SIMULATION OF UNCONVENTIONAL RESERVOIRS WITH COMPLEX FRACTURE CONDITIONS

A Dissertation

by

YU JIANG

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	John E. Killough
Committee Members,	Eduardo Gildin
	Kan Wu
	Vivek Sarin
Head of Department,	Jeff Spath

May 2022

Major Subject: Petroleum Engineering

ABSTRACT

The unconventional reservoirs have been a major source of the modern oil and gas industry. However, simulation of unconventional reservoirs with complex fracture conditions remains to be a difficult problem. The nano-scale porous system has increased the difficulty in modeling the multi-physics, multi-phase, multi-medium, and multi-component flow in fractured unconventional reservoirs. The fracture network dominates flow in the unconventional reservoirs but is hard to characterize due to its high complexity and uncertainty. The global Jacobian matrix is usually large in dimensions and ill-conditioned, for which specific techniques are required to enhance the simulator performance.

This dissertation has proposed a workflow to simulate unconventional reservoirs with complex fracture conditions. A multi-component multi-phase reservoir simulator is developed based on the GURU framework with a capacity of multi-porosity-permeability system model which accounts for multiple flow mechanisms including convection, diffusion, and adsorption/desorption. A fully implicit Jacobian system is set up by coupling the reservoir matrix with fracture and wellbore components. High-performance computation techniques are applied to the simulator to enhance the computational performance by CPU- and GPU-based parallelization on the linear solvers and Jacobian construction. A parallel multi-stage preconditioner with respect to the corresponding subspaces of wellbore, fractures, saturation, and pressure is developed with adaptive settings which provides a better convergence rate for iterative solvers. An

LGR-EDFM model is developed to capture the large fractures in complex fracture networks. The hydraulic fractures are modeled using a novel imaging-based micro-seismic interpretation method while the natural fracture network is characterized by a fractal model with empirical risk minimization calibration through histograms. All of these govern a numerical simulator with the capability to solve fractured unconventional reservoir problems.

More specific research is also performed to further increase the knowledge and improve the accuracy of fractured unconventional simulations. A pre-stage initialization method is developed based on the fracking schedule simulation, which provides a better estimation of initial pressure and saturation profiles. Water flowback and hydrocarbon flow mechanisms are studied to get a better knowledge of unconventional reservoir physics. A non-Darcy water flux model is proposed based on the boundary layer theory to better describe the water flux in nano-pores. A domain decomposition solver is developed based on the EDFM fracture network to further increase the capability and computational performance. The fracture closure effects are studied using a dynamic EDFM method to better simulate the pressure depletion behavior during production.

DEDICATION

To my family and friends for their kind support and love.

ACKNOWLEDGEMENTS

I would like to emphasize my deepest appreciation to my advisor, Dr. John Killough. He is such a knowledgeable person who has helped me quite a lot during all my Ph.D. years. I would not be able to achieve my goal without his generous guide and support.

I would like to thank Dr. Eduardo Gildin, Dr. Vivek Sarin, and Dr. Kan Wu for their kind acceptance to be my dissertation committee. They have given out various precious and valuable opinions which helped me to further improve my research.

My thanks also go to my colleagues, the department faculty and staff, and everyone in the Killough research group. You have made my Ph.D. years such a nice and unforgettable experience.

My special thanks go to Mr. Yongzheng Cui from China University of Petroleum (East China) and Dr. Jin Tang from Texas A&M University for their kind sharing of thoughts and ideas as well as collaborations.

At last, I would like to express my thanks to my beloved family, for your help, support, and understanding during all these years and all my life.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee consisting of Dr. John E. Killough as committee chair, Dr. Eduardo Gildin, and Dr. Kan Wu of the Department of Petroleum Engineering and Dr. Vivek Sarin of the Department of Computer Science and Engineering.

Funding Sources

Part of this work was funded by a research assistantship provided by Marathon Oil to the Petroleum Engineering Department of Texas A&M University, from year 2017 to year 2019.

Part of this work was supported by the scholarship provided by Dr. John E. Killough, from year 2020 to year 2021.

NOMENCLATURE

$\emptyset$	Porosity
S	Saturation
τ	Phase composition
t	Time
m_L	Langmuir isothermal adsorption coefficient
k	Permeability
φ	Phase potential
μ	Viscosity
q, Q	Flux rate
ρ	Density
x_i, y_i	Multiphase concentration
σ_i	Transmissibility shape factor
C_i	Component concentration
P	Pressure
V	Volume
Z	Real gas compressibility factor
G	Non-Darcy startup potential gradient
T	Temperature
δ	PR-EoS coefficients
$G(\mathbf{n})$	Phase Gibbs energy

f_{ij}	Fugacity
K	VLE equilibrium coefficient
L	Fracture dimension
R_i^k	Residual for component i at timestep k
WI	Wellbore index
J	Jacobian matrix
Q	Iterator
S	Smoother
M	Decoupling multiplier matrix
U	Subspace converter
$\mathrm{T_i}$	Transmissibility
D_{frac}	Fracture fractal dimension
W_i	Weighting parameter

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

1.1. Research background and motivation

1.1.1. Unconventional reservoirs

The unconventional reservoirs have been playing a major role in the modern oil and gas industry, and their contribution to total production has been on an increasing historical trend (**Figure 1.1**) due to the demand for global fossil energy consumption (Holditch, 2003). However, efficient reservoir development and production from unconventional reservoirs has always been a hard problem due to its physical nature of high complexity and low permeability. Novel techniques like horizontal drilling with single-to-multistage hydraulic fracking (Ozkan et al., 2011; Rahim et al., 2012) has been introduced into the industry to enhance production performance in low permeability formations like tight oil sandstones and oil/gas shales by injecting a certain amount of liquid volume to raise pore pressure and cause shear failures in bulk rock. This not only breaks the rock to create multiple sets of larger flow channels inside the reservoir matrix known as hydraulic fractures but also opens up the pre-existing reservoir natural fracture (Zhao et al., 2021) and thus form a fracture network that is usually complex in geometry due to the reservoir heterogeneity (Sierra, 2016).

The porous system in unconventional reservoirs adds another layer of complexity in unconventional reservoirs. Compared with the common case while only pore channels are considered, unconventional reservoirs can have multiple porous systems including hydraulic fractures, natural fractures, inorganic and organic matrices. The hydraulic and

natural fractures serve as pathways for the reservoir fluids and are thus flow dominators. On the other hand, the source matrix is partitioned into organic kerogen and inorganic clay particles due to their composing nature and physical behaviors (Odusina et al., 2011; Passey et al., 2010). In such cases, the corresponding pore channels are considerably much smaller than in the conventional reservoirs, which are usually in nanoscale. Thus the wall effect (Tian et al., 2017) and hydrocarbon molecule movement behaviors (Wang et al., 2021) become non-negligible. All these multi-dimensional and multi-physics parts are then imported into the reservoir simulation models and raise the problem complexity by a great amount.

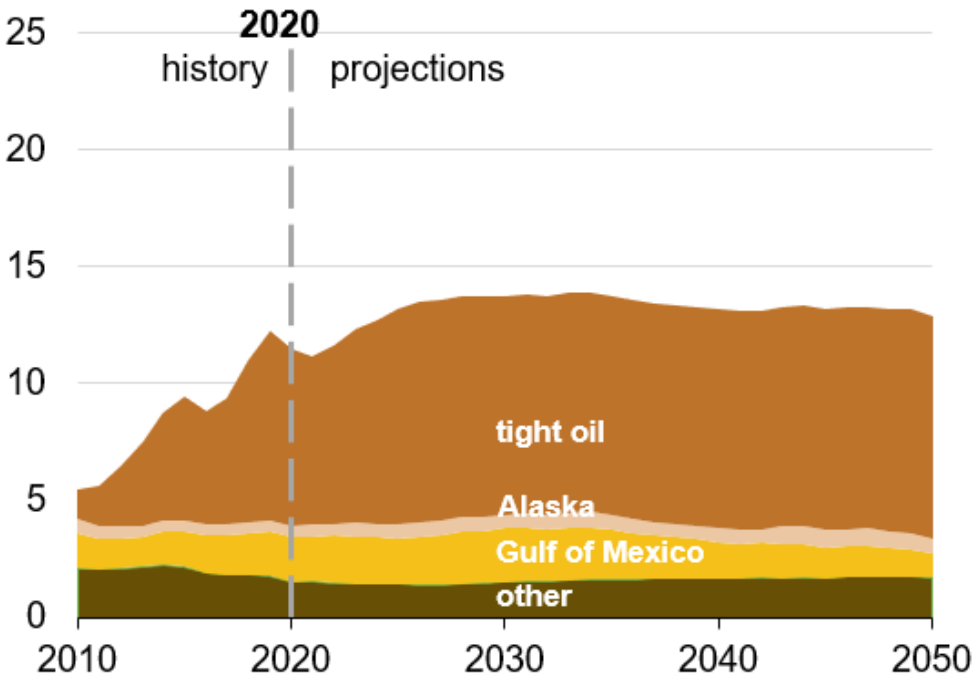

Figure 1.1 US domestic crude oil production by source, MMSTB/D. (EIA, 2021)

1.1.2. Unconventional simulation with fractures

Reservoir simulation is the technique to mimic the subsurface flow behavior through porous media using numerical mathematical models. It has been applied as one of the proven and efficient development and management methods for reservoir engineers to gain production estimations and predictions. Compared with conventional methods, simulation in unconventional reservoirs with fractures is usually far more difficult and complicated due to the multi-porosity multi-physics behavior in tight sandstones and shales (Feast et al., 2015) as well as the complex geometry of fracture network (Ren et al., 2016; Xiaogang et al., 2017). Challenges raised in modern unconventional reservoir simulations with complex fractures can be grouped into two categories of accuracy and performance, which will be discussed below.

The first part is the accuracy of the simulation. The complex porosity system in fractured unconventional reservoirs has already been discussed a bit in the subsection above, which usually requires a multi-porosity reservoir characterization model to fully describe the reservoir behavior. Another issue is that the Darcy flow in unconventional reservoirs is usually limited. Non-Darcy flow models, especially for gas, are relatively common in unconventional simulations to account for different flow mechanisms including slippage flow, Knudsen diffusion, absorption/desorption, etc. (Cao et al., 2015; Fathi & Akkutlu, 2012; Zhang et al., 2017). The traditional black-oil approach can be used in some cases to describe said fluid behavior in limited conditions, but compositional simulators with nano-scale vapor-liquid equilibrium (VLE) calculations usually give better results.

It is also worth addressing that the fracture network characterization adds more issues to simulation accuracy. While the state-of-the-art approach in the industry uses micro-seismicity datapoints to set up hydraulic fracture networks (Cipolla et al., 2011), it still contains very high uncertainties and usually requires an extensive amount of history matching work (Anbalagan, 2019) to get a useable fracture distribution. Adding natural fractures into the network makes the problem even more complex, which has a significant impact on production behavior but is rarely studied.

Computational performance usually behaves as the bottleneck in unconventional simulations, especially with complex fracture conditions. The compositional simulation itself is already a time-consuming process, in which the time cost grows dramatically with problem size and the number of components. Despite the relatively large problem size in unconventional simulations, adding fracture networks to bulk matrix usually result in an ill-conditioned sparse matrix (Norbeck et al., 2014), which can make the common linear solving approach unusable or cause significant computational performance issues. Thus, modern high-performance techniques like vectorization and parallelization as well as advanced linear solver methods become a must in unconventional reservoirs with complex fracture conditions.

1.2. Study scopes

The main scope of this dissertation is to give out a workflow to unconventional simulation with complex conditions based on the issues discussed above. The detailed scopes are listed below:

1) Develop a fully compositional, fully implicit reservoir simulator with multi-physics and multi-continuum settings to simulate the production/injection process in unconventional reservoirs.
2) Apply performance optimization techniques on the developed simulator for the capacity of large and complex unconventional problems with extreme fracture conditions.
3) Develop reservoir modeling methods for complex fracture network characterization based on combined data analysis.
4) Develop specific techniques to further analyze the unconventional problem and improves simulation accuracy.

The general workflow described is illustrated in Figure 1.2.

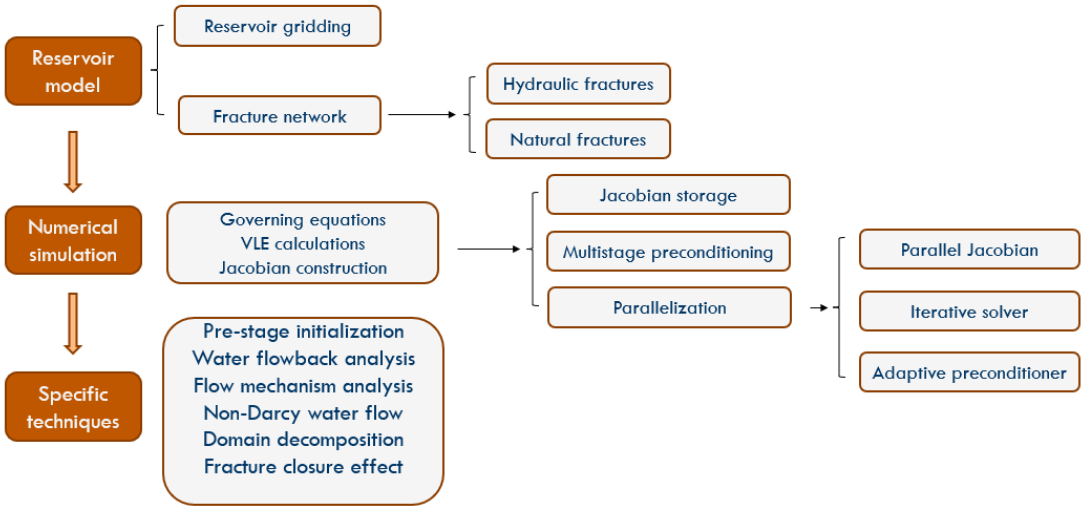

Figure 1.2 General workflow chart for fractured unconventional simulation

1.3. Chapter overview

The organization of this dissertation is listed as follows: the 1st chapter discusses the background of this research area as well as technical difficulties to overcome; the 2nd chapter introduces the development process of an in-house simulator designed specifically for unconventional fractured simulations; the 3rd chapter applies performance optimization methods to the simulator and improves its computational efficiency; the 4th chapter shows a detailed approach to model unconventional reservoirs with complex fracture conditions; the 5th chapter gives out a few methods specifically designed for fractured unconventional simulation to further analyze the problem and improves accuracy, and finally the 6th chapter summarizes the work and talks about the conclusions drawn from this study.

2. UNCONVENTIONAL RESERVOIR NUMERICAL SIMULATOR

2.1. Overview

The numerical simulation method is an important approach used to investigate the multi-phase, multi-component, multi-physics fluid flow in porous media systems. Various research has been done to develop simulation methods for conventional reservoirs with decent pore size and high permeability. However, research on unconventional fractured reservoirs is relatively rare and still an ongoing process due to its high complexity and uncertainty.

The geometry for fractured reservoirs is usually piecewise complex, for which the traditional block-centered Cartesian grid is no more capable to describe the boundary shapes. Therefore, common finite-difference methods fail to apply at the request of high simulation accuracy, and the finite-volume method (FVM) (Cao, 2002) with two-point-flux-approximation (TPFA) is more favored due to its capability of handling unstructured meshes and complex geometries. On the other hand, the compositional simulation model has been performing better than the traditional black-oil model as it better describes the fluid (especially gas) behavior in nano-pores (Alharthy et al., 2016).

In this dissertation, the code for unconventional reservoirs with complex fracture conditions has been developed based on an in-house reservoir simulator framework named General Unstructured Reservoir Utility (GURU) developed at Texas A&M University (Yan, 2017). It has the following capabilities:

1) Fully implicit method for time discretization.

2) FVM with TPFA for spatial discretization.
3) Multi-physics flow behavior with Darcy/non-Darcy convection flow, multi-component adsorption/desorption, Knudsen diffusion, and gas slippage.
4) Multi-phase multi-component VLE calculation for oil and gas.
5) Multi-porosity multi-permeability model with fracture network implementations.

2.2. Physical and mathematical model

The first problem to accurately describe the fractured unconventional reservoirs is the flow mechanisms in the bulk matrix through nano-sized pores and inside complex fracture networks. The bulk unconventional matrix usually falls into two different categories: organic and inorganic (**Figure 2.1**). The inorganic matrix consists of different minerals such as clay, quartz, and pyrite, and the organic matrix is known as porous kerogen particles distributed along with the inorganic matrix (Ambrose et al., 2011). Nano-sized pores develop in the organic matrix (Wang & Reed, 2009) and these pores contain hydrocarbon molecules, which proves as a combination of desorption from the surface of organic matters and diffusion from kerogen bulk. On the other hand, pore spaces in the inorganic matrix are hydrophilic and easily blocked by water (Zhang et al., 2012). Although connected with organic pores and are commonly larger, the inorganic matrix only takes desorbed hydrocarbon from the surface of the clay minerals into pore spaces and has a much weaker absorption capacity (Ji et al., 2012). Due to the intrinsic difference between these two shale matrices, it is appropriate to divide the matrix into

separate pore systems. A more detailed description of this multi-porosity system will be discussed in the next subsection.

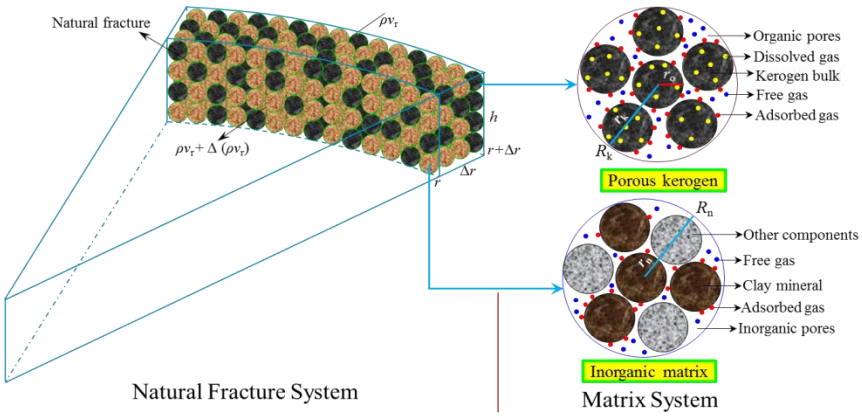

Figure 2.1 Organic and inorganic matters in fractured unconventional reservoirs, edited from Teng et al., 2016.

From the said physical model, the generalized mass balance equation for a specific component at a specific simulation grid is written as:

$$\frac{\partial \emptyset_i \sum S_j \tau_{i,j}}{\partial t} + \frac{\partial (1-\emptyset_i) m_L}{\partial t} = \nabla \cdot \left(\sum k_i \frac{k_{ri} \tau_{i,j}}{\mu_i} \nabla \varphi_{i,j} \right) + \sum q_{d,i} + \sum q_{s,i} + \sum q_{f,i} \quad (2.1)$$

In which case $\tau_{i,j}$ is the multiphase composition. For water, $\tau = \rho_w$ and for hydrocarbon,

$$\tau_{i,j} = x_i \rho_{i,j} + y_i \rho_{i,j} \quad (2.2)$$

In Equation 2.1, the different terms are described from left to right as follows:

1) The mass accumulation over the time domain.
2) The adsorption/desorption.
3) Darcy/non-Darcy convectional flow.
4) The Knudsen diffusion.

