

Three-Dimensional Projection-Based Embedded Discrete-Fracture Model for Compositional Simulation of Fractured Reservoirs

Olufemi Olorode, Bin Wang, and Harun Ur Rashid, Louisiana State University

Summary

Most unconventional oil and gas reservoirs are known to have several natural fractures in different orientations, which are consistent with the prevailing stresses when they were created. The accurate and efficient modeling of natural and hydraulic fractures presents a significant computational challenge. In this work, we show the limitations of the embedded discrete-fracture model (EDFM) and present the first 3D projection-based EDFM (pEDFM) algorithm and compositional simulation studies with realistic fracture networks in a fully 3D space.

The simulation results from this work indicate that the pEDFM presented can model realistic fractured unconventional reservoirs accurately and efficiently. To validate the model, we present some simplistic fracture cases that can be meshed and modeled easily using explicit-fracture modeling in commercial-reservoir simulators. From the cases studied, we observe that using progressively finer grids near the hydraulic-fracture surfaces helps to improve model accuracy because this allows us to capture the sharp pressure drops expected near these fracture surfaces. The simulation results show that, unlike EDFM, the robust pEDFM algorithm presented here is accurate even at the low fracture-conductivity values expected in many of these ubiquitous natural fractures.

In this paper, we present the first full 3D compositional modeling with pEDFM. We demonstrate that our model can accurately and efficiently model multiply fractured horizontal wells in unconventional reservoirs, which have complex networks of thousands of fractures at various orientations.

Introduction

Virtually all petroleum reservoirs are naturally fractured to some extent. Of these, unconventional oil and gas (UOG) reservoirs (also commonly referred to as shale oil/gas reservoirs) and fractured carbonate reservoirs have proven to be particularly significant. The U.S. Energy Information Administration (2019) shows that tight/shale gas reservoirs contribute approximately 75% of the total U.S. dry natural-gas production, whereas tight/shale oil accounts for approximately 50% of the total oil production from the United States today. The U.S. Energy Information Administration also predicts that the relative contribution of these UOG reservoirs to total U.S. oil and gas production will increase over the next 30 years. Although this work will focus on the modeling of fractured UOG reservoirs, the algorithm we propose can be used to model any fractured reservoir.

The simplest approach to model fractures explicitly involves representing each segment of a fracture with a gridblock. This brute-force method is referred to as the explicit-fracture model. It requires the matrix cells to conform to the geometry and orientation of every fracture in the simulation domain. Therefore, the mesh must be unstructured to model fractures with any arbitrary orientations and geometry. It is computationally expensive because of the enormous number of gridblocks that will be required to model such a system. Olorode et al. (2013) demonstrated using such an unstructured mesh to model hydraulically fractured shale gas wells explicitly. For reservoirs that have dense and interconnected fractures with uniform distribution, orientation, and aperture, it is common to use dual-continuum models, such as the dual-porosity and dual-permeability models (Warren and Root 1963). To account for the transient release of fluids from the matrix into the fractures in tight rocks, Pruess and Narasimhan (1985) developed the multiple interacting continua model, which involves subgridding the matrix gridblocks. Zimmerman et al. (1993) reported that using the multiple interacting continua model, we would need up to 10 submatrix cells to obtain accurate results over all time scales. To address this computational challenge, Zimmerman et al. (1993) presented the use of a transient shape factor instead of the pseudosteady-state shape factor in the standard dual-porosity model.

Although the dual-continuum model is typically used to model naturally fractured reservoirs today, it is limited in the sense that the matrix and fractures are assumed to be continuous, uniformly distributed, well connected, and with constant fracture properties and geometry (Norbeck et al. 2016). The marked variability in the production performance of different wells and fractures within the same shale play indicates that these assumptions are usually not applicable in UOG reservoirs. UOG reservoirs typically have fractures at multiple scales with different lengths, apertures, orientation, and conductivity. These heterogeneities in the fracture distribution and properties could be linked to the marked differences in the amount of production obtained from different wells in the same reservoir. Zoback and Kohli (2019) show that approximately 20% of UOG wells produce more than 80% of the total field production, and they suggest that the most productive regions of the reservoir (referred to as sweet spots) could be interpreted as regions with well-connected and conductive natural fracture networks. In addition, outcrop studies show that natural fractures exhibit significant variation in their aperture, length, height, conductivity, and spacing (Gillespie et al. 1993; Odling 1997; Odling et al. 1999). These variations in the fracture properties infer that using dual-continuum models could be questionable in these naturally fractured reservoirs. Another shortcoming of the multicontinuum models is that it is difficult to determine the shape factor for real reservoirs with capillarity and gravity effects, as well as multiple hydrocarbon components and fluid phases (Geiger et al. 2013).

The discrete-fracture model (DFM) was developed to account for the effect of each individual fracture on fluid flow and transport in these fractured reservoirs. The DFM models the fractures at a dimension of $n - 1$, where n is the number of dimensions of the model

(Kim and Deo 2000; Karimi-Fard and Firoozabadi 2001). This makes DFM faster than an explicit-fracture simulation, where very small n -dimensional gridblocks are used to represent each fracture segment in the domain. However, the DFM requires the grids in the simulation domain to conform to the geometry of all the fractures in the domain, making it impractical for the 3D modeling of UOG reservoirs with thousands of natural and hydraulic fractures (Moinfar 2013). To combine the advantage of the detailed fracture description from the DFM with the computational efficiency of the dual-continuum model, Gong (2007) developed the multiple subregion method. This method basically upscales a high-resolution DFM into an effective coarse-scale continuum model. Other authors (Gong et al. 2017; Li and Voskov 2018) have also demonstrated using a multilevel DFM, which models the large-scale fractures with the DFM and upscales the small-scale fractures using flow-based upscaling.

The EDFM was developed by Li and Lee (2008) to improve the computational efficiency in the modeling of natural fractures. It is based on a hierarchical approach (Lee et al. 1999, 2001), where very small-scale fractures are homogenized by changing the effective flow properties (such as permeability and transmissibility) of the matrix cells. In the EDFM, the matrix and large-scale fractures are discretized independently. Therefore, we can model the matrix cells with structured grids because they do not need to conform to the geometry and orientation of the fractures. To facilitate the practical use of this method in commercial simulators, Fumagalli et al. (2016) and Xu et al. (2019) extended EDFM to work with corner-point grids. EDFM has also been used to model UOG reservoirs, which are known to contain multiscale pores and fractures. Some authors (Moinfar et al. 2013; Li et al. 2017) have used a combination of EDFM and dual-continuum methods to model shale oil/gas reservoirs, whereas others (Diaz Campos et al. 2009; Devegowda et al. 2012; Zhang et al. 2017; Yu et al. 2019) have accounted for the confinement effects expected in these shale plays.

The major limitation of the EDFM is its inability to model fractures with low fracture conductivities. This led to the development of the pEDFM in Tene et al. (2017) and Tene (2018). The 2D pEDFM algorithm presented in Tene et al. (2017) and Jiang and Younis (2017) is limited to 2D or extruded 2D (2.5D) systems, and it implicitly assumes that the fractures will be vertical and fully penetrating from the top to the bottom of the reservoir. This rarely happens because the orientation of the natural fractures is controlled by the stress states at the time (in geologic history) when these fractures were created (Shafiei et al. 2018). Consequently, it is normal to have fractures with different orientations in space. In the recent paper by Rao et al. (2020), the authors stated that the 2D approach presented in Jiang and Younis (2017) will be complicated and impractical for 3D cases. Therefore, they proposed a simpler 3D algorithm, referred to as the microtranslation method. This method basically translates the fracture centroid in the direction of the normal vector, which makes either an acute or right angle with the vector from the cell centroid to the fracture centroid.

In this paper, we propose a robust 3D pEDFM that demonstrates the practicality of the extension of the 2D approach that was presented in Jiang and Younis (2017). We believe that this work provides significant improvement in 3D pEDF modeling because it prevents the unphysical configurations described in Figs. 13 and 14 of Jiang and Younis (2017). In addition, we demonstrate the first fully 3D pEDFM simulation of UOG reservoirs with more than 1,000 fractures in any arbitrary orientation. Some of the other advantages of using a 3D pEDFM include the following:

- Ability to account for each individual fracture's properties.
- Ability to perform uncertainty analyses on the fracture properties, distribution, and so on.
- Ability to perform robust optimization for infill well location, well spacing, cluster spacing, and so on, while accounting for the uncertainty in the location, distribution, and properties of the natural fractures.
- Ability to incorporate data from microseismic, distributed temperature sensing, distributed acoustic sensing, image logs, core samples, flow tests, and so on.
- In combination with a compositional reservoir-simulation model (which is discussed in the next section), 3D pEDFM provides the ability to efficiently model enhanced/improved oil recovery in unconventional oil reservoirs.

There is no technology available to determine the exact orientation and location of all the fractures in the subsurface today. Therefore, our goal is not to model the exact system of natural fractures in the subsurface but to quantify the uncertainty introduced by these fractures with arbitrary orientation.

Governing Equations for Compositional Reservoir Simulation

Cao (2002) and Voskov and Tchelepi (2012) discuss some of the mathematical formulations for compositional reservoir simulation. These include the molar variables (Fussell and Fussell 1979; Young and Stephenson 1983; Chien et al. 1985), volume-balance (Acs et al. 1985), overall composition variables (Collins et al. 1992; Voskov and Tchelepi 2012), and natural variables (Coats 1980) formulations. Møyner and Tchelepi (2017) provide details on the governing equations, discretization, and solution procedure for compositional simulation using the overall composition approach. As shown in this section, we also use the overall composition approach. We first provide a summary of these procedures in the context of a conventional reservoir simulator without a discrete fracture model. The next two sections will then extend these procedures to enable the modeling of fractured reservoirs using EDFM and pEDFM.

The mass-conservation equation for each hydrocarbon component i in the liquid (oil) and vapor (gas) phases can be expressed as

$$\frac{\partial}{\partial t} [\phi(\rho_l S_l X_l^i + \rho_v S_v X_g^i)] + \nabla \cdot (\rho_l X_l^i \vec{v}_l + \rho_v X_g^i \vec{v}_v) - (\rho_l X_l^i q_l + \rho_v X_g^i q_v) / V = 0. \quad (1)$$

In a similar form, we can write the governing equation for the conservation of water in the aqueous phase as:

$$\frac{\partial}{\partial t} (\phi \rho_w S_w) + \nabla \cdot (\rho_w \vec{v}_w) - \rho_w q_w / V = 0. \quad (2)$$

Here, ϕ represents porosity, and ρ_x , S_x , and q_x represent the mass density, saturation, and volumetric withdrawal/injection rate of phase x , respectively. The subscripts l and v refer to the liquid and vapor hydrocarbon phases, respectively. Therefore, the mass fractions of component i in the liquid and vapor phases are X_l^i and X_g^i , respectively. The symbols, \vec{v}_l and \vec{v}_v represent the Darcy velocity for the liquid and vapor hydrocarbon phases, respectively. In these conservation equations, we assumed that there is no mass transfer between the organic phases (oil and gas) and the aqueous (water) phase. We also divided the last term on the right-hand side of Eqs. 1 and 2 by bulk volume V to ensure that the equation is dimensionally consistent.

The phase velocities in Eqs. 1 and 2 can be obtained from the Darcy equation as follows:

$$\vec{v}_x = -\mathbf{K} \frac{k_x(S)}{\mu_x} (\nabla p_x - \rho_x g \nabla z) = -\mathbf{K} \lambda_x (\nabla p_x - \rho_x g \nabla z). \quad (3)$$

The phase mobility (λ_α) is defined as the ratio of the relative permeability of phase α to its phase viscosity μ_α , whereas \mathbf{K} is the absolute matrix permeability. The capillary pressure definitions, as well as saturation and composition constraints (i.e., saturations and compositions sum to 1) are also applied to the governing equations. In the overall composition approach, the primary variables are pressure, overall mole fraction of each component, and water saturation (p , z_1, \dots, z_{n-1} , and S_w), respectively. To obtain X_l^i and X_g^i , we solve the flash equations, which ensure that each component in the liquid phase is in chemical equilibrium with the corresponding component in the vapor phase, the sum of the number of moles of each component in the liquid and gas phases is equal to its corresponding overall composition, and that all mole fractions sum to 1. These three conditions are realized when the following equations are solved:

$$f_g^i(p, T, y_1, \dots, y_n) - f_l^i(p, T, x_1, \dots, x_n) = 0, \quad \text{for } i \in 1, \dots, n_c, \quad \dots \dots \dots (4)$$

$$z_i - Lx_i - (1 - L)y_i = 0, \quad \text{for } i \in 1, \dots, n_c, \quad \dots \dots \dots (5)$$

$$\sum_{i=1}^{n_c} x_i - y_i = 0. \quad \dots \dots \dots (6)$$

In these equations, f_g^i and f_l^i are the fugacities of each component in the gas and liquid phases, respectively. The solution to these flash equations will give the liquid fraction (L), as well as the mole fractions of each component in the gas and liquid phases (y_i and x_i). To compute the fugacities and phase compressibility factors (Z_g and Z_l), we use the Peng-Robinson equation of state (Peng and Robinson 1976). Firoozabadi (2015) provides more details on the equation of state, flash procedure, and the equations to compute the fugacities and compressibility factors. The results obtained from the solution of these flash equations are then used to compute the phase saturations, phase densities, and mass fractions, as discussed in Møyner and Tchelepi (2017).

