Two-Phase Compositional Flow Simulation in Complex Fractured Media by 3D Unstructured Gridding with Horizontal and Deviated Wells

All Zidane and Abbas Firoozabadi, Reservoir Engineering Research Institute

Summary
A higher-order numerical model for compositional two-phase flow in fractured media is presented in this paper. The simulation of horizontal and deviated wells is incorporated in the formation using unstructured grids. All commonly used types of finite elements are accounted for in the algorithm: quadrangular and triangular elements in 2D, and hexahedra, prisms and tetrahedra elements in 3D. The fracture crossflow equilibrium (FCFE) approach is applied to model the flow exchange between the fractures and the matrix. FCFE is combined with the hybridized form of the mass conservative mixed finite element (MHFE) and the higher-order discontinuous Galerkin (DG) method. A computer-aided design (CAD) interface is developed that connects the mesh generator to the CAD software. The interface allows to design, mesh, and incorporate horizontal and deviated wells into the higher-order simulator. The algorithm allows flow simulation in fractures in all ranges of permeability values as opposed to the embedded discrete fracture matrix (EDFM) approach where low permeable fractures affect the accuracy of the results. The efficiency, accuracy, and strengths of the model are demonstrated in comparison to alternatives including the embedded discrete fracture approach in different examples. Detailed incorporation of complex wells is presented in this work.

Introduction
A substantial amount of the hydrocarbon reserves is in the naturally fractured reservoirs. Efficient exploitation of these reservoirs is facilitated using compositional reservoir simulators that have accuracy and computational efficiency.

Modeling of fractured reservoirs is challenging because fractures impose a large range of spatial properties. In addition, fractures may make the geometry more complex. Different approximations have been made in the literature to model fractured reservoirs (Bastian et al. 2000; Bogdanov et al. 2003; Geiger et al. 2004; Martin et al. 2005; Hoteit and Firoozabadi 2008; Reiter et al. 2012; Zidane and Firoozabadi 2014, 2015, 2017; Makedonska et al. 2015; Abushaikha et al. 2015; Ahmed et al. 2015a; Bahranin and Danesh 2014; Chen et al. 2015; Hyman et al. 2015; Nejati et al. 2015). Structured grids may not describe various geometrical complexities. Unstructured gridding is the method of choice to describe complex fractured porous media (Heinemann et al. 1991; Naccache 1997; Karypis and Kumar 1998; Becket et al. 2001, 2006; Usadi et al. 2007; Liu et al. 2009).


One variant of the DFM approach is the EDFM. The EDFM approach was first proposed by Lee et al. (2001) and later adopted by Li and Lee (2008), Hajibeygi et al. (2011), Moinfar et al. (2014), and Yan et al. (2016). In this work, we refer to EDFM and DFM as separate entities. In the DFM models, the matrix domain is discretized to describe the fracture positions. The matrix grid cells conform to the fracture elements. The matrix–fracture exchange fluxes are modeled explicitly (Hui et al. 2018).

In the EDFM models, the fracture elements are embedded in the matrix grid cells, and virtual grids are created to represent the fractures (Yu et al. 2017). Four types of connections are considered in the EDFM models: flow between matrix grid cells and the fracture elements, flow between fracture elements within one fracture, flow between intersecting fracture elements, and flow between fractures and the well (Yu et al. 2017). In EDFM, a structured Cartesian grid may be used even if the fractures have different orientations. This will allow the EDFM model to be implemented in finite difference (FD) simulators. EDFM enables the modeling of irregular fracture geometry in Cartesian grids, but in case of high-saturation contrast, an adaptive grid refinement is required which affects the efficiency (Jiang and Younis 2016; Hui et al. 2018). To overcome the limitations of the DP models, a hybrid model that combines DP and EDFM has been proposed (Amir and Sun 2017; Weirong et al. 2017).

Unstructured grids have gained popularity in flow modeling and simulation (Aavatsmark et al. 1998, 2010; Edwards and Zheng 2010; Lamine and Edwards 2010; Moortgat and Firoozabadi 2016) with many advantages over the Cartesian grids. These include the ease to simulate fractured reservoirs in which fractures have different orientations and incorporation of deviated wells (DWs) in the domain.

In this work, we use the DFM concept as in the past. Our algorithm includes the commonly used finite elements (FEs) in 2D and 3D, namely, quadrangular and triangular elements in 2D and hexahedral, prismatic, and tetrahedral elements (structured and unstructured) in 3D. The fracture aperture is assumed to be small compared to the matrix scale (Noorishad and Mehran 1982; Granet et al. 1998; Martin et al. 2005; Hoteit and Firoozabadi 2008). The fractures are, therefore, represented by a lower dimension compared to the dimension in the matrix domain. We apply the FCFE approach in our DFM mode (Zidane and Firoozabadi 2017). In FCFE, we assume a
constant pressure across the fracture width. This alleviates the need for small elements in the matrix domain near the fractures. The fractures are represented by the edges of the FEs in FCFE. This makes FCFE readily applicable to unstructured grids in 2D and 3D.

