Localized Nonlinear Solution Strategies for Efficient Simulation of Unconventional Reservoirs

Jiamin Jiang
Chevron Energy Technology Co.
1500 Louisiana St., Houston, TX 77002, USA

Abstract

Accurate and efficient numerical simulation of unconventional reservoirs is challenging. Long period of transient flow and steep potential gradients occur due to the extreme conductivity contrast between matrix and fracture. Detailed near-well/near-fracture models are necessary to provide sufficient resolution, but they are computationally impractical for field cases with multiple hydraulic-fracture stages.

Previous works in the literature of unconventional simulations mainly focus on the gridding level that adapts to wells and fractures. Limited research has been conducted on nonlinear strategies that exploit locality across timesteps and nonlinear iterations. It was reported that an individual Newton update is typically sparse and nonlinear convergence is constrained by a small portion of the model. To perform localized computations, an a-priori strategy is essential to first determine the active subset of simulation cells for the subsequent iteration. The active set flags the cells that will be updated, and then the corresponding localized linear system is solved.

The objective of this work is to develop localization methods that are readily applicable to complex fracture networks and flow physics in unconventional reservoirs. By utilizing the diffusive nature of pressure updates, an adaptive algorithm is proposed to make adequate estimates for the active domains. In addition, we further develop a localized solver based on nonlinear domain decomposition (DD). Comparing to a standard DD method, domain partitions are dynamically constructed. The new solver provides effective partitioning that adapts to flow dynamics and Newton updates.

We evaluate the developed methods using several complex problems with discrete fracture networks. The problems consider multi-phase and compositional fluid systems with phase changes. The results show that large degrees of solution locality present across timesteps and iterations. Comparing to a standard Newton solver, the new solvers enable superior computational performance. Moreover, the Newton convergence behavior is preserved, without any impact on solution accuracy.

1. Introduction

Unconventional reservoirs have received great attention as a primary energy resource in the past decade worldwide. Economic production from these reservoirs depends on effective stimulation by means of hydraulic fracturing. Micro-seismic measurements and other evidence suggest the creation of complex fracture networks that connect huge reservoir surface areas to the wellbore (Maxwell et al. 2002; Fisher et al. 2002; Mayerhofer et

*Corresponding author
Email address: jiamin.jiang@chevron.com (Jiamin Jiang)
In terms of reservoir development and management, numerical simulation continues to play a critical role in evaluating and optimizing the stimulation and production processes (Cipolla et al. 2010b; Weng et al. 2014). Because of the ultra-low permeability of matrix, a long period of transient flow occurs in unconventional formation. The extreme contrast in conductivity between matrix and fracture also results in steep potential gradients that are difficult to capture. Therefore, detailed near-well/near-fracture models are necessary to provide sufficient resolution for the matrix-fracture interactions (Mayerhofer et al. 2010; Cipolla et al. 2010a; Cipolla et al. 2010b). However, a fine grid simulation requires too much CPU time and it is impractical to perform for the entire domain in field cases with multiple hydraulic-fracture stages (Ding et al. 2014; Weng et al. 2014; Panfili et al. 2015).

Several modeling approaches for fractured-well have been proposed in the literature, attempting to improve the solution efficiency while maintaining the accuracy. One simple approach is applying local grid refinement (LGR) on a coarse background grid (Mallison et al. 2010; Cipolla et al. 2010b; Artus and Fructus 2012; Panfili et al. 2015; Jiang and Younis 2015b; Jiang and Younis 2016; Xue et al. 2019). A number of meshing algorithms are available to generate adaptive and optimized mesh with good quality around discrete fractures. Pruess and Narasimhan (1985) developed a sub-gridding method called multiple interacting continua (MINC) by subdividing each matrix cell according to the distance from fractures. The MINC method is intended to improve the classical dual-porosity model (Warren and Root 1963; Kazemi et al. 1976), and subsequently better characterize the transient effects with large solution gradients. In recent years, MINC is widely applied and extended for the simulations of unconventional reservoirs (Hui et al. 2013; Wu et al. 2014; Jiang and Younis 2015a; Jiang and Younis 2016; Farah et al. 2015; Ren et al. 2016; Ding et al. 2018). In addition, Ding et al. (2014) proposed a coupled modeling method that combines a coarse-grid reservoir model with detailed near-fracture models. The two models are solved, and the associated boundary conditions are updated in an alternate mode. The coupled method can be viewed as dynamic upscaling that computes the time-dependent fracture index for the coarse domain. Although some promising results were presented, the detailed models may still take a large fraction of the overall computational expense.

In reservoir simulation, the Fully Implicit Method (FIM) is often the method of choice for the temporal discretization of the conservation equations (Aziz and Settari 1979; Coats 1980). FIM offers unconditional stability, but it requires the solution of large coupled nonlinear systems. For a target timestep, a sequence of Newton iterations is performed until convergence. This iterative process is expensive and can account for a significant fraction of the total cost. Here define n as the total number of degrees of freedom in a system. Considering the costs of computing the residual vector, Jacobian matrix, and thermodynamic properties, the overall complexity of a nonlinear iteration is generally superlinear in n (Younis et al. 2010).

Previous works in the literature of unconventional simulations mainly focus on the gridding level that adapts to wells and fractures. Limited research has been conducted on solver techniques that exploit locality across timesteps and nonlinear iterations. The benefit of this type of methods is evident: the solution accuracy is maintained, because neither discretization scheme nor spatial mesh will be modified (Younis et al. 2010; Sheth and Younis 2017). Simulation studies have shown that flow dynamics evolve quite slowly within the ultra-tight formation. Pressure drop may remain in the vicinity of fractures even after years of production (Cipolla et al. 2010b; Ding et al. 2014; Jiang and Younis 2015b). Conceivably, a significant speedup can be achieved if performing adaptive computations only for the locales that are undergoing changes. In addition to the timestep level, a large degree of locality also presents on the nonlinear (Newton) level. It was reported that an individual Newton update is typically sparse and the nonlinear convergence is constrained by a small portion of the model (Younis et al. 2010; Lu and Beckner 2011; Sheth and Younis 2017). To exploit the locality at each iteration, an a priori
strategy is essential to first identify the active subset of simulation cells. The active set flags the cells that will be updated, and then the corresponding localized linear system is solved. Lu and Beckner (2011) observed that over the course of several iterations, the sparsity pattern of the Newton updates was related to that of the discrete residual vector. They proposed to use non-zero entries in the residual vector as an estimate of the active set for the subsequent iteration. It should be mentioned that their heuristic strategy may suffer from an efficiency issue due to overly conservative estimate. Sheth and Younis (2017) have shown that missing any non-zero update during the localization process may lead to worse nonlinear convergence, comparing to the standard Newton method. A theoretical framework was then developed to predict the sparsity pattern of Newton updates. Analytical derivations were made to ensure a conservative estimate of the active sets. The results in Sheth and Younis (2017) demonstrated that their localization method performs quite well for several challenging models.