To solve the continuous equations (Eqs. 1 and 2) numerically, we first perform temporal discretization using the backward Euler scheme to obtain

$$\frac{1}{\Delta t} [(\phi \rho_l S_l X_l^i + \phi \rho_v S_v X_v^i)^{n+1} - (\phi \rho_l S_l X_l^i + \phi \rho_v S_v X_v^i)^n] + \nabla \cdot (\rho_l X_l^i \vec{v}_l + \rho_v X_v^i \vec{v}_v) - (\rho_l X_l^i q_l + \rho_v X_v^i q_v) / V = R_i, \quad \dots \dots \dots (7)$$

$$\frac{1}{\Delta t} [(\phi \rho_w S_w)^{n+1} - (\phi \rho_w S_w)^n] + \nabla \cdot (\rho_w \vec{v}_w) - \rho_w q_w / V = R_w. \quad \dots \dots \dots (8)$$

In these equations, $n + 1$ represents the current timestep, and n represents the previous timestep. All other terms without these superscripts are evaluated at the current timestep. We discretize the flux terms in space using the finite-volume method with two-point flux approximation. This method involves integrating Eqs. 7 and 8 over a control volume, as discussed in the MATLAB® (The MathWorks, Inc., Natick, Massachusetts, USA) reservoir simulation book (Lie 2019). Section 4.4 in this book explains how the transmissibility (T) and flux terms (in the second term of both equations) are computed using the discrete divergence and gradient operators. In terms of these discrete operators, the discretized form of the equation can be written as

$$\frac{V}{\Delta t} [(\phi \rho_l S_l X_l^i + \phi \rho_v S_v X_v^i)^{n+1} - (\phi \rho_l S_l X_l^i + \phi \rho_v S_v X_v^i)^n] + \text{div}(\rho_l X_l^i \vec{v}_l + \rho_v X_v^i \vec{v}_v)^{n+1} - (\rho_l X_l^i q_l + \rho_v X_v^i q_v)^{n+1} / V = R_i^{n+1}, \quad \dots \dots \dots (9)$$

$$\frac{V}{\Delta t} [(\phi \rho_w S_w)^{n+1} - (\phi \rho_w S_w)^n] + \text{div}(\rho_w \vec{v}_w)^{n+1} - (\rho_w q_w)^{n+1} / V = R_w^{n+1}, \quad \dots \dots \dots (10)$$

and

$$\vec{v}_\alpha = -T_{ik} \lambda_\alpha^{n+1} [\text{grad}(p_\alpha^{n+1}) - \rho_\alpha^{n+1} g \text{grad}(z)], \quad \dots \dots \dots (11)$$

where

$$T_{ik} = [T_{i,k}^{-1} + T_{k,i}^{-1}]^{-1} \quad \dots \dots \dots (12)$$

and

$$T_{i,k} = A_{i,k} \mathbf{K}_i \frac{\vec{c}_{i,k} \cdot \vec{n}_{i,k}}{|\vec{c}_{i,k}|^2}. \quad \dots \dots \dots (13)$$

The div and grad functions in Eqs. 9 through 11 are the discrete divergence and gradient operators that are implemented as functions in Lie (2019). The cell volume and face areas are represented as V and $A_{i,k}$, respectively. The symbol, $\vec{n}_{i,k}$ is the unit normal that points from the centroid of cell i to the face between cells i and k , and $\vec{c}_{i,k}$ is the vector from the cell centroid to the face centroid. T_{ik} is face transmissibility, and $T_{i,k}$ is the contribution of a cell to the face transmissibility. It is referred to as a half-transmissibility because a pair of cells contributes to the transmissibility of each face in the two-point flux approximation scheme. Unlike the conventional definition of transmissibility in structured grids (Aziz and Settari 1979), Eqs. 12 and 13 apply to both structured and unstructured grids when the two-point flux approximation scheme is used. It is worth noting that the temporal and spatial discretization schemes are numerical approximations to the governing partial differential equations and lead to a mass imbalance, which is referred to as the residual (R). The Newton-Raphson method involves applying the Taylor expansion to the residual at the current timestep and current Newtonian iteration to obtain

$$\frac{\partial R^{k+1}}{\partial X} \Delta X = -R^{k+1}(X). \quad \dots \dots \dots (14)$$

The partial derivative of the residuals with respect to each of the primary variables ($\frac{\partial R^{k+1}}{\partial X}$) is referred to as the Jacobian matrix.

This work uses automatic differentiation to compute the Jacobian matrix, using the MATLAB Reservoir Simulation Toolkit (MRST) by Lie (2019). The compositional module in MRST uses automatic differentiation to compute the derivatives of the flash equations with respect to the primary variables, as explained in Møyner and Tchelepi (2017). Aziz and Settari (1979) and Lie (2019) provide

more details on the application of the Newton-Raphson iteration scheme to solve the nonlinear system of equations that govern flow in petroleum reservoirs. The resulting linearized system of equations can be solved with an appropriate linear solver to obtain the changes in the primary variables (ΔX) at each Newtonian iteration. In this work, we use a bi-conjugate gradient-stabilized linear solver, with an algebraic multi-grid preconditioner. The changes in the primary variables are then added to the previous value of the primary variable (X), and the process is repeated until the system converges. We then move on to the next timestep and repeat this Newton-Raphson iterative procedure.

Compositional Simulation of Fractured Reservoirs with EDFM

EDFM uses the concept of non-neighboring connections (NNCs) to couple the flow of fluids in a fracture cell to that of its host matrix cell. The implementation is such that we simply add q_i^{nnc} to the semidiscrete form of the governing equation (Eq. 7) as follows:

$$\frac{1}{\Delta t} [(\phi \rho_l S_l X_l^i + \phi \rho_v S_v X_g^i)^{n+1} - (\phi \rho_l S_l X_l^i + \phi \rho_v S_v X_g^i)^n] + \nabla \cdot (\rho_l X_l^i \vec{v}_l + \rho_v X_g^i \vec{v}_v) - (\rho_l X_l^i q_l + \rho_v X_g^i q_v) / V + q_i^{nnc} / V = R_i. \quad \dots \quad (15)$$

Here, q_i^{nnc} is the mass rate of component i , which is exchanged through the NNC (in units of mass per time). It is defined as

$$q_i^{nnc} = \sum_{m=1}^{N_{nnc}} A_m^{nnc} \sum_{\alpha=1}^{n_p} \frac{k_m^{nnc} k_{r\alpha}}{\mu_\alpha} \rho_\alpha X_\alpha^i \left[\frac{(p_\alpha - \rho_\alpha g z) - (p_\alpha - \rho_\alpha g z)_m^{nnc}}{d_m^{nnc}} \right]. \quad \dots \quad (16)$$

In this equation, subscript m is an index from 1 through N_{nnc} , which is the total number of NNCs for each cell. The terms $(p_\alpha - \rho_\alpha g z)$ and $(p_\alpha - \rho_\alpha g z)_m^{nnc}$ are the flow potentials of a cell and its non-neighboring cell, respectively. The symbols A^{nnc} , k^{nnc} , and d^{nnc} represent the area, permeability, and distance of the NNCs, respectively. These parameters are used to determine the transmissibility factor between any pair of cells that interact through a NNC. The transmissibility factor for an NNC is given as

$$T^{nnc} = \frac{k^{nnc} A^{nnc}}{d^{nnc}}. \quad \dots \quad (17)$$

The expressions for A^{nnc} , k^{nnc} , and d^{nnc} vary depending on the type of NNC under consideration. Given that the only difference between Eqs. 7 and 15 is the addition of the q_i^{nnc} / V , Eq. 15 will be discretized as discussed in the previous section (Eqs. 9 through 14). Moinfar (2013) provides more details on the equations presented in this section, as well as the expressions for the three types of NNCs identified in EDFM. In the next four subsections, we outline all four possible types of connections between cells in EDFM. This allows us to clearly explain the additional NNCs that are needed to implement our 3D pEDFM algorithm.

Matrix/Matrix Connectivity. This refers to the standard connectivity that is obtained by virtue of a matrix cell being a neighbor of another matrix cell. Therefore, it is not an NNC (in a strict sense), and we do not need expressions for A^{nnc} , k^{nnc} , d^{nnc} , and T^{nnc} in this case.

Matrix/Fracture Connectivity. This refers to the NNC between a fracture cell and its host matrix cell. The first image in **Fig. 1** illustrates this as the interaction between the blue fracture and the peach matrix cell in which it is located. For this case, the parameters A^{nnc} , k^{nnc} , and d^{nnc} in Eq. 17 are defined as

$$A^{nnc} = 2A_f, \quad \dots \quad (18)$$

$$k^{nnc} = \frac{k_m k_f}{k_m + k_f}, \quad \dots \quad (19)$$

$$d^{nnc} = \frac{\int_v x_n dv}{V}. \quad \dots \quad (20)$$

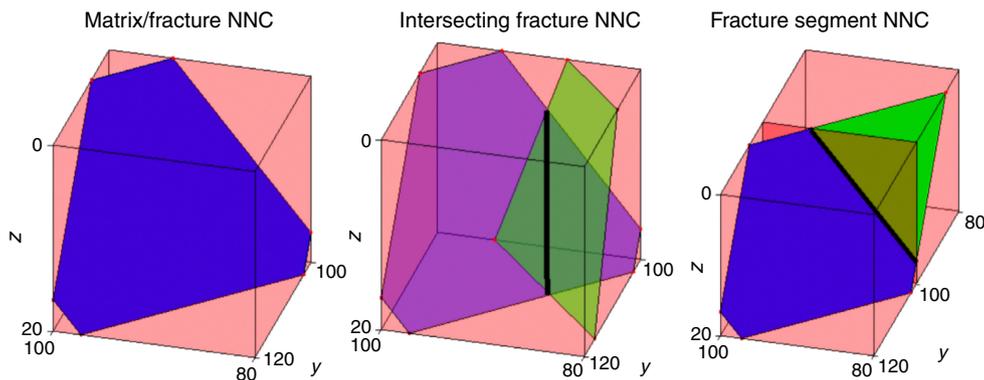


Fig. 1—Types of NNCs in EDFM. Image is courtesy of Moinfar (2013).

Here, A_f is the fracture area, and k_m and k_f are the matrix and fracture permeabilities, respectively. The symbols dv , x_n , and v represent the volume element, normal distance of the element from the fracture, and cell volume, respectively.

Fracture/Fracture Connectivity. This refers to the connection between two cells of an individual fracture plane. Although it is referred to as a fracture segment NNC in the third image in Fig. 1, it could be seen as a standard connection that can be obtained by virtue of a fracture cell being a neighbor to another fracture cell within the same fracture plane. Like the matrix/matrix connection, this is obtained by default in a reservoir simulator, and there is no need to define any NNC parameters in this case.

Intersecting Fracture Connectivity. This refers to the NNC that results from the intersection of a fracture cell with another fracture cell from a different fracture plane. As illustrated in the second image in Fig. 1, the intersection between two fracture cells from different fracture sets gives the solid black line shown. For this case, the NNC transmissibility is defined as

$$T^{nnc} = \frac{T_1 T_2}{T_1 + T_2}, \dots \dots \dots (21)$$

where T_1 and T_2 are the half transmissibilities of the two intersecting fractures

$$T_1 = \frac{k_{f1} \omega_1 L_{int}}{d_{f1}}, \dots \dots \dots (22)$$

and

$$T_2 = \frac{k_{f2} \omega_2 L_{int}}{d_{f2}}. \dots \dots \dots (23)$$

Here, L_{int} represents the length of the line of intersection between two fracture cells within a matrix cell. The symbols ω_f and k_f represent the fracture aperture and fracture permeability, respectively, and d_{f1} and d_{f2} represent the distances from the centroids of fracture cells 1 and 2, respectively, to the intersection line.

Introduction of the pEDFM

Tene et al. (2017) presented results that showed that EDFM is not accurate when the permeability of a fracture is less than that of the matrix. They also showed that in cases where fractures lie exactly at the interface between two matrix cells, EDFM basically ignores the properties of the fracture in between these matrix cells. They developed a 2D pEDFM to fix these limitations of the EDFM. As will be shown in the next section, pEDFM extends the EDFM approach by modifying the matrix/matrix and fracture/matrix transmissibilities in the neighborhood of the fractures. The major challenge in implementing this algorithm is the determination of the matrix cells whose transmissibilities need to be modified. The matrix cell that hosts the fracture cell is referred to as a host matrix cell or host cell, and the neighboring matrix cells into which the fracture cell will be projected are referred to as the projection matrix cells or projection cells. For the rare case in which fractures lie exactly at the interfaces between two pairs of matrix cells, pEDFM simplifies into the DFM, as discussed in Tene et al. (2017). When fracture cells do not lie at matrix cell interfaces, they cut into the interior of the matrix cell(s) and are referred to as interior fracture cells.

Although Tene et al. (2017) provided the mathematical equations to compute the matrix/matrix and fracture/matrix transmissibilities in the 2D pEDFM, the authors stated that the development of a generic algorithm to determine the projection cells is beyond the scope of their paper. Jiang and Younis (2017) presented this algorithm for a 2D system, but an extension of this algorithm to a fully 3D system is not trivial. This paper presents a robust 3D pEDFM algorithm, which allows us to model realistic fractures that could have any arbitrary orientation in a structured 3D Cartesian grid. It is worth mentioning that the pEDFM model proposed could also be applied for improved speed and accuracy in the modeling of UOG reservoirs that have no natural fractures but with hundreds of hydraulic fractures. Raterman et al. (2018) presents an actual observation of limited natural fracturing but up to 966 hydraulic fractures in a region within the Eagle Ford Shale. The modeling of these many hydraulic fractures using the DFM or an explicit-fracture model would be computationally prohibitive, but our 3D pEDFM model will provide the necessary computational speed and flexibility to model this UOG reservoir with hydraulic fractures striking at N060°E and dipping at 75 to 80°SE (with a variation of up to $\pm 20^\circ$ in both the strike and dip).

Robust Algorithm for 3D pEDFM

Fig. 2 shows the algorithm to find the projection matrix cells for 3D Cartesian grids with fractures in any arbitrary orientation. Although our algorithm could be adapted for simple corner-point grids with six faces, we leave this (as well as more complicated corner-point and unstructured grids) to a future study. The four cases discussed in this section cover all possible orientations of a fracture in a structured hexahedral gridblock. However, to simplify our discussion, we use the simplest examples in each of these four cases. The algorithm starts with a loop through all interior fracture cells. Within this loop, we compute the distance between the fracture centroid and the centroids of all the six faces of the host matrix cell. Because this is a 3D algorithm, there will be a pair of faces in each of the three possible (X-, Y-, and Z-) directions. The procedure to determine the projection matrix cell involves finding which of the pairs of the host matrix faces is the projection face for each of the three directions. This is because a projection face will always separate a host matrix cell from a projection cell.

