aziz and Wong\(^1\) reviewed various models in detail. Based on the primary sets of these models (i.e. on the Jacobian variables) Aziz and Wong found two major categories of solution methods. One group of methods use pressure, saturation, (temperature), and phase compositions as primary variables. According to their study, this first group stems from a class of models where equilibrium K-values are not composition-dependent. Another group - using pressure, (temperature), and overall system composition - stems from a class of models in which
compositional effects are important. They mentioned, however, that there has certainly been some overlap of techniques and processes.

There are many simplifications when formulating the governing differential equations. Generally, an important simplification of the models is the assumption of an instant and local thermodynamic equilibrium. The solution techniques can be classified according to the calculation of local thermodynamic equilibrium. One group of models is referred to as the "coupled" models. In this type of models the equilibrium calculations are coupled with the flow calculations, and the fluid properties are consistent only at the end of the iteration process of flow calculation. Another type of model is known as the "decoupled" model. In the decoupled models the flow calculations are separated from the equilibrium calculations, and the primary sets are suitable to serve as input data for the pvT or EOS calculations.

The effectiveness of a solution technique essentially depends on two factors: (1) What is the "cost" of one iteration step i.e. the calculation of the Jacobian matrix? (2) How "fast" is the convergence? In the following we examine these questions for 3P IMPES-type black oil models. Three primary sets are applied to simulate 1D water injection. The first primary set (iIMPEM) is an intensive type IMPEM set; this set is equivalent to the primary set of Young and Stephenson. The second primary set is an IMPES one, and the third one is the extensive type IMPEM (eIMPEM) primary set of Acs et al. Wong and Aziz have already compared the intensive and extensive type IMPEM models (Ref. 2. and Ref. 3.) for compositional cases. Our intention now is to compare the nonlinear behaviour of the solutions procedures when using: (1) saturations (compositions); (2) overall composition/densities; (3) extensive properties among the primary variables.

A C++ program was developed for this comparison. The basic C++ classes and member functions of this model are presented in the Appendix.

LINEARIZATION OF THE MODEL EQUATIONS

The discretized form of the model equation is

$$ E_T(X_T) = 0, $$

where the subscript T indicates the total system, i.e. $X_T$ denotes the vector of all thermodynamic properties of all the volume elements we intend to determine. The unknown vector $X_T$ is obtained using an iteration procedure. The unknown variables can be divided into two parts. The first part is called the primary set, and is generally expressed by the Newton method from the flow equations and the saturation/volume constraint as the first step of the solution procedure. Then the additional or secondary variables are derived from the purely algebraic subsystem. Thus

$$ E = \{E_{\Phi}, E_{\Delta} \} \quad \text{and} \quad X_T = \{X, X_A\} $$

can be written, where $E$ denotes the flow and saturation/volume constraint equations, while $X$ denotes the primary set. We assume that

$$ X = \{X_1, \ldots, X, \ldots, X_N\}, $$

where $NE$ is the number of volume elements. We examine the primary sets, our nonlinear system is

$$ F(X) = 0 \quad \text{(1)} $$

We name $X$ the natural or basic variable set of the system, and assume

$$ X = (r_1, \ldots, r_i, \ldots, r_{NC}, \rho) \quad \text{(2)} $$

where $r_i$ is the overall density of component $i$, and $NC$ is the number of the components. Let us select another primary set $Y$. Because of the nonlinear nature of the problem generally $H(X, Y) = 0$. If $X$ can be explicitly expressed from the nonlinear system

$$ G(Y) = X \quad \text{(3)} $$

and

$$ J_G \Delta Y = \Delta X \quad \text{(4)} $$

can be written, where $J_G$ is the Jacobian matrix of the $G$ transformation, and the subscript (superscript) $1$ refers to the iteration level, and for example $\Delta Y = X^{1+l} - X^1$.