5) The source and sink term for wellbore flux.

6) Fracture-matrix convection and diffusion.

For which the diffusion term is written as:

$$q_{d,i} = \sigma_i S_i D_i \nabla C_i \tag{2.3}$$

And the fracture-matrix flux has its convection and diffusions,

$$q_{f,\mathrm{i}} = \sigma \left(k_{i,j} \frac{k_{ri} \tau_{i,j}}{\mu_i} \Delta \varphi_{i,j} \right) + q_{d,i,f} \tag{2.4}$$

The m_L stands for Langmuir adsorption/desorption parameter, with the form:

$$m_L = \rho_b \rho_{r,i} V_{L,i} \frac{\frac{P_{i,j}}{Z_i}}{\frac{P_{i,j}}{Z_i} + P_{L,i}} \tag{2.5}$$

The $\nabla \varphi_{i,j}$ is the potential gradient in simulation grids. To account for the non-Darcy start-up pressure gradient effect (Xiao et al., 2010), the gradient is written in the corresponding piecewise function form:

$$\nabla \varphi_{i,j} = \begin{cases} 0, \nabla \varphi_{i,j} < G \\ \left(1 - \frac{G}{\nabla \varphi_{i,j}}\right) \nabla \varphi_{i,j}, \nabla \varphi_{i,j} > G \end{cases} \tag{2.6}$$

The equation-of-state for hydrocarbons in the simulator is calculated using the Peng-Robinson model (PR EoS) (Peng & Robinson, 1976).

$$p_\alpha = \frac{RT}{V_\alpha - b_\alpha} - \frac{a_\alpha(T)}{(V_\alpha + \delta_1 b_\alpha)(V_\alpha + \delta_2 b_\alpha)} \tag{2.7}$$

Where the real-gas factor is given in a cubic form:

$$\begin{aligned} &Z_\alpha^3 + [(\delta_1 + \delta_2 - 1)B_\alpha - 1]Z_\alpha^2 + [A_\alpha + \delta_1 \delta_2 B_\alpha^2 - (\delta_1 + \delta_2)B_\alpha(B_\alpha + 1)]Z_\alpha \\ &-[A_\alpha B_\alpha + \delta_1 \delta_2 B_\alpha^2 (B_\alpha + 1)] = 0 \end{aligned} \tag{2.8}$$

Vapor-liquid-equilibrium (VLE) is used to determine the phase state and the compositional molecular concentration for each oil and gas component with respect to the composition, pressure, and temperature (**Figure 2.2**). In this dissertation, a few assumptions are made to simplify the calculations:

1) Water will always be a separate liquid phase compared to oil and gas.
2) The other components are not soluble in the water phase.

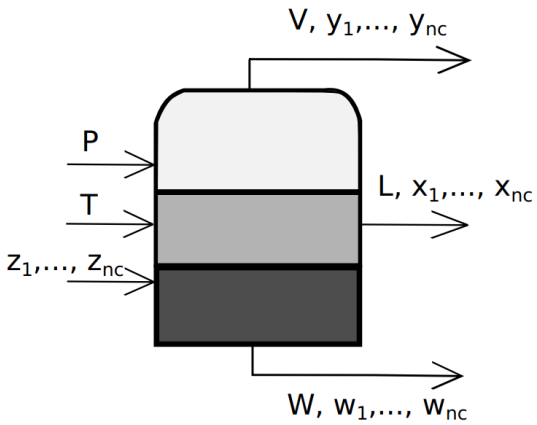

Figure 2.2 Multi-phase VLE with flash calculation

Two sets of calculations are two be made for VLE calculations, known as the stability analysis and the flash calculation. Stability analysis is used to measure whether a specific phase is stable enough and is not going to split into more phases. It is only necessary to test the heaviest component among the multi-phase multi-component model to measure the overall stability of the mixture. Flash calculation is another method that calculates the molecular fraction of every component in every single phase as well as the molecular fraction of every phase. Both methods come from the theory of minimizing

the Gibbs free energy (Ebel et al., 2000). For a multi-phase multi-component system, the isothermal constant-pressure equation of Gibbs energy minimization can be written as:

$$\frac{G(\mathbf{n})}{RT} = \sum_{j=1}^{np} \sum_{i=1}^{nc} n_{ij} \ln f_{ij} \tag{2.9}$$

This dissertation uses an implicit method to solve for phase-component VLEs by combining Newton-Raphson iteration with initial values calculated by successive substitution (NR-SS). A brief illustration of the residual Jacobian matrix is shown as Equation 2.10.

$$J_{VLE} = \begin{bmatrix} \frac{\partial R_{f,1}}{\partial \ln K_1} & \frac{\partial R_{f,1}}{\partial \ln K_2} \cdots & \frac{\partial R_{f,1}}{\partial \ln K_n} & \frac{\partial R_{f,1}}{\partial f_v} \\ \frac{\partial R_{f,2}}{\partial \ln K_1} & \frac{\partial R_{f,2}}{\partial \ln K_2} \cdots & \frac{\partial R_{f,2}}{\partial \ln K_n} & \frac{\partial R_{f,2}}{\partial f_v} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial R_{f,n}}{\partial \ln K_1} & \frac{\partial R_{f,n}}{\partial \ln K_2} \cdots & \frac{\partial R_{f,n}}{\partial \ln K_n} & \frac{\partial R_{f,n}}{\partial f_v} \\ \frac{\partial R_r}{\partial \ln K_1} & \frac{\partial R_r}{\partial \ln K_2} \cdots & \frac{\partial R_r}{\partial \ln K_n} & \frac{\partial R_r}{\partial f_v} \end{bmatrix} \tag{2.10}$$

Further analysis shows that the Newton-Raphson iterations used in VLE calculation has generally a smaller region of convergence despite fast converging. In rare cases when several components have almost similar properties, the Newton-Raphson can simply fail. Thus, to further stabilize the VLE process, it is necessary to save the solution at every solving step.

The compositional fluid properties are calculated using separate correlations. Water viscosity is calculated using the Kestin model (Kestin et al., 1981) shown below:

$$\mu(p, T, m) = \mu^0(T, m)[1 + \beta(T, m) \cdot p] \tag{2.11}$$

The oil and gas viscosities are calculated by Lohrez-Bray-Clark correlation (Lohrenz et al., 1964), shown in Equation 2.12.

$$\mu = \mu^0 + \frac{1}{\varepsilon}[(\alpha_1\rho^4 + \alpha_2\rho^3 + \alpha_3\rho^2 + \alpha_4\rho^1 + \alpha_5)^4 - 0.0001] \quad (2.12)$$

The permeability calculations are more complex due to the usage of the multi-porosity-permeability model. It will be discussed in the next subsection.

2.3. Multi-porosity-permeability model with fracture network

As mentioned in **Figure 2.1**, unconventional reservoirs have more complex lithology and pore structures compared with conventional plays. In fractured unconventional systems, the fracture network dominates fluid transportation while the matrix holds major fluid storage. Thus, common single-porosity models are replaced by multi-porosity settings (**Figure 2.3**) to better describe the coexistence of different pore systems (Ambrose et al., 2010; Sondergeld et al., 2010).

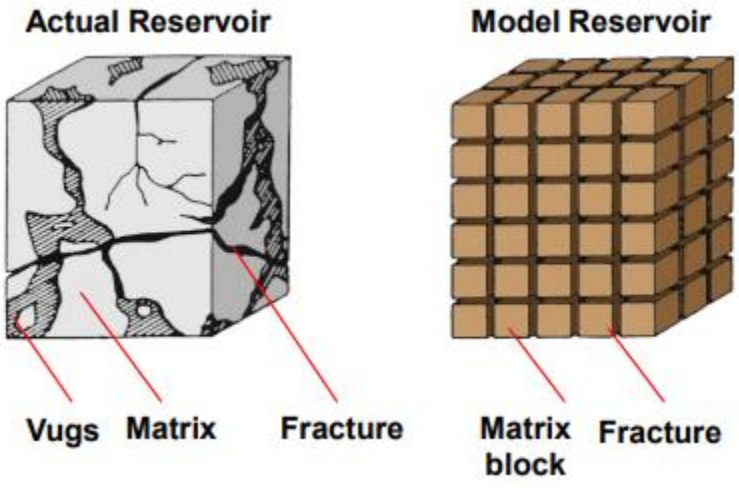

Figure 2.3 The multi-porosity model setting (Sondergeld et al., 2010)

In this dissertation, the dual-porosity model from Warren and Root (Warren & Root, 1963) is further extended into a generalized form (Yan et al., 2013) and the matrix is further spliced by organic kerogen and inorganic clay matters to form a triple-porosity dual-permeability reservoir model with smaller-than-grid-dimension micro fractures and coupled with embedded discrete fracture model (EDFM), shown in **Figure 2.4**.

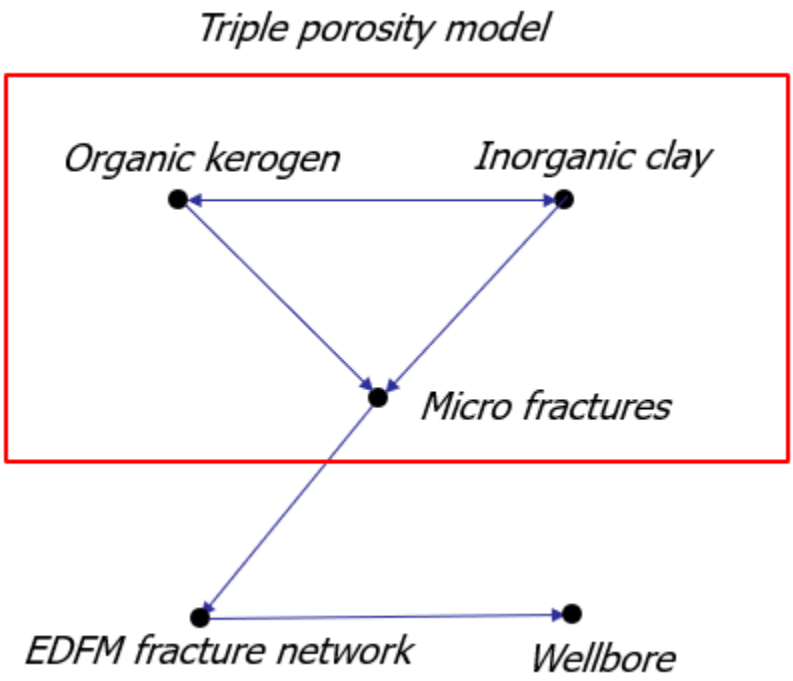

Figure 2.4 The triple-porosity dual-permeability model with fracture coupling

For computational performance, we further define the fluid flow pattern inside different porosity types. Fluid flow can appear from organic to inorganic pores and vice-versa, while only the one-dimensional flow is allowed for fluids to enter the micro-fractures. The micro-fracture system is then combined with the EDFM system which serves as the large-scale flow pathway.

The shape factors for matrix-microfracture flow are derived from Kazemi's model (Kazemi et al., 1976) with the following form:

$$\sigma = 4\left(\frac{k_i}{{L_i}^2} + \frac{k_j}{{L_j}^2} + \frac{k_k}{{L_k}^2}\right) \tag{2.13}$$

Besides the Darcy convection, Langmuir absorption/desorption, and diffusion process mentioned above, more fluid flow mechanisms are also implemented with respect to this multi-porosity-permeability model. Fluid follows Darcy flow inside fractures, and slippage flow takes control when flowing from matrix pores to fractures (Teng et al., 2016). This gives us a continuous, linear superposition model for gas flow in shale (Civan, 2010) considering slip flow and Knudsen diffusion. Water flow in shale is less studied compared with hydrocarbon. Using steady-state Darcy flow in both pore space and fractures (Sun & Schechter, 2015; R.-h. Zhang et al., 2019) can significantly simplify the calculation and saves computational time with the cost of accuracy. This Darcy versus non-Darcy effect will be further discussed in Chapter 5.

Fractures with dimensions larger than grid size are modeled separately using EDFM by injecting fracture planes into the matrix system and connecting them together using non-neighbor connections (NNC). A more detailed description of the EDFM method, as well as transmissibility calculations, can be found in Chapter 4.

2.4. Global Jacobian system with wells and fractures

To obtain the unconditionally stable nature with respect to time-step size, the fully implicit method (FIM) is used to manage the time discretization. The said method requires solving a large Jacobian system of linear equations at every Newton-Raphson

iteration step. The residual Jacobian can be derived based on the governing mass-balance partial differential equations, where the residual has the form of:

$$R_i^{k+1} = \frac{1}{\Delta t}\left(N_i^{k+1} - N_i^k + D_i^{k+1} - D_i^k\right) + \sum q_{i,flux} + \sum q_{i,src} = 0 \qquad (2.14)$$

The constructed residual Jacobian for the reservoir with respect to the matrix and EDFM fracture systems has the form of **Figure 2.5**. With proper reordering of fracture-matrix items, it can be divided into different sub-matrices:

1) The fracture-to-fracture sub-matrix (top-left).
2) The matrix-to-matrix sub-matrix (bottom-right).
3) & 4) The fracture-matrix sub-matrix (top-right and bottom-left, similar by transposing).

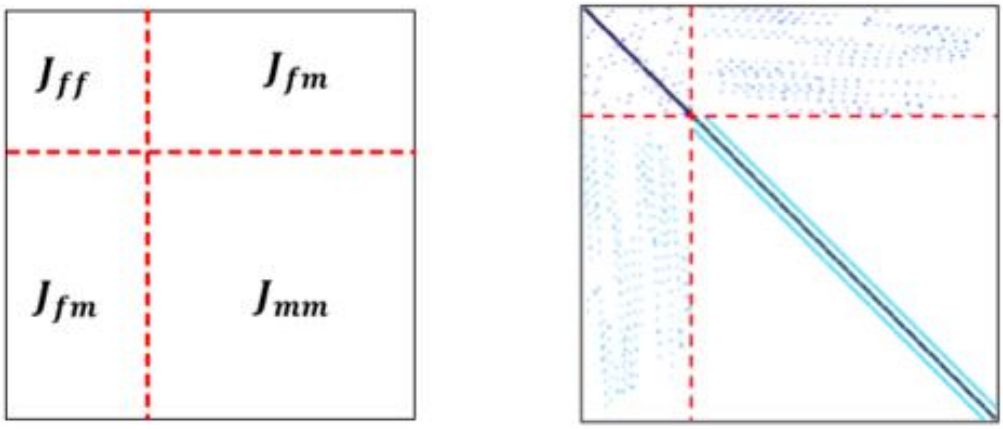

Figure 2.5 The residual Jacobian matrix for matrix-fracture system with sub-division (left) and illustration of datapoints (right), edited from Yan et al. 2017

Wellbore settings are calculated using the Peaceman model (Peaceman, 1990) with the form of Equation 2.15.

$$Q_i = WI \sum_{j=1}^{np} x_{i,\varepsilon_j} \frac{k_{rj}}{\mu_j} \left(P_j - P_{bh}\right) \tag{2.15}$$

We use the implicit well method to couple wellbore flow into the reservoirs. By constructing the well residual Jacobian and attaching them after the degree-of-freedom (DOF) of reservoir Jacobian, the following block-matrix can be generated:

$$\begin{bmatrix} J_{RR} & J_{RW} \\ J_{RW}{}^{T} & J_{WW} \end{bmatrix} \begin{bmatrix} u_R \\ u_W \end{bmatrix} = \begin{bmatrix} f_R \\ f_W \end{bmatrix} \tag{2.16}$$

Whereas J_{RR} is the reservoir residual Jacobian, J_{WW} is the well residual Jacobian, and J_{RW} is the coupled well-reservoir submatrix. An illustration of this global Jacobian system is shown in **Figure 2.6**.

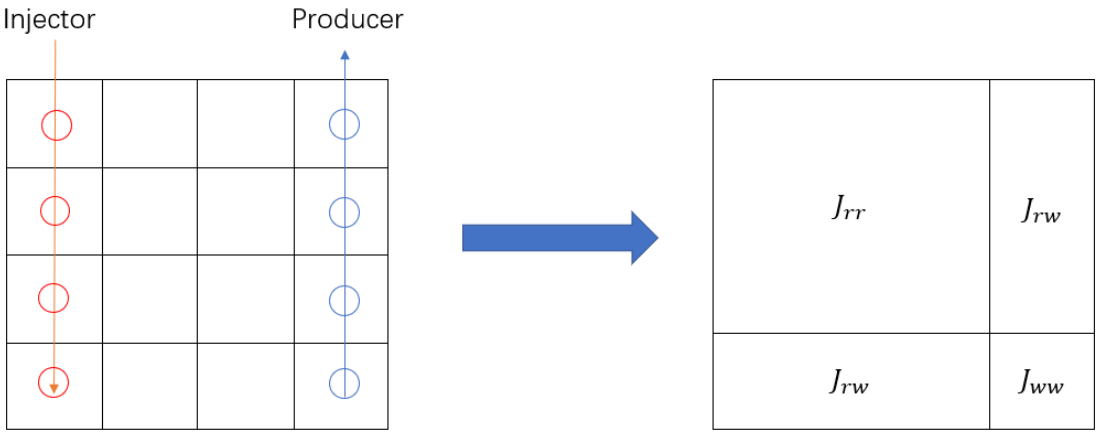

Figure 2.6 The illustration of global Jacobian with implicit well settings

It is worth noticing that coupling the well Jacobian using this implicit format will worsen the ellipticity of pressure functions as the wellbore equations behave differently compared with the matrix. Thus, the Jacobian matrix conditions badly due to the large differences between fracture and matrix flow as well as wellbore behavior, making the

system no longer diagonal dominating and ill-conditioned and requiring further preconditioning before the linear solving phase.

2.5. Case study

A case study is performed to test and further validate the simulator designed above using a single well model settings from the YOST reservoir. The original reservoir model contains 3.95 million corner-point grids as well as 168 large-scaled hydraulic fractures connected to a horizontal wellbore with reasonably high heterogeneity. The illustration of model properties is shown in **Figure 2.7** and **Figure 2.8**.