In this work we do not intend to rely on an analytical derivation or conservative estimate for the active sets. The objective is to develop localization methods that readily accommodate to complex fracture networks and flow physics for the simulations of unconventional reservoirs. Through aggressive localization, the computational speedup is expected to be greatly improved. We recently reveal that the Newton updates for pressure-driven problems exhibit diffusive and global behaviors, because of its parabolic nature (Jiang and Tchelepi 2018). By utilizing the nature of pressure updates, an adaptive algorithm is proposed to make adequate estimates for the active domains. In addition, we further develop a localized solver based on nonlinear domain decomposition (DD). Comparing to a standard DD method, domain partitions are dynamically constructed from the previous iterations. During the nonlinear DD process, the subproblems of an iteration are solved sequentially, and thus the localization can be naturally achieved. This leads to a reliable strategy that exploits the locality while preserving the convergence behavior of the standard Newton process. Note that the two methods developed involve different complexities and efforts for implementation. Subsequently, their applications depend on specific efficiency and implementation considerations.

We evaluate the localization methods using several complex problems with discrete fracture networks. The test problems consider multi-phase and compositional fluid systems with phase changes. The results show that large degrees of solution locality present across timesteps and iterations. Comparing to a standard Newton solver, the new solvers exhibit superior computational performance. Moreover, the Newton convergence behavior is preserved, without any impact on the solution accuracy.

2. Isothermal compositional model

We consider compressible gas-oil flow in porous media without capillarity. We ignore water that does not exchange mass with the hydrocarbon phases.

The conservation equations for the isothermal compositional problem containing \( n_c \) components are written as,

\[
\frac{\partial}{\partial t} \left[ \phi (x_c \rho_o s_o + y_c \rho_g s_g) \right] + \nabla \cdot (x_c \rho_o u_o + y_c \rho_g u_g) - q_c = 0,
\]

(1)

where \( c \in \{1, \ldots, n_c\} \). \( x_c \) and \( y_c \) are molar fractions of component \( c \) in the oil and gas phases, respectively. \( \phi \) is rock porosity and \( t \) is time. \( \rho_l \) is phase molar density. \( s_l \) is phase saturation. \( q_c \) is well flow rate.

Phase velocity \( u_l \) is expressed as a function of phase potential gradient \( \nabla \Phi_l \) using the extended Darcy’s law,

\[
u_l = -k_l \lambda_l \nabla \Phi_l = -k_l \lambda_l \left( \nabla p - \rho g \nabla h \right).
\]

(2)

where \( k \) is rock permeability. \( p \) is pressure. Capillarity is assumed to be negligible. \( g \) is gravitational acceleration and \( h \) is height. Phase mobility is given as \( \lambda_l = k_{rl}/\mu_l \). \( k_{rl} \) and \( \mu_l \) are relative permeability and viscosity, respectively.
In order to close the nonlinear system, additional equations are needed. These include the thermodynamic equilibrium constraints,

\[ f_{c,o}(p, x) - f_{c,g}(p, y) = 0, \]  

(3)

where \( p, T, \) and \( z_c \) denote pressure, temperature, and overall molar fraction, respectively. \( f_{c,l} \) is the fugacity of component \( c \) in phase \( l \).

We now write the phase constraints,

\[ \sum_{c=1}^{n_c} x_c - 1 = 0, \quad \sum_{c=1}^{n_c} y_c - 1 = 0, \]  

(4)

and the saturation constraint as,

\[ s_o + s_g - 1 = 0. \]  

(5)

The above system of equations provide a complete mathematical statement for two-phase multi-component flow. The local equilibrium constraints are enforced only when both phases are present.

3. Natural-variables formulation

An important aspect of any compositional formulation is the choice of dividing the equations and unknowns into primary and secondary sets. Here we employ the popular natural-variables set (Coats 1980; Voskov and Tchelepi 2012). The primary unknowns include pressure, saturations, and molar fractions,

1. \( p \) – pressure [1],
2. \( s_l \) – phase saturations [2],
3. \( x_c, y_c \) – phase compositions of each component [2\( n_c \)].

The size of each variable is given in square bracket.

The various coefficients can be obtained as functions of the base variables. For a two-phase cell, the molar phase fraction is related to saturation as follows,

\[ \nu_l = \frac{\rho_l s_l}{\sum_m \rho_m s_m} \]  

(6)

and overall molar fraction of component \( c \) is written as,

\[ z_c = x_c \nu_o + y_c \nu_g \]  

(7)

Note that for single-phase (\( l \)) mixture, \( \nu_l = s_l = 1 \), and \( x_{c,l} \equiv z_c \).

3.1. Variable substitution

An essential ingredient of the natural-variables formulation is the ‘variable substitution’ process (Aziz and Wong 1989; Cao 2002; Voskov and Tchelepi 2012). A common strategy for variable-switching between Newton iterations during a timestep is,

1. For any cell whose status in the previous iteration is single-phase, run the phase stability test (Michelsen 1982a) to check if the mixture becomes two-phase. For the mixture that splits into two phases, perform the flash to compute the phase compositions (Michelsen 1982b).

2. If a cell is already in the two-phase state, the thermodynamic constraints are included in the nonlinear system as part of the global Jacobian.

3. If a phase saturation, or phase fraction, becomes negative between two successive iterations, the phase disappears, and appropriate variable-switching is performed.

The system of conservation equations is solved for single-phase regimes, and the combination of conservation equations and thermodynamic constraints is solved for the two-phase regime.
3.2. Phase behavior

Phase behavior computation is usually a stand-alone procedure for detecting phase changes. For a mixture of \( n \) components and two phases, the mathematical model describing the thermodynamic equilibrium is (Voskov and Tchelepi 2012),

\[
f_{c,o}(p, x) - f_{c,g}(p, y) = 0, \quad (8)
\]

\[
z_c - \nu_o x_c - (1 - \nu_o) y_c = 0, \quad (9)
\]

\[
\sum_{c=1}^{n_c} (x_c - y_c) = 0. \quad (10)
\]

where \( \nu_l \) is molar fraction of phase \( l \). We assume that \( p, T, \) and \( z_c \) are known. The objective is to find all the \( x_c, y_c \) and \( \nu_l \).

Phase behavior of a hydrocarbon mixture is commonly described using an Equation of State (EoS) model.