For all volume elements

$$ G(X) = Y \quad \text{(5)} $$

We use the Newton method to derive $X$ from Eq. 1,

$$ \Delta X = - J_E^{-1} F(\hat{X}) \quad \text{(6)} $$

Eq. 4 can be substituted into Eq. 6 resulting in equation

$$ \Delta Y = - J_G^{-1} J_E^{-1} F(X) \quad \text{(7)} $$

In Eq. (7) $J_E$ is the Jacobian matrix referring to the original system applying the basic variables. The $J_G$ blockdiagonal matrix is the Jacobian of the variable transformation. The convergence of the Newton method of the original problem depends on the nonlinearity of $J_E$, while the convergence of the Newton method when using the primary set $Y$ depends on the nonlinearity of matrix product $J_E J_G$. 

208
3P Black Oil Equations

A 3P black-oil model is used to illustrate the above transformation. For simplicity, the nonlinearities of the accumulation and transport terms are assumed. The capillary-gravity expressions and rock compressibility are neglected. Only the gas component dissolution in the oil phase is assumed. The gas phase contains a gas component, the water phase contains a water component only. An IMPES-type technique is applied. Hereafter, subscripts of capital letter G, O, and W refer to the components, and the subscripts of lower case letters g, o, and w refer to the phases.

Three primary sets were applied.

iIMPEM method: X (Eq. 2) is an intensive type IMPEM set.

IMPES method: \( Y \) is the conventional primary set of a 3P black-oil model

\[ Y_{\text{IMPES}} = (S_g, S_o, S_w, p) \]

eIMPEM method: \( Y \) is the primary set of the original extensive type IMPEM VB model

\[ Y_{\text{eIMPEM}} = (m_o, m_o, m_w, p) \]

The phases and components and the primary set of the three techniques are shown in Table 1. The saturation/volume constraint equations of the three models are summarized in Table 2.

In the following the difference equations of a volume element will be given for each of the three methods. We use subscript 0 to denote this general volume element, and subscript k (k=1, .. ,M) to denote its neighbors.

iIMPEM (intensive type IMPEM) method:
The mass conservation equations are

\[
F_i^{\text{iIMPEM}} = V_p \frac{\Delta r_{i,0}}{\Delta t} - \sum_{k=1}^{M} \left( \frac{A_K}{A} \sum_{j=1}^{N_p} \frac{k}{d} \frac{C_i}{\rho_j} \right) \rho_j (Q_i^{n+1} - Q_i^{n-1}) - \Delta t Q_i^o = 0, \quad i = G, O, W
\]

We can concisely write the above equations as

\[
F_i^{\text{iIMPEM}} = V_p \frac{\Delta r_{i,0}}{\Delta t} - \Delta t \sum_{k=1}^{M} T_{i,k} (Q_i^{n+1} - Q_i^{n-1}) - \Delta t Q_i^o = 0, \quad i = G, O, W
\]

The saturation constraint equation is:

\[
F_4^{\text{iIMPEM}} = \frac{r_g}{\rho_g} + \frac{r_o}{\rho_o} + \frac{r_w}{\rho_w} - 1 = 0
\]

IMPES method:
The component conservation equations are

\[
F_i^{\text{IMPES}} = V_p \frac{S_i}{R_i} \frac{\rho_{i,0}^{n+1}}{\rho_{i,0} - \Delta t Q_{i,0}^o = 0, \quad i = G, O, W
\]

The saturation constraint equation is:

\[
F_4^{\text{IMPES}} = S_g + S_o + S_w - 1 = 0
\]

eIMPEM (extensive type) method:
The component conservation equations are

\[
F_i^{\text{eIMPEM}} = \Delta m_{i,0} - \Delta t \sum_{k=1}^{M} T_{i,k} (Q_i^{n+1} - Q_i^{n-1}) - \Delta t Q_{i,0}^o = 0, \quad i = G, O, W
\]

The volume constraint equation is:

\[
F_4^{\text{eIMPEM}} = V_f - V_p = 0
\]

A diagonal block of each Jacobian matrix can be seen in Fig. 1 - Fig. 3. Due to the IMPES-type handling of component balance equations, the linearization of the flow terms result in the same terms of the Jacobian matrices for the three methods. An off-diagonal block of the Jacobians is shown in Fig. 4.
We can explicitly express $X$ when transforming the iIMPEM primary variables to IMPES ones.

$$G(Y_{\text{IMPES}}) = X \quad \quad \quad (17)$$

A Jacobian block of this transformation is shown in Fig. 5. If we right multiply the Jacobian matrix $J_G$ of the iIMPEM method (Fig. 1) with the Jacobian matrix $J_F$ of the $G$ transformation (Fig. 5) we obtain the Jacobian matrix of the IMPES method (Fig. 2).