In addition to fractures, multiphase flow in vertical and horizontal wells is of general interest in subsurface formation flows. In many cases, the wellbore is deviated and, in some cases, highly inclined (Stanislav et al. 1990). DWS increase the reservoir reach and improve deliverability (Gahari et al. 2011). Wellbore deviation adds more complexity to multiphase flow modeling (Hasan and Kabir 1988). Various studies have been conducted to study the pressure drop and flow patterns in inclined tubes and DWS (Beggs and Brill 1973; Barnea et al. 1985; Hasan and Kabir 1988; Stanislav et al. 1990). There is an extensive use of DWS with high deviation ramps in some formations. The hole deviation could exceed 70°, and the ramp lengths may reach more than 3000 m (Escarón 1983; Kruger 2007). There is a need for reservoir simulation of complex wells and accurate production forecast.

Well modeling is traditionally accomplished through an idealization by Dirac line sources (Aziz and Settari 1979). The flow rate in well elements is calculated through the transmissibility terms on the basis of effective mobility. The pressure gradient used in modeling of flow around the well is the pressure difference between the bottomhole pressure and the average pressure in the gridblock. Collins well elements is calculated through the transmissibility terms on the basis of effective mobility. The pressure gradient used in modeling of flow around the well is the pressure difference between the bottomhole pressure and the average pressure in the gridblock. Collins et al. (1992) presented a model to simulate wellbore dynamics. Nghiem et al. (1992) discussed the important factors that influence the flow around the well.

In compressible two-phase flow, the mass transport equations for component $i$ in an $n_c$-component mixture are given by

$$
\phi \frac{\partial c_i}{\partial t} + \nabla \cdot \left( \sum_j c_i x_{i,j} \mathbf{u}_j \right) = F_i, \quad i = 1, \ldots, n_c \quad \text{in} \quad \Omega \times (0, T),
$$

and

$$
\sum_{i=1}^{n_c} \xi_i = 1, \quad \text{in} \quad \Omega \times (0, T).
$$

The algorithm includes an interface to CAD software and to tetgen (Si 2011) to design and generate complicated domain geometries, fractures, and DWS. The details of well implementation are presented in Appendix A.

Model Description

We briefly outline the essence of the algorithm as follows:

- The mass balances in the matrix domain are discretized using the mass conservative DG method coupled with a slope limiter to remove the nonphysical oscillations (Chavent and Jaffré 1986; Hoteit et al. 2004). The degrees of freedom in DG are the number of nodes at the element level. In addition, DG is a convenient method to capture the discontinuities in phase composition at the element interfaces. In the fracture network, the finite volume (FV) method is used.
- The mass balance equations in the matrix domain are discretized using the mass conservative DG method coupled with a slope limiter to remove the nonphysical oscillations (Chavent and Jaffré 1986; Hoteit et al. 2004). The degrees of freedom in DG are the number of nodes at the element level. In addition, DG is a convenient method to capture the discontinuities in phase composition at the element interfaces. In the fracture network, the finite volume (FV) method is used.
- Quadrangular and triangular elements are used in 2D and hexahedral, prismatic, and tetrahedral elements in 3D.
- The algorithm includes an interface to CAD software and to tetgen (Si 2011) to design and generate complicated domain geometries, fractures, and DWS. The details of well implementation are presented in Appendix A.
- Local thermodynamic equilibrium and phase stability are implemented (Li and Firoozabadi 2012). The phase and volumetric oscillations that could be produced from the two-point flux approximation (TPFA) and the multipoint flux approximation (MPFA). Later, MPFA implementations do not suffer from the induced oscillations (Aavatsmark et al. 2010; Edwards and Zheng 2010; Lamine and Edwards 2010; Sandve et al. 2012). In MHFE, the traces of pressure at the element interfaces and the pressure at the cell centers are calculated. The MHFE is a natural choice for implementation in unstructured gridding (Darlow et al. 1984; Mosé et al. 1994; Hoteit and Firoozabadi 2008; Ackerer and Younes 2008; Younes et al. 2011, 2014, 2015; Zidane et al. 2012, 2014a, 2014b; Zidane and Firoozabadi 2018a, 2018b). The effect of capillary pressure is not included in this work. This is justified by the low interfacial tension (IFT) in compositional two-phase (oil and gas) flow at high pressure.
- The mass balance equations in the matrix domain are discretized using the mass conservative DG method coupled with a slope limiter to remove the nonphysical oscillations (Chavent and Jaffré 1986; Hoteit et al. 2004). The degrees of freedom in DG are the number of nodes at the element level. In addition, DG is a convenient method to capture the discontinuities in phase composition at the element interfaces. In the fracture network, the finite volume (FV) method is used.
- The mass balance equations in the matrix domain are discretized using the mass conservative DG method coupled with a slope limiter to remove the nonphysical oscillations (Chavent and Jaffré 1986; Hoteit et al. 2004). The degrees of freedom in DG are the number of nodes at the element level. In addition, DG is a convenient method to capture the discontinuities in phase composition at the element interfaces. In the fracture network, the finite volume (FV) method is used.

Mathematical Model

For completeness, the main equations that govern the compositional multiphase flow in fractured media are presented in this section. The governing equations in the matrix domain and the fractures are treated separately.

Matrix Domain. In compressible two-phase flow, the mass transport equations for component $i$ in an $n_c$-component mixture are given by
where \( \phi \) denotes the porosity; \( v_x \) is the velocity of Phase \( x \); \( c \) is the overall molar density of the mixture; \( z_i \) and \( F_i \) are the overall mole fraction and the sink/source term of Component \( i \) in the mixture, respectively; \( c_j \) is the molar density of Phase \( x \); \( x_{j,i} \) is the mole fraction of Component \( i \) in Phase \( x \); \( \Omega \) is the computational domain; \( \tau \) denotes the simulation time; and \( n_c \) is the number of components. We neglect diffusion in Eq. 1.