Case 0: No Equidistant Pair of Faces. Fig. 3 presents a case where a fracture cell is closer to one of the pair of faces in all three directions. In this case, the closer face is selected for each pair of faces, and that gives us the three projection cells for the X-, Y-, and Z-directions. The left image in the figure shows the 3D view of a case with no equidistant pair of faces, and the middle and right images show the corresponding front and top views, respectively. It is easy to see that the fracture is closer to the right, back, and bottom faces in the X-, Y-, and Z-directions, respectively. In this case in which there is no equidistant pair of faces in any direction, we simply compute the projection matrix areas and transmissibilities and then proceed to the next interior fracture cell.

If there is at least one equidistant pair of faces in any of the three directions, we first compute the area and transmissibility for the faces we can select on the basis of the nearness to the fracture centroid and then proceed to the steps shown in the lower half of Fig. 2. This corresponds to the remaining three cases discussed in the next three subsections. For the scenario with one equidistant pair of faces, we would have selected two faces based on proximity to the fracture centroid. Similarly, one face would have been selected in the case of two equidistant pairs of faces.

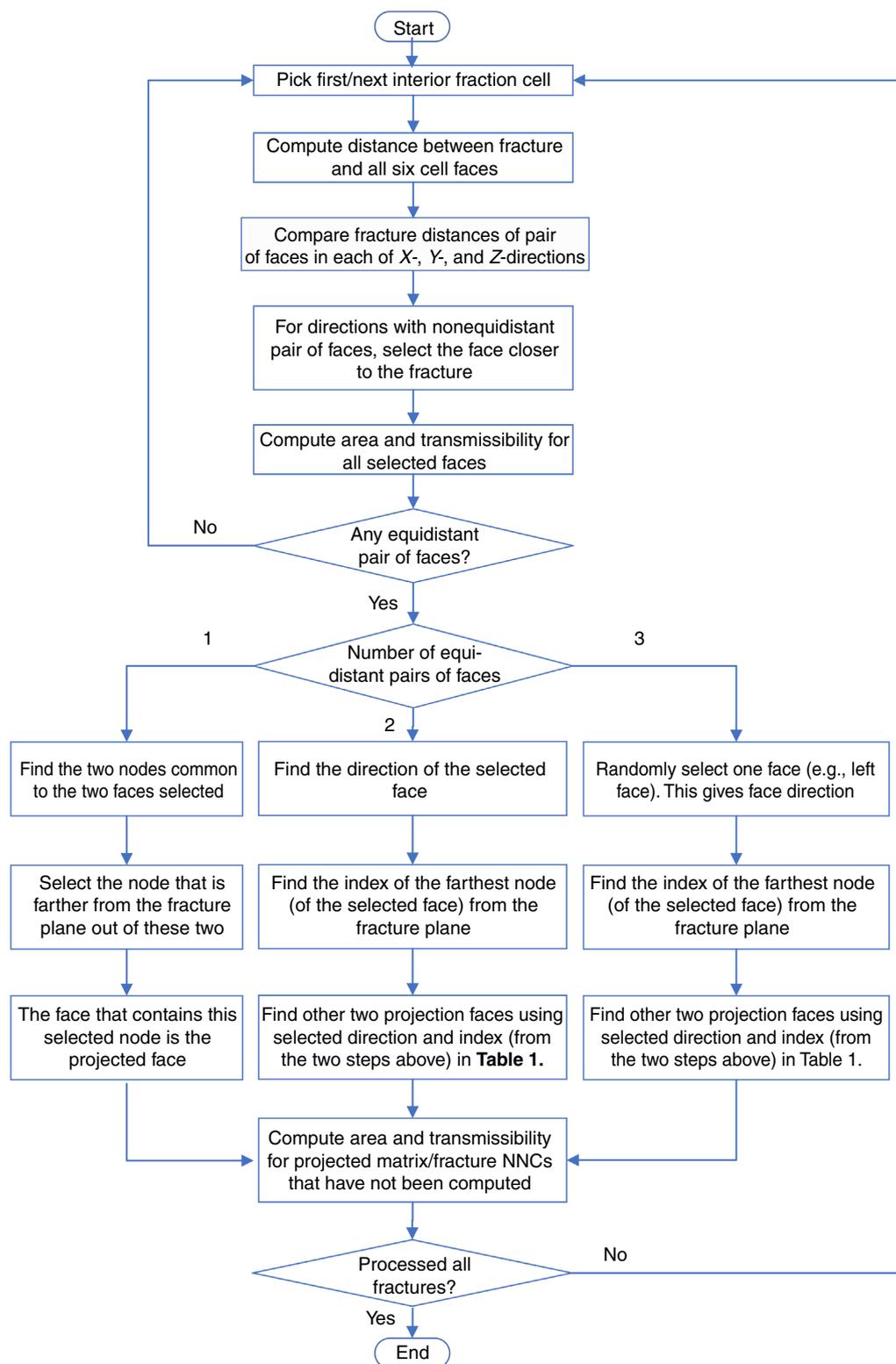


Fig. 2—Proposed algorithm automatically determines the three neighboring cells, whose transmissibilities and connection areas need to be modified in 3D pEDFM.

Case 1: One Equidistant Pair of Faces. The left image in Fig. 4 shows the 3D view of a case with one equidistant pair of faces, and the middle and right images show its corresponding front and top views, respectively. It is easy to see that the fracture centroid is equidistant from both the front and back faces but closer to the right and bottom faces in the X- and Z-directions, respectively. The algorithm we use in this case is such that we find the coordinates of the two nodes that are common to the two faces, which would have been selected from the previous steps in the algorithm. Next, we find the distance between each of these two nodes and the plane of the fracture cell. The node with the larger distance is selected, and this node will be one of the four nodes that bound the third face to be selected. This approach, like its 2D corollary (Jiang and Younis 2017), prevents the selection of faces that intersect in the path of the fracture plane. The algorithm then continues with the computation of the area and transmissibility for this newly (third) selected face, before proceeding to the next interior fracture cell.

Node Index	Face 1	Face 2	Face 1	Face 2	Face 1	Face 2
1	Front	Bottom	Left	Bottom	Left	Front
2	Back	Bottom	Left	Top	Right	Front
3	Back	Top	Right	Top	Right	Back
4	Front	Top	Right	Bottom	Left	Back

Table 1—Selection of remaining two faces based on a combination of the node index and the direction of the only selected face.

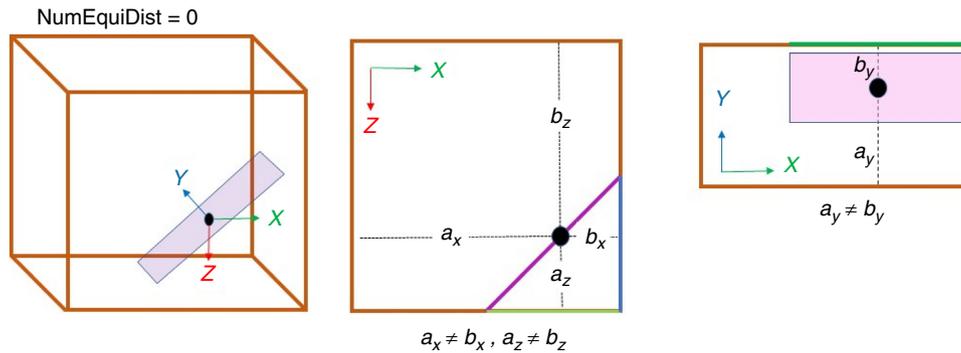


Fig. 3—In this case, where there is no pair of faces that is equidistant from the fracture centroid in any direction, the nearer faces will be selected in all three directions. For the case shown here, the fracture is closer to the right, back, and bottom faces in the X-, Y-, and Z-directions, respectively. These will be the three faces selected in the algorithm.

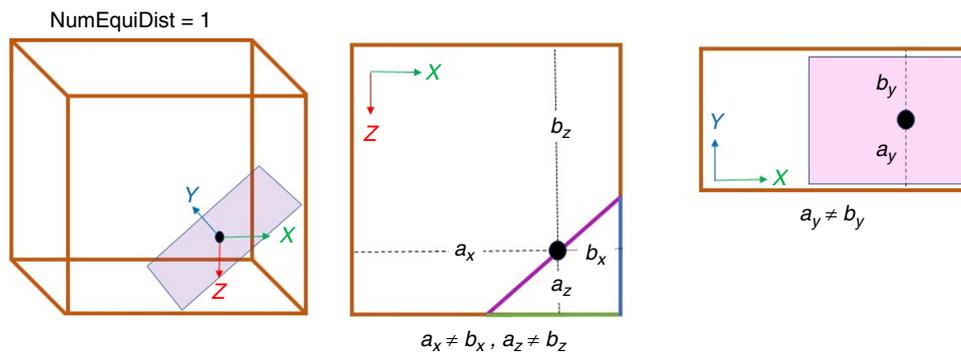


Fig. 4—In this case, we select the right and bottom faces in the X- and Z-directions, respectively. We cannot select either the front or back faces because they are equidistant from the fracture centroid. In such cases, where there is only one equidistant pair of faces, the third face will be selected based on the common nodes of the selected faces. The node with the maximum distance from the fracture plane will be used in the face selection.

Case 2: Two Equidistant Pairs of Faces. The left image in Fig. 5 shows the 3D view of a case with two equidistant pairs of faces, and the middle and right images show its corresponding front and top views, respectively. It is easy to see that the fracture centroid is equidistant from the front and back faces and also equidistant from the top and bottom faces, but closer to the right face in the X-direction. In this case, where there are two pairs of equidistant host cells out of the three directions, we first find the direction of the only selected face. We then find which of the four nodes of this face is farthest from the plane of the fracture. The combination of the index of the farthest node (out of the four nodes bounding the selected face) and the direction of the previously selected face is used to select the remaining two faces such that they do not intersect in the path of the fracture plane. The selection of these two faces based on the selected face direction and the index of the furthest node can be simplified using Table 1. This table is based on using the coordinate system shown in Fig. 6, as well as the node-ordering scheme in Figs. 7a to 7c for the left/right, front/back, and top/bottom faces, respectively. Also, for each cell, the faces are ordered from face 1 to 6, representing the left, right, front, back, top, and bottom faces, respectively. The algorithm then continues with the computation of the area and transmissibility for these two newly (second and third) selected faces, before proceeding to the next interior fracture cell.

Case 3: Three Equidistant Pairs of Faces. The left image in Fig. 8 shows the 3D view of a case where all three pairs of host cell faces are equidistant from the fracture cell centroid, and the middle and right images show its corresponding front and top views, respectively. In this case, we randomly select one out of the cell faces. This gives the face direction for the first out of three projection cell surfaces. We then compute the area and transmissibility for this selected projection face and find the index of the farthest node (of the selected face) from the fracture plane. As in Case 2, the combination of the index of the farthest node (out of the four nodes bounding the selected face) and the direction of the previously selected face is used to select the remaining two faces, such that they do not intersect in the path of the fracture plane. We again use Table 1 as in the previous case. The algorithm then continues with the computation of the area and transmissibility for these two newly (second and third) selected faces, before proceeding to the next interior fracture

cell. Although Figs. 3 through 5 and 7 only show rectangular projections of the fracture on the faces, the algorithm presented is robust enough to handle fractures with any arbitrary orientations, where the projected area could be any convex and coplanar polygon. We later demonstrate this using a Bakken shale oil simulation case that has 1,000 fractures in any arbitrary orientation.

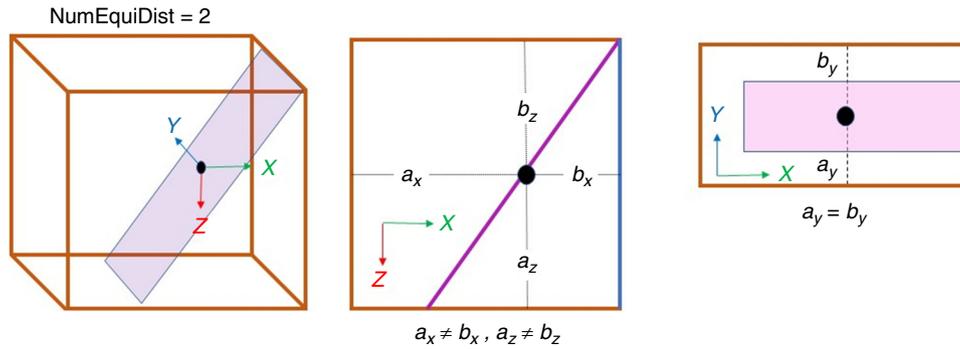


Fig. 5—In this case, we are only able to select the right face based on proximity to the fracture centroid in the X direction. In such cases, where there are two equidistant pairs of faces, one of the four nodes of the selected face will be used to select the remaining two faces.

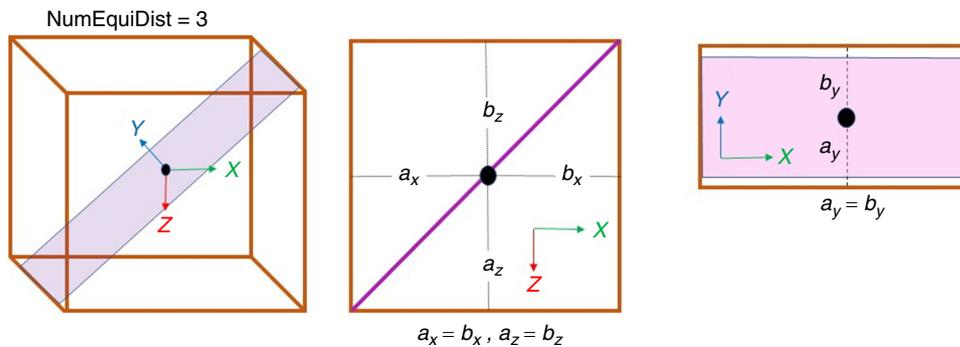


Fig. 6—Cell faces are numbered from 1 to 6 in the following order: left (1), right (2), front (3), back (4), top (5), and bottom (6). The direction of the X-, Y-, and Z-axes is also given.