Similarly, if we right multiply the Jacobian matrix of the IMPES method with the inverse Jacobian of the $G$ transformation, we obtain the Jacobian matrix of the iIMPEM method. Here, it is very important to note that $\Delta X$ (masses) can always transform to $\Delta Y_{\text{IMPES}}$ (saturation). The transformation $\Delta Y_{\text{IMPES}}$ set to $\Delta X$ set is valid, however, only in three phase cases.

As for the IMPES primary sets, there are differences not only because $m_i = V_{p} r_i$ or $m_i = V_{f} r_i$, $i=G,O,W$, but also because the iIMPEM model uses saturation constraint, and the eIMPEM model uses volume constraint equations.

SIMULATION RESULTS

The boundary conditions of the 1D simulation were: constant rate water injection at $x=0$, while the original pressure was maintained at the $x=L$, where $L$ was 1000 m. The number of the grid block was 80. Initial pressure was 192 bar, and initial saturations were $S_{g0} = 1$, $S_{o0} = 6$, $S_{w0} = 3$. The initial hydrocarbon pore volume was 450 m$^3$ $m^{-3}$, $S_g = 0.5$, $S_o = 0.25$, and water injection rate was 1 m$^3$/day. The results on Figs. 7 - 12, refer to 270 days when the injected water was 60% of the initial hydrocarbon PV.

Figs. 7, 9, and 11, show results without iterations. Figs. 8, 10, and 12, show results with one iteration. One can see, that only the IMPES results are different in Figs. 7, 9, and 11. After one iteration the three techniques give the same solutions. Nevertheless, only the results of IMPES calculations were different after the iteration; the two IMPEM models gave practically the same results as those resulted in the first step. The results of the IMPES calculations are practically the same, and both methods worked without material balance errors. Yet the two solution techniques differ. The iIMPEM primary variables are dependent, as for isothermal cases, the number of independent properties is only NC. The saturation and volume constraint was not satisfied exactly. However, in this simple example the errors were negligible. The primary variables of the eIMPEM method are independent. During the simulation process the saturation constraints were satisfied, but very small volume balance errors appeared and these were corrected in the subsequent timesteps. The IMPES calculation resulted in material balance errors the saturation and volume constraint equations were satisfied. The above statements are summarised in Table 3.

TRANSFORMATION OF THE PRIMARY SETS

The Jacobian matrix of the IMPES technique from Eq. (7)

$$J_{\text{IMPES}} = J_F J_G \quad \quad \quad (18)$$

We can calculate $\Delta X$ from Eq. 4. The pressure is invariable when applying this transformation and we can get correct masses of the iIMPEM technique, and a corrected saturation set can be calculated from the new masses. After this transformation the saturation and volume constraint equations are no longer satisfied exactly in the first step, as we switch to the iIMPEM method. With this transformation we get the same results as the iIMPEM methods in the first step. (See IMPES to IMPES solution on Figs. 7, 9, and 11.) The additional work is only the back transformation of the primary set by volume elements (Fig. 5). We can give a similar transformation for two phase cases (Fig. 6).

Starting from the iIMPEM Jacobian matrix and transforming $Y_{\text{IMPES}}$ to $X$ after the solution of the system of linear equations, we have the IMPES solution with the material balance errors for one-step solutions.

Summarizing the above, we can obtain iIMPEM results starting from jIMPEM Jacobian matrix or starting from IMPES Jacobian matrix and transforming back the primary set at the end of the IMPES step. The pressures are the same for both cases but the techniques are different with regard to how the saturations (concentrations) are calculated, i.e. whether we calculate $r_i$ from $S_j$ ($i=G,O,W$, $j=G,O,W$), or $S_j$ from $r_i$.

DISCUSSION

The primary set selection of IMPES models has a primarily traditional cause. The problems of IMPES technique come from the original transformation of basic primary set.