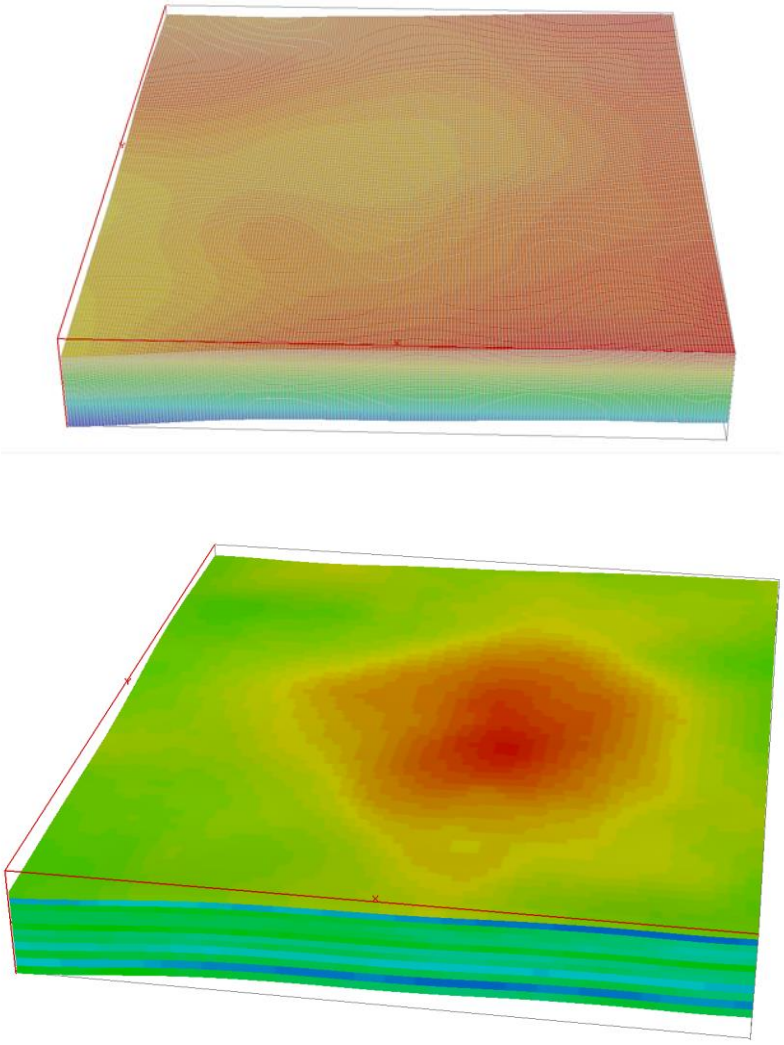

Figure 2.7 The illustration of YOST well model depth (top) and porosity (bottom)

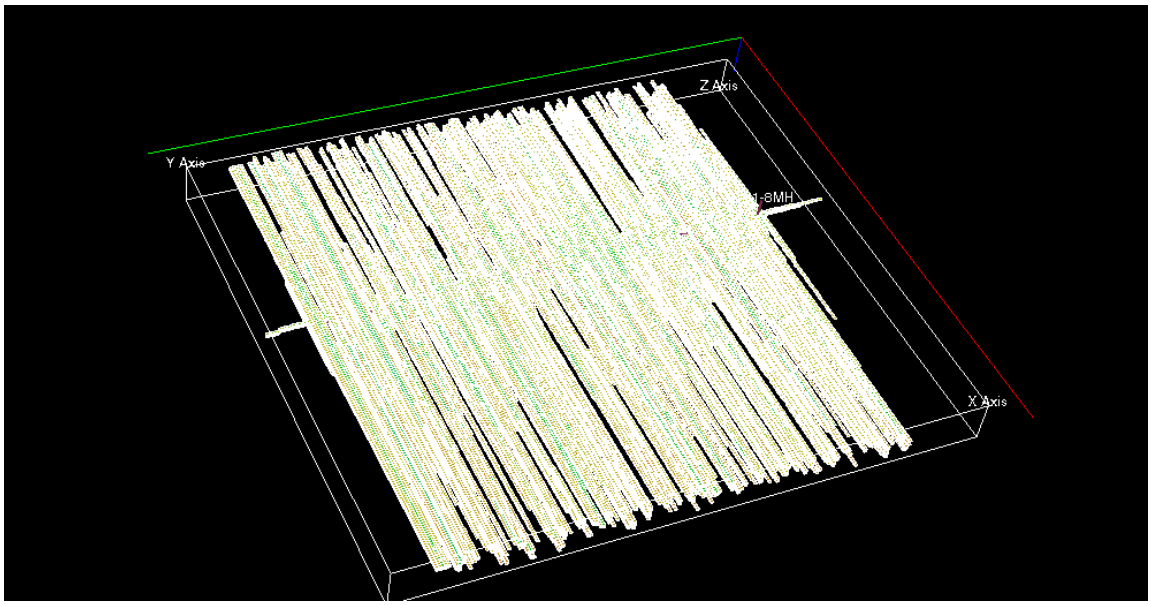

Figure 2.8 The illustration of YOST wellbore with hydraulic fractures

The simulation is performed with GURU, and the results are shown in **Figure 2.9**. A decent match is observed between the history data and simulated well bottom-hole pressure, which provides a good validation for GURU's accuracy and capability.

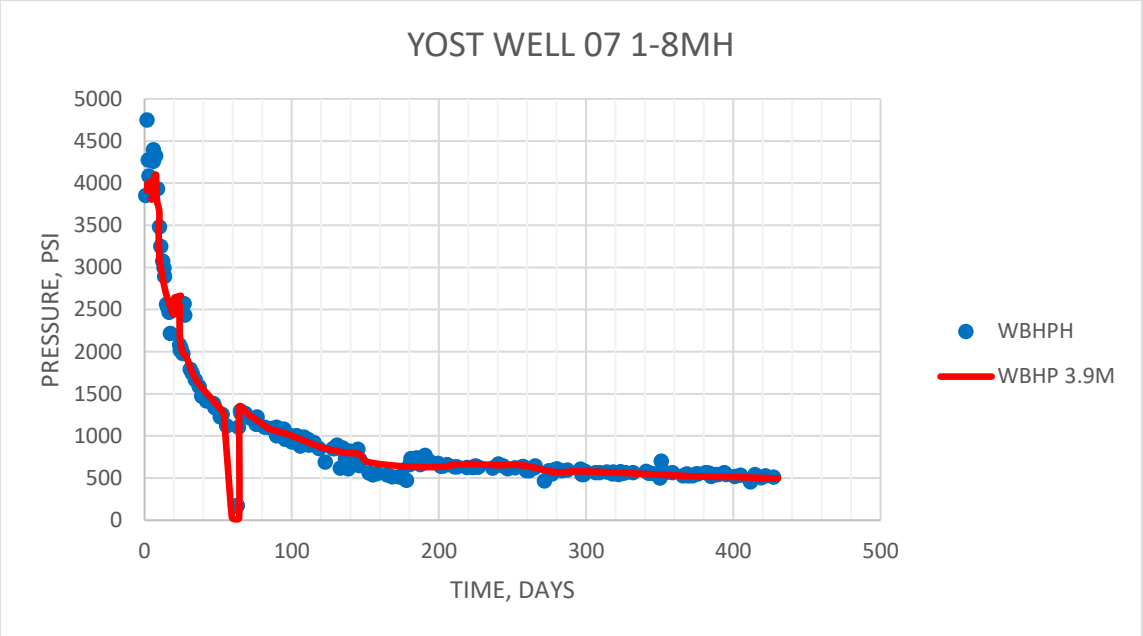

Figure 2.9 Yost well model validation.

Another case study is performed to further test GURU's robustness by performing upscaling to the YOST well model. The upscaled 1.5 million and 0.39 million cell models match with the original model (**Figure 2.10)**, proving that GURU has enough accuracy, robustness, and capability to hold unconventional reservoir simulations with complex fracture conditions.

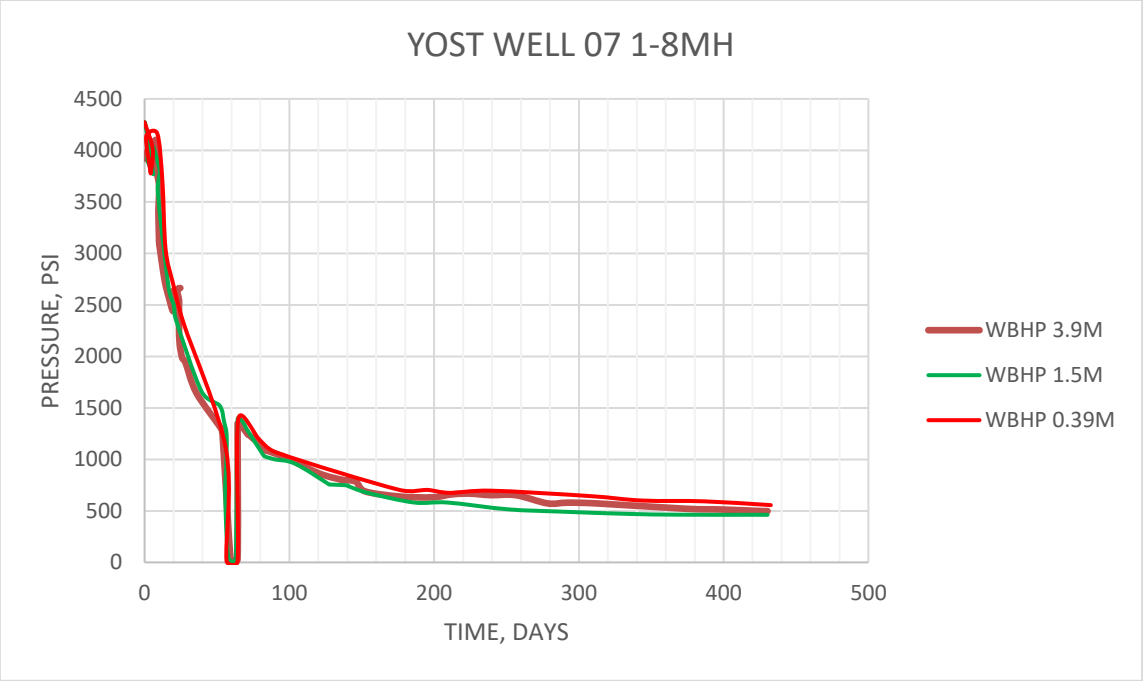

Figure 2.10 Yost well model upscaling case study

3. UNCONVENTIONAL PERFORMANCE OPTIMIZATION

3.1. Overview

The time-consuming part of unconventional reservoir simulation comes down to mathematics. When we tune the GURU with test cases, it is well found that the most CPU-extensive tasks come from the Newton-Raphson iterations used in FIM discretization (Douglas et al., 1959; Monteagudo & Firoozabadi, 2007). Although FIM is the state-of-the-art mathematical method in the petroleum industry due to its unconditional stability and low restriction on timestep settings, it requires to construct and solve a large, usually sparse, highly coupled, ill-conditioned Jacobian linear system of equations at every Newton step. This Jacobian construction can easily take up to 50% of CPU time, as well as linear solving which can go up to 80% in some extreme cases. Thus, the performance in unconventional simulations with complex fracture conditions is all about optimizing the time-cost for these two steps.

The linear solving method can be divided into two categories: direct and iterative methods (Saad, 2003). The direct method like Gaussian elimination solves the linear system of equations without mathematical error and does not require a set-up stage, thus can be effective for problems with dense matrix and limited size like VLE calculations. Iterative methods have a generally lower request of memory and better scalability compared to the direct methods. Thus, they have been widely used to solve the global Jacobian matrix. For reservoir simulators without strict symmetricity, Krylov subspace methods like GMRES and BiCGstab are more favored. When the Jacobian becomes

complex and the condition number worsens, preconditioning methods (Xu, 1992) are used to accelerate the converging speed for such iterative solvers.

Due to the further development of modern multi-core computer architectures, developers can further accelerate the unsatisfactory simulator performance by making full use of the hardware's computing power. Converting from the serial program to parallel may also require redesigning the parallel algorithms from previous serial ones. Much research has been proposed in this field (Mesbah et al., 2019; Werneck et al., 2019; Yang et al., 2019), and the dissertation will talk about the method developed and used for GURU using both multi-core CPU and GPU accelerations.

3.2. Iterative solvers

3.2.1. General linear iterative methods

This section talks about a basic problem in the field of scientific computation. Given a large sparse real matrix $A \in R^{N\times N}$ of any form and a corresponding real vector $x \in R^N$, it is expected to find a real vector $b \in R^N$ that satisfies,

$$Ax = b \tag{3.1}$$

The iterative solver converts the linear system of equations into a new form. By picking up a set of initial guesses for $x_i^{(0)}(i = 1,2,\cdots,N)$, results get updated based on calculations from every iteration and gradually converge to an approximate solution with satisfactory tolerance. The general form of iterative solver can be written as:

$$x^{k+1} = x^k + Q(b - Ax^k) \tag{3.2}$$

Whereas the matrix Q is known as the iterator. Preconditioning methods are developed such that the obtained $Q \approx A^{-1}$ to simplify the iteration calculations. More

detailed discussions on solvers and preconditioners are listed in the following subsections.

3.2.2. Jacobian matrix storage methods

The global Jacobian for unconventional reservoirs with EDFM fractures can be written in the following form with reordering (Yamazaki et al., 2011).

$$J_R = \begin{bmatrix} J_{mm} & J_{mf} & J_{mw} \\ {J_{mf}}^T & J_{ff} & J_{mw} \\ {J_{mw}}^T & {J_{fw}}^T & J_{ww} \end{bmatrix} \tag{3.3}$$

Whereas any element of J for a specific grid can be written as a block matrix of residuals with respect to pressure temperature and component concentrations. This matrix can be extremely large but sparse, thus it is necessary to design specific methods to store it inside memory properly.

The common approach is to store the sparse Jacobian matrix using the compressed sparse row (CSR) method or block-compressed sparse row (BCSR) method (Saad, 2011). Both methods save the non-zeros of Jacobian and keep dual lists for their column numbers and the starting index of row numbers while the BCSR further split the matrix into smaller dense submatrices. However, there can be differences in performance depending on how the data in the matrix are ordered and stored. For a specific 2-D two-phase problem, it is possible to store the non-zeros either according to its CSR DoF or its BCSR grid pattern, which is shown in **Figure 3.2**.

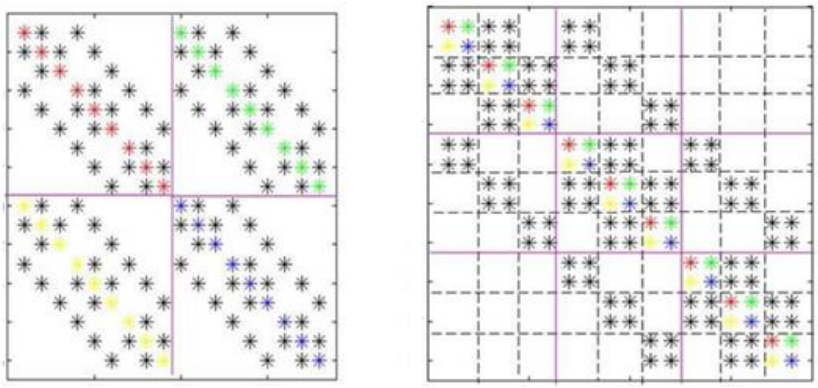

Figure 3.1 The non-zeros for a 2-dimensional 2-phase problem sparse Jacobian based on CSR DoF (left) and BCSR grid pattern (right)

Tests have been performed on both cases, and the grid pattern storage mode generally gives out better computational efficiency. This is due to the local storage of submatrix for the DoFs on every grid unit give out better memory access and less cache missing. Thus, BCSR with grid pattern is implemented in GURU to replace CSR for better performance.

3.3. The multi-stage preconditioner

The multi-porosity-permeability model with multi-component settings can be solved for the following DoFs:

1) Pressure, P.
2) Temperature (will be reduced in isothermal models), T.
3) Saturation (concentration) for every component, $S = [S_1, S_2, \cdots, S_n]^T$.

As the different DoF are measured with separate equations, their mathematical quality also differs. In this dissertation, every DoF is assigned with its corresponding auxiliary space, and preconditioners can be designed based on the subspaces separately.

3.3.1. Schur complement methods for preconditioning

Preconditioning can be a key factor to improve the computational efficiency, especially in reservoir cases where the eigenvalue distributions of Jacobian can be poorly distributed. An illustration of a simple test sample eigenvalue distributions is shown for 20x20x6 grids with 2 wells in **Figure 3.3**. It can be easily noticed that this Jacobian is poorly conditioned. When it comes to the even more complex Jacobian in fractured unconventional reservoirs, preconditioner becomes an urgent need.

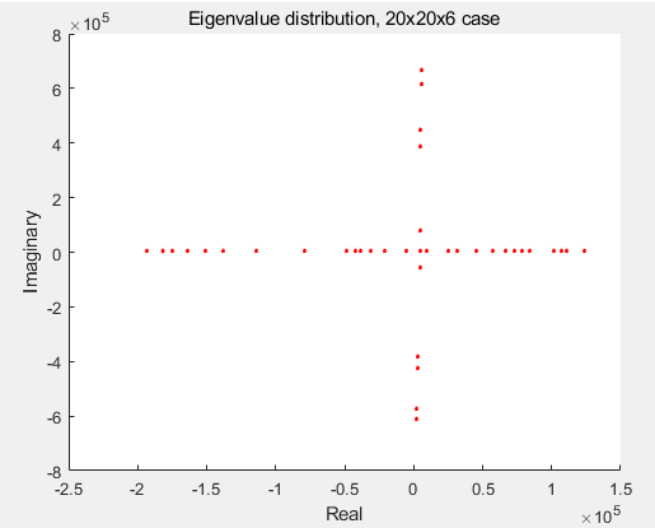

Figure 3.2 The eigenvalue distribution for a test case 10x10x3

There have been a few common methods for sparse matrix preconditioning, including the incomplete LU factorization (ILU) (Concus et al., 1985; Meyerink, 1983), the multigrid method (MG), and the corresponding algebraic multigrid (AMG) (Brandt, 1984; Falgout, 2006; Notay, 2010). The ILU preconditioner can be easy to design and have good efficiency in medium-sized single-phase simulations. However, its convergence gets gradually worsens with enlarged problem sizes which can cause

problems. AMG is more suitable for the elliptical pressure equations and has better capability than ILU. But as the non-symmetricity and heterogeneities get larger in unconventional reservoirs as well as the preconditioners applied, the ellipticity gets destroyed and the AMG performance can significantly degenerate or simply becomes non-usable.

The constrained pressure residual (CPR) method (Wallis, 1983; Wallis et al., 1985) takes advantage of both ILU and AMG based on the mathematical nature that the pressure functions are elliptical (without the effect of capillary pressure) while the concentration/saturation functions are more towards hyperbolic (Roy et al., 2019). It serves as a decent decoupling method in reservoirs where the pressure subblocks are more dominating by reducing the coupled relationship between pressure and concentration blocks and solving the approximate Schur complement (Smith, 1997) pressure system. The basic equation of CPR is shown as below by finding a decoupling matrix M as the multiplier for Equation 3.1,

$$Q_{\mathrm{CPR}} = S(I - AM) + M \tag{3.4}$$

The general idea of the pressure-saturation (concentration) decoupling process can be described as below:

$$\begin{bmatrix} I & -J_{PS}J_{SS}^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} J_{PP} & J_{PS} \\ J_{SP} & J_{SS} \end{bmatrix} = \begin{bmatrix} J_{PP} - J_{PS}J_{SS}^{-1}J_{SP} & 0 \\ J_{SP} & J_{SS} \end{bmatrix} \tag{3.5}$$

For which the modified matrix reduces the Schur complement of concentration/saturation submatrix to zero. The S in Equation 3.4 represents the smoother, for which ILU-based methods are commonly used. Common approaches to generate the M matrix includes alternative block factorization (ABF), quasi-IMPES, and

true-IMPES methods (Cusini et al., 2015). Among the three methods, the ABF generates a strongly nonsymmetric pressure submatrix and destroys its ellipticity in full. The IMPES-based methods keep the symmetricity better, while, the true-IMPES approach keeps the ellipticity of pressure equations by most although is also harder to build up compared with the quasi method. A table representing tests of different decoupling approach results is shown in **Table 3.1**. The classical AMG is used to solve the original pressure functions directly and results are compared with the decoupled pressure sub-matrix from the three CPR methods to compare the time consumption and numbers of iterations. It is found that the ABF method greatly increases the iteration numbers of pressure functions, while quasi- and true-IMPES methods tend to keep up with the classical AMG iterations. Thus, the quasi-IMPES is selected in this dissertation to be the desired CPR decoupling method for the pressure-concentration problem.