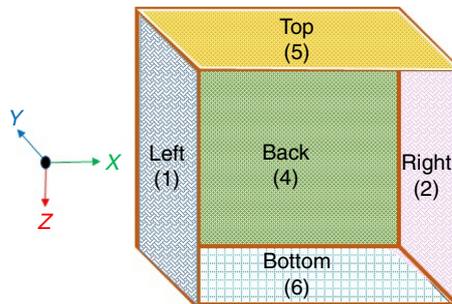


Fig. 7—Node ordering scheme: (a) left/right, (b) front/back, and (c) top/bottom faces.

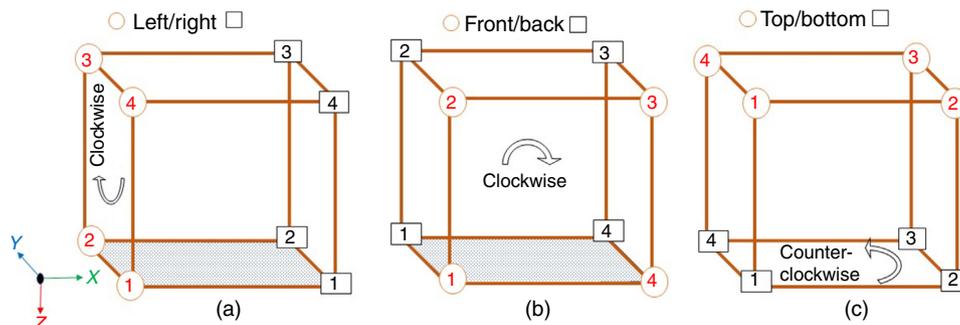


Fig. 8—In this case, where all the three pairs of host-cell faces are equidistant from the fracture, we arbitrarily select one face in any direction and then revert to the algorithm for Case 2.

The modifications needed to extend an EDFM into a pEDFM involves first determining the projection cells, and then computing the transmissibilities for the two distinct groups of NNCs that are absent in the EDFM algorithm. These two pEDFM NNCs are discussed in the next two subsections.

Projection Matrix/Fracture Transmissibility. This refers to the NNC transmissibility between a projection matrix cell and a fracture cell. For convenience, it is abbreviated as the pM-F transmissibility. Tene et al. (2017) gives the expression for this transmissibility as

$$T_{pM-F}^{nnc} = \frac{A_{if\perp\vec{x}} k_{pM-F}^{nnc}}{d_{pM-F}^{nnc}}, \quad \dots \dots \dots (24)$$

where

$$k_{pM-F}^{nnc} = \frac{k_{pM} k_f}{k_{pM} + k_f}. \quad \dots \dots \dots (25)$$

Here, $A_{if\perp\vec{x}}$ is the area of the fracture projections along each dimension, \vec{x} represents the X-, Y-, and Z-coordinates, and d_{pM-F}^{nnc} is the distance between the fracture centroid and the centroid of the projection cell.

Projection Matrix/Matrix Transmissibility. This refers to the NNC transmissibility between the host matrix cells and their corresponding projection matrix cells (pM-M transmissibilities). The expression for this transmissibility is given as

$$T_{pM-M}^{nnc} = k \frac{A_{ii} - A_{if\perp\vec{x}}}{\Delta\vec{x}_e}. \quad \dots \dots \dots (26)$$

Here, $\Delta\vec{x}_e$ refers to the gridblock sizes in each spatial direction. To facilitate the implementation of this transmissibility calculation in code, we can take advantage of the matrix/matrix transmissibilities already computed in standard reservoir simulation by reformulating the equation as

$$T_{pM-M}^{nnc} = T_{M-M} \frac{A_{ii} - A_{if\perp\vec{x}}}{A_{ii}}. \quad \dots \dots \dots (27)$$

Here, T_{M-M} is the standard matrix/matrix transmissibility that can be written as

$$T_{M-M} = k \frac{A_{ii}}{\Delta\vec{x}_e}. \quad \dots \dots \dots (28)$$

The interested reader can confirm that Eq. 27 can be obtained from a combination of Eqs. 26 and 28. The fractional term in Eq. 27 is then implemented as a transmissibility multiplier.

When multiple fracture segments are collocated within a host matrix cell, the total projected area is computed as a union of the projection areas for each of the fracture segments. This is expressed mathematically as

$$A_{ij}^p = \bar{A}_i^p \cup \bar{A}_j^p. \quad \dots \dots \dots (29)$$

Here, \bar{A}_i^p and \bar{A}_j^p are the projection areas of fracture segments i and j , respectively (in one direction), and A_{ij}^p is the total projection area in the same direction. Jiang and Younis (2017) provides more details on this case with multiple fractures within a host matrix cell.

Validation of the 3D pEDFM

In this section, we will present two simple validation cases. The first case will validate our 3D pEDFM against published 2D pEDFM results, whereas the second case is a compositional case that will validate our 3D pEDFM against high-resolution explicit fracture model results from both the MRST (Lie 2019) and a commercial compositional simulator. All the pEDFM equations and algorithms described in this paper were implemented as an extension to the existing modules available in the MRST.

Validation against Published Results. Hajibeygi et al. (2011) provide a classic test case that can be used to compare our pEDFM simulation results with the corresponding results from EDFM and a high-resolution explicit-fracture model. It basically involves modeling single-phase flow of a slightly compressible fluid across a square domain with two sealing fractures. These fractures intersect at 90° in the middle of the reservoir domain, as shown in Fig. 9. The flow from the left- to the right-hand side of the domain is caused by setting the pressure to a higher value on the left-hand side of the domain.

We used a $27 \times 27 \times 3$ Cartesian grid with a homogeneous permeability of 1 darcy and a porosity of 0.3 everywhere in the reservoir. Although this case is a 2.5D problem (because the 2D top view of the grid is basically extruded down three layers and the two fractures are fully vertical), the next two sections will show other cases with fractures that can have any arbitrary orientation in 3D space. To validate the accuracy of our pEDFM code in modeling fractures with both low and high permeability values, we run two cases: one with a fracture permeability of 10 nd and another with a permeability of 1,000 darcy.

To drive the flow of water from the west to the east of the domain, we specify a Dirichlet boundary condition with a fixed pressure of 10 bars on the west boundary and a Dirichlet boundary condition with a fixed pressure of 0 bars on the east boundary. We ran the simulation for a period of 60 days, and the simulation results are shown in Fig. 10. The image on the left shows the Dirichlet boundary conditions in red, and the two intersecting fractures are shown in blue. The second image gives the pressure profile.

To obtain results similar to those in Figs. 3 and 4 of Tene et al. (2017), we also provide a high-resolution explicit-fracture solution for reference. Instead of the $27 \times 27 \times 3$ Cartesian grid used in the EDFM and pEDFM cases, this reference solution uses a $225 \times 225 \times 3$ Cartesian grid. All the other parameters of the high-resolution explicit-fracture model are identical with those from the EDFM and pEDFM cases. Fig. 11 presents the pressure profiles for the reference, pEDFM, and EDFM cases with the fractures acting as flow barriers (fracture permeability is 10 nd). The results are similar to the published results in Tene et al. (2017). Observe that the pEDFM results match the reference solution, whereas EDFM does not. The EDFM pressure profiles seem to completely ignore the presence of these two intersecting flow barriers.

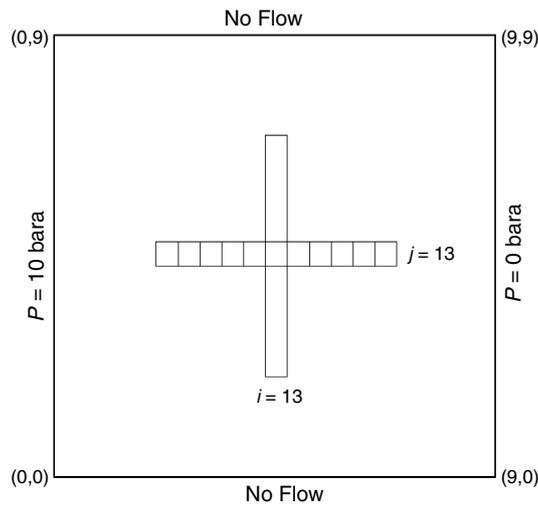


Fig. 9—This sketch illustrates the plus-shaped intersection of two fractures in the middle of the 2D domain. The image is courtesy of Hajibeygi et al. (2011).

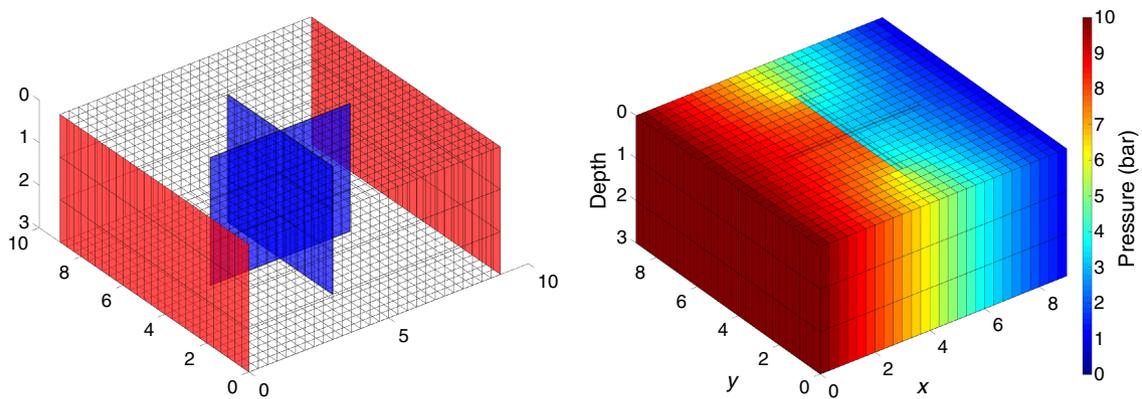


Fig. 10—In the left plot, faces with Dirichlet boundary conditions are marked red, and the fracture surfaces are marked blue. The right plot shows the resulting pressure distribution.

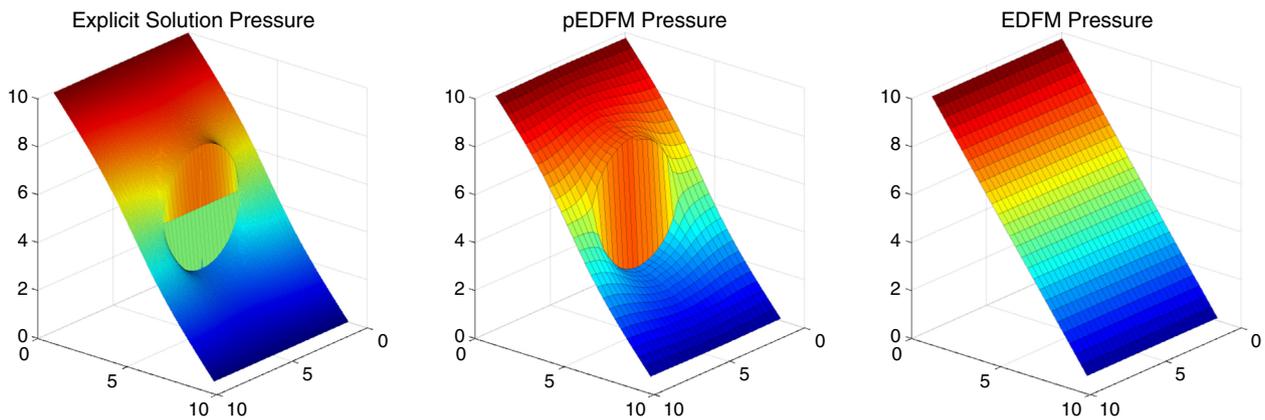


Fig. 11—At the low fracture-conductivity values in this model, EDFM is unable to capture the pressure variation near the flow barrier, but pEDFM is able to capture the sharp pressure jump present in the high-resolution reference solution.

In addition, Fig. 12 presents the simulation results for the case where the two intersecting fractures are highly conductive (fracture permeability is set to 1,000 darcy). In this case, both EDFM and pEDFM match the high-resolution results.

Fig. 13 provides a quantitative comparison of both the EDFM and pEDFM simulations to the reference solutions obtained from the high-resolution explicit-fracture simulation model. The results shown in this figure were obtained by extracting the pressure in the middle cells (in the J - or Y -direction) running from the west to the east of the reservoir-simulation domain. The results from Figs. 11 through 13 are consistent with the observations in the literature that EDFM seems to be accurate at high fracture conductivities but inaccurate at low fracture conductivities. Conversely, pEDFM seems accurate at both low and high fracture conductivities.

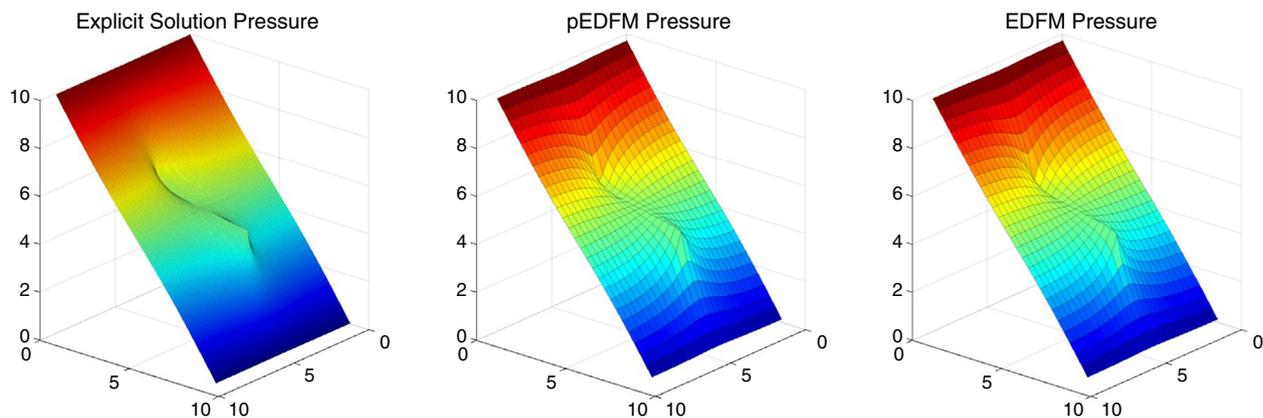


Fig. 12—At the high fracture-conductivity values in this model, both the EDFM and pEDFM simulation results seem to match the high-resolution reference solution. They are both able to capture the pressure variation near the fracture surfaces in this case.