4. Nonlinear solution strategies

The spatial and temporal discretization schemes used for the compositional flow model are summarized in Appendix A.

4.1. Newton method

At each timestep of a FIM simulation, given the current state \( u^n \), and a fixed timestep size \( \Delta t \), we seek to obtain the new state \( u^{n+1} \).

The nonlinear residual system is solved by the Newton method,

\[
F(u^{n+1}) = 0 \quad (11)
\]

The Newton method generates a sequence of iterates, \( u^\nu, \nu = 0, 1, ..., \) each involving the construction of a Jacobian matrix and solution of the resulting linear system,

\[
J(u^\nu) \delta u^\nu = -F(u^\nu) \quad (12)
\]

where

\[
\delta u^\nu = u^{\nu+1} - u^\nu \quad (13)
\]

and \( J(u) = \frac{\partial F}{\partial u} (u) \) denotes the Jacobian matrix of \( F \) with respect to \( u \).

Here we assume that entries of Newton update such that \( |\delta u_i| < \epsilon \) are essentially negligible. Given a Newton iteration, the support set is defined for the indices of cells that exhibit non-zero update,

\[
\text{supp} \delta u = \{ i : |\delta u_i| \geq \epsilon, \quad i = 1, ..., n \} \quad (14)
\]

4.2. Locality within solution processes

In this work we focus on the solution process for the pressure-driven production problem with multi-phase multi-component fluid. We aim to exploit two levels of locality for improving computational efficiency. The first is on the timestep level. Because of the ultra-low matrix permeability in unconventional formation, transient flow within matrix may last a long period. As a result, flow dynamics (e.g. pressure propagation) evolve slowly and locally. During early stage of production, only a small portion of domain undergoes considerable variable changes.

For a timestep, the solution update is the sum of all the corresponding Newton updates. Conceivably, the locality also presents on the nonlinear (Newton) level, even if most of the domain is affected over the timestep. Previous works showed that an
individual update computed for flow and transport problems is typically sparse and constrained by a small subset of cells (Younis et al. 2010; Lu and Beckner 2011; Sheth and Younis 2017).

Here we show an example with two-fracture to demonstrate the solution behavior. The details of the model will be given in the result section. The flagging profiles for the Newton iterations of a timestep are plotted in Fig. 1. The cells that exhibit non-zero pressure updates are flagged in color blue. As we can see, the first iteration reaches the maximum area, and the region gets smaller as the iterations proceed. In the last iteration, the support set of the updates localizes to just a few cells.

Figure 1: Flagging of the pressure updates for the three Newton iterations of a timestep.

Lu and Beckner (2011) proposed an adaptive Newton strategy that solves localized systems. Their method identifies unconverged cells and their neighbors as the active subset to be updated. The unconverged set is inflated to a heuristic extent (e.g. plus one layer of neighbor cells), as a safety measure.

From the simulation studies, we also observe that under a convergent Newton sequence, the support set of updates always shrink after each iteration, such that,

$$\text{supp } \delta u^{\nu+1} \subseteq \text{supp } \delta u^{\nu}$$

(15)

Therefore, the support set from a current Newton iteration becomes an adequate estimate for the subsequent iteration.

The adaptive algorithm by Lu and Beckner (2011) is simple to implement. However, the algorithm starts with the entire domain at the first iteration of a timestep, to ensure conservative estimates for the active set. This will considerably contribute to the overall computational cost.

Our recent studies revealed that Newton iterations are closely tied to the underlying physics problem (Jiang and Tchelepi 2018). The updates for the hyperbolic transport problem may have local support that propagates through the domain as the Newton process goes forward. By comparison, the support of the pressure (flow) problem shows diffusive and global behaviors, due to its parabolic nature. To exploit this mechanism, here we propose a localized Newton strategy, which is aggressive in the way that it does not require an initial conservative estimate for the active domain. We observe that the support of pressure updates tends to reach the solution front at the first iteration. Therefore, the localized algorithm can start with a moderate active domain $\Omega_A$, and expand it if necessary using the outermost (boundary) layer of $\Omega_A$.

4.3. Localized Newton algorithm

We describe the algorithm based on the proposed localized Newton strategy. Consider a nonlinear system of equations $F = (F_1, ..., F_n)^T$ with unknowns $u = (u_1, ..., u_n)^T$. Let $V = \{1, ..., n\}$ be an index set; i.e., there is one integer for each $F_i$ and unknown $u_i$. Let $V_A$ be the index set that contains the active cells, and $n_A$ be the dimension of $V_A$. We
define $V_B \subseteq V_A$ as the cell set for the outermost (boundary) layer of the active domain. Further define the set $V_{i,m}^{nbr}$ for the neighbors of cell $i$, and $m$ is the number of layers that are incorporated. The illustration for $V_{i,m}^{nbr}$ with $m = 1$ and $m = 2$ is plotted in Fig. 2. The neighbor cells are flagged in yellow.

Let $R_A$ be a Boolean matrix of dimension $n_A \times n$. $R_A$ corresponds to the restriction operator from $V$ to $V_A$. The transpose matrix $R_A^T$ is an extension operator from $V_A$ to $V$. Then the nonlinear function and the local unknowns of the active set $V_A$ can be expressed as $F_A = R_A F$ and $u_A = R_A u$, respectively. The Jacobian of $F_A$ is,

$$J_A = R_A J R_A^T$$

(16)

Note that the submatrix $J_A$ can be directly constructed, and thus the full matrix $J$ is never needed.

The localization method takes as input a state at time level $n$ and outputs the updated state. The algorithmic details can be described as,

Step 1. Set the iteration counter $\nu$ to zero, and initialize $u^\nu$ to the current state $u^n$. Construct the initial active set $V_A$ and the associated boundary set $V_B$.

Step 2. Perform localization: solve the reduced linear system over the active domain, and update the solution.

Step 3. For each cell $i$ in $V_B$, if the update is larger than the cutoff value, inflate $V_A$ with the neighbor subset $V_{i,m}^{nbr}$.

Step 4. If there is any non-negligible update in $V_B$, the localization status stays in the ‘expand’ mode; otherwise, switch to the ‘shrink’ mode, and $V_A$ is specified as the support set of $\delta u_A$.

Step 5. Check convergence criteria of Newton iterations. If not converged, go back to Step 2. Otherwise, the timestep is finished.

The proposed method is also outlined in Algorithm 1. We can see that the localization process determines the active set that needs to be solved for the subsequent iteration. The process consists of two stages. In the first stage, $V_A$ does not fully cover the actual support of the timestep. Based on the diffusive nature of pressure updates, $V_B$ is used to detect and expand the active domain. The localized algorithm is self-adaptive in a sense that Newton update already provides an adequate estimate.