Table 3.1 Comparison of the different CPR decoupling methods.

	Classical AMG		ABF		quasi-IMPES		true-IMPES	
Case	Time	Iters	Time	Iters	Time	Iters	Time	Iters
1	6.01	11	19.01	31	7.97	13	8.02	13
2	10.21	5	15.57	12	10.43	6	10.31	6
3	28.6	8	37.6	10	36.4	10	39.17	10
4	114.9	13	175.2	20	138.6	15	142.9	15
5	50.83	11	158.7	33	102.4	19	98.12	18

The decoupling matrix for the quasi-IMPES method is described as below:

$$M_{quasi-IMPES} = \begin{bmatrix} I & -diag(J_{PS})diag(J_{SS})^{-1} \\ 0 & I \end{bmatrix} \tag{3.6}$$

The CPR method can be reorganized into a multi-stage multiplicative preconditioner form in pressure auxiliary space, which is written as:

$$I - Q_{\mathrm{CPR}}A = (I - SA)(I - U_P Q_P U_P^T A) \tag{3.7}$$

Whereas Q_P is the Schur complement for pressures generated using the quasi-IMPES method, and the U_P is the auxiliary space convertor for pressure-to-global space with the following form:

$$U_P = \begin{bmatrix} I_P \\ 0 \end{bmatrix} \in R^{N \times N_P} \tag{3.8}$$

For the more complex problem within unconventional reservoirs with complex fracture conditions, the traditional CPR method is no more capable enough to hold the major difference in wellbore flux as well as fracture networks in the multi-porosity-permeability model. Thus, a further developed multi-stage preconditioner is developed based on a similar but more advanced approach. The detailed process is discussed in the next subsection.

3.3.2. General multi-stage pressure-saturation preconditioner

It is easy to further extend the CPR multiplicative preconditioner into a more generalized case with N different auxiliary spaces by creating multiple individual Schur complements, for which a general form of such CPR-type multi-stage multiplicative preconditioner can be written in the following form:

$$I - Q_{\text{ms}}A = (I - SA)\prod_{i=1}^{N}(I - U_i Q_i U_i^T A) \tag{3.9}$$

The multi-component multi-phase model has a complex saturation (concentration) part, which requires an extra stage of preconditioning for both pressure and saturation space. Thus, A preconditioner is written as:

$$I - Q_{\text{ms-ps}}A = (I - SA)(I - U_P Q_P U_P^T A)(I - U_S Q_S U_S^T A) \tag{3.10}$$

The pseudo-code description of this two-stage preconditioning can be shown as below:

Table 3.2 The two-stage preconditioner for pressure-saturation problem

Step 1. Given the residual vector r and initial set of vectors x_0.

Step 2. Solve the saturation (concentration) problem based on the following equation:

$$x_1 \leftarrow x_0 + U_S B Q_S U_S^T (r - Ax_0) \tag{3.11}$$

Step 3. Solve the pressure problem based on the following equation:

$$x_s \leftarrow x_1 + U_P Q_P U_P^T (r - Ax_1) \tag{3.12}$$

Step 4. Smoothing the solved vector x_s with Gauss-Seidel smoothing.

$$x \leftarrow x_s + S(r - Ax_s) \tag{3.13}$$

Step 5. Global preconditioning the vector x with ILU (0).

In the table above, the Q_P part is solved approximately using a single-stage AMG V-cycle, while the Q_S part is obtained with one iteration of the Gauss-Seidel iteration. Global smoother S is also taken by Gauss-Seidel, followed by another stage of global Jacobian preconditioning generated using ILU after the smoothing in Equation 3.13.

Tests have been performed to validate such preconditioner with implementations using a 7-component conceptual reservoir model with different problem sizes solved by PGMRES and tolerance error 10^{-3}. Comparisons are made to the common ILU (0) preconditioned PGMRES approach, and results are shown in **Figure 3.3**.

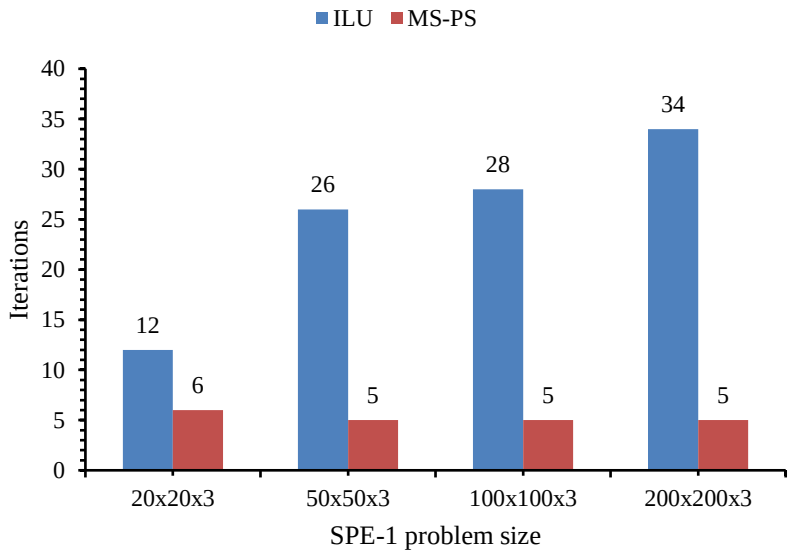

Figure 3.3 Comparison of two-stage preconditioner to ILU preconditioner

It is worth noting that the ILU preconditioner is still capable to solve the problem effectively in such cases. However, the iteration number gets much larger with respect to the increased problem size. On the other hand, the $Q_{\text{ms-ps}}$ preconditioner performs better with much lower iteration numbers, and the iteration number keeps stable for different sized problems. This can be effective in large and complex systems. However, small matrices may not benefit much from the reduced iteration numbers as the set-up phase for this multi-stage preconditioner requires an extra amount of time.

3.3.3. Multi-stage preconditioner for fractures and wells

Adding fracture systems and wells make the system more complex as they hold different flow equations, and the numerical quantity differs which significantly worsens the matrix condition. In this dissertation, fractures and wells are added into the matrix Jacobian by extending the global matrices, shown in **Figure 3.4**.

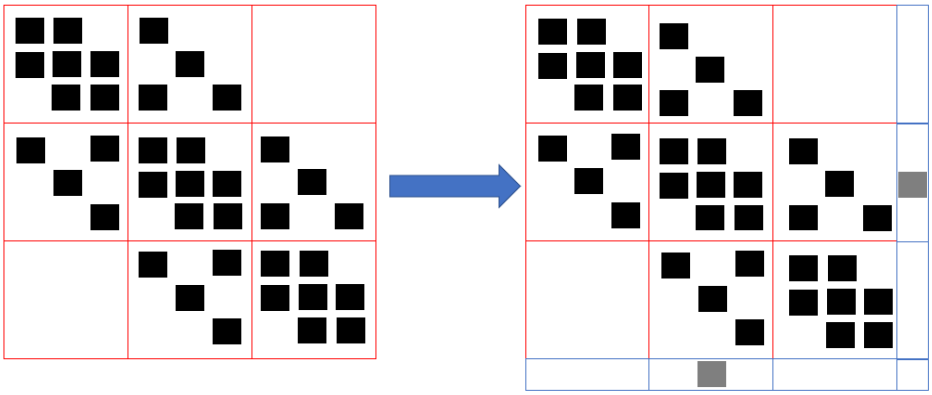

Figure 3.4 Extending the global matrix for fractures and well systems

The fracture numbers and well connections are usually much less than the reservoir matrix. However, they have significant contributions to the reservoir flow model and are coupled deeply in the global Jacobian. Thus, the $Q_{\text{ms-ps}}$ preconditioner does not give out a satisfactory convergence rate. It is necessary to set up a new CPR-style multi-stage preconditioner accounting for both the wells and fractures based on the $Q_{\text{ms-ps}}$ for the global Jacobian, in the form below:

$$I - Q_{\text{msg}}A = (I - SA)(I - U_P Q_P U_P^T A)(I - U_S Q_S U_S^T A)(I - U_F Q_F U_F^T A)(I - U_W Q_W U_W^T A) \quad (3.14)$$

The pseudo-code description of this multi-stage preconditioning can be shown as below:

Table 3.3 The multi-stage preconditioner for fracture and well system

Step 1. Given the residual vector r and initial set of vectors x_0.

Step 2. Solve the well system based on the following equation:

$$x_1 \leftarrow x_0 + U_W Q_W U_W^T (r - Ax_0) \tag{3.15}$$

Step 3. Solve the fracture system based on the following equation:

$$x_2 \leftarrow x_1 + U_F Q_F U_F^T (r - Ax_1) \tag{3.16}$$

Step 2. Solve the saturation (concentration) problem based on the following equation:

$$x_3 \leftarrow x_2 + U_S Q_S U_S^T (r - Ax_0) \tag{3.17}$$

Step 3. Solve the pressure problem based on the following equation:

$$x_s \leftarrow x_3 + U_P Q_P U_P^T (r - Ax_1) \tag{3.18}$$

Step 4. Smoothing the solved vector x_s with Gauss-Seidel smoothing.

$$x \leftarrow x_s + S(r - Ax_s) \tag{3.19}$$

Step 5. Global preconditioning the vector x with ILU (0).

Equation 3.17 counts back into the method mentioned in **Table 3.2** by performing calculations separately for pressure and saturation using Equation 3.11 and 3.12. A global Gauss-Seidel is also performed on Equation 3.18 as well as a next-stage ILU preconditioning to obtain the final preconditioner Q_{msg}.

The Q_W and Q_F here require explicit care in this system. In the form of an extended matrix, their decoupling process can also be calculated using the Schur complement method to obtain the desired stage-preconditioner with satisfactory accuracy. Specifically for the wells and fractures, both the inverse of their Schur

complements and the inverse of the corresponding subspace matrices are approximated using a two-stage AMG V-cycle.

It is also worth noting that the order used to solve the multiple subspaces is important. In unconventional simulations with fractures and wells, it is chosen that we perform the preconditioning firstly to wells, then to fractures, move to saturations, and the last to pressures, and smooth the global Jacobian only once after solving all subspace problems. This provides higher robustness in reservoir simulation tests.

A case study has been performed on a reservoir with 200x200x3 grids and a 7-component fluid model. Multiple fracture models are created to validate the multi-stage preconditioners and comparisons are made with respect to the ILU (0). Results of iterations and CPU times are shown in **Figure 3.5** and **Figure 3.6**, respectively.

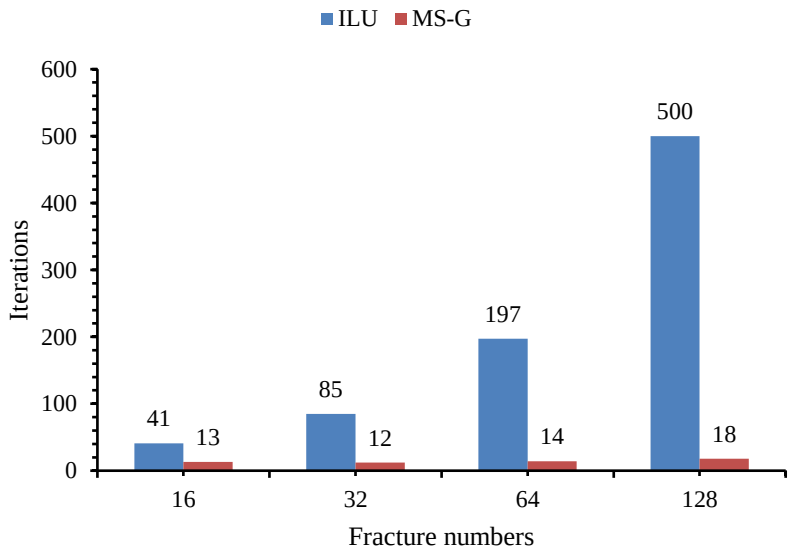

Figure 3.5 Comparison of multi-stage preconditioner to ILU preconditioner, iteration numbers. The ILU preconditioned case failed to converge for the 128-fracture case after 500 iterations, thus only the finished iteration number is listed.

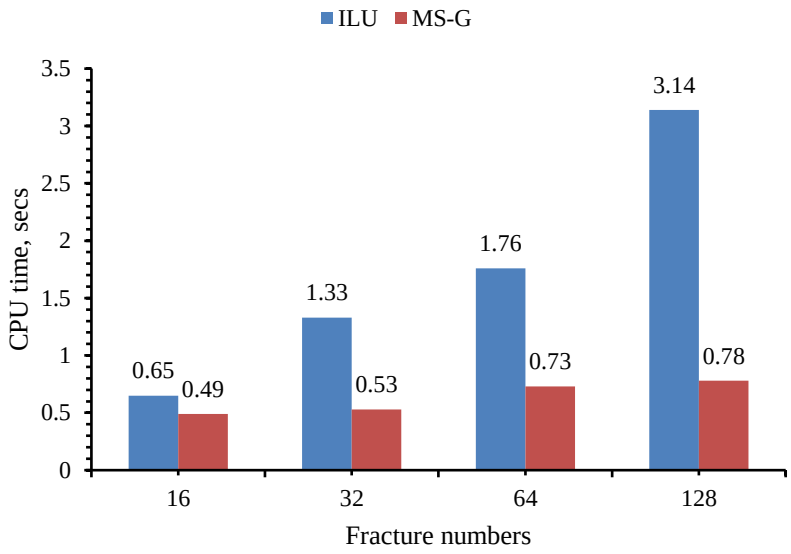

Figure 3.6 Comparison of multi-stage preconditioner to ILU preconditioner, CPU time cost. The ILU preconditioned case failed to converge for the 128-fracture case after 500 iterations, so the result for only these 500 are listed.

It is worth noting that adding fractures and wells significantly worsens the Jacobian condition. ILU becomes significantly slower and even unusable when fracture numbers are large. On the other hand, the iteration number of the global multi-stage preconditioner slightly increases but mostly stays stable, and it significantly reduces the iteration number and has a much smaller CPU time cost.

3.3.4. Linear solvers for the subspace problems

The pressure subspace and saturation subspace behave differently based on their mathematical form of equations. Thus, it is necessary to develop different solving strategies for them specifically.

The pressure subspace is usually large in DoFs. We would like to keep its ellipticity by using CPR-based quasi-IMPES decoupling, and thus the system can be approximated fast using classical AMG methods.

The saturation (concentration) subspace, on the other hand, is hyperbolical, and the saturation profile is mostly dominated by the pressure distribution. Due to this physical behavior, it is appropriate to perform a reorder to obtain a downwind scheme (Courant et al., 1952), which can theoretically generate a lower triangular matrix. A model with two components and 2-D grids is illustrated for the saturation matrix as below:

$$J_S = \begin{bmatrix} J_{S_1,S_1} & J_{S_1,S_2} \\ J_{S_2,S_1} & A_{S_2,S_2} \end{bmatrix} \tag{3.19}$$

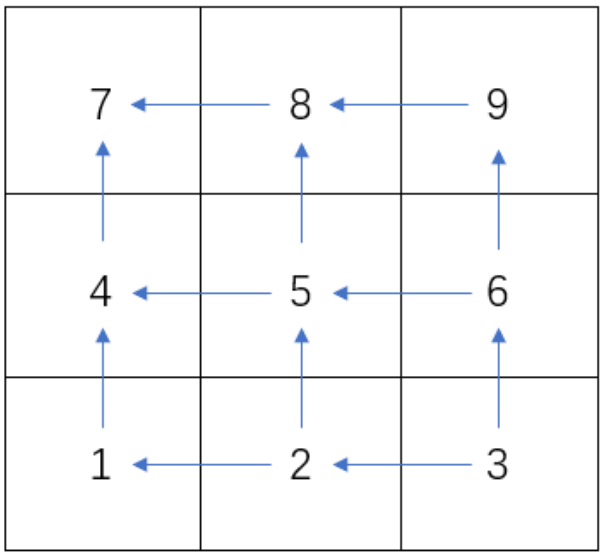

Figure 3.7 Reordering method for downwind scheme with respect to the pressure profile. The blue arrows show the pressure difference among the grids.

The reordering method based on pressure difference is shown in **Figure 3.7**. A comparison between the grid-index ordering and downwind ordering is shown in **Figure 3.8**.

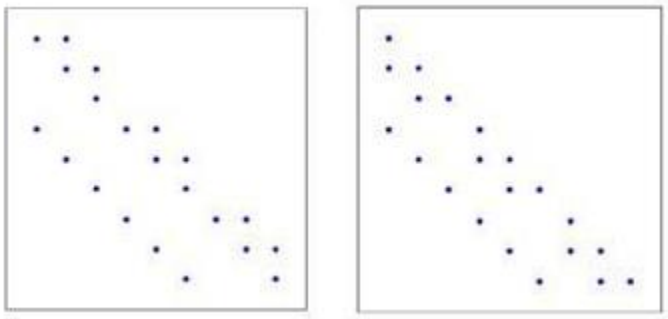

Figure 3.8 Matrix non-zeros for the illustrated case in Figure 3.7 with grid-index ordering (left) and downwind ordering (right).

Compared to the grid-index method, the downwind reordering provides a better pattern, which can be solved effectively using the Gauss-Seidel method.

3.4. Parallelization of Unconventional Simulator

Parallel computation is a method to execute multiple instructions at a single time. The design of parallelization is to accelerate the computational speed and raise the capability of solving larger and more complex problems (**Figure 3.9**). Compared with the serial mode, the parallel computation has the following aspects:

1) A problem is broken into discrete parts that can be solved concurrently.
2) Each part is further broken down into a series of instructions.
3) Instructions from each part execute simultaneously on different processors.
4) An overall control/coordination mechanism is employed.