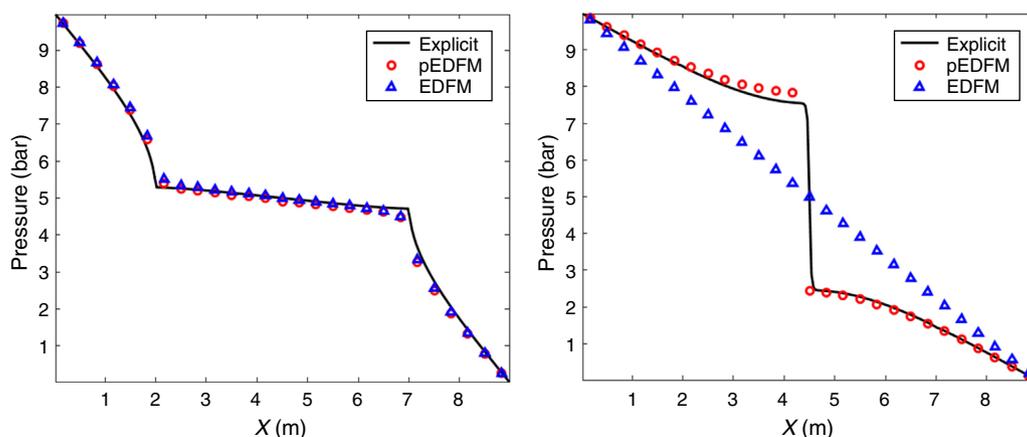


Fig. 13—Comparison of the pressure profiles from pEDFM and EDFM to the reference solution indicates that both models are accurate at high fracture conductivities (left). At low fracture conductivities (right), EDFM is inaccurate, whereas pEDFM still works accurately.

Considering that the natural fractures in UOG reservoirs were generated over several different geological periods of time, some are active at the stress state of the rock today, whereas others are not. In addition, some of the older natural fractures could be sealed with cementing materials and act as sealing fractures/faults. The pEDFM code developed as part of this work provides the computational speed and accuracy required to model natural fractures/faults with high or low fracture conductivities and in any arbitrary orientation in 3D.

Validation against a Commercial Compositional Simulator. In this section, we present simulation results that validate our 3D pEDFM model against explicit-fracture simulation results from MRST and a commercial compositional simulator. For comparison, we also show EDFM simulation results within the same context. The cases studied involve modeling a single hydraulically fractured well in the middle of two sealing barriers, which are basically two vertical fractures with very low fracture conductivity values. The motivation to provide a case study with a hydraulic fracture connected to a well (hydraulically fractured well) is to quantify the significance of the pEDFM modification on production rates and cumulative production. Unlike the previous subsection, where we modeled the pEDFM and EDFM cases with a more coarse grid than the explicit case, we modeled all cases using the same grid in this case study. This allows us to isolate the errors that could be attributed to using coarser grids in EDFM and pEDFM (in comparison with the grids used in the explicit-fracture model).

Fig. 14 illustrates our use of geometrically spaced grids to capture the expected transient flow behavior near the fracture surfaces. To obtain the results shown in **Figs. 15 and 16**, we used a three-component mixture of methane (C_1), ethane (C_2), and propane (C_3) at mole fractions of 0.991, 0.0088, and 0.0002, respectively. **Fig. 15** compares the pEDFM, EDFM, and MRST explicit-fracture model results to the explicit-fracture model results from a commercial simulator and uses identical grids for the pEDFM, EDFM, and explicit-fracture models. However, in **Fig. 16**, we use a structured grid for the pEDFM and EDFM model results, as is typical in these models, and show the same MRST explicit-fracture model results from **Fig. 15** for reference. The differences between the pEDFM and EDFM results in **Fig. 15** and those in **Fig. 16** are entirely because the structured grids (used in **Fig. 16**) are unable to capture the transient behavior near the fracture surfaces. We observe that EDFM significantly overestimates production in both **Figs. 15 and 16**, and this implies that the overestimation of the production is not attributable to the coarseness of the structured grid used in **Fig. 16**. However, the 6% underestimation of the expected ultimate recovery (EUR) by the pEDFM model in **Fig. 16** is entirely because the coarse structured grid used in this case is unable to account for the sharp pressure drops expected near the fracture surfaces.

To evaluate the accuracy of our 3D pEDFM approach in a truly compositional case with multiple fluid phases, we use the Bakken Shale compositional data given by Nojabaei (2015) instead of the three-component gas mixture used to obtain **Figs. 15 and 16**. **Fig. 17** shows the EDFM, pEDFM, and explicit-fracture model pressure profiles obtained on the left, middle, and right, respectively, and **Fig. 18** provides the corresponding saturation profiles. The pEDFM approach predicts a pressure profile that matches that from the

high-resolution explicit-fracture model. Thus, it shows that pEDFM can correctly account for the effect of the two flow barriers, whereas EDFM is unable to account for the presence of the flow barriers in the domain, although a high-resolution grid is used in the matrix (for all model results in Figs. 17 and 18). Similarly, the oil-saturation profiles presented in Fig. 18 show that pEDFM is able match the reference explicit-fracture solution, whereas EDFM cannot account for the flow barriers accurately. We do not show the corresponding gas saturation because this is a two-phase (oil-gas) simulation, and the gas-saturation values are simply 1 minus the oil saturation values presented.

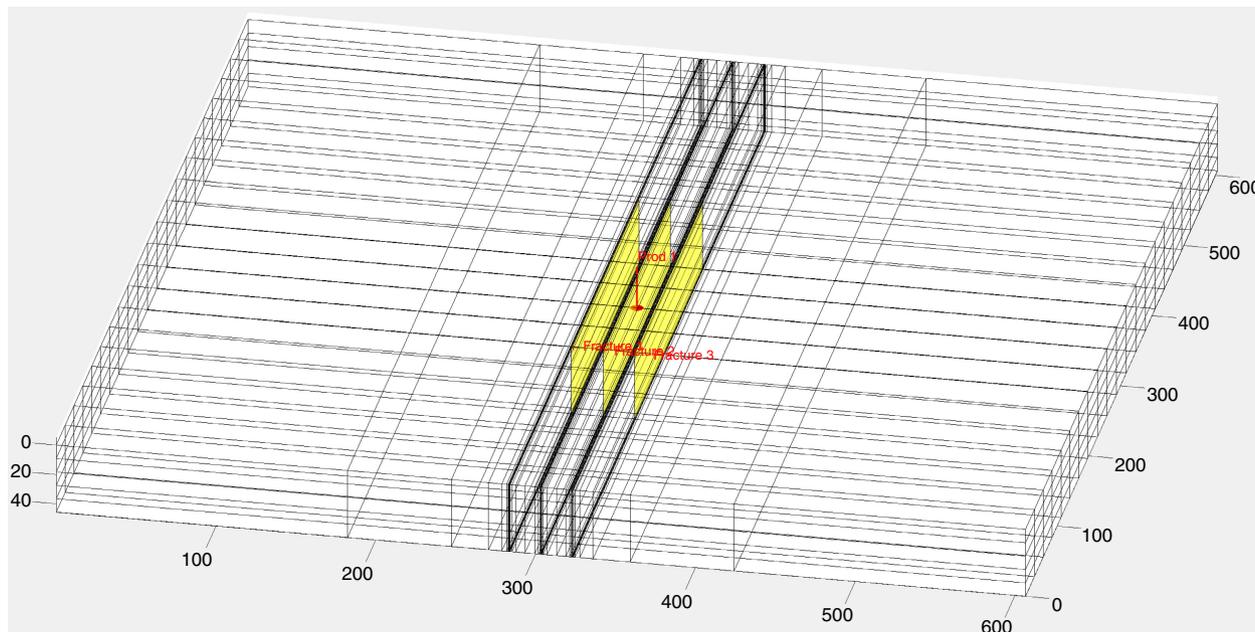


Fig. 14—All three fractures are marked in yellow. The middle fracture is the hydraulic fracture, which is connected to a well, whereas the other two fractures are modeled as sealing flow barriers with low fracture-permeability values.

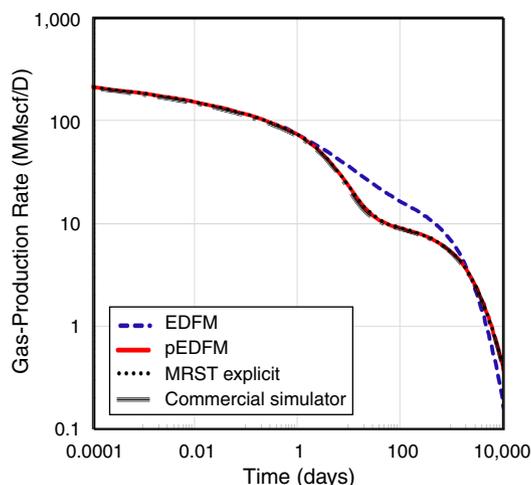


Fig. 15—Gas-production rates show that the pEDFM and MRST explicit-fracture models match the explicit-fracture simulation results from a commercial simulator, but EDFM overestimates gas production. The cumulative gas production shows a perfect match between the pEDFM and explicit-fracture models, whereas EDFM overestimates the gas EUR by 9%.

Fig. 19 shows the significance of accurately modeling the flow barriers in terms of the oil- and gas-production rates. It is well known that log-log rate plots tend to compress the late-time production profile, so we provide the corresponding oil and gas cumulative production on a Cartesian plot in **Fig. 20**. The results show that EDFM overestimates the EURs of oil and gas by 38% and 67%, respectively. This is because it is unable to account for the presence of the flow barriers in the domain. Because we use the same grid as in the explicit-fracture case, all of this error can be attributed to the EDFM model. Our 3D pEDFM model, in contrast, matches the explicit fracture simulation results perfectly. To clarify, the results presented in Figs. 19 and 20 were obtained using the same grid shown in Fig. 14, but the Bakken fluid was used instead of the C_1 , C_2 , C_3 gas mixture used in Fig. 15.

Applications of 3D pEDFM to UOG Reservoir Case Studies

Here, we discuss how to apply our 3D pEDFM to model realistic fractured-reservoir systems that could have thousands of natural fractures in any arbitrary orientations in 3D. The focus is on the modeling of UOG reservoirs, which are typically naturally fractured and

tend to exhibit compositional effects. Although we focus on the modeling of specific realizations of a stochastic fracture network, we understand that there is currently no technology available to determine the exact location of all the natural fractures in the subsurface. Our goal is to provide a tool that can be used to study and evaluate the sensitivity of our model forecasts to the uncertainty in the amount, distribution, geometry, and orientation of these natural fractures.

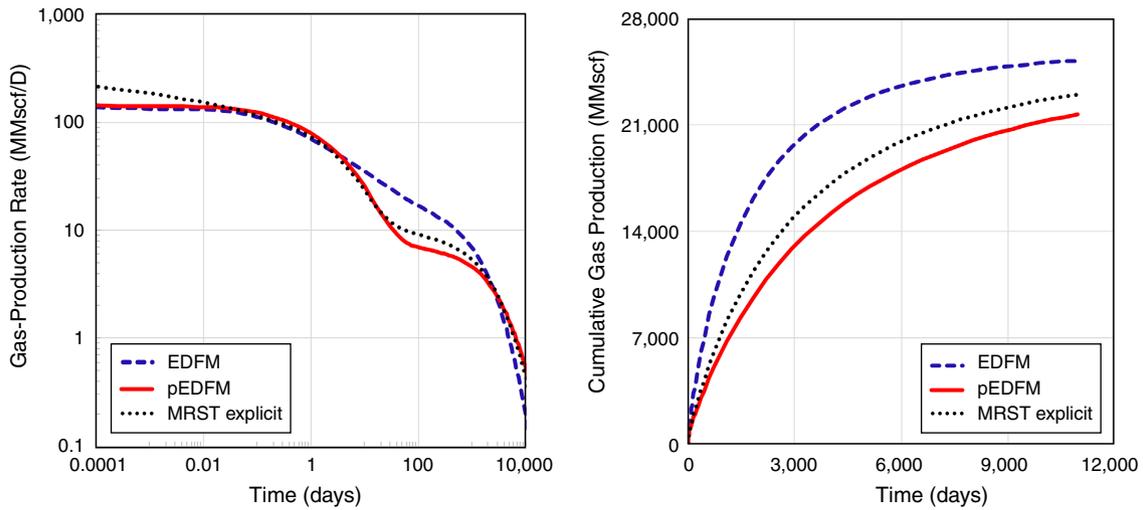


Fig. 16—This result shows that pEDFM underestimates production (in this case) because the structured grids used are coarse and unable to account for the sharp drops in pressure near the fracture. The pEDFM underestimates gas EUR by 6%, whereas EDFM overestimates it by 10%.