The initial active set for the algorithm can be constructed to comprise the cells in the vicinity of wells or fractures. Subsequently, computational speedup will be greatly
Algorithm 1 Localized Newton

1: \( \nu = 0 \) \hspace{1cm} u^\nu = u^n
2: Initialize \( V_A \) and \( V_B \).
3: \textbf{while} is\_expand or not\_converged \textbf{do} \hspace{1cm} \triangleright \text{Newton loop}
   4: \hspace{1cm} Solve reduced linear system,
   \hspace{1cm} \( J_A \delta u_A = -F_A \)
   5: \hspace{1cm} \( u^{\nu+1} = u^\nu + R_A^T \delta u_A \)
   6: \hspace{1cm} Check convergence criteria.
   7: \hspace{1cm} \textbf{for} \( i \in V_B \) \hspace{1cm} \triangleright \text{Determine active set}
   8: \hspace{1cm} \hspace{1cm} \textbf{if} \( |\delta u_{i,A}| \geq \epsilon \) \textbf{then}
   9: \hspace{1cm} \hspace{1cm} Obtain \( V_{i,m}^{\text{nbr}} \) of cell \( i \),
   10: \hspace{1cm} \hspace{1cm} \( V_A = V_A \cup V_{i,m}^{\text{nbr}} \)
   11: \hspace{1cm} \hspace{1cm} \textbf{end if}
   12: \hspace{1cm} \textbf{end for}
   13: \hspace{1cm} \textbf{if} \( \|\delta u_B\|_\infty < \epsilon \) \hspace{1cm} \triangleright \text{Shrink mode}
   14: \hspace{1cm} \hspace{1cm} is\_expand \leftarrow \text{false}
   15: \hspace{1cm} \hspace{1cm} \( V_A = \text{supp} \delta u_A \)
   16: \hspace{1cm} \hspace{1cm} \textbf{else}
   17: \hspace{1cm} \hspace{1cm} is\_expand \leftarrow \text{true}
   18: \hspace{1cm} \hspace{1cm} \textbf{end if}
   19: \hspace{1cm} Update \( V_B \) from \( V_A \).
   20: \hspace{1cm} \( \nu \leftarrow \nu + 1 \)
21: \textbf{end while}

increased. On the other hand, the iterations taken during the ‘expand’ mode may degrade the nonlinear convergence, comparing to the standard Newton method. In practice, a balance needs to be achieved between more aggressive localization and increased number of iterations.

It is worth to note that the iterative process for expanding active domain can be viewed as a nonlinear domain decomposition (DD) problem. Each subproblem is solved with Dirichlet boundary (constant pressure) conditions. Similarly to nonlinear preconditioning, the subdomain solutions can provide better initial guesses for Newton iterations. As a result, the localized solver shows satisfying convergence performance from the simulation cases.

4.4. Adaptive Nonlinear Domain Decomposition

For a localized Newton algorithm, more iterations may be required, if the estimate for the support set is not conservative. On the other hand, an excessively conservative estimate will reduce the speedup gained from the localized computations.

In this work we further develop an adaptive method that provides aggressive localization, while preserving the convergence behavior of the standard Newton process. To present the method, consider first a nonlinear domain decomposition (DD) with non-overlapping partitions,

\[
\bigcup_{k=1}^{N} V_k = V, \hspace{1cm} V_j \cap V_k = \emptyset \text{ if } j \not= k, \hspace{1cm} \text{and } V_k \subset V. \hspace{1cm} (17)
\]

Let \( n \) be the total number of unknowns and \( n_k \) be the total number of unknowns associated with the subset \( V_k \). The restrictions of \( u \) and \( F \) to \( V_k \) are \( u_k = R_k u \) and \( F_k = R_k F \), respectively. The Jacobian of the subproblem \( k \) is given as,

\[
J_k = R_k^T J_k R_k^T \hspace{1cm} (18)
\]
with \( k = 1, \ldots, N \). The boundary conditions for a subproblem are Dirichlet-type and taken from the neighboring subdomains.

A global solution of the nonlinear DD method is obtained by solving first subproblems and then gluing them together (Dolean et al. 2016),

\[
u^{\nu+1} = \sum_{k=1}^{N} R_k^T u_k^{\nu+1}
\]

It is common to apply the additive (Jacobi) form of the Schwarz methods. For a standard DD, static partitions of simulation grid are performed in a preprocessing step (Cai et al. 1998; Skogestad et al. 2013; Dolean et al. 2016).

To exploit the localized behaviors of flow problems, we propose instead an adaptive DD solver based on dynamic partitions. Utilizing the diffusive nature of pressure updates, subdomains are constructed from the previous iterations. To achieve localized computations, the subproblems of a nonlinear DD iteration are solved sequentially. This leads to a multiplicative (Gauss-Seidel) Schwarz method.

The algorithmic details of the adaptive DD method for a timestep are given as,

Step 1. Set the iteration counters \( \nu \) and \( k \) to zero, and initialize \( u^\nu \) to the current state \( u^n \). Construct the initial active set \( V_A^k \), and the associated boundary sets \( V_B^k \) and \( V_{\partial B}^k \). Define \( \partial B \) as the outer layer adjacent to \( \Omega_A \), such that \( V_{\partial B}^k \cap V_A^k = \emptyset \).

Step 2. Start expanding the active domain, and perform localized computations. Construct the subdomains, and the associated cell sets, using the Newton updates. Specifically, for each cell \( i \) in \( V_B^k \), if the update is larger than the cutoff value, inflate \( V_A^{k+1} \) and the total active set \( V_T \) with the neighbor subset \( V_{i,m}^{nbr} \).

Step 3. If there is any non-negligible update in \( V_B^k \), remain in the ‘expand’ mode, and obtain the boundary sets of \( V_A^{k+1} \). Otherwise, switch to the ‘shrink’ mode, indicating that the maximum support set over the timestep is reached.

Step 4. Set \( N \) as the number of the constructed subdomains. Start the nonlinear DD loop, with the counter \( \nu \) denoting the outer iteration. For each \( k \), first collect \( \delta u_{\partial B}^k \) over \( V_{\partial B}^k \), from the latest Newton updates. If there is any non-negligible element in \( \delta u_A^k \) or \( \delta u_{\partial B}^k \), perform localized computation and update the solution.

Step 5. Check convergence criteria. Repeat the outer iteration until all the subdomains are converged.

The new adaptive solver is also outlined in Algorithm 2. Note that a subdomain needs to be solved only when the solution is not yet converged or the boundary values change. Therefore the localization is naturally achieved during the nonlinear DD process. This leads to a reliable strategy to exploit the locality prior to each outer iteration.