The iIMPEM and eIMPEM methods are more effective than the IMPES method, even in black oil case. The calculation of one iteration step requires a similar computing time for all three cases, but the convergence of iIMPEM methods is faster than that of the IMPES method. The fluid properties of the iIMPEM method are consistent only at the end of the iteration process of a timestep. However, sometimes one can get the result without iterations. The fluid properties of the eIMPEM method are always consistent, but numerical errors appear in the
volume balance errors. The timesteps of the eIMPEM technique are chosen in such a way that the volume balance errors should be kept small enough, and the small volume balance errors are corrected in the subsequent timestep.

A transformation technique is introduced to switch one simulation technique to another. One can improve the results of existing IMPES model based on this transformation technique. This modification can be realized by volume elements after the solution of an IMPES timestep. According to our examinations the iIMPEM and IMPES techniques resulted in the same pressure distribution. Thus, another improvement of IMPES technique can be realized, if we use only the pressure of IMPES solution and calculate the component mass changes from the mass balance equations as, in the eIMPEM method. This technique can be convenient for compositional cases, when the Jacobian matrix of the primary set transformation $J_G$ can be very complicated. More recently, a similar technique has been applied by Coats et al.\textsuperscript{5} in their fully implicit compositional model.

The examination of fully implicit models is the next task to be carried out. Then, we can solve a more nonlinear system than in the IMPES/IMPEM cases. The transmissibility derivatives will modify the structure and of the Jacobian matrix, too.

**THE COMPUTER MODEL**

A C++ model has been developed for this comparison. The main classes, together with the member functions, are presented in the Appendix.

**CONCLUSION**

We compare the solution techniques when using "saturation/phase composition" and "overall concentration/density"-type primary sets. Our results, and the good experiences of the last decade using IMPEM models have proved the superiority of "overall concentration/density"-type primary sets in every respect.

The examinations of this paper have cleared up the origin of the material balance errors of the IMPES method. If we consider the nonlinear coupling of Eq. 17, and calculate the component masses from phase saturations applying Eq. 4 (see Fig. 5) there will be no material balance errors. By using this correction, however, we switch from the IMPES technique to an IMPEM one.

The transformation technique presented in this paper can be applied to examine other solution methods, and it may help to develop an optimal linearization technique for reservoir simulators.

**NOMENCLATURES**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>area, $m^2$</td>
</tr>
<tr>
<td>B</td>
<td>formation volume factor, $m^3/m^3$</td>
</tr>
<tr>
<td>C</td>
<td>mass fraction, $kg/kg$</td>
</tr>
<tr>
<td>d</td>
<td>distance between grid points, $m$</td>
</tr>
<tr>
<td>k</td>
<td>relative permeability</td>
</tr>
<tr>
<td>K</td>
<td>absolute permeability, $m^2$</td>
</tr>
<tr>
<td>L</td>
<td>transmissibilities (mass rates, $kg/s.Pa$)</td>
</tr>
<tr>
<td>m</td>
<td>mass, $kg$</td>
</tr>
<tr>
<td>M</td>
<td>the number of neighboring elements</td>
</tr>
<tr>
<td>NC</td>
<td>the number of components</td>
</tr>
<tr>
<td>NE</td>
<td>the number of volume elements</td>
</tr>
<tr>
<td>NJ</td>
<td>the number of Jacobian variables</td>
</tr>
<tr>
<td>NP</td>
<td>the number of phases</td>
</tr>
<tr>
<td>r</td>
<td>overall component density, $kg/m^3$</td>
</tr>
<tr>
<td>Rs</td>
<td>dissolved gas–oil ratio, $m^3/m^3$</td>
</tr>
<tr>
<td>Q</td>
<td>mass source, $kg/s$</td>
</tr>
<tr>
<td>t</td>
<td>time, $s$</td>
</tr>
<tr>
<td>V</td>
<td>volume, $m^3$</td>
</tr>
<tr>
<td>$\mu$</td>
<td>viscosity, $Pa.s$</td>
</tr>
<tr>
<td>$\rho$</td>
<td>density, $kg/m^3$</td>
</tr>
</tbody>
</table>