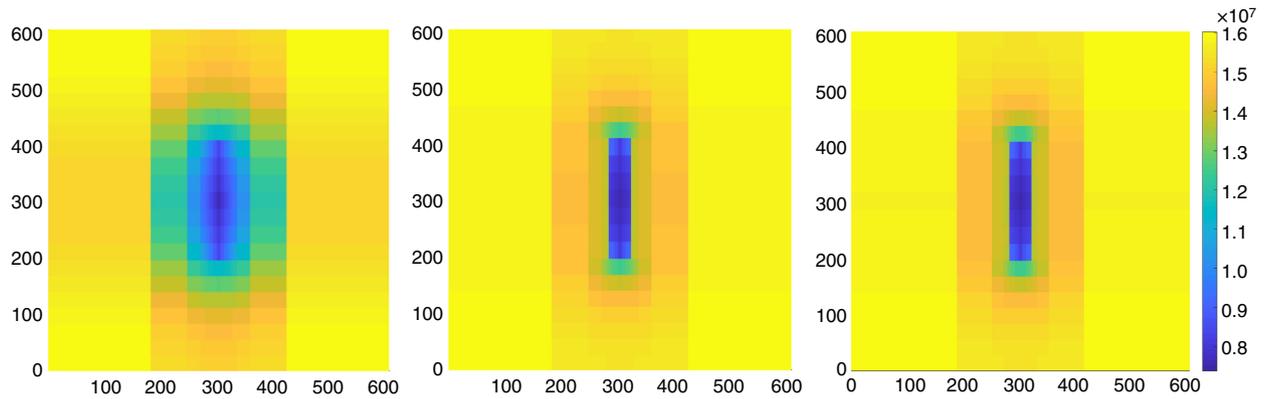


Fig. 17—This figure shows the EDFM, pEDFM, and explicit-fracture model pressure profiles on the left, middle, and right, respectively. The pEDFM pressure profile shows a perfect match with the explicit-fracture simulation results, whereas the EDFM pressure profile does not match the explicit-fracture model pressure profile because EDFM cannot account for the presence of the flow barrier.

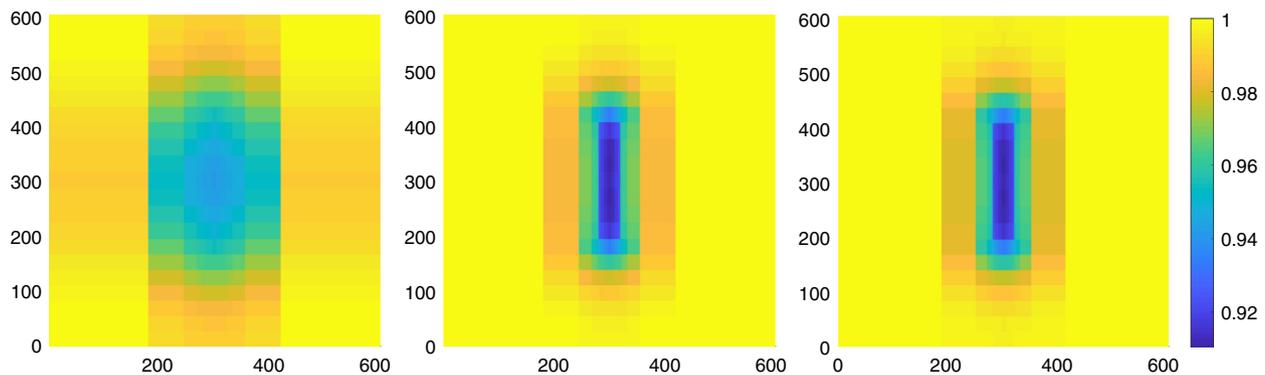


Fig. 18—This figure shows the EDFM, pEDFM, and explicit-fracture model oil-saturation profiles on the left, middle, and right, respectively. The pEDFM oil-saturation profile matches the explicit-fracture model saturation profile. As expected, EDFM oil saturation does not match the explicit-fracture simulation results because EDFM is unable to account for the presence of the flow barrier.

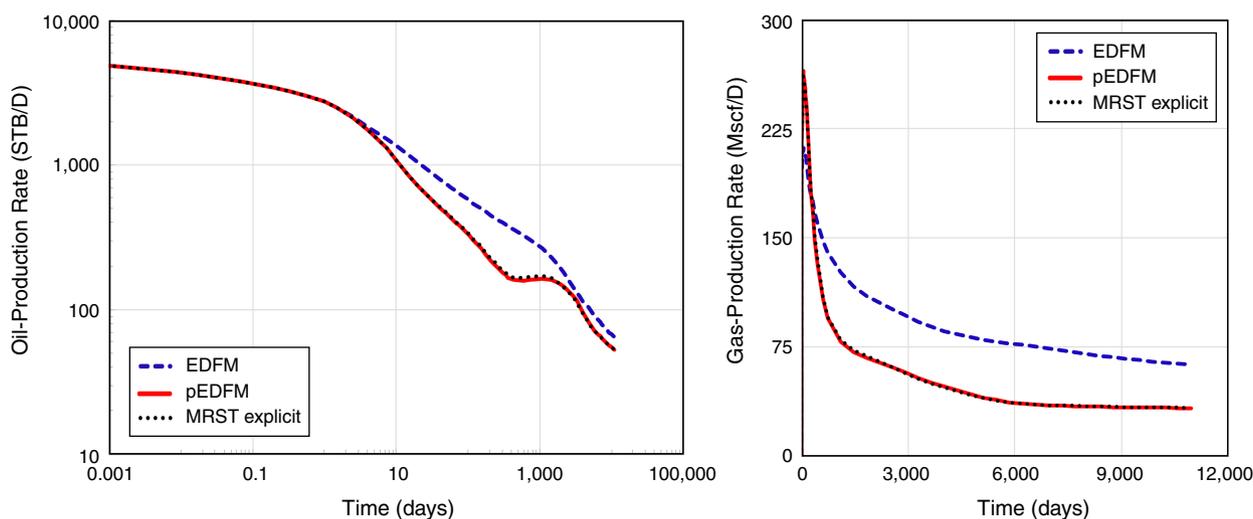


Fig. 19—Oil- and gas-production rates show that pEDFM matches the explicit-fracture simulation results, whereas EDFM overestimates both oil and gas production. To clarify, all results in this figure were obtained using the same mesh shown in Fig. 14, but the Bakken fluid was used instead of the C_1 , C_2 , C_3 gas mixture used in Fig. 15.

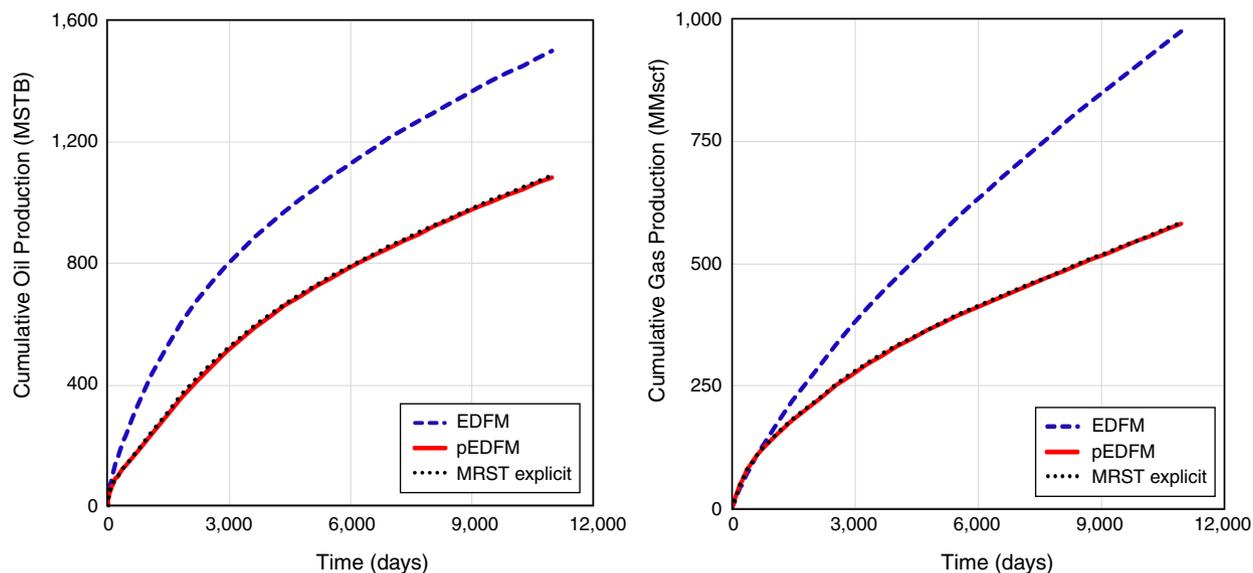


Fig. 20—Cumulative oil and gas production shows a perfect match between the pEDFM and explicit-fracture models, whereas the EDFM simulation overestimates the oil and gas EURs by 38% and 67%, respectively.

Stochastic Generation of Fractures with Arbitrary Orientations in 3D. To facilitate the generation of stochastic fracture networks in 3D, we use an opensource MATLAB code, called Alghalandis Discrete Fracture Network Engineering (Alghalandis Computing 2017). For a detailed tutorial on the stochastic generation of fractures in Alghalandis Discrete Fracture Network Engineering, we refer the reader to Alghalandis (2018). We used Alghalandis Discrete Fracture Network Engineering to generate 1,000 natural fractures with arbitrary orientations in 3D. Fig. 21 shows the resulting grid we obtained after meshing the hydraulic fractures and natural fractures. The figure shows a system with eight fracture stages and three fractures per stage. The horizontal well is shown as the blue line that runs through all the hydraulic fractures. A close examination of the natural fractures in Fig. 21 shows that there are two fracture sets at two different orientations. Half of the fractures were modeled as conductive faults, whereas the other half were modeled as flow barriers.

Bakken Shale Oil Reservoir Simulation. Table 2 summarizes the model parameters used in this Bakken shale case study. The compositional data and binary interaction constants used were obtained from Nojabaei (2015) and are tabulated in Appendix A. The simulation was run for a period of 30 years, and the production performance plots given in Figs. 22 and 23 show that EDFM tends to overestimate both oil- and gas-production rates and cumulative production. Fig. 22 presents the oil-production rates on a log-log plot and the associated gas-production rate on a Cartesian plot. The cumulative oil and gas production presented in Fig. 23 shows that EDFM overestimates the oil and gas EURs by 10% and 9%, respectively (in comparison with pEDFM). Recall that half of the natural fractures were modeled as flow barriers, so it is normal to expect EDFM to overestimate production in this case. However, it is important to note that the hydraulic fractures were modeled at the interface between matrix cells. This means that pEDFM simplifies to DFM in this case, but this is not the case with EDFM.

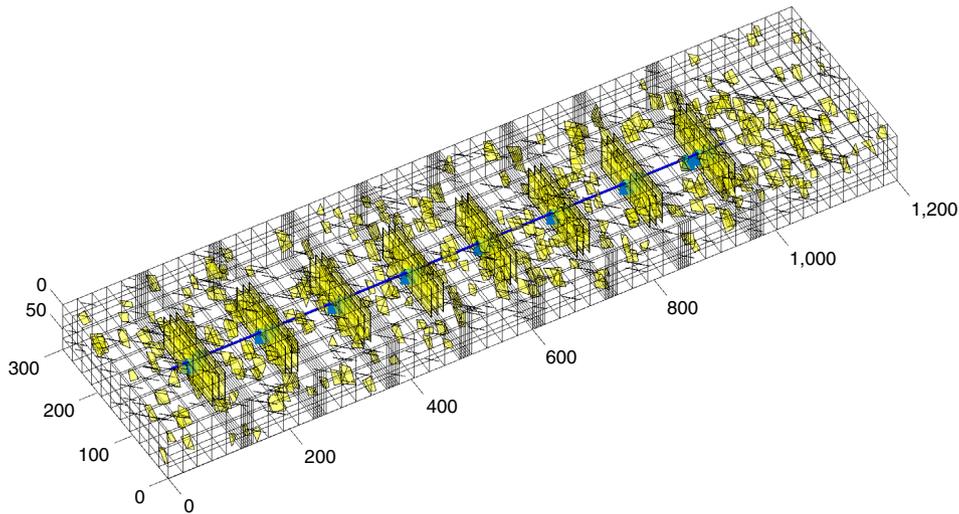


Fig. 21—This figure illustrates our simulation domain for the Bakken shale case study. It shows a system with eight fracture stages and three fractures per stage.

Parameters	Field Unit	SI Unit
Fracture half-length (x_f)	229.659 ft	70 m
Fracture width (w_f)	0.00984 ft	0.003 m
Reservoir thickness (h)	295.276 ft	90 m
Cluster spacing	10 ft	3.05 m
Fracture spacing	100 ft	30.5 m
Matrix permeability (k_m)	0.01 md	$9.9 \times 10^{-18} \text{ m}^2$
Fracture permeability (k_f)	1,000 md	$9.9 \times 10^{-14} \text{ m}^2$
Matrix porosity (ϕ)	0.04	0.04
Fracture porosity (ϕ_f)	0.5	0.5
Temperature (T)	175°F	352.594 K
Well radius (r_w)	0.32 ft	0.1 m
Water saturation (S_w)	0.23	0.23
Initial reservoir pressure (p_i)	5,700 psia	39,300,117 Pa
Flowing bottomhole pressure (p_{wf})	2,000 psia	$1.379 \times 10^7 \text{ Pa}$

Table 2—Representative Bakken Shale reservoir parameters.

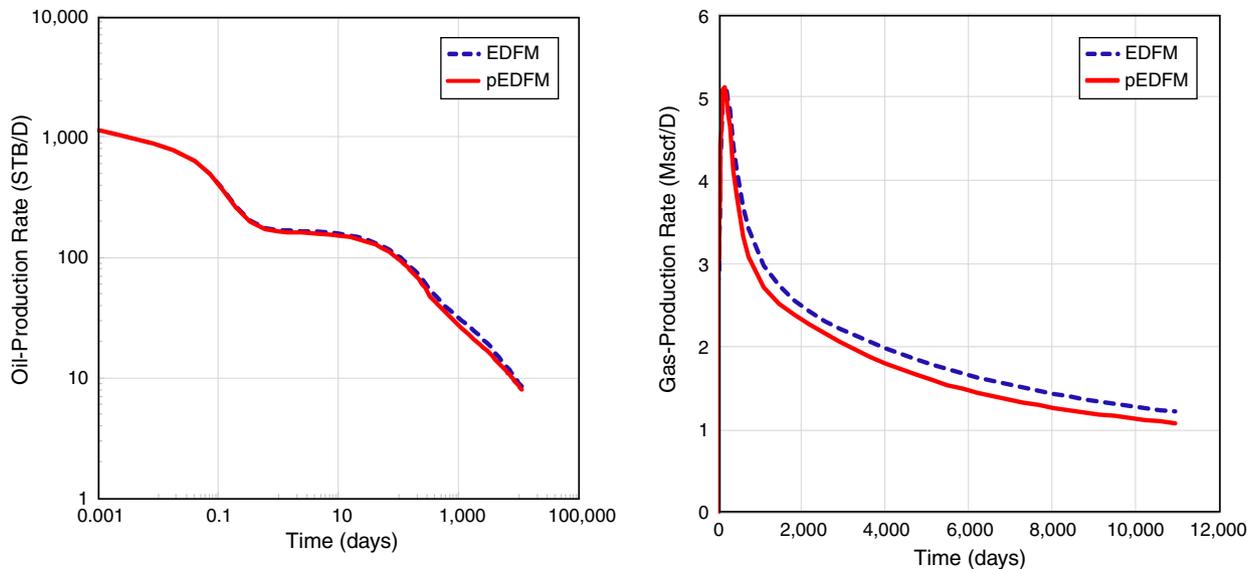


Fig. 22—Results show that the estimated oil- and gas-production rates from EDFM are higher than those from pEDFM.