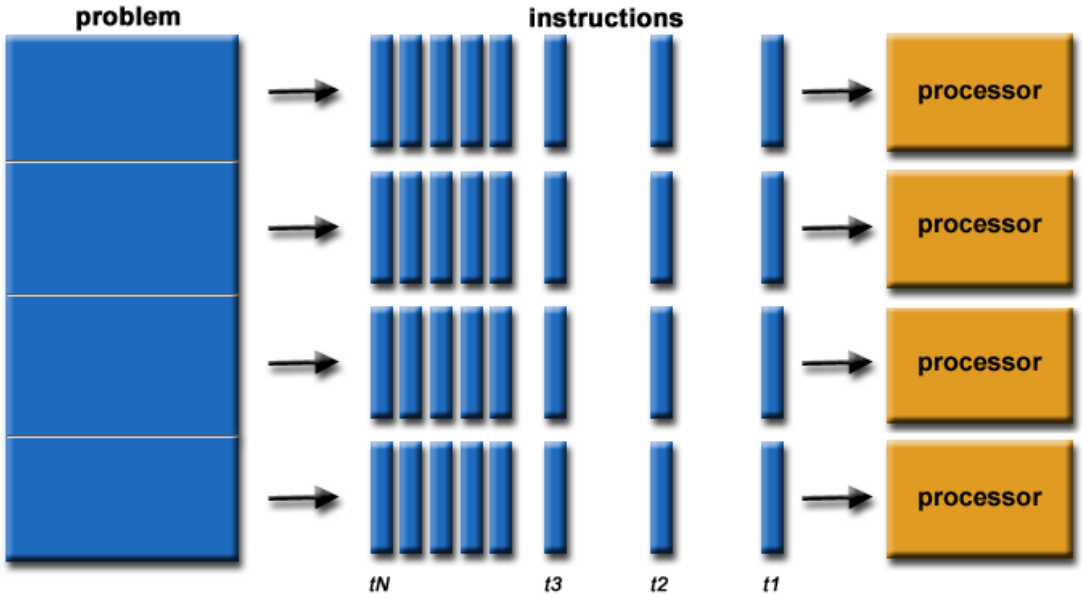

Figure 3.9 Illustration of parallel computation method, edited from *Introduction to Parallel Computation* (Grama et al.).

3.4.1. Parallel programming models

The computational efficiency benefits a lot from parallel computation, especially by making more use of hardware capability. The modern CPUs and GPUs are designed with multi-core architectures. Thus, it becomes necessary to design programs running parallelly. Multiple methods and interfaces have been developed to allow parallel programming on different hardware, for which the dissertation will give a brief review of a few widely used models.

OpenMP (Dagum & Menon, 1998) is a parallel programming model designed for shared-memory multi-core CPU architectures. By providing a high-level abstract description of control functions and a fork-join model, it allows the compiler to automatically build parallel programs. It is easy to design and use, but not efficient for

cases with complex thread communication. Nor can it be used in distributed-memory architectures, for which MPI is more preferred.

MPI (Forum, 1994) is a standard for message-passing interfaces between processes. It provides a model for designing parallel programs with message-passing models. This model has the capability to be used on both shared and distributed architectures and has much wider portability and better CPU efficiency compared to OpenMP. MPI is one of the key techniques in the field of large-scale numerical simulation, however, programs need to be designed with much care and effort.

CUDA (Nickolls et al., 2008) is a software and hardware architecture for GPU computing designed by NVIDIA. Compared with the multi-core CPUs, GPUs have much larger numbers of computing processors (cores) using a SIMD architecture **(Figure 3.10)** which allows much higher floating-point performance. However, GPU is only more effective for computation with less logical complexity since GPU does not support advanced mechanisms like branch prediction and the cores are generally weaker compared to CPUs.

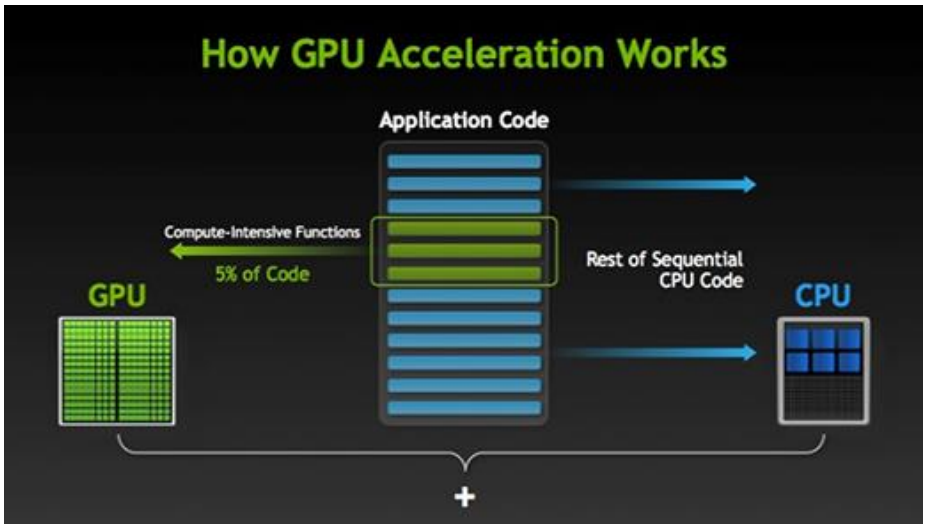

Figure 3.10 Illustration of GPU acceleration (Goyat & Sahoo, 2019)

3.4.2. Optimize the Jacobian construction

The Jacobian construction is one of the most time-consuming parts in unconventional compositional simulations as massive calculations are expected for every single grid at every Newton iteration step. However, the process only reads connection data from the last calculated Newton step, for which calculations can be made individually without data dependency. Thus, the Jacobian construction process in the unconventional simulation itself is a vectorized process and can be highly parallelizable conceptually. A test case with 200x200x3 grids has been made to test the parallelized performance for Jacobian matrix construction, and the results are shown in **Figure 3.11**. However, raising the thread numbers to a much higher value does not give out satisfactory performance increases due to the complexity involved in VLE calculations.

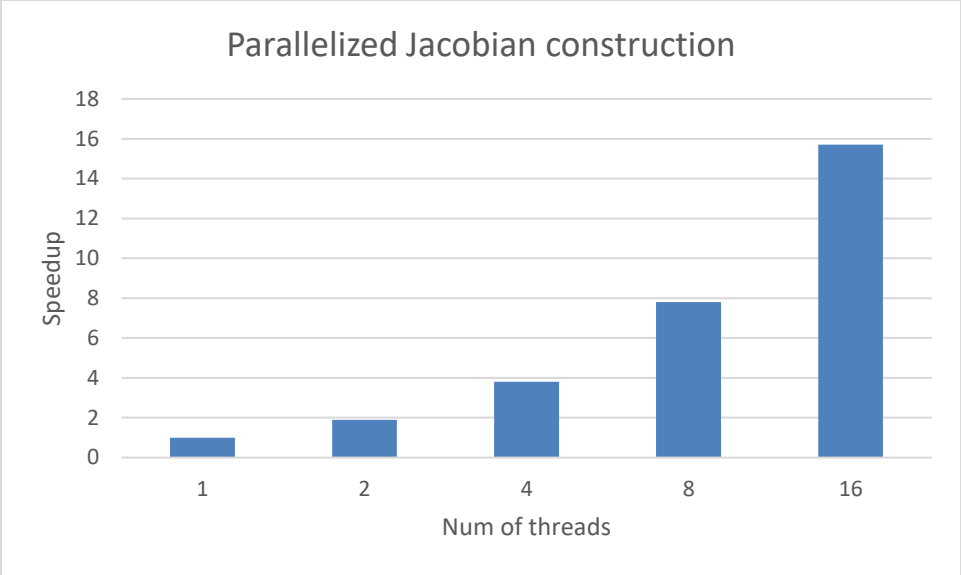

Figure 3.11 Case study of parallelized Jacobian matrix construction.

It is also worth noting that the Jacobian construction can be further optimized since the geometry of the Jacobian does not change with time in the case of static connection profiles. Thus, it is appropriate to create a hash table (Cormen et al., 2009) for the reordered BCSR Jacobian method with respect to grid indexing and avoid further access and operations on matrix storage and ordering.

3.4.3. Parallelization of Gauss-Seidel

The Gauss-Seidel (GS) is one of the most common iterative methods in matrix computations. It is used in GURU both as the smoother for global Jacobian and the pressure subspace AMG V-cycle as well as the solving method for saturation subspace. It serves as an improved method to the Jacobi iteration, with the following form of linear equations and matrix-vector operations:

$$\begin{cases} x^{(0)} = \left(x_1^{(0)}, x_2^{(0)}, \cdots, x_n^{(0)}\right)^T \text{(initial vector)} \\ x_i^{(k+1)} = \dfrac{\left(b_i - \sum_{j-1}^{i-1} A_{ij}\, x_j^{(k)} - \sum_{j=i}^{n} A_{ii} x_j^{(k+1)}\right)}{A_{ii}} \\ i = 1,2,\cdots,n; \\ k = 0,1,\cdots, \text{the iteration numbers} \end{cases} \tag{3.20}$$

$$\vec{x}^{k+1} = \vec{x}^{k} + (D + L)^{-1}\left(\vec{b} - A\vec{x}^{k}\right) \tag{3.21}$$

As in the GS process, at iteration step $k + 1$, the calculated variables $x_j^{(k+1)}, j = 1,2,\cdots, i-1$ are used to get the i-th component $x_i^{(k+1)}$. This significantly improves the convergence rate compared to Jacobi iterations, while on the other hand, also makes the GS process a conceptually serial algorithm.

Implementations are made to obtain better parallel performance for GS with respect to lowering the convergence rate. The most famous method is the black-red Gauss-Seidel (Koester et al., 1994). The structured grids are partitioned into multiple groups and marked as different colors, and the vertex set is divided into subgroups for which iterations are performed parallelly on the same color. For example, in the black-red GS of **Figure 3.12**, firstly all the red vertices are computed in parallel, and then all the black vertices are calculated in parallel with respect to the red in serial. The four-color or more colored diagrams can also be used in this manner.

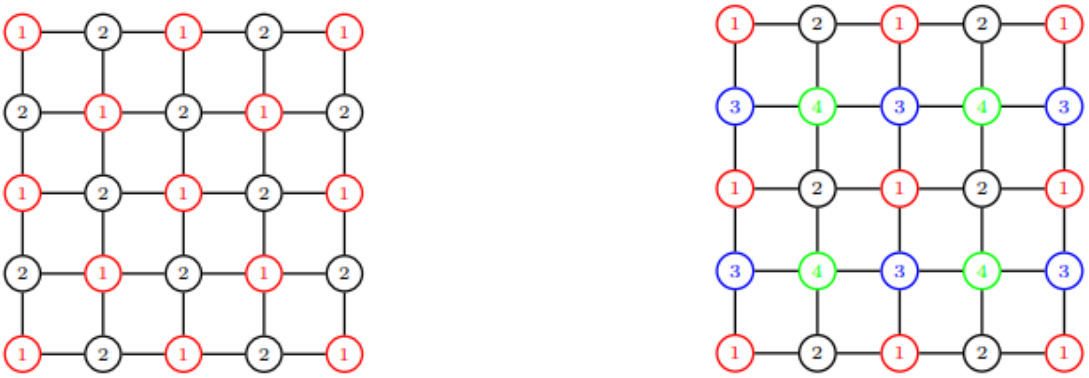

Figure 3.12 Multicolor GS method based on grid structures: black-red color (left) and four-color (right).

It is worth noting that the maximum number of multi-color groups cannot exceed the total number of non-zero entries. It is not desirable to have too many groups, as the grouping algorithms become inefficient, and the performance of the parallel part is weakened. At the most extreme condition, the multi-color GS reduces to Jacobi iteration, which is fully parallelizable but slow to converge.

Further research on the grouping method shows that the non-zeros with smaller absolute values in a row show less impact in Gauss-Seidel iterations (Vaněk et al., 1996). Thus, it is possible to build up a connection matrix based on the neighborhood coupling effect to further improve the parallel performance of multi-color GS. The connection matrix is generated as below:

$$C_{ij} = \begin{cases} w, & |a_{ij}| > \frac{1}{n}\sum_{k=1}^{n}|a_{ik}| \\ 0, & |a_{ij}| \leq \frac{1}{n}\sum_{k=1}^{n}|a_{ik}| \end{cases} \quad \forall\, i,j = 1,2,\cdots,n, i \neq j \qquad (3.22)$$

In which case the weight factor value of $w > 0$ shows that the value is strongly coupled while the value of 0 is weakly coupled. This allows further regrouping, and the parallel performance is enhanced. A case study has been performed on the multi-color GS with connection matrix regrouping (MS-GS-CR) and normal multicolor GS (MS-GS) in the test reservoir model solved by classical AMG with different thread numbers. The results are shown in **Table 3.4**.

Table 3.4 Case study of the multi-color GS method

	Threads	1	2	4	8	16
MS-GS	Newton	151	151	151	157	191
	Linear	1378	1378	1378	1723	1935
GS-CR	Newton	137	137	134	134	133
	Linear	1416	1416	1405	1402	1398

The comparison shows that the multicolor GS with connection regrouping is mostly stable with strong scalability. It is also worth noting that the regular multicolor GS performs significantly better under low thread numbers while the regrouping method has much better parallel performance. Thus, picking the appropriate method according to the hardware architecture can be important in the Gauss-Seidel iterations.

3.4.4. Parallelizing the linear solver

While the Krylov subspace linear iterative solvers are not conceptually fully parallelizable at every step, there are quite a few matrix-vector and vector-vector operations which can significantly benefit from the parallelization. This subsection gives

out an algorithm on a parallelized version of PGMRES in **Table 3.5**, which is used in GURU as the linear solver.

Table 3.5 The parallel PGMRES algorithm description

Step 1. Select the initial guess vectors x_0. Calculate the residual vector:

$$r_0 = \mathrm{b} - A_{x_0} \tag{3.23}$$

Step 2. Calculate the p_1 vector:

$$p_1 = \frac{r_0}{\|r_0\|} \tag{3.24}$$

Step 3. Solve the Arnoldi process for vector P_n and matrix $\widetilde{H}_n$. For $j = 1, 2, \cdots, n$,

$$\bar{p} = AQ\vec{p}_j \tag{3.25}$$

$$\hbar_{i,j} = (\bar{p},\ p_i), \quad i = 1, 2, \cdots j \tag{3.26}$$

$$\tilde{p}_{j+1} = \bar{p} - \sum_{i=1}^{j} \hbar_{i,j}\, p_i \tag{3.27}$$

$$\hbar_{j+1,\ j} = \left\| \tilde{p}_{j+1} \right\| \tag{3.28}$$

$$p_{j+1} = \frac{\tilde{p}_{j+1}}{\hbar_{j+1,j}} \tag{3.29}$$

Step 4. Solve the minimalization problem for y:

$$\left\| \beta e_1 - \widetilde{H}_n y \right\| = \min_{z \in R^n} \left\| \|p_1\| e_1 - \widetilde{H}_n z \right\| \tag{3.30}$$

Step 5. Solve the residual r_n:

$$x_n = x_0 + QP_n y_n \tag{3.31}$$

$$r_n = b - Ax_n \tag{3.32}$$

Table 3.5 Continued The parallel PGMRES algorithm description

Step 6. If $\|r_n\| \leq tol$, end the iteration. Else, go to Step 3 with updated x_0 and p_1.

$$x_0 = x_n \tag{3.33}$$

$$p_1 = \frac{r_n}{\|r_n\|} \tag{3.34}$$

It is worth noting that many of these equations in this PGMRES can be parallelized, including Equation 3.24, 3.25, 3.31 as matrix-vector multiplication and Equation 3.26 as a vector-vector inner product. The algorithm is coded using OpenMP and CUDA on CPU and GPU systems, and a case study has been performed to measure the effect of parallelization using the strong scalability test on a reservoir problem. The results are shown in **Table 3.6**.

Table 3.6 Case study of the parallelized PGMRES algorithm

Threads	Time	Speedup
1	2898.25	1.00
2	1937.83	1.50
4	1305.67	2.22
8	1099.91	2.63
16	571.02	5.08
GPU	234.99	12.33

An increasing trend is obtained based on the thread number. CUDA on GPU performs significantly better compared to OpenMP on CPU. The production and

pressure curves are matched on every run, proving that the parallel algorithm is stable and robust.

3.4.5. Parallelizing the multi-stage preconditioner

The CPR-style multi-stage preconditioner can also be further optimized. The Gauss-Seidel process can be parallelized using the multi-color method mentioned in the previous subsection. In the specific algorithm provided by **Table 3.3**, the AMG V-cycles in the subspace calculations are parallelized using traditional four-color GS as well as the saturation subspace. The connection reordering GS is further applied to the pressure subspace for better parallelization performance.

On the other hand, the preconditioner is further developed using a two-step adaptive setup method. The first step is used to reduce the unnecessary AMG set-up cost in the computation of subspace stage-preconditioners by reusing the calculated preconditioner from the last AMG iteration. A limitation coefficient δ is applied for classical AMG with respect to the convergence and residual of the current stage:

$$\frac{\|r^i\|}{\|r^{i-1}\|} \leq \delta \tag{3.35}$$

Whereas r^i is the residual vector of the i-th iteration. This dissertation also provides a definition of δ for automatic processing with respect to the iteration error tolerance ε as:

$$\delta = \varepsilon \frac{\|r^0\|}{\|r^i\|} \tag{3.36}$$

The second step of this adaptive method is applied to the global multi-stage preconditioner. It is observed that the Jacobian matrix within a certain timestep does not

vary much between the adjacent Newton-Raphson iterations. Thus, it can be unnecessary to regenerate the preconditioner B matrix at every Newton step. An adaptive coefficient μ is used as the control variable with respect to the number of linear solver iterations. The algorithm is described in **Table 3.7**.

Table 3.7 The adaptive multi-stage preconditioner for a NR-iteration

Step 1. Generate preconditioner Q_i.

Step 2. Solve the Newton step at i and obtain the iteration number k_i:

$$x_{i+1} = x_i + Q_i(b_i - A_i x_i) \tag{3.37}$$

Step 3. For $i = i + 1$, If $k_i < \mu$, go to step 2 with:

$$Q_{i+1} = Q_i \tag{3.38}$$

Else, go to step 1.

It is worth noting that when $\mu = 0$ is selected, this method simplifies to the general multi-stage preconditioning form. Picking up the μ value can be important for simulation performance optimization to find the balance between preconditioner set-up time cost and the linear solver acceleration. Case studies have been performed to analyze the performance of this method, shown in **Table 3.8**.

Table 3.8 Case study of the parallelized PGMRES algorithm

μ	0		50		100	
Threads	Time	Speedup	Time	Speedup	Time	Speedup
1	2898.25	1.00	2798.21	1.04	2920.01	0.99
2	1937.83	1.50	1827.21	1.59	1926.66	1.50
4	1305.67	2.22	1201.67	2.41	1395.09	2.08
8	1099.91	2.63	1067.11	2.72	1160.34	2.50
16	571.02	5.08	541.99	5.35	581.01	4.99

Unfortunately, there has been no easy way to obtain the desired μ value to obtain peak performance as the coefficient can be affected by the Jacobian structure, the problem size, the parallel thread numbers, etc. It is necessary to run a few numerical tests and hand-pick a value for performance optimization.

3.5. Case study

A few case studies have been performed on the real field reservoir data to further test the simulator's performance. Strong scalability tests have been performed on a Hansen well model with 1 horizontal wellbore, 100 fractures, and 80 thousand grid blocks. An illustration of this field model is shown in **Figure 3.13**. Tests are performed for different CPU threads as well as GPU, and the numerical simulation speedups are compared with respect to the serial case at a total of 561 seconds, shown in **Figure 3.14**.