**Subscripts**

<table>
<thead>
<tr>
<th>Subscript</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>o</td>
<td>general volume element</td>
</tr>
<tr>
<td>f</td>
<td>fluid</td>
</tr>
<tr>
<td>G</td>
<td>gas component</td>
</tr>
<tr>
<td>g</td>
<td>gas phase</td>
</tr>
<tr>
<td>i</td>
<td>component, $i=1,...,NC$</td>
</tr>
<tr>
<td>j</td>
<td>phases, $j=1,...,NP$</td>
</tr>
<tr>
<td>k</td>
<td>neighboring volume element, $k=1,...,M$</td>
</tr>
<tr>
<td>O</td>
<td>oil component</td>
</tr>
<tr>
<td>o</td>
<td>oil phase</td>
</tr>
<tr>
<td>p</td>
<td>pore</td>
</tr>
<tr>
<td>W</td>
<td>water component</td>
</tr>
<tr>
<td>w</td>
<td>water phase</td>
</tr>
</tbody>
</table>

**Superscripts**

<table>
<thead>
<tr>
<th>Superscript</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>iteration</td>
</tr>
<tr>
<td>u</td>
<td>upstream</td>
</tr>
<tr>
<td>n</td>
<td>old time level</td>
</tr>
<tr>
<td>n+1</td>
<td>new time level</td>
</tr>
</tbody>
</table>

**REFERENCES**


APPENDIX

Main classes of the C++ model

// BLACK OIL PARAMETERS
struct black_oil
{  int i;  // identity number
   char name[20];
   float Pb, rho_std[NC];
   float c_bo1, c_bo2, c_bo3, c_bo4;
   float c_bg1, c_bg2, c_bg3, c_bg4, c_rs1, c_rs2;
   // member function
   black_oil(FILE *in, int ii, char *nam);
};

// TRANSMISSIBILITY & CAPILLARITY REGION
struct trans_cap
{  int i;  // identity number
   int type;
   float Sor, Sgr, Swr, Pcg, Pcw, ...
   // member functions
   void init(FILE *in, int ii, int tip, float Sor, float Sgr, float Swr);
   void kr(float *s, float *k);
   void mu(float ppes, float Temp, float *visc);
   void trans(float p, float T, float s[], float krmu[]);
   void cap(float *s);
};

// ROCK PROPERTIES
struct rock
{  int i;  // identity number
   float phii, phi;
   float pi, cr, V;
   // member functions
   void init(mt i, float ff, float c);
   void init2(float pi, float Vpi, FILE *, int);
};

// FLUID SYSTEM
struct fluid
{  int i;  // identity number
   int type;
   black_oil *b;  // connect black oil parameters
   trans_cap *tc;  // connect trans-cap region
   // primary
   float p, T, m[NC], r[NC];
   // secondary
   float Vf, C[NC][NP], ro[NP], S[NP];
   // black_oil
   float bo1, bo2, bw, rs;
   float bop, bgp, bwp, rwp;
   // transmissibilities
   float krmu[NP];
   // changes
   float dpt, dme[NC], dm[NC], dr[NC], ds[NP];
   // member functions
   void init(int i, int tip, float pi, float Ti,
             float *m, rock R, black_oil *b, trans_cap *tc);
   void f(float pres, black_oil *b);
   void lU(FILE *out, int); // calculate the upstream one
};

// WELL CALCULATIONS
struct well{ ...
}

// A JACOBIAN BLOCK
struct JB
{  float J[NJ][NJ];  // J[row][column]
   void clear(float ii);
}

// A BOUNDARY SURFACE
struct boundary
{  int ib;  // identity number
   float A, d, K;
   float AKd;
   fluid *NB1, *NB2, *U;
   // member functions
   void init(int ib, float A, float d, float dz, float K);
   void assign_neighbors(fuid *nbl, fluid *nb2);
   void LU(FILE *out, int); // calculate the upstream composition L
};

// A VOLUME ELEMENT
struct volume_element
{  int ie;  // identity number
   float x, y, z;
   float rv, W;
   fluid *F;
   rock *R;
   int M;
   fluid **N;
   boundary **B;
   JE *JD;
   float residual[NJ];
   float X[NJ];  // primary set
   float deltaX[NJ];
   float Y[NJ];  // transformed primary set
   // member functions
   void connectFRNB(void); // initialization of connections
   void clear_dm_dpt(void);
   void qchang (void); // well calculations
   void get_X(FILE *out, int); // generate the Jacobian blocks
   void get_Y(FILE *out, int); // generate transformation block
};

enum techniques {iIMPEM, from_iIMPEM_to_IMPES, IMPES, from_IMPES_to_iIMPEM, eIMPEM};
Table 1. The masses of components and phases, and the primary variables of the three methods