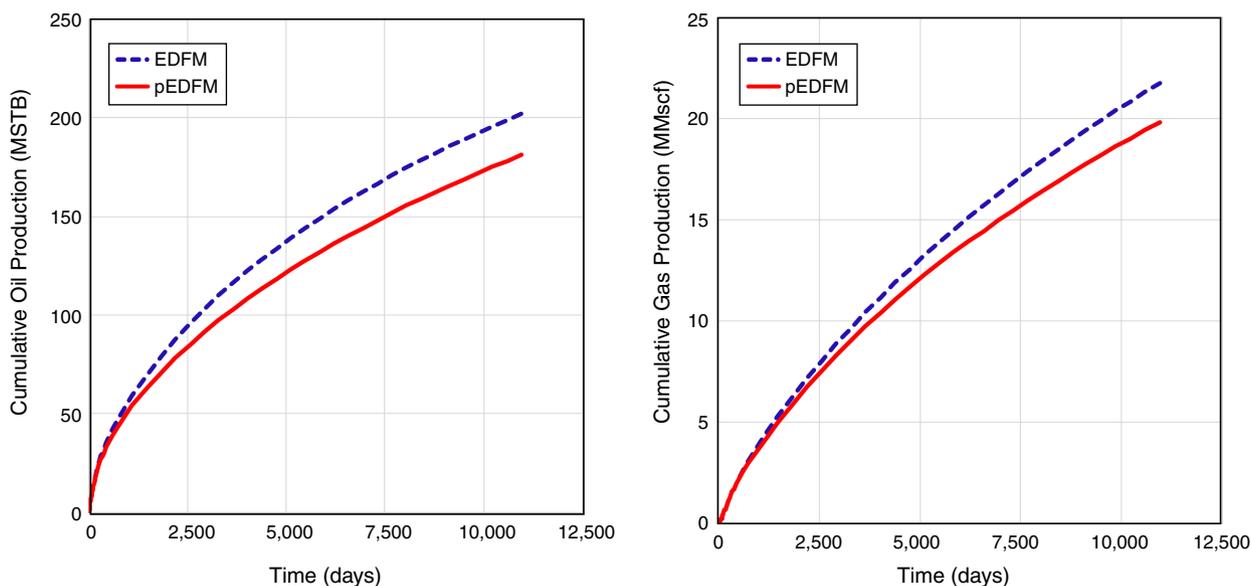


Fig. 23—Results show that EDFM overestimates the oil and gas EURs by 10% and 9%, respectively (in comparison to pEDFM).

Fig. 24 presents the pressure profile we obtained after visualizing our pEDFM simulation results in ParaView® (Ahrens et al. 2005). As expected, there is a sharp drop in pressure near the surfaces of the hydraulic fractures. Fig. 25 shows the corresponding water- and oil-saturation profiles obtained after 30 years of simulated production. The gas saturation is not shown because it is obtained as 1 minus the sum of the water- and oil-saturation values. In this simulation, it remains constant at the specified initial value of 7%. The saturation profiles are presented simply to show the capability of modeling three-phase compositional simulation using our 3D pEDFM model.

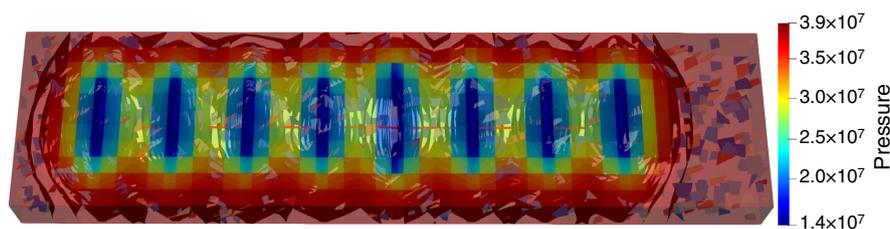


Fig. 24—Pressure profile (showing isocontours and grid) after 30 years of production.

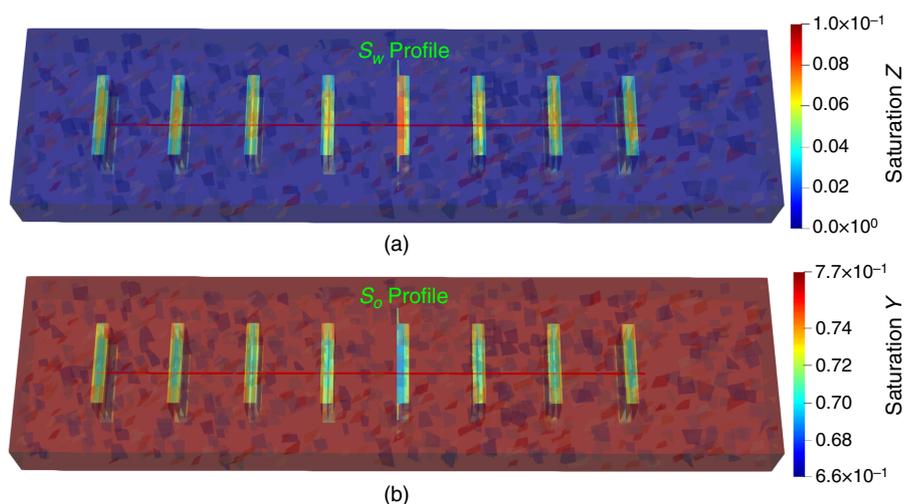


Fig. 25—Plot of (a) water- and (b) oil-saturation profiles after 30 years of production.

Although this Bakken shale oil case is a compositional simulation of eight hydrocarbon components, we only show the profiles for the first two hydrocarbon components in Fig. 26. This figure is included only to demonstrate the capability of simulating several hydrocarbon components and visualizing the mole fractions for each of these components.

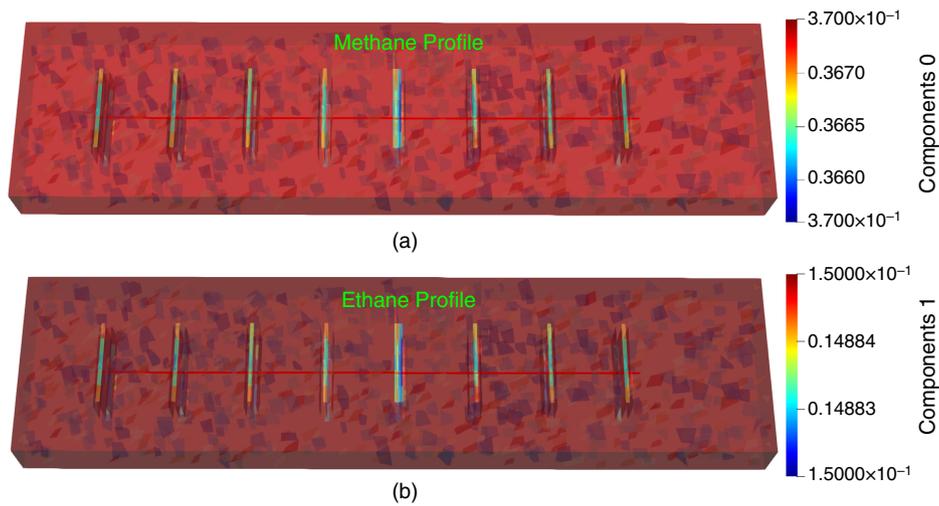


Fig. 26—Profiles of mole fraction of C_1 and C_2 after 30 years of production. The other six components are not shown for brevity.

Conclusions

This work addresses the issue of speed and accuracy in the modeling of realistic fractured UOG resources with the presentation of a robust 3D pEDFM algorithm. This algorithm was implemented in a compositional simulator and applied to model Bakken shale oil reservoir with more than 1,000 natural fractures in addition to the hydraulic fractures. Our simulation results indicate the feasibility of modeling several hydraulic and natural fractures (with arbitrary orientations) efficiently and accurately using a 3D pEDFM model.

The simulation results from this work quantify the errors introduced by the modeling of low-conductivity fractures with EDFM. We also provided several case studies that show the accuracy of the 3D pEDFM by comparing its production performance, pressure, and saturation profiles with those from a high-resolution reference solution. In all cases, the 3D pEDFM shows superior performance and is only slightly slower than the EDFM. This work can readily be applied to perform numerical studies of enhanced/improved oil recovery in fractured UOG reservoirs and fractured carbonate reservoirs with heterogeneous fracture properties.

Although we restricted our implementation of the pEDFM to structured hexahedral meshes, we realize that an extension of this algorithm to the more complex cases with corner-point and unstructured grids will further improve the usefulness of this work. For corner-point grids with six faces, our 3D algorithm could be easily adapted. However, an extension to fully unstructured grids and complex corner-point grids that pinch out (along fault surfaces) could be challenging. This is because the number of neighboring cells could be more or less than six in these cases. These further extensions of the presented algorithm will be a subject for future studies.

Nomenclature

- $A_{if \perp x}$ = area of fracture projections along each dimension, L^2 , m^2
 A_{ij}^p = projection area, L^2 , m^2
 A^{nnc} = area of a non-neighboring connection, L^2 , m^2
 d^{nnc} = non-neighboring connection distance, L , m
 f = fracture
 if = interaction between matrix cell i and fracture cell f
 i, j = cell indices
 k_f = fracture permeability, L^2 , m^2
 k_m = matrix permeability, L^2 , m^2
 k^{nnc} = permeability of a non-neighboring connection, L^2 , m^2
 $k + 1$ = current timestep
 m = matrix
 $M - M$ = interaction between two different host matrix cells
 nnc = non-neighboring connections
 $pM - F$ = interaction a projection matrix and a fracture cell
 R = residual, $MT^{-1}L^{-3}$, $kg^{-1}m^{-3}$
 S_α = saturation of phase, α
 T_i = half transmissibility, L^3 , m^3
 T^{nnc} = transmissibility factor for an NNC, L^3 , m^3
 \vec{x} = X -, Y -, and Z -coordinates, L , m
 \vec{x}_e = gridblock sizes in the X -, Y -, and Z -directions, L , m
 X_i = mass fraction of component i in the liquid phase
 Y_i = mass fraction of component i in the gas phase
 Z_i = overall mass fraction of component i
 α = fluid phase
 ϕ = porosity
 ρ = density, M/L^3 , kg/m^3

Acknowledgments

The authors recognize the support of the Craft and Hawkins Department of Petroleum Engineering at Louisiana State University. This work was facilitated using the computational resources and software packages available in the department and at the Center for Computation and Technology. We also appreciate the comments and suggestions from all the reviewers.