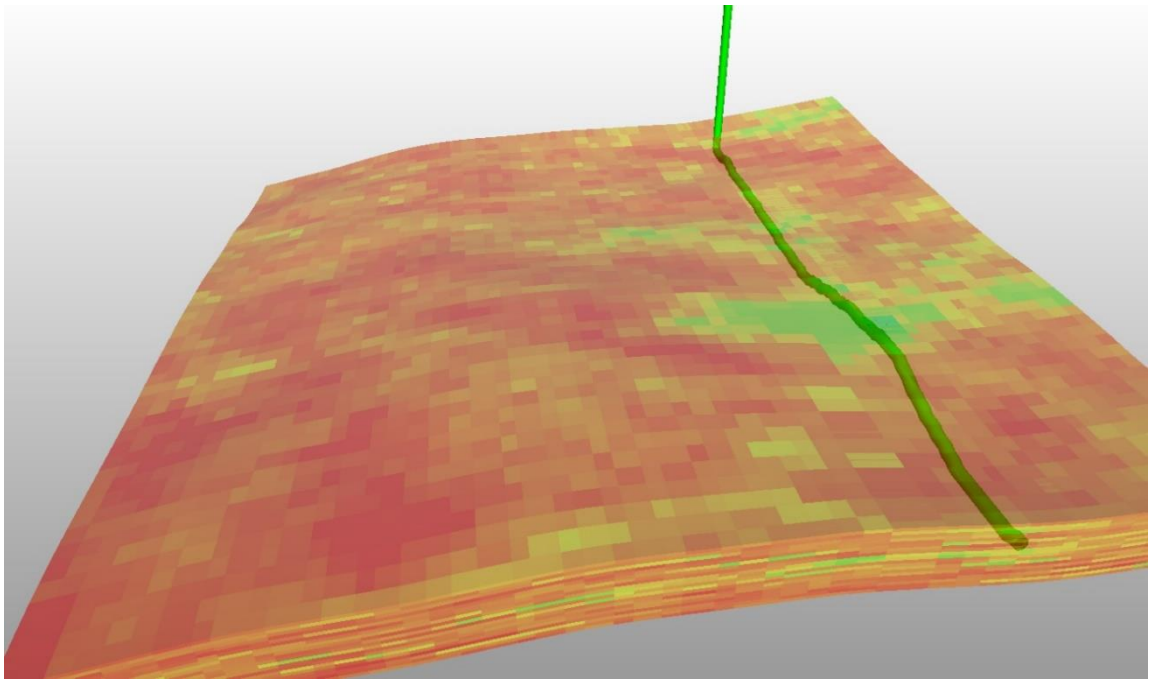

Figure 3.13 Illustration of Hansen well model.

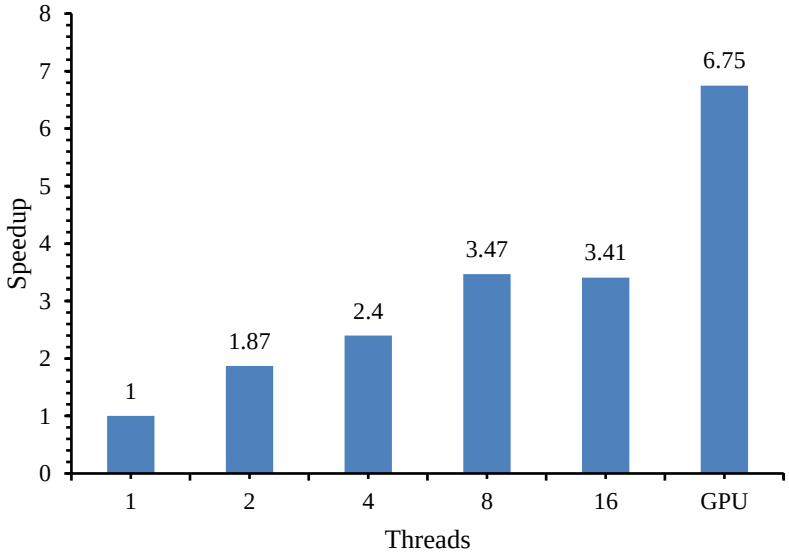

Figure 3.14 Hansen well strong scalability test.

Significant although not ideal speedups for CPU multi-process performance have been obtained on the Hansen well model case study. GPU gives out peak performance which significantly outperforms all the other CPU cases. However, the CPU case with 16 threads does not work as well as expected, which may be caused by the communication delay and setup cost in such a specific case.

Another weak scalability test has also been performed on the YOST model described in Chapter 2. The results are shown in **Figure 3.15**.

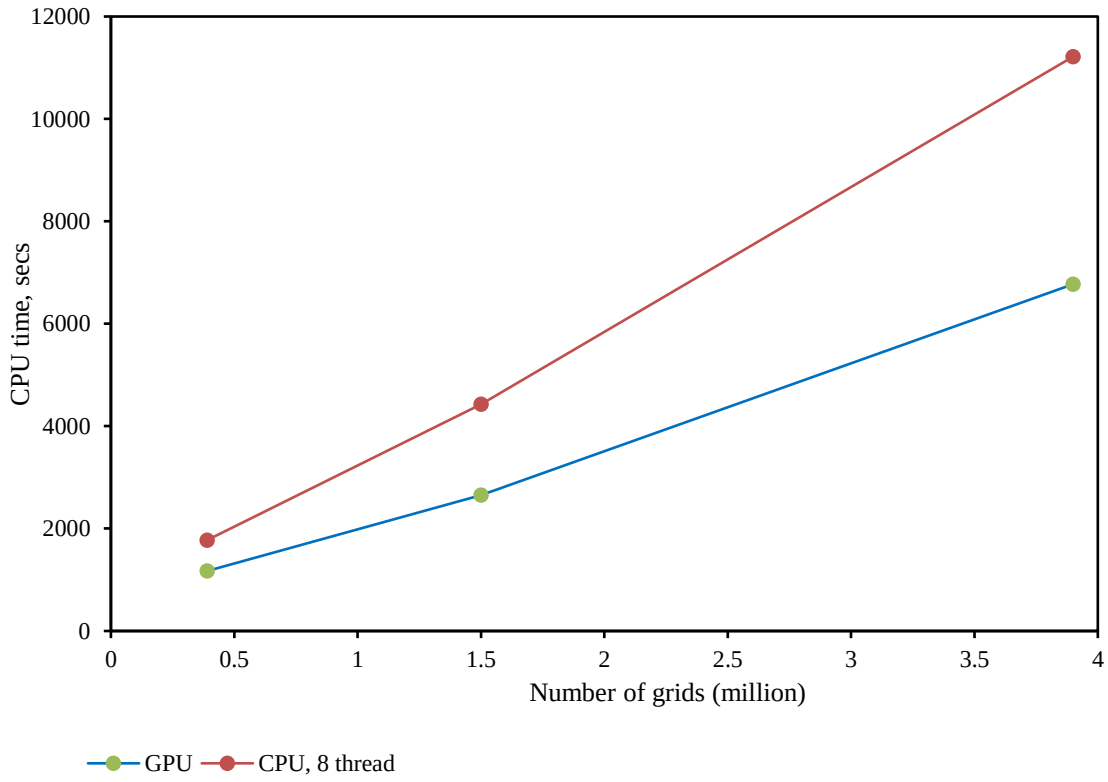

Figure 3.15 YOST model weak scalability test.

The results obtained for 3 different cases with 0.39, 1.5, and 3.9 million cells are plotted, and trends with mostly linear behavior have been obtained. This means that the

parallelization and preconditioning methods developed in this chapter are computationally stable and scalable with respect to this fractured unconventional reservoir simulator. GPU still outperforms the CPU in all cases, but it is expected to change with much smaller cases.

A few other optimization approaches based on the parallelization and domain decomposition methods are also proposed to further enhance the performance for unconventional reservoirs with more extreme fracture conditions with respect to larger fracture matrix size and complex fracture network connections. However, they will not be able to be analyzed without the prerequisite knowledge of the developed EDFM method. Thus, they are not implemented in the case studies in this chapter and will be discussed in Chapter 5.

4. FRACTURED UNCONVENTIONAL RESERVOIR MODELING

4.1. Overview

Fractured unconventional reservoirs usually have a complex fracture network consisting of natural and hydraulic fractures, which serve as the dominating flow paths in hydrocarbon production. The shape, dimensions, and connections in said fracture network have significant impacts on unconventional simulation results. Thus, it is necessary to propose a method to characterize the fracture network precisely and efficiently with respect to the complexity and uncertainty as well as the bulk reservoir.

The multi-porosity model has already been discussed in this dissertation for its capability of modeling fracture flows. However, this method is derived based on a few very specific and uniform assumptions and therefore is not suitable for the large and complex fracture networks in unconventional reservoirs. Actually, significant errors can occur while using it for fracture-dominated reservoir performances (Gale et al., 2014; Ouillon et al., 1996).

In the past decades, a few other different approaches have been proposed to better model the fractures, which fall into three different categories: local grid refinement (LGR), discrete fracture network (DFN), and embedded discrete fracture model (EDFM).

LGR fracture model is introduced into the petroleum industry firstly as an extended version of the continuous-continuum model (Azim & Abdelmoneim, 2013). This approach models the fractures by reshaping the local grids into smaller dimensions

and creating the fracture flows using permeability multipliers for equivalent flux volumes (**Figure 4.1**). This approach tends to give very fast simulation results but usually fails to preserve the nature of fracture geometry and complex flow behaviors.

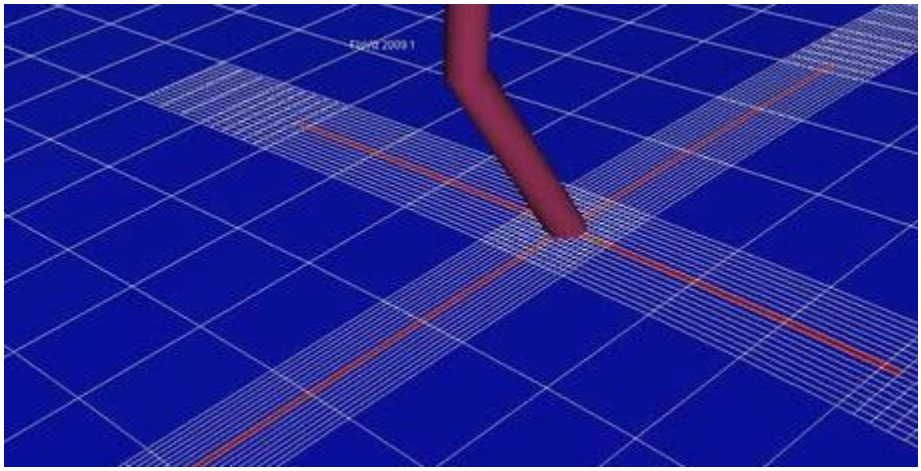

Figure 4.1 Illustration of LGR fractures.

DFN is another approach based on unstructured gridding and discretizing methods, which explicitly requires the unstructured grids to conform to the shape and size of fractures (Sun & Schechter, 2015). This forms a global grid system without the need to assume anything for fracture network geometries, which allows the precise modeling of flux inside fractures and between fracture and matrix (**Figure 4.2**). However, DFN results in an excessively large number of grid cells from the fact that grid refinement is required close to the fracture regions for convergence, which can be challenging to compute efficiently.

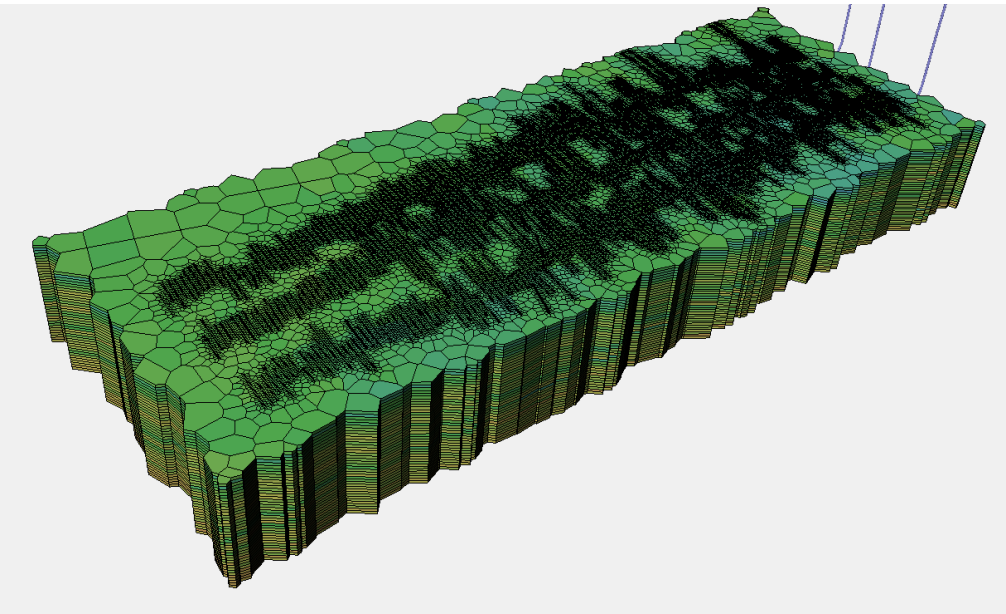

Figure 4.2 Illustration of DFN fractures.

EDFM (Lee et al., 2000) takes the accuracy advantage from DFN to models the fractures discretely. But it does not require the matrix grids to conform to fracture geometry. Instead, it embeds the fractures directly into the matrix system using none-neighbor connections (**Figure 4.3**). Much research has been done to further extend the usage of the EDFM method by implementing the EDFM model into 3-dimensional cases (Moinfar et al., 2013) and coupling the fracture grids based on grid subdivisions (Yang et al., 2018). However, convergence issues may occur when two or more fractures interact in a coarse matrix grid (Ţene et al., 2017), for which further modifications on the EDFM is needed to provide precise and efficient solutions.

Due to its capability of computational efficiency as well as the flexibility to create fracture networks, it is selected as the desired fracture modeling method in this dissertation.

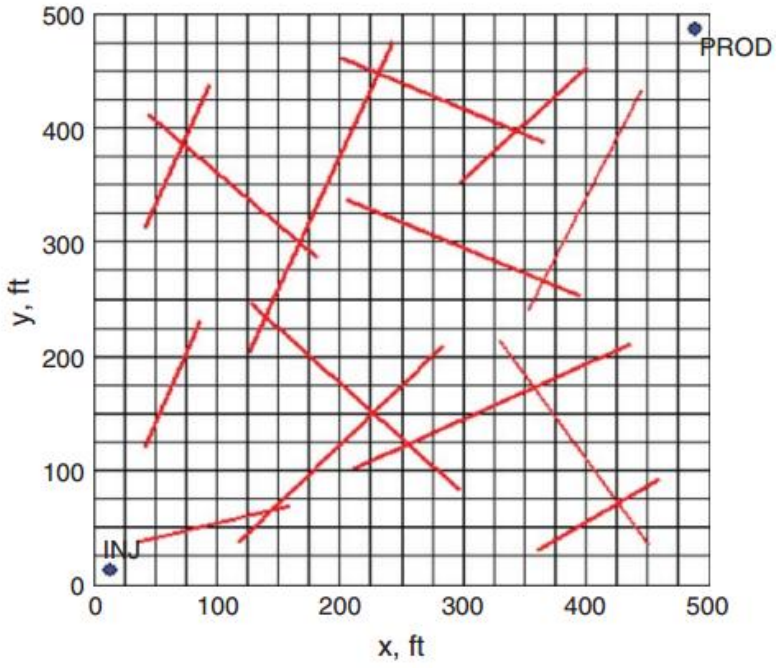

Figure 4.3 Illustration of EDFM fractures.

4.2. Matrix model

As with EDFM fracture models, matrix gridding becomes less of an issue in unconventional reservoirs as the regional flow is always dominated by the fracture network. However, the idea of orthogonal connections still comes into consideration when building up the matrix grids to improve the convergence rate of the global matrix. The grid orientation effect also has a remarkable influence on the pressure and concentration profiles dominated by fracture-matrix connections using TPFA. Thus, a point-based gridding method is used to create a PEBI-like (Helnemann et al., 1991) unstructured system for the global matrix grid.

A given shape of reservoir region can be divided into subsystems of n convex hulls. For every convex hull, grid center points are placed under a centroid Voronoi tessellation (CVT) organization (Du et al., 1999) to preserve the best grid similarity. An illustration is shown in **Figure 4.4** for the creation of uniform PEBI grids using this method.

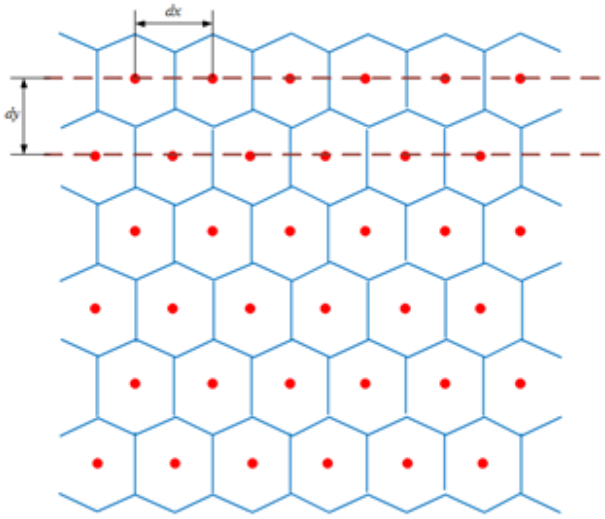

Figure 4.4 Illustration of CVT-based matrix gridding.

Reservoir boundaries are captured using another method named restricted Voronoi diagram (RVD) (Merland et al., 2011). It cuts the Voronoi cells directly with surfaces outside the central region and reconstructs the grid boundaries (**Figure 4.5**). This allows for easy implementation of complex reservoir layers and geological bodies like faults and pinch-outs.

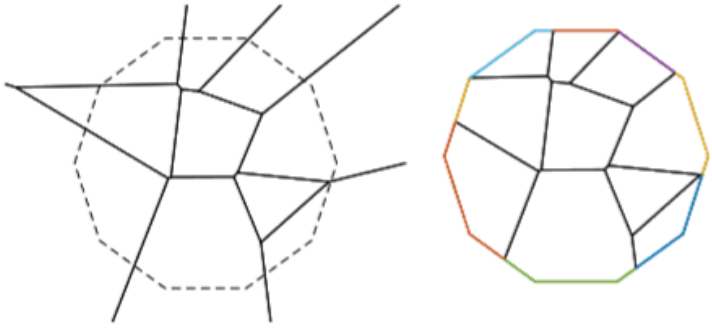

Figure 4.5 Illustration of RVD-based grid boundaries.

Case studies have been performed on a few real reservoir models. The result of a layered reservoir with multiple faults is shown in **Figure 4.6**.