An outer iteration of the DD method can be written in a fixed-point form,

\[
u^{\nu+1} = \sum_{i=1}^{N} R_i^T G_i(u^\nu) =: \mathcal{G}(u^\nu) \tag{20}
\]

where the solution operator for a subproblem is,

\[
u^{\nu+1}_{ik} = G_i(u^\nu) \tag{21}
\]

As can be seen, evaluation of the function \( \mathcal{G}(u) \) involves the solution of all the subproblems \( (1, \ldots, N) \). Despite its simple form, the fixed-point method may suffer from slow
Algorithm 2 Adaptive Nonlinear Domain Decomposition

1: $\nu = 0$, $k = 0$
2: $u^{\nu} = u^n$
3: Initialize $V^k_A$, $V^k_B$ and $V^k_{\partial B}$.
4: $V_T = V^k_A$
5: while is expand do $\triangleright$ Newton loop
6: Local solve over $V^k_A$,
   $J_A \delta u_A = -F_A$
7: $u^{\nu+1} = u^{\nu} + R^T_A \delta u_A$
8: $V^{k+1}_A \leftarrow \{\}$
9: for $i \in V^k_B$ do $\triangleright$ Determine active set
10: if $|\delta u_{i,A}| \geq \epsilon$ then
11: Obtain $V_{i,m}^{i,m}$ of cell $i$,
12: $V_T = V_T \cup V_{i,m}^{i,m}$
13: $V^{k+1}_A = V^{k+1}_A \cup V_{i,m}^{i,m}$
14: end if
15: end for
16: if $\|\delta u_B\|_{\infty} < \epsilon$ then $\triangleright$ Shrink mode
17: is expand $\leftarrow$ false
18: else
19: is expand $\leftarrow$ true
20: Obtain $V^k_B$ of $V_T$,
21: $V^{k+1}_B = V^k_B \cap V^{k+1}_A$
22: Obtain $V^{k+1}_{\partial B}$ of $V^{k+1}_A$.
23: end if
24: $k \leftarrow k + 1$
25: end while
26: $N = k$
27: $k = 0$, $\nu = 1$
28: while not converged do $\triangleright$ Nonlinear DD loop
29: for $k < N$ do
30: Collect $\delta u_{\partial B}^k$ over $V_{\partial B}^k$.
31: if $\|\delta u_A^k,\|_{\infty} \geq \epsilon$ or $\|\delta u_{\partial B}^k\|_{\infty} \geq \epsilon$ then
32: Local solve over $V^k_A$,
33: $u^{\nu+1} = u^{\nu} + R^T_A \delta u_A$
34: Check convergence criteria.
35: end if
36: end for
37: $\nu \leftarrow \nu + 1$
38: end while
convergence, or even divergence (Skogestad et al. 2013; Dolean et al. 2016). Recently we proposed several ways of accelerating the nonlinear DD process (Jiang and Tchelepi 2019). The nonlinear acceleration techniques greatly improve the outer convergence behavior, while requiring little additional cost. The investigation on the outer convergence of the adaptive DD solver is subject to a future work.

5. Results

We evaluate the efficacy of the localization methods using several test problems with discrete fracture networks. The problems include an oil-water system and a two-phase compositional system with phase changes. A 2D synthetic model is generated to contain a single-stage hydraulically-fractured horizontal well at the center of a reservoir. The fractures are assumed to fully penetrate the formation.

An embedded discrete fracture model (EDFM) is employed to explicitly describe the discrete fractures. Lee et al. (2001), Li and Lee (2008), Hajibeygi et al. (2011) and Moinfar et al. (2014) introduced and extended EDFM, which does not require simulation grid to conform to fracture geometry. Recent works on the implementations of EDFM for various types of problems include Panfili et al. (2015), Jiang and Younis (2016, 2017), Ren et al. (2018), Hui et al. (2019), Xue et al. (2019), and Rey et al. (2019).

A simple time-stepping strategy is employed: starting with a small initial value, timestep sizes gradually increase to the maximum value. Newton convergence is based on the following criterion: solution (pressure) delta (increment) $\|\delta p\|_\infty < \epsilon_p$ between iterations. The specification of the base model is given in Table 1.

| Parameter             | Value | Unit  |
|-----------------------|-------|-------|
| Initial pressure      | 2500  | psi   |
| Matrix porosity       | 0.05  |       |
| Rock compressibility  | 3.4e-4| 1/psi |
| Matrix permeability   | 1e-19 | m$^2$ |
| Fracture permeability | 1e-10 | m$^2$ |
| Fracture aperture     | 1e-3  | m     |
| Production BHP        | 1000  | psi   |
| Total simulation time | 1500  | day   |
| Max timestep size     | 100   | day   |

5.1. Grid sensitivity

We first test a 100m $\times$ 100m model with two fractures to demonstrate the effect of transient flow. Initial water saturation is set as the connate saturation, so that water is immobile during simulations. Newton tolerance has the value of $\epsilon_p = 0.3$ psi. Simulations are run for three different levels of grid resolution.

Pressure profiles at the end of simulation are shown in Fig. 3. Oil rates are plotted in Fig. 4. As we can see, oil productions are largely underestimated by the coarse grid systems. The large cell sizes of coarse grid are not adequate for the sharp pressure variations in the vicinity of the fractures. On the other hand, the fine-grid case involves a large number of cells and thus requires significant computational efforts.

5.2. Localized Newton method

5.2.1. Case 1

We test the model with 200 $\times$ 200 grid level. At every iteration, the localized Newton algorithm 1 provides the active set to be updated, and then solves the reduced linear system. Convergence tolerance of $\epsilon_p = 0.3$ psi is employed as the cutoff value for the
(a) $20 \times 20$
(b) $50 \times 50$
(c) $200 \times 200$

Figure 3: Pressure profiles for the three levels of grid resolution.

![Pressure profiles for the three levels of grid resolution.](image)

![Pressure profile of Case 1.](image)

Pressure profile is shown in Fig. 5. We can observe that only a small fraction of the cells around the fractures undergoes significant changes in pressure. Oil rates are plotted in Fig. 6. As expected, the solution from the localization method exactly matches

active and boundary sets. The neighbor set $V_{i,m}^{i,m}$ with $m = 2$ is used to expand active domain. The support set of the last timestep is taken as the initial active set for the current timestep.
the reference solution. This is because the iterative processes converge under the same convergence tolerance.

We plot the ratios of active domain (per timestep) versus simulation time in Fig. 7. Note that 2-4 iterations are taken at each timestep. Computational performance of Case 1 is summarized in Table 2. $M_A$ is defined as the ratio of the active to full sets. For the standard Newton, the total ratio $M_A$ is equal to the total iteration number, with the average ratio as 1.