References

- Acs, G., Doleschall, S., and Farkas, E. 1985. General Purpose Compositional Model. *SPE J.* **25**: 543–553. SPE-10515-PA. <https://doi.org/10.2118/10515-PA>.
- Ahrens, J., Geveci, B., and Law, C. 2005. Paraview: An End-User Tool for Large Data Visualization. In *The Visualization Handbook*, C. D. Hansen and C. R. Johnson (eds.), Chapter 36, 717–731. Burlington, Massachusetts, USA: Elsevier Butterworth-Heinemann. <https://doi.org/10.1016/B978-012387582-2/50038-1>.
- Alghalandis Computing. 2017. Alghalandis Discrete Fracture Network Engineering (ADFNE), <https://alghalandis.net/products/adfne/> (accessed June 2020).
- Alghalandis, Y. F. 2018. *DFNE Practices with ADFNE*. Toronto, Ontario, Canada: Alghalandis Computing.
- Aziz, K. and Settari, A. 1979. *Petroleum Reservoir Simulation*. London, UK: Applied Science Publishers.
- Bell, I. H., Wronski, J., Quoilin, S. et al. 2014. Pure and Pseudo-Pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library Coolprop. *Indust & Eng Chem Res* **53** (6): 2498–2508. <https://doi.org/10.1021/ie4033999>.
- Cao, H. 2002. *Development of Techniques for General Purpose Simulators*. PhD thesis, Stanford University, Stanford, California, USA.
- Chien, M. C. H., Lee, S. T., and Chen, W. H. 1985. A New Fully Implicit Compositional Simulator. Paper presented at the SPE Reservoir Simulation Symposium, Dallas, Texas, USA, 10–13 February. SPE-13385-MS. <https://doi.org/10.2118/13385-MS>.
- Coats, K. H. 1980. An Equation of State Compositional Model. *SPE J.* **20** (5): 363–376. SPE-8284-PA. <https://doi.org/10.2118/8284-PA>.
- Collins, D. A., Nghiem, L. X., Li, Y. K. et al. 1992. An Efficient Approach to Adaptive-Implicit Compositional Simulation with an Equation of State. *SPE Res Eng* **7** (2): 259–264. SPE-15133-PA. <https://doi.org/10.2118/15133-PA>.
- Devegowda, D., Sapmanee, K., Civan, F. et al. 2012. Phase Behavior of Gas Condensates in Shales due to Pore Proximity Effects: Implications for Transport, Reserves and Well Productivity. Paper presented at the SPE Annual Technical Conference and Exhibition, San Antonio, Texas, USA, 8–10 October. SPE-160099-MS. <https://doi.org/10.2118/160099-MS>.
- Diaz Campos, M., Akkutlu, I. Y., and Sigal, R. F. 2009. A Molecular Dynamics Study on Natural Gas Solubility Enhancement in Water Confined to Small Pores. Paper presented at the SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, USA, 4–7 October. SPE-124491-MS. <https://doi.org/10.2118/124491-MS>.
- Firoozabadi, A. 2015. *Thermodynamics and Applications in Hydrocarbon Energy Production*. New York, New York, USA: McGraw-Hill Education.
- Fumagalli, A., Pasquale, L., Zonca, S. et al. 2016. An Upscaling Procedure for Fractured Reservoirs with Embedded Grids. *Water Res Res* **52** (8): 6506–6525. <https://doi.org/10.1002/2015WR017729>.
- Fussell, L. T. and Fussell, D. D. 1979. An Iterative Technique for Compositional Reservoir Models. *SPE J.* **19** (4): 211–220. SPE-6891-PA. <https://doi.org/10.2118/6891-PA>.
- Geiger, S., Dentz, M., and Neuweiler, I. 2013. A Novel Multi-Rate Dual-Porosity Model for Improved Simulation of Fractured and Multiporosity Reservoirs. *SPE J.* **18** (4): 670–684. SPE-148130-PA. <https://doi.org/10.2118/148130-PA>.
- Gillespie, P., Howard, C., Walsh, J. et al. 1993. Measurement and Characterisation of Spatial Distributions of Fractures. *Tectonophysics* **226** (1–4): 113–141. [https://doi.org/10.1016/0040-1951\(93\)90114-Y](https://doi.org/10.1016/0040-1951(93)90114-Y).
- Gong, B. 2007. *Effective Models of Fractured Systems*. PhD thesis, Stanford University, Stanford, California, USA.
- Gong, B., Thakur, G., Qin, G. et al. 2017. Application of Multi-Level and High-Resolution Fracture Modeling in Field-Scale Reservoir Simulation Study. Paper presented at the SPE Reservoir Characterization and Simulation Conference and Exhibition, Abu Dhabi, UAE, 8–10 May. SPE-186068-MS. <https://doi.org/10.2118/186068-MS>.
- Hajibeygi, H., Karvounis, D., and Jenny, P. 2011. A Hierarchical Fracture Model for the Iterative Multiscale Finite Volume Method. *J Comput Physics* **230** (24): 8729–8743. <https://doi.org/10.1016/j.jcp.2011.08.021>.
- Jiang, J. and Younis, R. M. 2017. An Improved Projection-Based Embedded Discrete Fracture Model (pEDFM) for Multiphase Flow in Fractured Reservoirs. *Adv Water Res* **109**: 267–289. <https://doi.org/10.1016/j.advwatres.2017.09.017>.
- Karimi-Fard, M. and Firoozabadi, A. 2001. Numerical Simulation of Water Injection in 2D Fractured Media Using Discrete-Fracture Model. Paper presented at the SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, USA, 30 September–3 October. SPE-71615-MS. <https://doi.org/10.2118/71615-MS>.
- Kim, J.-G. and Deo, M. D. 2000. Finite Element, Discrete-Fracture Model for Multiphase Flow in Porous Media. *AIChE J* **46** (6): 1120–1130. <https://doi.org/10.1002/aic.690460604>.
- Lee, S. H., Jensen, C. L., and Lough, M. F. 1999. An Efficient Finite Difference Model for Flow in a Reservoir with Multiple Length-Scale Fractures. Paper presented at the SPE Annual Technical Conference and Exhibition, Houston, Texas, USA, 3–6 October. SPE-56752-MS. <https://doi.org/10.2118/56752-MS>.
- Lee, S. H., Lough, M. F., and Jensen, C. L. 2001. Hierarchical Modeling of Flow in Naturally Fractured Formations with Multiple Length Scales. *Water Res Res* **37** (3): 443–455. <https://doi.org/10.1029/2000WR900340>.
- Li, L. and Lee, S. H. 2008. Efficient Field-Scale Simulation of Black Oil in a Naturally Fractured Reservoir through Discrete Fracture Networks and Homogenized Media. *SPE Res Eval & Eng* **11** (4): 750–758. SPE-103901-PA. <https://doi.org/10.2118/103901-PA>.
- Li, L. and Voskov, D. 2018. Multi-Level Discrete Fracture Model for Carbonate Reservoirs. In *Conference Proceedings, ECMOR XVI - 16th European Conference on the Mathematics of Oil Recovery*, September 2018, Volume 2018, 1–17. Houten, The Netherlands: European Association of Geoscientists & Engineers. <https://doi.org/10.3997/2214-4609.201802164>.
- Li, W., Dong, Z., and Lei, G. 2017. Integrating Embedded Discrete Fracture and Dual-Porosity, Dual-Permeability Methods To Simulate Fluid Flow in Shale Oil Reservoirs. *Energies* **10** (10): 1471. <https://doi.org/10.3390/en10101471>.
- Lie, K.-A. 2019. *An Introduction to Reservoir Simulation Using MATLAB/GNU Octave: User Guide for the MATLAB Reservoir Simulation Toolbox (MRST)*. Cambridge, Massachusetts, USA: Cambridge University Press.
- Moinfar, A. 2013. *Development of an Efficient Embedded Discrete Fracture Model for 3D Compositional Reservoir Simulation in Fractured Reservoirs*. PhD thesis, University of Texas at Austin, Austin, Texas, USA.
- Moinfar, A., Varavei, A., Sepehrmoori, K. et al. 2013. Development of a Coupled Dual Continuum and Discrete Fracture Model for the Simulation of Unconventional Reservoirs. Paper presented at the SPE Reservoir Simulation Symposium, The Woodlands, Texas, USA, 18–20 February. SPE-163647-MS. <https://doi.org/10.2118/163647-MS>.
- Møyner, O. and Tchelepi, H. 2017. A Multiscale Restriction-Smoothed Basis Method for Compositional Models. Paper presented at the SPE Reservoir Simulation Conference, Montgomery, Texas, USA, 20–22 February. SPE-182679-MS. <https://doi.org/10.2118/182679-MS>.
- Nojabaei, B. 2015. *Phase Behavior and Flow Analysis of Shale Reservoirs Using a Compositionally Extended Black-Oil Approach*. PhD thesis, Pennsylvania State University, State College, Pennsylvania, USA.
- Norbeck, J. H., McClure, M. W., Lo, J. W. et al. 2016. An Embedded Fracture Modeling Framework for Simulation of Hydraulic Fracturing and Shear Stimulation. *Comput Geosci* **20** (1): 1–18. <https://doi.org/10.1007/s10596-015-9543-2>.
- Odling, N., Gillespie, P., Bourguine, B. L. et al. 1999. Variations in Fracture System Geometry and Their Implications for Fluid Flow in Fractures Hydrocarbon Reservoirs. *Pet Geosci* **5** (4): 373–384. <https://doi.org/10.1144/petgeo.5.4.373>.
- Odling, N. E. 1997. Scaling and Connectivity of Joint Systems in Sandstones from Western Norway. *J Struct Geol* **19** (10): 1257–1271. [https://doi.org/10.1016/S0191-8141\(97\)00041-2](https://doi.org/10.1016/S0191-8141(97)00041-2).

- Olorode, O. M., Freeman, C. M., Moridis, G. J. et al. 2013. High-Resolution Numerical Modeling of Complex and Irregular Fracture Patterns in Shale-Gas Reservoirs and Tight Gas Reservoirs. *SPE Res Eval & Eng* **16** (4): 443–455. SPE-152482-PA. <https://doi.org/10.2118/152482-PA>.
- Peng, D.-Y. and Robinson, D. B. 1976. A New Two-Constant Equation of State. *Indust Eng Chem Fundamentals* **15** (1): 59–64. <https://doi.org/10.1021/i160057a011>.
- Pruess, K. and Narasimhan, T. N. 1985. A Practical Method for Modeling Fluid and Heat Flow in Fractured Porous Media. *SPE J.* **25** (1): 14–26. SPE-10509-PA. <https://doi.org/10.2118/10509-PA>.
- Rao, X., Cheng, L., Cao, R. et al. 2020. A Modified Projection-Based Embedded Discrete Fracture Model (pEDFM) for Practical and Accurate Numerical Simulation of Fractured Reservoir. *J Pet Sci & Eng* **187**: 106852. <https://doi.org/10.1016/j.petrol.2019.106852>.
- Rateman, K. T., Farrell, H. E., Mora, O. S. et al. 2018. Sampling a Stimulated Rock Volume: An Eagle Ford Example. *SPE J.* **21** (4): 927–941. SPE-191375-PA. <https://doi.org/10.2118/191375-PA>.
- Shafiei, A., Dusseault, M. B., Kosari, E. et al. 2018. Natural Fractures Characterization and In Situ Stresses Inference in a Carbonate Reservoir—An Integrated Approach. *Energies* **11** (2): 312. <https://doi.org/10.3390/en11020312>.
- Ṭene, M., Bosma, S. B. M., Al Kobaisi, M. S. et al. 2017. Projection-Based Embedded Discrete Fracture Model (pEDFM). *Adv Water Res* **105**: 205–216. <https://doi.org/10.1016/j.adwatres.2017.05.009>.
- Ṭene, M. 2018. *Algebraic Multiscale Framework for Fractured Reservoir Simulation*. PhD thesis, Delft University of Technology, Delft, The Netherlands.
- U.S. Energy Information Administration. 2019. *Annual Energy Outlook 2019 with Projections to 2050*. Washington, DC, USA: U.S. Energy Information Administration.
- Voskov, D. V. and Tchalepi, H. A. 2012. Comparison of Nonlinear Formulations for Two-Phase Multi-Component EoS-Based Simulation. *J Pet Sci & Eng* **82–83**: 101–111. <https://doi.org/10.1016/j.petrol.2011.10.012>.
- Warren, J. E. and Root, P. J. 1963. The Behavior of Naturally Fractured Reservoirs. *SPE J.* **3** (3): 245–255. SPE-426-PA. <https://doi.org/10.2118/426-PA>.
- Xu, Y., Fernandes, B. R. B., Marcondes, F. et al. 2019. Embedded Discrete Fracture Modeling for Compositional Reservoir Simulation Using Corner-Point Grids. *J Pet Sci & Eng* **177**: 41–52. <https://doi.org/10.1016/j.petrol.2019.02.024>.
- Yaws, C. L. 1999. *Chemical Properties Handbook*. New York, New York, USA: McGraw-Hill.
- Young, L. C. and Stephenson, R. E. 1983. A Generalized Compositional Approach for Reservoir Simulation. *SPE J.* **23** (5): 727–742. SPE-10516-PA. <https://doi.org/10.2118/10516-PA>.
- Yu, W., Zhang, Y., Varavei, A. et al. 2019. Compositional Simulation of CO₂ Huff-n-Puff in Eagle Ford Tight Oil Reservoirs with CO₂ Molecular Diffusion, Nanopore Confinement, and Complex Natural Fractures. *SPE Res Eval & Eng* **22** (2): 492–508. SPE-190325-PA. <https://doi.org/10.2118/190325-PA>.
- Zhang, Y., Yu, W., Sepehrmoori, K. et al. 2017. Investigation of Nanopore Confinement on Fluid Flow in Tight Reservoirs. *J Pet Sci & Eng* **150**: 265–271. <https://doi.org/10.1016/j.petrol.2016.11.005>.
- Zimmerman, R. W., Chen, G., Hadgu, T. et al. 1993. A Numerical Dual-Porosity Model with Semianalytical Treatment of Fracture/Matrix Flow. *Water Res Res* **29** (7): 2127–2137. <https://doi.org/10.1029/93WR00749>.
- Zoback, M. D. and Kohli, A. H. 2019. *Unconventional Reservoir Geomechanics: Shale Gas, Tight Oil, and Induced Seismicity*. Cambridge, Massachusetts, USA: Cambridge University Press.

Appendix A—Chemical Properties for Compositional Simulation

Table A-1 provides the compositional data, and **Table A-2** presents the binary interaction constants for the Bakken shale oil. Both tables were culled from Nojabaei (2015). For the other simple fluid mixtures in this work, we use the chemical properties from Yaws (1999) and Bell et al. (2014).

Components	Mole Fraction	Critical Pressure (psia)	Critical Temperature (R)	Acentric Factor	Molar Weight (lb/lb-mol)	Critical Volume (ft ³ /lb-mol)	Parachor
C ₁	0.36736	655.02	335.336	0.0102	16.535	1.58	74.8
C ₂	0.14885	721.99	549.969	0.1028	30.433	2.34	107.7
C ₃	0.09334	615.76	665.97	0.152	44.097	3.25	151.9
C ₄	0.05751	546.46	759.208	0.1894	58.124	4.11	189.6
C ₄ –C ₆	0.06406	461.29	875.479	0.2684	78.295	5.39	250.2
C ₇ –C ₁₂	0.15854	363.34	1,053.25	0.4291	120.562	8.81	350.2
C ₁₃ –C ₂₁	0.0733	249.61	1,332.095	0.7203	220.716	15.19	590.0
C ₂₂ –C ₈₀	0.03704	190.12	1,844.491	1.0159	443.518	36	1,216.8

Table A-1—Compositional data for Bakken shale oil.

	C ₁	C ₂	C ₃	C ₄	C ₅ –C ₆	C ₇ –C ₁₂	C ₁₃ –C ₂₁	C ₂₂ –C ₈₀
C ₁	0	0.005	0.0035	0.0035	0.0037	0.0033	0.0033	0.0033
C ₂	0.005	0	0.0031	0.0031	0.0031	0.0026	0.0026	0.0026
C ₃	0.0035	0.0031	0	0	0	0	0	0
C ₄	0.0035	0.0031	0	0	0	0	0	0
C ₅ –C ₆	0.0037	0.0031	0	0	0	0	0	0
C ₇ –C ₁₂	0.0033	0.0026	0	0	0	0	0	0
C ₁₃ –C ₂₁	0.0033	0.0026	0	0	0	0	0	0
C ₂₂ –C ₈₀	0.0033	0.0026	0	0	0	0	0	0

Table A-2—Binary interaction constants for Bakken shale oil.