The velocity of Phase \( x \) is given by Darcy's law:

\[
v_x = -\frac{K k_{x,i}}{\mu_x} (\nabla p - \rho_x g), \quad x = o, g,
\]

where \( K \) is the absolute permeability; \( k_{x,i} \), \( \mu_x \), and \( \rho_x \) are the relative permeability, dynamic viscosity, and mass density of Phase \( x \), respectively, with \( \lambda_x = k_{x,i}/\mu_x \); \( p \) is the pressure; and \( g \) is the gravitational acceleration. We use the method from Lohrenz et al. (1964) to describe the phase viscosities.

The pressure equation on the basis of the total volume balance is given by (Acs et al. 1985; Watts 1986)

\[
\phi C_i \frac{\partial p}{\partial t} + \sum_{i=1}^{n_c} \nabla_i \cdot \left( \sum_x c_x x_{j,i} (v_x - v_{x,j}^f) \right) = \sum_{i=1}^{n_c} \nabla_i F_i,
\]

where \( C_i \) is the total compressibility and \( \nabla_i \) is the total partial molar volume of Component \( i \) (Firoozabadi 2015).

The phase equilibrium calculation is on the basis of initial guess from stability analysis. The minimum Gibbs free energy is warranted. From the local thermodynamic equilibrium, the equality of the fugacities of each component in the two phases (oil and gas) is given as follows:

\[
f_{o,j}(T, p, x_{j,o}) = f_{g,j}(T, p, x_{j,g}), \quad i = 1, \ldots, n_c; \quad j = 1, \ldots, n_c - 1.
\]

**Fracture Network.** In the fracture elements, the mass balance equations are given by

\[
\phi \frac{\partial z_i}{\partial t} + \nabla \cdot \left( \sum_x c_x x_{j,i} (v_x - v_{x,j}^f) \right) = F_i,
\]

where \( v_x \) is the velocity across the fracture width and \( v_{x,j}^f \) is the velocity in the direction of fracture length.

The pressure equation in the fractures is given by

\[
\phi C_i \frac{\partial p}{\partial t} + \sum_{i=1}^{n_c} \nabla_i \left\{ \nabla \cdot \left[ \sum_x c_x x_{j,i} (v_x - v_{x,j}^f) \right] - F_i \right\} = 0.
\]

The above equations are integrated along the fracture width.

**Discretization.** We briefly describe the discretization of the flow and transport equations in the following. The total velocity is discretized in the MHFE method. The flux at each interface is evaluated as a function of the traces of the pressure at the interface and the pressure at the center of each FE as follows:

\[
q_{K,E} = z_{K,E} \beta_{K,E} - \sum_{E \in \partial K} \beta_{K,E,E'} \tau_{K,E'} - \gamma_{K,E}.
\]

The coefficients \( z_{K,E} \), \( \beta_{K,E,E'} \), and \( \gamma_{K,E} \) depend on the geometry of the element; the details of the MHFE formulation can be found in previous studies (Chavent et al. 1990; Brezzi and Fortin 1991; Chavent and Roberts 1991; Mosé et al. 1994).

The FV integration of the pressure equation in the matrix domain gives

\[
\phi |K| C_i \frac{\Delta p}{\Delta t} + \sum_{i=1}^{n_c} \nabla_i \left[ \sum_{E \in \partial K} c_x x_{j,i} (v_x + G_x) \cdot n_E - F_i \right] = 0.
\]

And in the fracture network, we obtain

\[
\phi |K| C_i \frac{\Delta p}{\Delta t} + \sum_{i=1}^{n_c} \nabla_i \left[ \sum_{E \in \partial K} c_x x_{j,i} (v_x + G_x) \cdot n_E - |F_i| \right] = 0.
\]

To discretize the mass balance equations, the DG method is used in the matrix domain and the FV method in the fracture network. In DG, the mass balance equations are multiplied by a shape function and integrated over each matrix element \( K \); this gives

\[
\int_K \frac{\partial z_i}{\partial t} \phi_{K,j} + \sum_{E \in \partial K} \left( \sum_x c_x x_{j,i} n_E \phi_{K,j} - \sum_x (c_x x_{j,i}) \nabla \phi_{K,j} - \int_K \phi_{K,j} \right) = 0.
\]

The FV integration over the mass balance equations in the fractures gives

\[
\phi |K| c_x x_{j,i}^{t+1} - c_x x_{j,i}^t + \sum_{E \in \partial K} \left( c_x x_{j,i} x_{j,i}^{t+1} q_{x,j,k} - \nabla \phi_{K,j} \right) = |F_i|.
\]
For the temporal discretization, an explicit time scheme is used in the matrix and an implicit time scheme in the fractures. $\overline{\mathbf{Q}}_{ij}^{\text{fr}}$ in the above equation denotes the exchange flux of Component $i$ in Phase 2 between the fracture and the adjacent matrix elements.

When multiple fractures intersect (Fig. 1), the upstream value at the interface is evaluated using a combination of the conservation of mass and Kirchhoff’s law (Ahmed et al. 2015b; Zidane and Firoozabadi 2017).