The results show that the localization method exhibits good convergence performance. As we noted before, the iterations for expanding active domain can be viewed as nonlinear preconditioning which provides better initial guesses. The localized solver achieves an at least 13-fold reduction in computations, comparing to the standard solver. The actual simulation speedup depends on the scaling of computational complexity $O(n^3)$.

We re-run the case using the localized Newton for one timestep with the size of 50 days. The timestep size is equal to the total simulation time. The profiles for the flagging of cells over the 4 iterations are plotted in Fig. 9. Four types of cell sets are specified: 1. active set (in color yellow); 2. boundary set (black); 3. the active cells with non-negligible update (the support set), (blue); and 4. the boundary cells with non-negligible
Table 2: Computational performance of Case 1.

|               | Timesteps | Total iterations | Total ratio $M_A$ | Average ratio $M_A$ per iteration |
|---------------|-----------|------------------|-------------------|-----------------------------------|
| Localized Newton | 54        | 159              | 11.4              | 0.072                             |
| Standard Newton | 54        | 156              | 156               | 1                                 |

update (red). A color illustration for the cell types is given in Fig. 8.

![Figure 8: Illustration for the cell types: 1. active set (yellow); 2. boundary set (black); 3. the active cells with non-negligible update (blue); 4. the boundary cells with non-negligible update (red).](image)

As can be seen, the initial active set is small and not conservative, resulting in aggressive localization and thus high computational speedup. As the iterations proceed, the active domain expands until the maximum area is reached.

5.2.2. Case 2

We test a 300m × 300m model with 200 × 200 grid and a more complex fracture network. The model contains 4 secondary fractures with permeability of 1e-13 m². Pressure profile is shown in Fig. 10. We plot the ratios of active domain (per timestep) versus simulation time in Fig. 11.

Computational performance of Case 2 is summarized in Table 3. From the results we see that the localization method enables a significant reduction in computations, while preserving the original convergence behavior.

Table 3: Computational performance of Case 2.

|               | Timesteps | Total iterations | Total ratio $M_A$ | Average ratio $M_A$ per iteration |
|---------------|-----------|------------------|-------------------|-----------------------------------|
| Localized Newton | 54        | 155              | 10.1              | 0.065                             |
| Standard Newton | 54        | 153              | 153               | 1                                 |

We re-run the case for one timestep with the size of 50 days. The profiles for the flagging of cells over the 4 iterations are plotted in Fig. 12. The color illustration of cell sets is the same as specified in the previous section. As can be seen, the sparsity patterns of the updates vary largely from one iteration to the next. After two iterations, the nonlinear convergence is constrained to just several cells.
5.2.3. Case 3

We again consider the model from Case 2, but with $100 \times 100$ grid. The model uses a two-component fluid system where the initial oil is made of $\{C_1(50\% ), C_{10}(50\% )\}$. Phase density and viscosity depend on pressure and compositions. Phase molar density $\rho_l$ is evaluated based on the compressibility $(Z)$ factor from the PR EoS. Phase viscosity $\mu_l$ is computed by the correlation of Lohrenz et al. (1964). Simple relative permeabilities
given by quadratic function are used. Initial pressure is 2900 psi and temperature is 340 K. The total simulation time is 500 days, with the maximum timestep size as 50 days. The other parameters in the previous case remain unchanged.

The profiles for the phase status and saturation of gas are shown in Fig. 13 and Fig. 14 respectively. As the pressure drops below the bubble-point, gas starts to
appear around the fractures. The transient effects of saturation and phase dynamics can be difficult to capture using a coarse grid. From the results we observe that the saturation updates exhibit considerable locality. By comparison, the pressure updates affect a much larger region.

![Figure 13: Phase status of gas for Case 3.](image)

![Figure 14: Gas saturation for Case 3.](image)

We plot the ratios of active domain (per timestep) versus simulation time in Fig. 15. Computational performance of Case 3 is summarized in Table 4. The localized Newton method shows superior performance, with a small increase of iterations.

|                         | Timesteps | Total iterations | Total ratio $M_A$ | Average ratio $M_A$ per iteration |
|-------------------------|-----------|------------------|-------------------|-----------------------------------|
| Localized Newton        | 40        | 153              | 16.1              | 0.1                               |
| Standard Newton         | 40        | 148              | 148               | 1                                 |
5.3. Adaptive Nonlinear DD method

We study the adaptive nonlinear DD (Algorithm 2) which is based on dynamic partitions to perform localized computations. The developed solver can make adequate estimates of the active set for each inner iteration.

5.3.1. Case 2.1

We use the same model as specified in Case 1. Computational performance of the case is summarized in Table 5. The adaptive DD solver greatly reduces the computational cost, while taking much more iterations. This is because each outer iteration consists of multiple inner iterations in the nonlinear DD process. We note that the size of a subdomain system is relatively small and the total number of outer iterations is comparable to the standard Newton method.

Table 5: Computational performance of Case 2.1.

|                  | Timesteps | Total (inner) iterations | Total ratio $M_A$ | Average ratio $M_A$ per iteration |
|------------------|-----------|--------------------------|-------------------|-----------------------------------|
| Adaptive Nonlinear DD | 54        | 804                      | 20.8              | 0.026                             |
| Standard Newton   | 54        | 156                      | 156               | 1                                 |

We plot the ratios of active domain (per timestep) versus simulation time in Fig. 16.

We re-run the case for one timestep with the size of 50 days. The neighbor set $V_{nbr}$ is specified with $m = 4$. Flagging profiles of cells over the 4 iterations are plotted in Fig. 17. As we can see, the timestep converges with 2 outer iterations. The algorithm constructs the subdomains that adapt to the flow dynamics and pressure updates. During the nonlinear DD process, the subproblems are solved sequentially, to achieve localization. The results confirm that the support set of the timestep is contained in the union of all the flagged subsets.

5.3.2. Case 2.2

The model from Case 2 is used. Computational performance of the case is summarized in Table 6. We plot the ratios of active domain (per timestep) versus simulation time in Fig. 18. We can see that the adaptive solver obtains an at least 11-fold reduction in solution effort, without degrading the original Newton convergence.
This work develops a prototype algorithm for the adaptive DD solver, which can be further improved and optimized, e.g. exploiting the solution locality within each subdomain. The algorithm will be expected to prevent any overly conservative estimate. Subsequently, higher simulation speedup can be achieved.
Table 6: Computational performance of Case 2.2.

|               | Timesteps | Total (inner) iterations | Total ratio $M_A$ | Average ratio $M_A$ per iteration |
|---------------|-----------|--------------------------|--------------------|----------------------------------|
| Adaptive Nonlinear DD | 54        | 255                      | 13.6               | 0.053                            |
| Standard Newton   | 54        | 153                      | 153                | 1                                |

Figure 18: Ratios of active domain (per timestep) for Case 2.2.