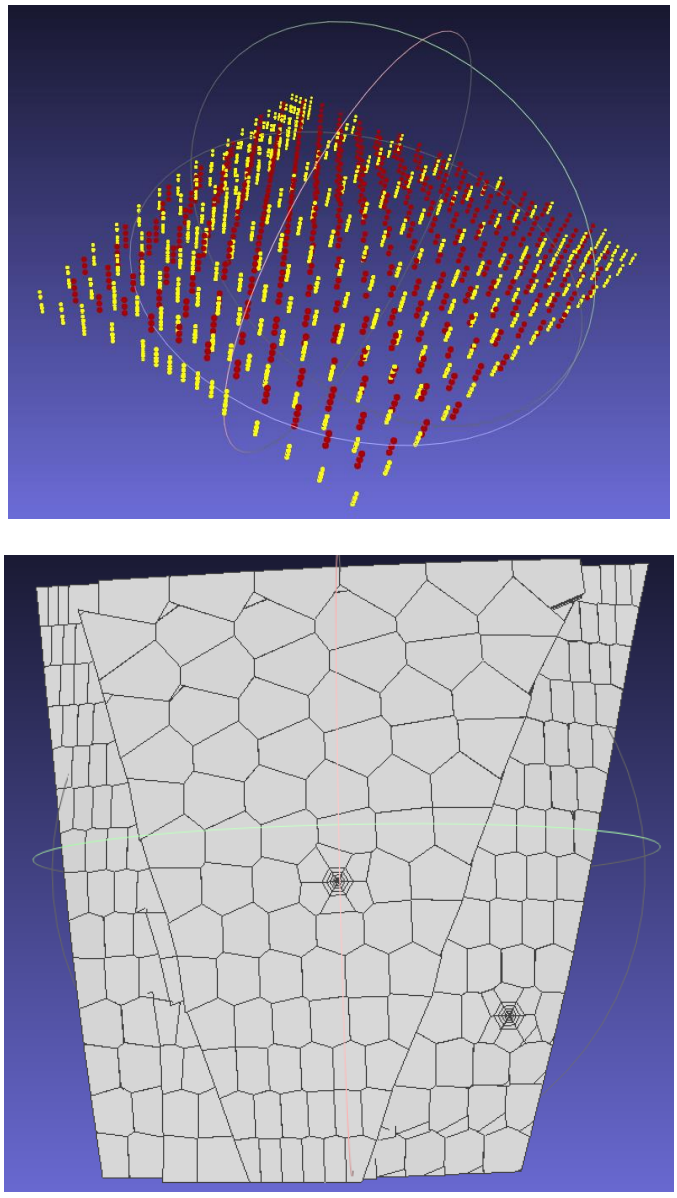

Figure 4.6 Case study of reservoir model with wells and faults using CVT point settings (top) and generated reservoir grid systems (bottom).

It is worth noting that, if the CVT points are placed in a uniform manner both row-wise and column-wise, this method degenerates into creating normal structured Cartesian grids. It can be convenient to have such ability for testing and case studies.

4.3. Fracture model

Based on the dimension of reservoir matrix grids, it is possible to split the reservoir fractures into different categories. In GURU, fractures with dimensions smaller than the matrix grid are modeled using the micro-fracture layer in the multi-porosity-permeability model, while fractures with larger dimensions are modeled explicitly using EDFM, which forms the connections into a fracture network.

The fracture model used in this dissertation is named LGR-EDFM which is based on the c-EDFM method (Chai et al., 2016) with further modifications and functional extensions into general unstructured grid systems as well as improved convergence. The transmissibility for fracture-fracture (F-F) and fracture-matrix (F-M) connections are listed in **Table 4.1**.

Table 4.1 Transmissibility formulations for c-EDFM, modified from Chai (2016).

Type	T_{nnc}	A_{nnc}	k_{nnc}	d_{nnc}
F-F	$\frac{1}{T_{f1}^{-1} + T_{f2}^{-1}}$	$l_{f1,f2}\min(w_{f1}, w_{f2})$	---	---
F-M	$\frac{A_{nnc}k_{nnc}}{d_{nnc}}$	$A_{F\text{-}M}$	$\frac{2}{k_F^{-1} + k_M^{-1}}$	$\frac{1}{V}\int d_f \, dv$

To further extend the usage of c-EDFM on fractured unstructured grid systems, a two-step LGR implementation has been performed to improve the convergence rate on traditional EDFM. The first step is to refine the connected matrix cells based on fracture surfaces (**Figure4.7**). For fractures that only partially thrust the matrix, an extension is performed so that the corresponding cell also gets split. This eliminates the hard part of

measuring the connection surfaces between the fracture plane and an arbitrary shaped unstructured matrix grid.

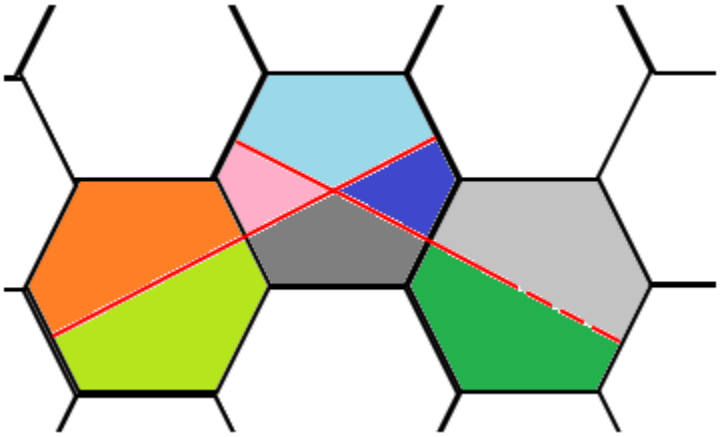

Figure 4.7 LGR-EDFM grid splitting. Notice that the bottom-right cell is partially thrust by fracture but still splits into multiple grids.

After the splitting process, the corresponding cells are then merged with the outer layer of surrounding unchanged matrix grids, which provides a space to perform the next step of local grid refinement. The LGR is processed is applied to the new grids based on a CVT-like point-based structure, creating smaller grids (**Figure 4.8**) which allow better pressure convergence between the matrix and fractures.

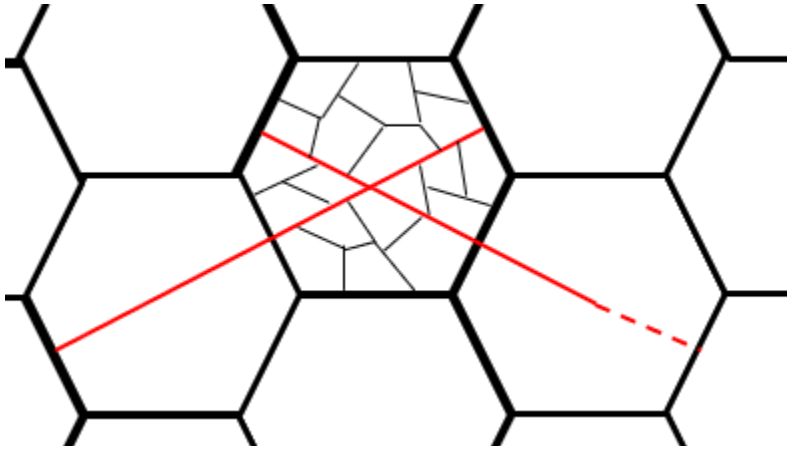

Figure 4.8 Grid LGR for EDFM. Smaller cells are created for better convergence.

A case study has been performed to test the efficiency of the LGR-EDFM method in a 200x200x3 reservoir with different numbers of intersecting fractures using the multi-stage preconditioned GURU. The results are shown in **Figure 4.9**. The LGR helps to get a better convergence rate by reducing the iteration numbers, which is much less sensitive compared to classical EDFM approaches. However, this does not always guarantee less CPU time as the enlarged grid numbers and Jacobian generally require more computations and thus may outperform the good of lessened iteration numbers in small cases.

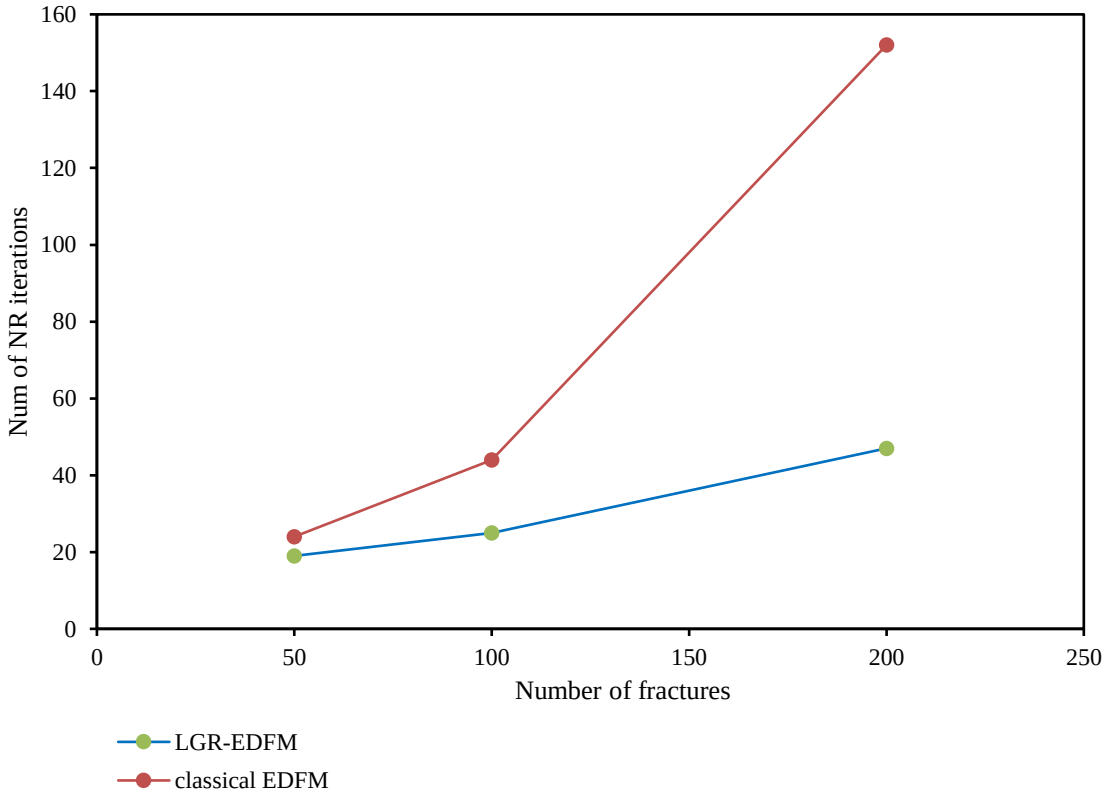

Figure 4.9 Grid LGR for EDFM. Smaller cells are created for better flux convergence.

4.4. Fracture network characterization

To better simulate the unconventional reservoir with complex fracture conditions, it is important to get an accurate characterization of the fracture network., which consists of both hydraulic fractures and natural fractures.

Various research has been performed on hydraulic fracture characterizations. The state-of-the-art approaches in the field practices build up hydraulic fractures by injecting planar surfaces according to micro-seismic data (Le Calvez et al., 2007; Maxwell et al., 2002; Warpinski et al., 1998) and well log information (Molenaar et al., 2012;

Sookprasong et al., 2014). On the other hand, research on the natural fracture part is relatively rare. The randomized fracture plot (L. ZHANG et al., 2019) has been used to describe the natural fracture distribution by generating randomized distributions based on fracture intensities using a Poisson process and modeling them together with bulk rock using DFN. As the fracture network is generated using randomized probabilistic methods, it is usually hard to preserve geological knowledge, and that are geologically unrealistic distributions can occur frequently. Due to the large uncertainty in fracture distributions, extensive usage of history matching has always been required to create models for both hydraulic and natural fractures, and the time cost can be extremely large to obtain a usable model.

In this dissertation, a novel numerical approach for characterizing the fracture network using a hybrid fracture model has been proposed. Hydraulic fractures are modeled using imaging methods from micro-seismic location point clouds, while the natural fractures are characterized using a fractal model. Calibration methods have been proposed to further reduce the uncertainty of generated fracture networks.

4.4.1. Hydraulic fracture characterization

The current hydraulic fracture characterization methods make full use of the micro-seismic data from the unconventional well fracking. The datapoint clouds are interpreted into a fracture network model by evaluating the distribution of micro-seismic events. A typical workflow is approached by using the micro-seismic event width to length aspect ratio (WLAR) to fit either planer surfaces or fracture networks by rotating (Shakiba & Sepehrnoori, 2015), shown in **Figure 4.10**. The green line here represents

horizontal wellbore, the blue represents a planar hydraulic fracture of the main fracture with low WLAR from micro-seismic clouds, while the red represents a fracture network model including main and second-level fractures with high WLAR from micro-seismic clouds.

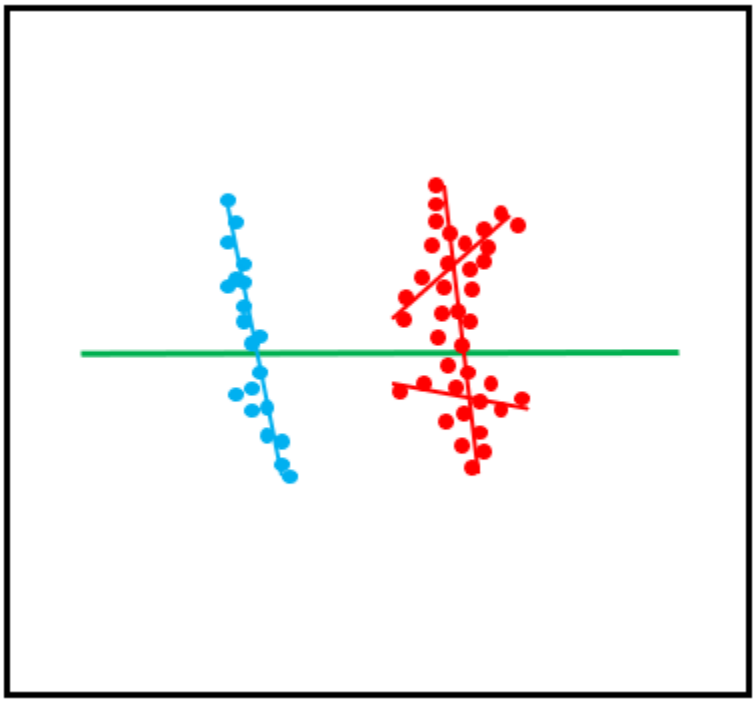

Figure 4.10 2-D Illustration of different hydraulic fracture models from micro-seismic clouds.

However, these characterization methods may fail to capture the complex nature of hydraulic fracture networks. On the other hand, they usually require high-quality data from multi-well array systems for accuracy, whereas single-well observations have shortcomings in the determination of accurate fracture geometries and locations (Seibel et al., 2010). To improve the situation, in this dissertation, an imaging-based data-driven hydraulic fracture characterization method is developed with automatic clustering, which

provides a more generalized model with the capability to model multi-well as well as single well systems with respect to multistage hydraulic fracturing micro-seismic data clouds.

There have been multiple imaging algorithms to build up planes from datapoint clouds. However, they cannot be directly imported into hydraulic fracture characterization. As the algorithms only select data based on statistics, they may create fracture planes with geologically unrealistic connections. Thus, a pre-build clustering stage becomes necessary to limit the imaging algorithms within specific regions. This dissertation uses an iteration clustering method based on the fracture spacing and fracking stage information to better construct the clustered regions for datapoints.

To perform the fracture network characterization, a filtering stage with a maximum fracture length is firstly applied to rule out the noise data. Then a guided initial guess of clustering is set up based on the fracture spacing information. K-mean clustering is performed for every staged data as an initial evaluation, and then a bi-wing iteration is performed with different cluster numbers to estimate a satisfactory clustering by adaptively combining and dividing the micro-seismic data from the neighboring stages to find a specific cluster number with minimized evaluating residual.

A more detailed workflow is listed below:

1) Plot the well and micro-seismic cloud events in 3-D space.
2) Perform filtering with respect to a given maximum allowed fracture length.
3) Along the wellbore, measure the start and endpoint of hydraulic fracking by projecting the micro-seismic data onto the wellbore line along the direction of

the primary strike angle and measure the outermost projecting locations to obtain the effective length L.

4) Set up an initial guess set of clustering based on the facture spacing information. The total number of clusters n_0 equals the effective length L divided by the fracture spacing d_f:

$$n_0 = \frac{L}{d_f} \tag{4.1}$$

The initial clustering centers are selected along the wellbore, at the perforation locations.

5) Perform an initial K-mean clustering (Lloyd, 1982) for every micro-seismic event datapoint in every staged fracking process. The evaluation function is shown below:

$$J_0 = \sum_{n=1}^{N} \sum_{k=1}^{n_0} r_{nk} \|x_n - \mu_k\|^2 \tag{4.2}$$

6) Perform Step 4) and 5) for a bi-wing iteration for adaptive selecting of a better cluster number, for which $n_1 = n_0 + 1$ and $n_{-1} = n_0 - 1$. Calculate J_1 and J_{-1}.

7) The stop evaluation criteria are set as:

$$J_0 \leq J_1 \; and \; J_0 \leq J_{-1} \tag{4.3}$$

If the stop criteria are met, satisfactory clustered results are obtained. Else, set n_0 to either n_{-1} or n_1 in response to the smaller J_{-1} and J_1 values, and run the new bi-wing iteration until the stop criteria are met.

8) For every clustered dataset and its neighbors, perform a RANSAC (Bolles & Fischler, 1981) and DBSCAN (Yu et al., 2021) to generate the planar fracture networks.

A case study has been performed using this workflow for a field Hansen well, and the processes and results are shown in the **Figure 4.11**, **Figure 4.12,** and **Figure 4.13**.

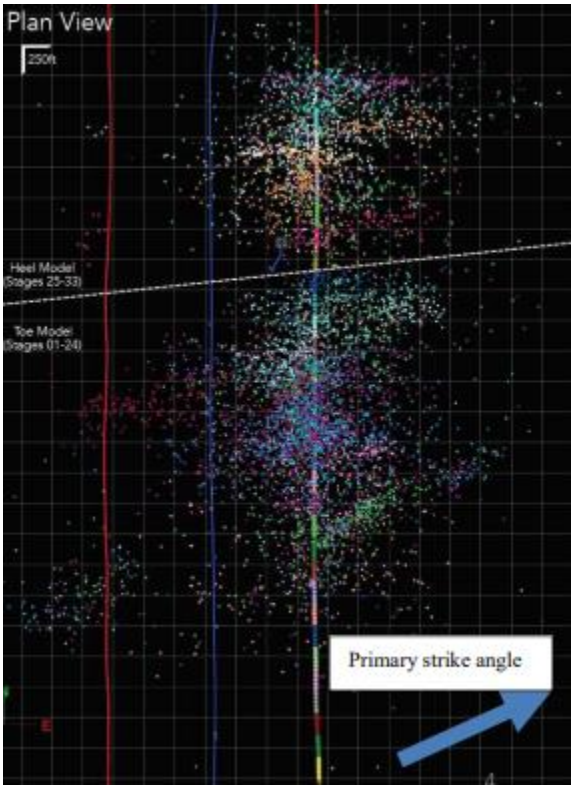

Figure 4.11 Planar illustration of a single Hansen well micro-seismic cloud information.