**Numerical Examples**

We present five numerical examples. In these examples, the components in the petroleum fluids vary from 2 to 8 in 2D and 3D domains. An Intel® Core™ i5 PC, with 3-GHz central processing unit (CPU) and 4-GB RAM, is used in all runs.

**Example 1: Quarter of Five Spot and Comparison with Finite Element–Finite Volume.** In this example, the effect of mesh refinement is examined in triangular elements in a quarter of a five spot with the fractures presented by Geiger et al. (2009). The domain contains 12 interconnected fractures as shown in Fig. 2. The matrix domain has a permeability of 100 md and 20% porosity. The fracture permeability is 1000 darcies. CO₂ is injected into the formation saturated with an equimolar mixture of ethane and propane. Injection and production wells are located at the origin and the diagonally opposite corner, respectively. Similar to Geiger et al. (2009), six different levels of mesh refinements are used. We show in Fig. 2 the different meshes with the number of elements in each mesh. The effect of mesh refinement is evaluated by computing the $L^2$ norm of error of gas saturation assuming the fine grid solution to be exact. We compare the $L^2$ norm of error in our model with the finite element–finite volume (FEFV) model presented in Geiger et al. (2009). The type of flow (compositional two-phase) in our model is different from the black oil model used in Geiger et al. (2009). However, comparing the $L^2$ norm of gas saturation in both models may give a picture on the overall performance in terms of number of elements. For simplicity, we evaluate the $L^2$ norm of error as a function of the relative FE area (i.e., area of FE divided by total area of the domain). As shown in Fig. 3, the $L^2$ norm of error of the gas saturation in our model is less than that in FEFV model for all levels of refinement. Material balance error (in terms of number of moles) is within a range of $10^{-13}$ in this example and will be discussed in more detail later.

In addition to the $L^2$ norm of error, the CPU time with different meshes in the FCFE and FEFV models is presented in Table 1. The large difference in the CPU time shown in Table 1 is partly related to the computational resources available at the time. However, indications are that our algorithm is faster than the FEFV model. The details of the simulation platform used are provided earlier. For reference, the gas saturation at 35% PVI is shown in Fig. 4. Note that assigning different permeability values across the normal and the lateral sides of the fractures is readily achieved in our algorithm as discussed in Appendix B.

**Example 2: Comparison with EDFM.** A comparison of the proposed algorithm with EDFM approach is presented in this example. The example is originally reported in Moinfar et al. (2014). A mixture of CO₂ and CH₄ is injected in a fractured domain saturated with a three-component oil. The compositions of the injected and reservoir fluids are presented in Table 2. The domain is shown in Fig. 5. The injection well is located at one corner and production is at the opposite corner where the pressure is kept constant. The relevant data are summarized in Table 3. The CPU time in the proposed algorithm for a simulation time of 600 days is 5.1 minutes in an Intel Core-i5 PC, with 3-GHz CPU and 4-GB RAM. In Moinfar et al. (2014), enough information was not provided on the platform used in the simulations; therefore, direct comparison of CPU time might not reflect a fair comparison of performance. Moinfar et al. (2014) report CPU time of 24.6 minutes. In Moinfar et al. (2014), a standard FD modeling was used in a domain that consists of $30 \times 15 \times 1$ matrix grids and 40 fracture grids. In the proposed algorithm, a higher-order discretization model is used. Furthermore, in compositional modeling, flash calculation is performed at all nodes of each FE in addition to flash calculation at the center of the element. In a hexahedral FE, there are nine flash calculations per FE. Added to the cost of DG and MHFE compared with FD, this demonstrates the efficiency of the proposed algorithm. For reference, the gas saturation at 35% PVI is shown in Fig. 6. Note that the same number of gridding is used as in Moinfar et al. (2014).

**Example 3: Comparison with the EDFM/DPDP Hybrid Model.** In this example, a fractured domain is considered that has the same properties of matrix and fractures as in Weirong et al. (2017) with the difference of CO₂ injection in propane instead of water injection in oil. The relevant data are presented in Table 4. The model in Weirong et al. (2017) is a hybrid combination of EDFM and DPDP.
It is known that the EDFM produces inaccurate results when the fractures permeability is low (Tene et al. 2017; Weirong et al. 2017). In the proposed algorithm, all range of fracture permeabilities can be simulated. A set of randomly distributed fractures with different lengths are generated. Impermeable fractures are added to demonstrate this feature in our model. The domain and the fracture network including the impermeable faults are shown in Fig. 7 (fractures shown in blue and impermeable faults in red). Geometric information for all fractures and impermeable faults is provided in Appendix C. CO₂ is injected in the middle of the domain. Production is from the four corners. Location of the injection and production wells is shown in Table 5, and using a mesh of 41,000 elements as a reference solution; further refinement did not affect the results. The different meshes are shown in Fig. 8.

![Fig. 2—Different mesh refinements: Example 1.](image_url)

![Fig. 3—Relative $L^2$ norm error of the gas saturation: Example 1.](image_url)
Table 1—CPU time for different mesh refinements: Example 1.

Table 2—Composition of injected and reservoir fluids (in mole %): Example 2.