<table>
<thead>
<tr>
<th>Components</th>
<th>Phases</th>
<th>Primary variables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>gas</td>
<td>oil</td>
</tr>
<tr>
<td></td>
<td>$m_{o,g}$</td>
<td>$m_{o,o}$</td>
</tr>
<tr>
<td></td>
<td>$r_{o} = \frac{m_{o,g}}{V_p}$</td>
<td>$r_{o} = \frac{m_{o,o}}{V_p}$</td>
</tr>
<tr>
<td>Gas</td>
<td>G</td>
<td>O</td>
</tr>
<tr>
<td>Oil</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Water</td>
<td>W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$r_{g} = \frac{m_{o,g}}{V_p}$</td>
<td>$r_{o} = \frac{m_{o,o}}{V_p}$</td>
</tr>
</tbody>
</table>

Table 2. Saturation/volume constraint equations of the methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Constraint equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>iIMPEM</td>
<td>$r_g/\rho_g + r_o/\rho_o + r_w/\rho_w - 1 = 0$</td>
</tr>
<tr>
<td>IMPES</td>
<td>$S_g + S_o + S_w - 1 = 0$</td>
</tr>
<tr>
<td>eIMPEM</td>
<td>$V_r - V_p = 0$</td>
</tr>
</tbody>
</table>

Table 3. Some features of the methods when calculating without iteration

<table>
<thead>
<tr>
<th>Method</th>
<th>Material balance errors</th>
<th>Saturation constraint errors</th>
<th>Volume constraint errors</th>
<th>Variable substitution is needed when gas or oil phases appear/disappear</th>
</tr>
</thead>
<tbody>
<tr>
<td>iIMPEM</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>IMPES</td>
<td>yes</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>eIMPEM</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>
\[ \begin{bmatrix}
V_p & 0 & 0 & -\Delta t \sum_{k=1}^{M} T_{O_k} \\
0 & V_p & 0 & -\Delta t \sum_{k=1}^{M} T_{O_k} \\
0 & 0 & V_p & -\Delta t \sum_{k=1}^{M} T_{W_k} \\
\end{bmatrix} \]

\[ \frac{B_g}{\rho_G} \frac{B_o - R_s B_g}{\rho_G} \frac{B_w}{\rho_G} \frac{r_G}{\rho_G} \frac{\partial B_g}{\partial p} \frac{r_o}{\rho_o} \frac{\partial B_o}{\partial p} \left( \frac{\partial B_g}{\partial p} - \frac{\partial R_s}{\partial p} B_g - \frac{\partial B_o}{\partial p} \right) \frac{r_w}{\rho_w} \frac{\partial B_w}{\partial p} \]

Figure 1. iMPEM method: a diagonal block of Jacobian matrix

\[ \begin{bmatrix}
V_p \frac{\rho_G^{ad}}{B_g} & V_p \frac{\rho_G^{ad}}{B_o} & 0 & V_p \left[ \frac{\partial R_s}{\partial p} B_o - \frac{\partial R_s}{\partial p} B_g \right] \frac{S_o}{B_o} - \frac{\partial B_o}{\partial p} \frac{S_o}{B_o} \frac{\partial B_g}{\partial p} \frac{\rho_G^{ad}}{\rho_G} - \Delta t \sum_{k=1}^{M} T_{O_k} \\
0 & V_p \frac{\rho_G^{ad}}{B_o} & 0 & - \frac{\partial B_o}{\partial p} \frac{S_o \rho_G^{ad}}{B_o \rho_G^{ad}} - \Delta t \sum_{k=1}^{M} T_{O_k} \\
0 & 0 & V_p \frac{\rho_G^{ad}}{B_w} & - \frac{\partial B_w}{\partial p} \frac{S_w \rho_G^{ad}}{B_w \rho_G^{ad}} - \Delta t \sum_{k=1}^{M} T_{W_k} \\
1 & 1 & 1 & 0 \\
\end{bmatrix} \]