6. Summary

We develop the localized solution strategies for efficient simulations of unconventional reservoirs. By utilizing the diffusive nature of pressure updates, an adaptive algorithm is proposed to make adequate estimates for the active sets to be solved. We also develop a localized solver based on nonlinear domain decomposition (DD). The solver provides effective partitioning that adapts to flow dynamics and Newton updates.

We test several complex problems with discrete fracture networks. The results show that large degrees of solution locality present across timesteps and iterations. Comparing to a standard Newton solver, the developed solvers enable superior computational performance. Moreover, the original Newton convergence is preserved, without any impact on the solution accuracy.

The new solvers can be extended to account for more complex fracture networks and flow physics. The incorporation of models with strong heterogeneity and field-scale fracture networks is a subject of our ongoing research.

Acknowledgements

This work was supported by the Chevron/Schlumberger/Total INTERSECT Alliance Technology. The author also thanks Chevron for permission to publish this paper.

Appendix A. Discretization methods

A standard finite-volume scheme is applied as the spatial discretization for the mass conservation equations. A two-point flux approximation (TFPA) is used to approximate the flux across a cell interface. The method of choice for the time discretization is the
fully-implicit scheme. The discrete form of conservation equation is given as,
\[
\frac{V}{\Delta t} \left[ (\phi \rho_T z_c)^{n+1} - (\phi \rho_T z_c)^n \right] - \sum_{ij} (x_c \rho_o F_o + y_c \rho_g F_g)^{n+1} - Q^{n+1}_c = 0. \tag{22}
\]
where superscripts denote timesteps, and $\Delta t$ is the timestep size. $V$ is the cell volume. All indices related to the cell numeration are neglected. The accumulation term involves the total density,
\[
\rho_T z_c = x_c \rho_o s_o + y_c \rho_g s_g
\]
and,
\[
\rho_T = \begin{cases} 
  s_o \rho_o(x) + s_g \rho_g(y), & \text{two phase}, \\
  \rho_l(z), & \text{one phase}. 
\end{cases} \tag{23}
\]
where $\rho_o(x)$ indicates the density computed at a composition $x$, and $\rho_l(z)$ is the density computed in the single-phase regime at a composition $z$.

The discrete phase flux across the interface $(ij)$ between two cells is written as,
\[
F_{l,ij} = \Upsilon_{ij} \lambda_{l,ij} \Delta \Phi_{l,ij} \tag{25}
\]
where subscript $(ij)$ denotes quantities defined at the cell interface. $\Upsilon_{ij}$ is the interface transmissibility. $\Delta \Phi_{l,ij} = \Delta p_{l,ij} - g_{l,ij}$ is the phase potential difference with the discrete weights $g_{l,ij} = \rho_{l,ij} \Delta h_{l,ij}$. The phase and compositional coefficients associated with the flux terms are evaluated using the Phase-Potential Upwinding (PPU) scheme.

References

Aziz, K., Settari, A., 1979. Petroleum Reservoir Simulation. Chapman & Hall.
Artus, V. and Fructus, D., 2012. Transmissibility corrections and grid control for shale gas numerical production forecasts. Oil & Gas Science and Technology, 67(5), pp.805-821.
Coats, K.H., 1980. An equation of state compositional model. Society of Petroleum Engineers Journal, 20(05), pp.363-376.
Cai, X.C., Gropp, W.D., Keyes, D.E., Melvin, R.G. and Young, D.P., 1998. Parallel Newton–Krylov–Schwarz algorithms for the transonic full potential equation. SIAM Journal on Scientific Computing, 19(1), pp.246-265.
Cipolla, C.L., Warpinski, N.R., Mayerhofer, M., Lolon, E.P. and Vincent, M., 2010a. The relationship between fracture complexity, reservoir properties, and fracture-treatment design. SPE production & Operations, 25(04), pp.438-452.
Cipolla, C.L., Lolon, E.P., Erde, J.C. and Rubin, B., 2010b. Reservoir modeling in shale-gas reservoirs. SPE reservoir evaluation & engineering, 13(04), pp.638-653.
Dolean, V., Gander, M.J., Kheriji, W., Kwok, F. and Masson, R., 2016. Nonlinear pre-conditioning: How to use a nonlinear Schwarz method to precondition Newton’s method. SIAM Journal on Scientific Computing, 38(6), pp.A3357-A3380.
Ding, D.Y., Wu, Y.S. and Jeannin, L., 2014. Efficient simulation of hydraulic fractured wells in unconventional reservoirs. Journal of Petroleum Science and Engineering, 122, pp.631-642.
Ding, D.Y., Farah, N., Bourbiaux, B., Wu, Y.S. and Mestiri, I., 2018. Simulation of matrix/fracture interaction in low-permeability fractured unconventional reservoirs. SPE Journal, 23(04), pp.1-389.
Fisher, M.K., Wright, C.A., Davidson, B.M., Goodwin, A.K., Fielder, E.O., Buckler, W.S. and Steinsberger, N.P., 2002, January. Integrating fracture mapping technologies to optimize stimulations in the Barnett Shale. In SPE annual technical conference and exhibition. Society of Petroleum Engineers.
Farah, N., Ding, D.Y. and Wu, Y.S., 2015, February. Simulation of the impact of fracturing fluid induced formation damage in shale gas reservoirs. In SPE Reservoir Simulation Symposium. Society of Petroleum Engineers.

Hui, M.H., Mallison, B.T., Fyrozjaee, M.H. and Narr, W., 2013, September. The upscaling of discrete fracture models for faster, coarse-scale simulations of IOR and EOR processes for fractured reservoirs. In SPE Annual Technical Conference and Exhibition. Society of Petroleum Engineers.

Hui, M.H., Dufour, G., Vitel, S., Muron, P., Tavakoli, R., Rousset, M., Rey, A. and Mallison, B., 2019, March. A Robust Embedded Discrete Fracture Modeling Workflow for Simulating Complex Processes in Field-Scale Fractured Reservoirs. In SPE Reservoir Simulation Conference. Society of Petroleum Engineers.

Jiang, J. and Younis, R.M., 2015a. A multimechanistic multicontinuum model for simulating shale gas reservoir with complex fractured system. Fuel, 161, pp.333-344.