Fig. 4—Gas saturation at different PVIs, and the velocity of gas phase at 60% PVI (white vectors): Example 1.
Table 3—Relevant data: Example 2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressure (bar)</td>
<td>69</td>
</tr>
<tr>
<td>Temperature (K)</td>
<td>344</td>
</tr>
<tr>
<td>Porosity (%)</td>
<td>10</td>
</tr>
<tr>
<td>Matrix permeability (md)</td>
<td>1</td>
</tr>
<tr>
<td>Fracture permeability (md)</td>
<td>$8 \times 10^4$</td>
</tr>
</tbody>
</table>

Table 4—Relevant data: Example 3.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix permeability (md)</td>
<td>1</td>
</tr>
<tr>
<td>Fracture permeability (darcies)</td>
<td>500</td>
</tr>
<tr>
<td>Fracture width (ft)</td>
<td>0.0065</td>
</tr>
<tr>
<td>Porosity (%)</td>
<td>5</td>
</tr>
<tr>
<td>Pressure at the bottom (bar)</td>
<td>25</td>
</tr>
<tr>
<td>Temperature (K)</td>
<td>311</td>
</tr>
<tr>
<td>Domain size (ft$^3$)</td>
<td>$1,000 \times 1,000 \times 20$</td>
</tr>
</tbody>
</table>
Example 4: Horizontal Well. This example is inspired from Tang et al. (2018) with some changes. The domain size is $1200 \times 800 \times 100$ m$^3$ and contains one horizontal well crossing the middle of the formation for CO$_2$ injection. The injection well of 5½ in. diameter has a total length of 800 m with 45° inclination. Four production wells are located at the four corners of the domain. The domain with the injection and production wells is shown in Fig. 11. CO$_2$ is injected at a constant rate of 0.1 PV/yr. The compositional data of the domain are shown in Table 6. Simulations were run with three different mesh refinements varying from 5,000 to 23,000 elements (Meshes 1–3). The refined mesh of 75,000 elements (Mesh 4) is a reference solution mesh. The reference mesh is used to calculate the relative $L^2$ norm of error of the gas saturation for the three different meshes. The total number of matrix and fracture elements is shown in Table 7. Fig. 12 reveals that the error drops to less than 5% for the 23,000-element mesh, an acceptable variation for a complicated problem with fully compositional flow. The CPU time in the aforementioned mesh is 7 hours. Note that the CPU time could be reduced by coarsening the well geometry as we discuss in Appendix A. Coarsening the well increases the size of the matrix elements near the well region, thus reduces the CPU time but affects the geometry of the well.

Example 5: Deviated Well. This example is inspired from Artus et al. (2017). The geometry of the boundary is kept the same as that in Artus et al. (2017); the domain dimensions in this example are 1000 $\times$ 1000 $\times$ 200 m$^3$ (Fig. 13a). The domain includes a DW inclined at three different angles in 3D with the same well diameter used as in the previous example. Production is performed at constant pressure at the bottom left and right corners of the domain as shown in Fig. 13a. The domain geometry is shown in Fig. 13a, and the well geometry and initial mesh (see Appendix A for details) are shown in Figs. 13b and 13c, respectively. Fractured and unfractured domains are studied. The fractures have different shapes and dimensions, and some cross the wellbore. Fig. 14 shows the domain with the fractures. The fractures have different shapes, dimensions, and orientations, and some cross the wellbore. The orientation of the fractures from vertical is as follows: for quadrangle fractures, the range assigned varies from 0° to 22.5° (randomly), and in circular shape fractures, the range of angles with vertical varies from 22.5° to 45° (randomly). The domain and fracture properties are the same as in Example 4. Fig. 15 shows the oil recovery with and without fractures, and as expected, the recovery is affected by the fractures. The oil recovery in the fractured media is 10% less than in unfractured media (to 1.5 PVI). The CPU time for a total of 45,000 elements is 13 hours for fractured media and 10.6 hours for unfractured media. The difference in CPU time between fractured and unfractured media demonstrates that the simulation of a fractured media in the proposed algorithm is nearly within the same range of unfractured media. As mentioned earlier, CPU time could be reduced by coarsening the well geometry or adapting a different time discretization scheme. Depending on the simulation purpose, either one or both options could be implemented. Material balance error is within a range of 10$^{-13}$ during the simulation time as shown in Fig. 16.
Fig. 8—Different mesh refinements in matrix and fractures (details of all meshes are shown in Table 5): Example 3.

Fig. 9—Oil recovery with different meshes: Example 3.
Fig. 10—Oil saturation at 45% PVI in (a) the matrix domain and (b) the fracture network: Example 3.

Fig. 11—Domain and location of injection and production wells. Horizontal well crossing the fractures is shown by dashed line: Example 4.

<table>
<thead>
<tr>
<th>Component</th>
<th>Overall Mole Fraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO₂</td>
<td>0.0086</td>
</tr>
<tr>
<td>N₂</td>
<td>0.0028</td>
</tr>
<tr>
<td>C₁</td>
<td>0.4351</td>
</tr>
<tr>
<td>C₂–C₃</td>
<td>0.1207</td>
</tr>
<tr>
<td>C₄–C₅</td>
<td>0.0505</td>
</tr>
<tr>
<td>C₆–C₁₀</td>
<td>0.1328</td>
</tr>
<tr>
<td>C₁₁–C₂₄</td>
<td>0.166</td>
</tr>
<tr>
<td>C₂₅+</td>
<td>0.0735</td>
</tr>
</tbody>
</table>

Table 6—Oil composition: Example 4.
Table 7—Domain properties: Example 4.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix permeability (md)</td>
<td>100</td>
</tr>
<tr>
<td>Porosity (%)</td>
<td>20</td>
</tr>
<tr>
<td>Temperature (K)</td>
<td>403.15</td>
</tr>
<tr>
<td>Pressure at the bottom (bar)</td>
<td>276</td>
</tr>
<tr>
<td>Fracture width (mm)</td>
<td>1</td>
</tr>
<tr>
<td>Fracture permeability (md)</td>
<td>$10^6$</td>
</tr>
</tbody>
</table>

Table 8—Total number of matrix and fracture elements in different meshes used: Example 4.