Figure 2. IMPES method: a diagonal block of Jacobian matrix

\[ \begin{bmatrix}
1 & 0 & 0 & -\Delta t \sum_{k=1}^{M} T_{O_k} \\
0 & 1 & 0 & -\Delta t \sum_{k=1}^{M} T_{O_k} \\
0 & 0 & 1 & -\Delta t \sum_{k=1}^{M} T_{W_k} \\
\end{bmatrix} \]

\[ \frac{B_g}{\rho_G} \frac{B_o - R_s B_g}{\rho_G} \frac{B_w}{\rho_G} \frac{m_G}{\rho_G} \frac{\partial B_g}{\partial p} \frac{m_o}{\rho_o} \frac{\partial B_o}{\partial p} \left( \frac{\partial B_g}{\partial p} - \frac{\partial R_s}{\partial p} B_g - \frac{\partial B_o}{\partial p} \right) \frac{m_w}{\rho_w} \frac{\partial B_w}{\partial p} \]

Figure 3. eIMPEM method: a diagonal block of Jacobian matrix
\[
J_k = \begin{bmatrix}
0 & 0 & 0 & -\Delta t \ T_{G,k} \\
0 & 0 & 0 & -\Delta t \ T_{O,k} \\
0 & 0 & 0 & -\Delta t \ T_{W,k} \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

Figure 4. An off-diagonal block of Jacobian for iIMPEM, IMPES, and eIMPEM methods; 1D case \( k=1 \) or 2.

\[
J_G \Delta Y_{\text{IMPES}} = \begin{bmatrix}
\frac{\rho_G^{\text{std}}}{B_g} & \frac{\rho_G^{\text{std}}}{B_o} R_s & 0 & \left( \frac{\partial R_B}{\partial p} \frac{\partial B_e}{\partial p} \right) S_o - \frac{\partial B_g}{\partial p} S_g & \frac{S_o}{B_o^2} \rho_G^{\text{std}} & \Delta S_g & \Delta r_G \\
0 & \frac{\rho_O^{\text{std}}}{B_o} & 0 & -\frac{\partial B_o}{\partial p} S_o \rho_O^{\text{std}} & \frac{S_o}{B_o^2} & \Delta S_o & \Delta r_O \\
0 & 0 & \frac{\rho_w^{\text{std}}}{B_w} & -\frac{\partial B_w}{\partial p} S_w \rho_w^{\text{std}} & \frac{S_w}{B_w^2} & \Delta S_w & \Delta r_w \\
0 & 0 & 0 & 1 & \Delta p & \Delta p
\end{bmatrix}
\]

Figure 5. A Jacobian block of IMPES transformation; 3P system

\[
J_G \Delta Y_{\text{IMPES-2P}} = \begin{bmatrix}
\frac{\rho_G^{\text{std}}}{B_o} S_o & \frac{\rho_G^{\text{std}}}{B_o} R_s & 0 & \left( \frac{\partial R_B}{\partial p} \frac{\partial B_e}{\partial p} \right) S_o \rho_G^{\text{std}} & \frac{S_o}{B_o^2} & \Delta R_s & \Delta r_G \\
0 & \frac{\rho_O^{\text{std}}}{B_o} & 0 & -\frac{\partial B_o}{\partial p} S_o \rho_O^{\text{std}} & \frac{S_o}{B_o^2} & \Delta S_o & \Delta r_O \\
0 & 0 & \frac{\rho_w^{\text{std}}}{B_w} & -\frac{\partial B_w}{\partial p} S_w \rho_w^{\text{std}} & \frac{S_w}{B_w^2} & \Delta S_w & \Delta r_w \\
0 & 0 & 0 & 1 & \Delta p & \Delta p
\end{bmatrix}
\]

Figure 6. A Jacobian block of IMPES transformation; 2P/undersaturated system
Figure 7. Gas saturations (no iteration)

Figure 8. Gas saturations (one iteration)

Figure 9. Oil saturations (no iteration)

Figure 10. Oil saturations (one iteration)

Figure 11. Water saturations (no iteration)

Figure 12. Water saturations (one iteration)