Jiang, J. and Younis, R.M., 2015b. Numerical study of complex fracture geometries for unconventional gas reservoirs using a discrete fracture-matrix model. Journal of Natural Gas Science and Engineering, 26, pp.1174-1186.

Jiang, J. and Younis, R.M., 2016. Hybrid coupled discrete-fracture/matrix and multicontinuum models for unconventional-reservoir simulation. SPE Journal, 21(03), pp.1-009.

Jiang, J. and Younis, R.M., 2017. An improved projection-based embedded discrete fracture model (pEDFM) for multiphase flow in fractured reservoirs. Advances in Water Resources, 109, pp.267-289.

Jiang, J. and Tchelepi, H.A., 2018. Dissipation-based continuation method for multiphase flow in heterogeneous porous media. Journal of Computational Physics, 375, pp.307-336.

Jiang, J. and Tchelepi, H.A., 2019. Nonlinear acceleration of sequential fully implicit (SFI) method for coupled flow and transport in porous media. Computer Methods in Applied Mechanics and Engineering, 352, pp.246-275.

Kazemi, H., Merrill Jr, L.S., Porterfield, K.L. and Zeman, P.R., 1976. Numerical simulation of water-oil flow in naturally fractured reservoirs. Society of Petroleum Engineers Journal, 16(06), pp.317-326.

Karimi-Fard, M., Durlofsky, L.J. and Aziz, K., 2003, January. An efficient discrete fracture model applicable for general purpose reservoir simulators. In SPE Reservoir Simulation Symposium. Society of Petroleum Engineers.

Lohrenz, J., Bray, B.G. and Clark, C.R., 1964. Calculating viscosities of reservoir fluids from their compositions. Journal of Petroleum Technology, 16(10), pp.1-171.

Lee, S.H., Lough, M.F. and Jensen, C.L., 2001. Hierarchical modeling of flow in naturally fractured formations with multiple length scales. Water resources research, 37(3), pp.443-455.

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 Reservoir Evaluation & Engineering, 11(04), pp.750-758.

Lu, P. and Beckner, B.L., 2011, January. An adaptive Newton’s method for reservoir simulation. In SPE Reservoir Simulation Symposium. Society of Petroleum Engineers.

Michelsen, M.L., 1982a. The isothermal flash problem. Part I. Stability. Fluid phase equilibria, 9(1), pp.1-19.

Michelsen, M.L., 1982b. The isothermal flash problem. Part II. Phase-split calculation. Fluid phase equilibria, 9(1), pp.21-40.

Maxwell, S.C., Urbancic, T.I., Steinsberger, N. and Zimno, R., 2002, January. Microseismic imaging of hydraulic fracture complexity in the Barnett shale. In SPE annual technical conference and exhibition. Society of Petroleum Engineers.

Mayerhofer, M.J., Lolon, E., Warpinski, N.R., Cipolla, C.L., Walser, D.W. and Rightsmire, C.M., 2010. What is stimulated reservoir volume?. SPE Production & Operations, 25(01), pp.89-98.
Mallison, B.T., Hui, M.H. and Narr, W., 2010, September. Practical gridding algorithms for discrete fracture modeling workflows. In ECMOR XII-12th European Conference on the Mathematics of Oil Recovery (pp. cp-163). European Association of Geoscientists & Engineers.

Moinfar, A., Varavei, A., Sepehrnoori, K. and Johns, R.T., 2014. Development of an Efficient Embedded Discrete Fracture Model for 3D Compositional Reservoir Simulation in Fractured Reservoirs. SPE Journal, 19(02), pp.289-303.

Pruess K. and Narasimhan, TN., 1985. A practical method for modeling fluid and heat flow in fractured porous media. Soc Petrol Eng J.

Panfili, P., Colin, R., Cominelli, A., Giamminonni, D. and Guerra, L., 2015, September. Efficient and effective field scale simulation of hydraulic fractured wells: methodology and application. In SPE Reservoir Characterisation and Simulation Conference and Exhibition. Society of Petroleum Engineers.

Ren, G.T., Jiang, J.M. and Younis, R.M., 2016, August. XFEM-EDFM-MINC for coupled geomechanics and flow in fractured reservoirs. In ECMOR XV-15th European Conference on the Mathematics of Oil Recovery (pp. cp-494). European Association of Geoscientists & Engineers.

Ren, G., Jiang, J. and Younis, R.M., 2018. A Model for coupled geomechanics and multiphase flow in fractured porous media using embedded meshes. Advances in Water Resources, 122, pp.113-130.

Rey, A., Schembre, J. and Wen, X.H., 2019, October. Calibration of the Water Flow-back in Unconventional Reservoirs with Complex Fractures using Embedded Discrete Fracture Model EDFM. In SPE Liquids-Rich Basins Conference-North America. Society of Petroleum Engineers.

Skogestad, J.O., Keilegavlen, E. and Nordbotten, J.M., 2013. Domain decomposition strategies for nonlinear flow problems in porous media. Journal of Computational Physics, 234, pp.439-451.

Sheth, S.M. and Younis, R.M., 2017. Localized linear systems in sequential implicit simulation of two-phase flow and transport. SPE Journal, 22(05), pp.1-542.

Sheth, S., Moncorg, A. and Younis, R., 2019. Localized linear systems for fully implicit simulation of multiphase multicomponent flow in porous media. Computational Geosciences, pp.1-17.

Voskov, D.V. and Tchelepi, H.A., 2012. Comparison of nonlinear formulations for two-phase multi-component EoS based simulation. Journal of Petroleum Science and Engineering, 82, pp.101-111.

Warren, J.E. and Root, P.J., 1963. The behavior of naturally fractured reservoirs. Society of Petroleum Engineers Journal, 3(03), pp.245-255.

Wu, Y.S., Li, J., Ding, D., Wang, C. and Di, Y., 2014. A generalized framework model for the simulation of gas production in unconventional gas reservoirs. Spe Journal, 19(05), pp.845-857.

Xue, X., Yang, C., Onishi, T., King, M.J. and Datta-Gupta, A., 2019. Modeling Hydraulically Fractured Shale Wells Using the Fast-Marching Method With Local Grid Refinements and an Embedded Discrete Fracture Model. SPE Journal.

Younis, R., Tchelepi, H.A. and Aziz, K., 2010. Adaptively Localized Continuation-Newton Method–Nonlinear Solvers That Converge All the Time. SPE Journal, 15(02), pp.526-544.

Weng, X., Kresse, O., Chuprakov, D., Cohen, C.E., Prioul, R. and Ganguly, U., 2014. Applying complex fracture model and integrated workflow in unconventional reservoirs. Journal of Petroleum Science and Engineering, 124, pp.468-483.