<table>
<thead>
<tr>
<th>Mesh</th>
<th>Matrix Elements</th>
<th>Fracture Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mesh 1</td>
<td>5,222</td>
<td>812</td>
</tr>
<tr>
<td>Mesh 2</td>
<td>14,390</td>
<td>1,473</td>
</tr>
<tr>
<td>Mesh 3</td>
<td>23,126</td>
<td>2,189</td>
</tr>
<tr>
<td>Mesh 4</td>
<td>76,457</td>
<td>2,516</td>
</tr>
</tbody>
</table>

**Fig. 12**—Relative $L^2$ norm error: Example 4.

**Fig. 13**—(a) Domain geometry, (b) well geometry, and (c) well mesh: Example 5.
Fig. 14—Domain with fractures: Example 5.

Fig. 15—Oil recovery with and without fractures: Example 5.

Fig. 16—Variation of material balance error during the simulation: Example 5.
Conclusions
This paper presents an efficient algorithm for compositional two-phase flow in fractured media with DWs. The model includes all commonly used types of FEs in 2D and 3D. The tools developed in the algorithm allow the generation of DWs in complex 3D geometries with fractures. This feature is a result of the interfaces that has been developed in the CAD framework and coupled to unstructured tetrahedrons. The proposed well discretization scheme preserves the shape of the well circumference at the expense of smaller grids in the immediate well vicinity. The fractures are readily set to impermeable faults due to the implementation of the flux calculation at the interfaces in 2D and 3D. The proposed algorithm is compared with different approaches with various degrees of complexity. Comparison with recent models in DFM and EDFM framework demonstrates the efficiency and accuracy of the proposed algorithm. The work presented in this paper has set the stage of extension to simulation of hydraulic and CO₂ fracturing.

Nomenclature

- \( c \) = overall molar density of the mixture
- \( C_t \) = total compressibility
- \( c_a \) = molar density
- \( E \) = grid edge
- \( f_a \) = fractional flow function
- \( f_i \) = sink/source term
- \( f_{\alpha} \) = fugacity
- \( g \) = gas phase
- \( g \) = gravitational acceleration
- \( i \) = component index
- \( k_{\alpha \beta} \) = relative permeability
- \( K \) = grid element
- \( K \) = absolute permeability
- \( n_c \) = number of components
- \( n_e \) = outward normal vector
- \( o \) = oil phase
- \( p \) = pressure
- \( q_{K E} \) = total flux across edge \( E \) in element \( K \)
- \( q_{r \beta}^{fr} \) = matrix–fracture exchange flux
- \( T \) = temperature
- \( v_a \) = velocity of phase \( \alpha \)
- \( v_{\alpha}^{fr} \) = velocity across the fracture length
- \( \nabla_i \) = total partial molar volume
- \( x_{i,\alpha} \) = mole fraction
- \( z_i \) = overall mole fraction
- \( \alpha \) = phase index
- \( \Delta t \) = timestep
- \( \mu_s \) = dynamic viscosity
- \( \rho_s \) = mass density
- \( \tau \) = simulation time
- \( \phi \) = porosity
- \( \phi_{K,j} \) = DG basis function
- \( \Omega \) = computational domain

Acknowledgment
This work was supported by the member companies of the Reservoir Engineering Research Institute (RERI) whose support is gratefully acknowledged.

References


Appendix A: Well Implementation

Well implementation in our model is by fully unstructured grids. As opposed to prismatic and Voronoi grids, unstructured tetrahedra can readily describe complex geometries in 3D. One restriction that the well imposes on the simulation model relates to the computational time. Well diameter could be in the range of few centimeters, when the reservoir is in kilometer scale. Because of the Courant–Freidrich–Levy (CFL) condition, the size of timestep is, therefore, restricted by the small elements in the immediate well vicinity. The size of matrix grids near the well region is ensured to be at their maximum allowed size honoring the Delaunay tetrahedralizations. Coarsening the well geometry allows for larger matrix elements near the well region; however, it affects the well meanders. The well effect in terms of injection/production is accounted for in the simulation through the boundaries of the well elements that are in contact with the simulation domain. The well elements are not included in the simulation. When matching the interfaces of the well elements with the matrix domain, the traces of the pressure are, therefore, evaluated by the MHFE method. Fluxes at the matrix/well interfaces are treated as boundary condition flux. Imposing the boundary condition at the well/matrix boundary is made possible because well elements are not part of the simulation domain.
The well with its inclination is first designed in a CAD platform (Fig. A-1a) and then discretized with unstructured tetrahedra (Fig. A-1b shows cross section of the well with unstructured tetrahedra). The size of matrix finite elements discretized without the well is kept large (Fig. A-1c), and the accuracy is guaranteed by the higher-order discretization. When including the well, the mesh is gradually refined near the well region (Fig. A-1d). The triangular interfaces of the tetrahedra elements in the matrix domain should match the interfaces of the well elements at the same coordinates. In the last step, the well elements are removed from the original mesh, and the final mesh is then ready for simulation. We note that with this procedure, the effect of the well in the simulation is included by the interfaces that match the well wall, and the well rate is distributed along the total number of interfaces for elements at the well–matrix boundary. In our future investigation in well modeling, the well hydraulics will be taken into consideration.

One advantage of our well modeling approach is the ease to change/adapt the mesh size at the well location. A deviated well requires more refinement to capture the deviation meanders. With higher refinement, the CPU time is increased for two reasons: (i) the CFL condition in the matrix domain and (ii) the overall increase in the total number of elements to preserve the well details. To overcome this limitation, our modeling technique allows using large elements at the well location. Depending on the type of simulation, a tradeoff should be made between well shape and efficiency. The proposed modeling technique allows to choose a different order of refinement/coarsening. To demonstrate this capability, we show in Fig. A-2 the mesh in discretization of the well discussed earlier and in Fig. A-3 different refinements for the same well. Fig. A-3 reveals that the coarser the grids are at the well, the more of its meander details are lost.

![Fig. A-1—(a and b) Discretization mesh of well, size enlarged for clarity; (c) reservoir with the well; and (d) reservoir grids near the well.](image)

![Fig. A-2—Well discretization mesh.](image)
Appendix B: Fractures Anisotropy

One of the features in our model is the ease to assign anisotropic permeability across the normal and lateral sides of the fractures. To demonstrate this ability, we rerun the simulation of Example 1 in the text by having a normal permeability value of the fractures to be two orders of magnitude less compared with the lateral side. Note that the fractures are represented by \((n - 1) - D\) in our algorithm; therefore, having low permeability across the normal side reduces the matrix–fracture exchange significantly. This is due to the negligible flow from the fracture interface \((n - 2) - D\) to the matrix domain \((n - D)\). For reference, we show in Fig. B-1 the gas saturation at 60% PVI for the problem discussed in Example 1 in the text and with the permeability values discussed above.

![Fig. A-3](image)

Fig. A-3—Different coarsening levels for the well from (a) the finer mesh to (d) the coarse mesh.

Appendix B: Fractures Anisotropy

One of the features in our model is the ease to assign anisotropic permeability across the normal and lateral sides of the fractures. To demonstrate this ability, we rerun the simulation of Example 1 in the text by having a normal permeability value of the fractures to be two orders of magnitude less compared with the lateral side. Note that the fractures are represented by \((n - 1) - D\) in our algorithm; therefore, having low permeability across the normal side reduces the matrix–fracture exchange significantly. This is due to the negligible flow from the fracture interface \((n - 2) - D\) to the matrix domain \((n - D)\). For reference, we show in Fig. B-1 the gas saturation at 60% PVI for the problem discussed in Example 1 in the text and with the permeability values discussed above.

![Fig. B-1](image)

Fig. B-1—Gas saturation at 60% PVI for anisotropic permeability in the fractures: Example 1 in the text.

Appendix C: Fracture and Impermeable Fault Coordinates

In the following, we provide geometric coordinates in feet for all fractures and impermeable faults of Example 3 in the text. Table C-1 shows the starting and end coordinates for all fractures and impermeable faults. Note that the fractures are vertical extending from top to bottom of the domain.

<table>
<thead>
<tr>
<th>X_start</th>
<th>Y_start</th>
<th>X_end</th>
<th>Y_end</th>
<th>Description</th>
<th>X_start</th>
<th>Y_start</th>
<th>X_end</th>
<th>Y_end</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>755.58</td>
<td>55.78</td>
<td>801.63</td>
<td>25.81</td>
<td>Fracture</td>
<td>767.67</td>
<td>598.84</td>
<td>938.23</td>
<td>882.01</td>
<td>Impermeable fault</td>
</tr>
<tr>
<td>688.89</td>
<td>21.6</td>
<td>755.58</td>
<td>55.78</td>
<td>Fracture</td>
<td>642.23</td>
<td>698.87</td>
<td>837.24</td>
<td>771.94</td>
<td>Impermeable fault</td>
</tr>
<tr>
<td>755.58</td>
<td>55.78</td>
<td>802.03</td>
<td>79.58</td>
<td>Fracture</td>
<td>27.54</td>
<td>824.82</td>
<td>88.58</td>
<td>706.52</td>
<td>Impermeable fault</td>
</tr>
<tr>
<td>495.28</td>
<td>796.57</td>
<td>565.59</td>
<td>876.43</td>
<td>Fracture</td>
<td>37.71</td>
<td>559.58</td>
<td>146.84</td>
<td>731.48</td>
<td>Impermeable fault</td>
</tr>
<tr>
<td>588.77</td>
<td>840.43</td>
<td>638.87</td>
<td>863.93</td>
<td>Fracture</td>
<td>47.68</td>
<td>507.43</td>
<td>393.8</td>
<td>470.65</td>
<td>Impermeable fault</td>
</tr>
<tr>
<td>638.87</td>
<td>863.93</td>
<td>840.74</td>
<td>958.63</td>
<td>Fracture</td>
<td>423.99</td>
<td>246.72</td>
<td>531.34</td>
<td>393.17</td>
<td>Impermeable fault</td>
</tr>
</tbody>
</table>

Table C-1—Geometric coordinates in feet for all fractures and faults in Example 3.
<table>
<thead>
<tr>
<th>X_start</th>
<th>Y_start</th>
<th>X_end</th>
<th>Y_end</th>
<th>Description</th>
<th>X_start</th>
<th>Y_start</th>
<th>X_end</th>
<th>Y_end</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>549.15</td>
<td>921.76</td>
<td>583.76</td>
<td>889.61</td>
<td>Fracture</td>
<td>557.93</td>
<td>227.59</td>
<td>875.82</td>
<td>296.65</td>
<td>Impermeable fault</td>
</tr>
<tr>
<td>583.76</td>
<td>889.61</td>
<td>614.3</td>
<td>861.24</td>
<td>Fracture</td>
<td>737.61</td>
<td>239.56</td>
<td>872.13</td>
<td>180.63</td>
<td>Impermeable fault</td>
</tr>
<tr>
<td>583.76</td>
<td>889.61</td>
<td>603.29</td>
<td>897.9</td>
<td>Fracture</td>
<td>402.08</td>
<td>982.71</td>
<td>792.78</td>
<td>953.69</td>
<td>Impermeable fault</td>
</tr>
<tr>
<td>589.29</td>
<td>911.27</td>
<td>603.29</td>
<td>897.9</td>
<td>Fracture</td>
<td>375.34</td>
<td>12.83</td>
<td>499.74</td>
<td>80.95</td>
<td>Impermeable fault</td>
</tr>
<tr>
<td>603.29</td>
<td>897.9</td>
<td>638.87</td>
<td>863.93</td>
<td>Fracture</td>
<td>957.05</td>
<td>23.55</td>
<td>932.93</td>
<td>251.1</td>
<td>Impermeable fault</td>
</tr>
<tr>
<td>638.87</td>
<td>863.93</td>
<td>685.23</td>
<td>819.66</td>
<td>Fracture</td>
<td>28.97</td>
<td>54.44</td>
<td>56.23</td>
<td>205.29</td>
<td>Impermeable fault</td>
</tr>
<tr>
<td>55.11</td>
<td>913.42</td>
<td>175.86</td>
<td>969.41</td>
<td>Fracture</td>
<td>636.61</td>
<td>913.1</td>
<td>719.2</td>
<td>947.12</td>
<td>Fracture</td>
</tr>
<tr>
<td>175.86</td>
<td>969.41</td>
<td>220.14</td>
<td>909.81</td>
<td>Fracture</td>
<td>627.05</td>
<td>774.27</td>
<td>685.23</td>
<td>819.66</td>
<td>Fracture</td>
</tr>
<tr>
<td>220.14</td>
<td>909.81</td>
<td>250.21</td>
<td>925.55</td>
<td>Fracture</td>
<td>95.28</td>
<td>546.57</td>
<td>165.59</td>
<td>626.43</td>
<td>Fracture</td>
</tr>
<tr>
<td>229.36</td>
<td>946.34</td>
<td>250.21</td>
<td>925.55</td>
<td>Fracture</td>
<td>188.77</td>
<td>590.43</td>
<td>238.87</td>
<td>613.93</td>
<td>Fracture</td>
</tr>
<tr>
<td>250.21</td>
<td>925.55</td>
<td>287.8</td>
<td>888.06</td>
<td>Fracture</td>
<td>238.87</td>
<td>613.93</td>
<td>440.74</td>
<td>708.63</td>
<td>Fracture</td>
</tr>
<tr>
<td>250.21</td>
<td>925.55</td>
<td>288.88</td>
<td>945.78</td>
<td>Fracture</td>
<td>149.15</td>
<td>671.76</td>
<td>183.76</td>
<td>639.61</td>
<td>Fracture</td>
</tr>
<tr>
<td>288.88</td>
<td>945.78</td>
<td>337.95</td>
<td>971.46</td>
<td>Fracture</td>
<td>183.76</td>
<td>639.61</td>
<td>214.3</td>
<td>611.24</td>
<td>Fracture</td>
</tr>
<tr>
<td>288.88</td>
<td>945.78</td>
<td>330.7</td>
<td>894.05</td>
<td>Fracture</td>
<td>183.76</td>
<td>639.61</td>
<td>203.29</td>
<td>647.9</td>
<td>Fracture</td>
</tr>
</tbody>
</table>

Table C-1 (continued)—Geometric coordinates in feet for all fractures and faults in Example 3.
Ali Zidane is a researcher at the Reservoir Engineering Research Institute. His research interests are in the theory and advanced numerical modeling of miscible and immiscible multiphase flows in homogeneous and fractured media and higher-order reservoir simulation of complex subsurface flows. He holds a PhD degree in Earth Sciences from Basel University, Switzerland.

Abbas Firoozabadi is the director of the Reservoir Engineering Research Institute in Palo Alto, California, USA, and a distinguished research professor at Rice University, Houston, Texas, USA. His main research interests are in higher-order reservoir simulations, molecular simulations and molecular engineering in relation to hydrocarbon energy production, and the stewardship of the environment. He holds a BS degree from the Ababan Institute of Technology, Abadan, Iran, and a PhD degree from the Illinois Institute of Technology, Chicago, Illinois, USA.