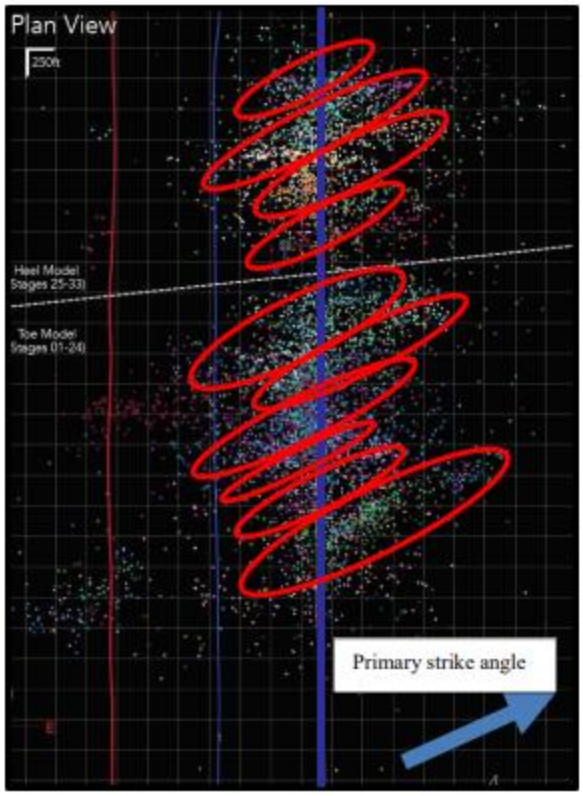

Figure 4.12 Planar illustration of a single Hansen well micro-seismic cloud information with adaptive clustering (red). Ten clusters are found with the algorithm.

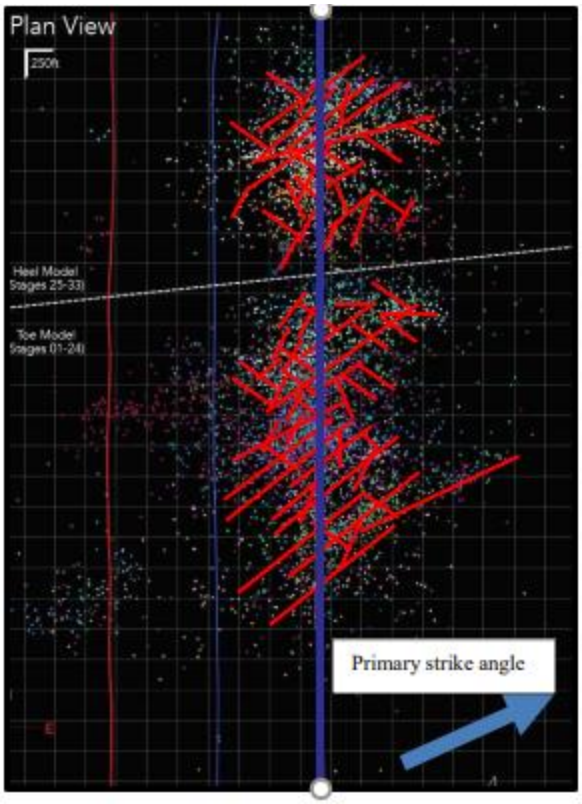

Figure 4.13 Planar illustration of a single Hansen well micro-seismic cloud information with RANSAC hydraulic fracture network (red).

Simulation has been performed to test the efficiency and accuracy of this method. A comparison of well BHP is shown in **Figure 4.14**. A decent bottom-hole pressure curve is obtained without the requirement of excessive creating of fracture models and history matchings, which is desired in unconventional simulation with hydraulic fractures.

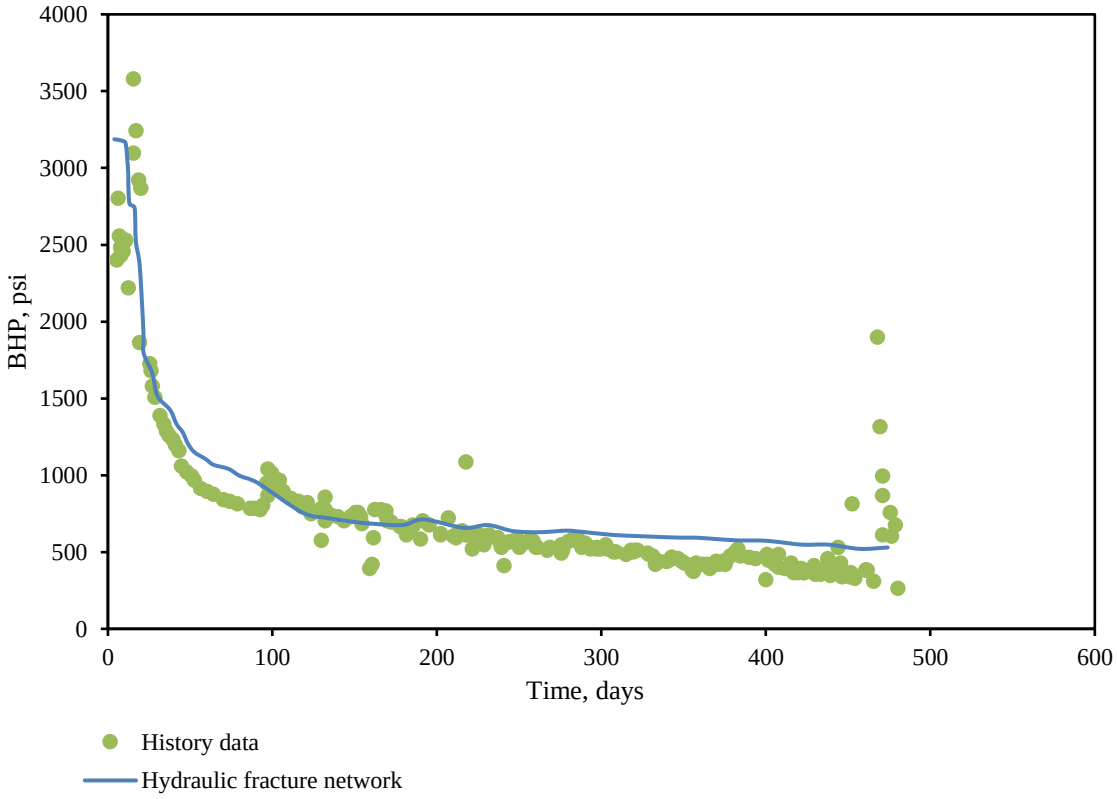

Figure 4.14 Simulation results with the hydraulic fracture network.

4.4.2. Natural fracture characterization

Studies have shown that the natural fractures in unconventional reservoirs contain highly self-similarity. Thus, it is appropriate and efficient to characterize the natural fracture distribution in unconventional reservoirs using fractal geometry theory. In the unconventional reservoir system, the fracture number and length are observed to follow the fractal scaling law (Hu et al., 2020; Miao et al., 2015):

$$N_{frac} = (L_{max}/L_{min})^{D_{frac}} \tag{4.4}$$

Thus, the probability density function can be derived in the form shown below:

$$f(L_{frac}) = D_{frac} \cdot {L_{min}}^{D_{frac}} \cdot {L_{frac}}^{-D_{frac}-1} \tag{4.5}$$

Whereas D_{frac} is the fractal dimension of natural fracture.

The fracture intensity can then be calculated as:

$$d_{frac} = \frac{\left(2 - D_{frac}\right) \cdot \emptyset \cdot \left[1 - \emptyset^{\frac{1-D_{frac}}{2-D_{frac}}}\right]}{\alpha \cdot \left(1 - D_{frac}\right) \cdot (1 - \emptyset) \cdot L_{max}} \tag{4.6}$$

Whereas the $\emptyset$ is fracture porosity and α is a fracture proportionality coefficient. The correlated fracture length in the global system can be then obtained using the following equation:

$$L_{frac} = \frac{L_{min}}{\left(1 - P(rand[0{\sim}1])\right)^{D_{frac}}} \tag{4.7}$$

While P is the cumulative probability function and $rand[0{\sim}1]$ stands for a randomized seed between 0 and 1, which is used to generate probabilistic natural fracture length.

The orientation of fractures is assumed to form a normal distribution along the mean direction (Balberg et al., 1991). Thus, all fractures are assigned with randomized deviation from the locally measured fracture orientation distribution method (Priest, 1993) for fractures.

The fracture centers in the system are generated using randomized distributions. Once the centers are generated, the 2-D probabilistic map of natural fracture distribution can then be established, and natural fracture properties can be calculated based on fracture length and fractal dimensions. A sample of a natural fracture system is shown in **Figure 4.15**.

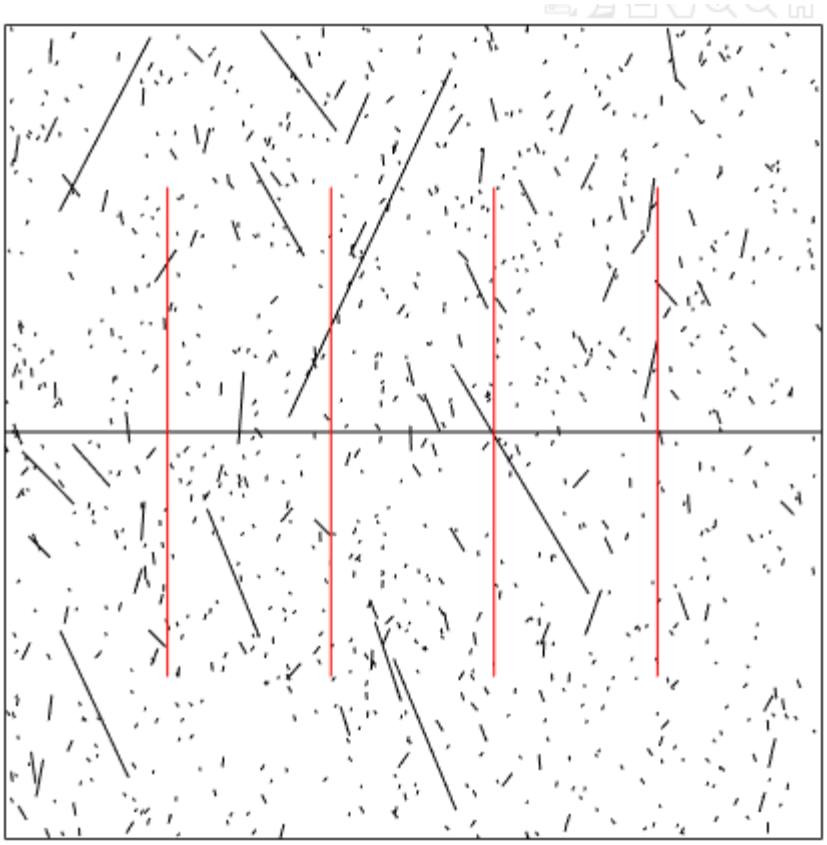

Figure 4.15 Illustration of a generated natural fracture network.

However, this fractal map still contains a large amount of uncertainty (Niven & Deutsch, 2012). To further develop the fracture network based on known geostatistical information, an empirical risk minimization method is performed to further calibrate the model and reduce the uncertainty.

In this calibration process, one set of fracture distribution realization is generated as the initial guess, and the loss function is developed as shown below with respect to fracture spacing, fracture density, and fracture length for every individual fracture in the whole dataset:

$$L_i = W_{spac} \sum_i (S - S_i)^2 + W_{den} \sum_i (D - D_i)^2 + W_{len} \sum_i (L - L_i)^2 \quad (4.8)$$

Whereas the S, D, and L are the histogram data of local fractures.

The empirical risk for the dataset is obtained as:

$$R = \frac{1}{N} \sum_{i=1}^{N} L_i \quad (4.9)$$

An iteration phase is started by trying to replace a current existing fracture with another newly generated one and thus obtaining a smaller global empirical risk. The specific current existing fracture is picked out using a loop for all fractures and the one with the largest risk is removed from the dataset. This process gives out a nicely matched fracture realization with respect to the given histograms of the fracture properties.

Sensitivity analysis has been performed for different properties of the fractal natural fracture model. Multiple realizations of fracture maps have been generated using different fractal dimensions, for which the maps are shown in **Figure 4.16**. Simulation results are shown in **Figure 4.17**.

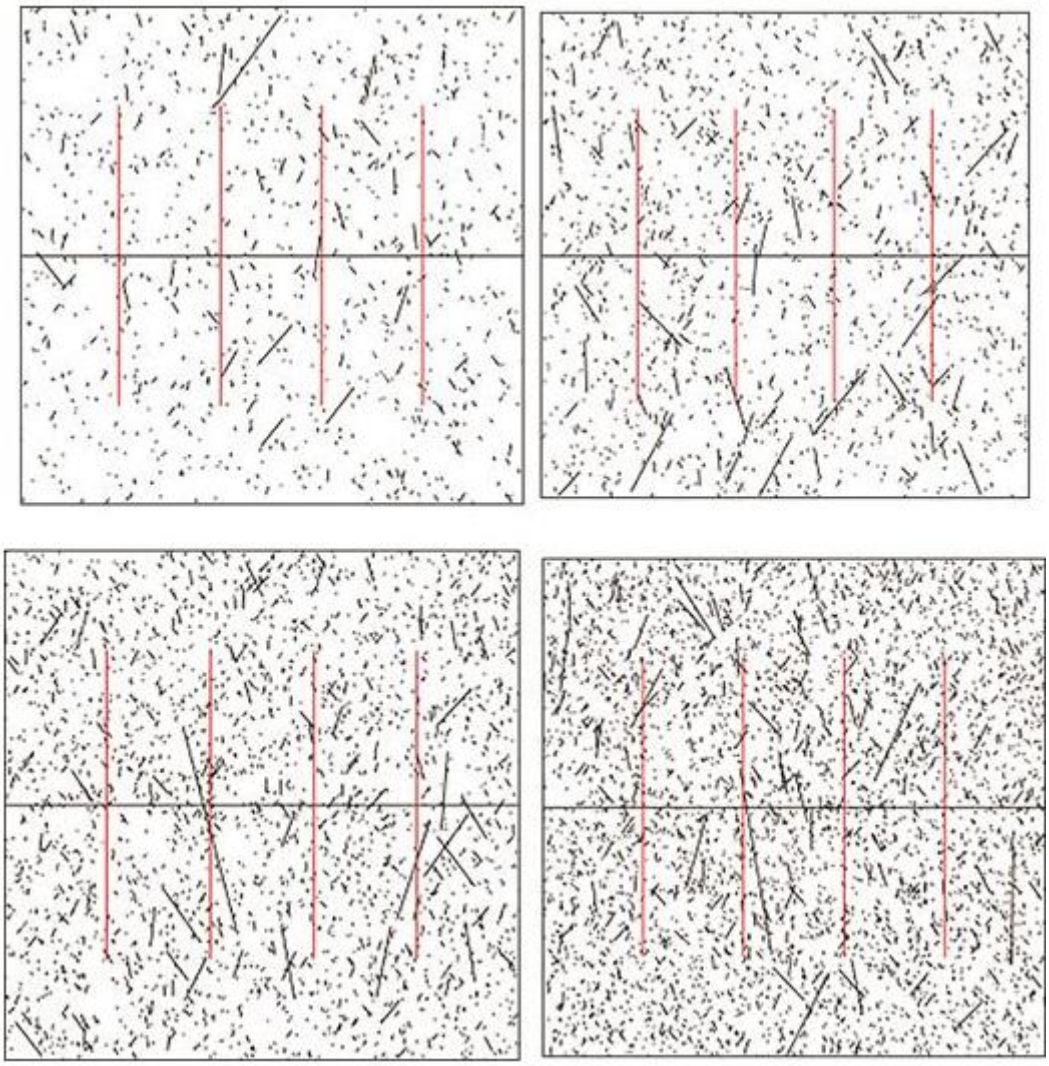

Figure 4.16 Numerical simulation case on natural fracture fractal dimensions. Lmin=0.2, Lmax=20, Dt=1.5, 1.6, 1.7, and 1.8, respectively.

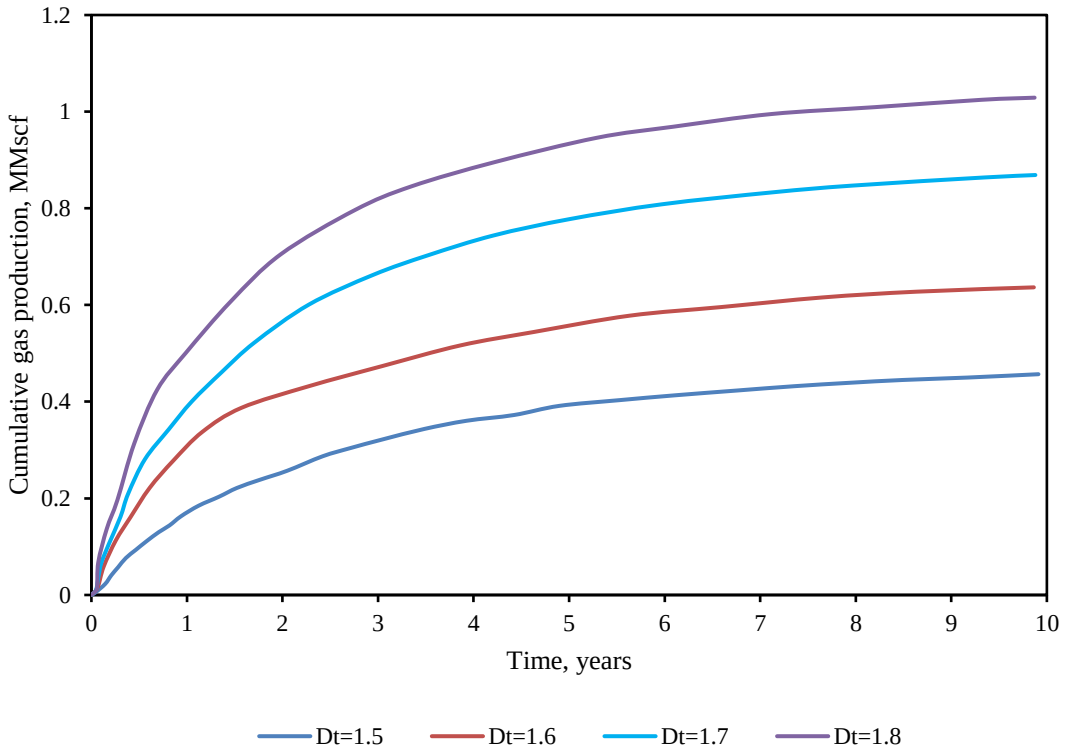

Figure 4.17 Numerical simulation results on natural fracture fractal dimension sensitivity analysis.

As seen from the results, the fracture fractal dimension has a significant impact on total reservoir production. The higher the fractal dimension is, the larger the production can be, and the relationship approximately follows a near-linear trend. This is because the fractal dimension is directly tied to the total number of fractures, which dominates the overall performance of the shale reservoir.

Another sensitivity analysis is performed on the fracture length. The maps are shown in **Figure 4.18**. Simulation results are shown in **Figure 4.19**.

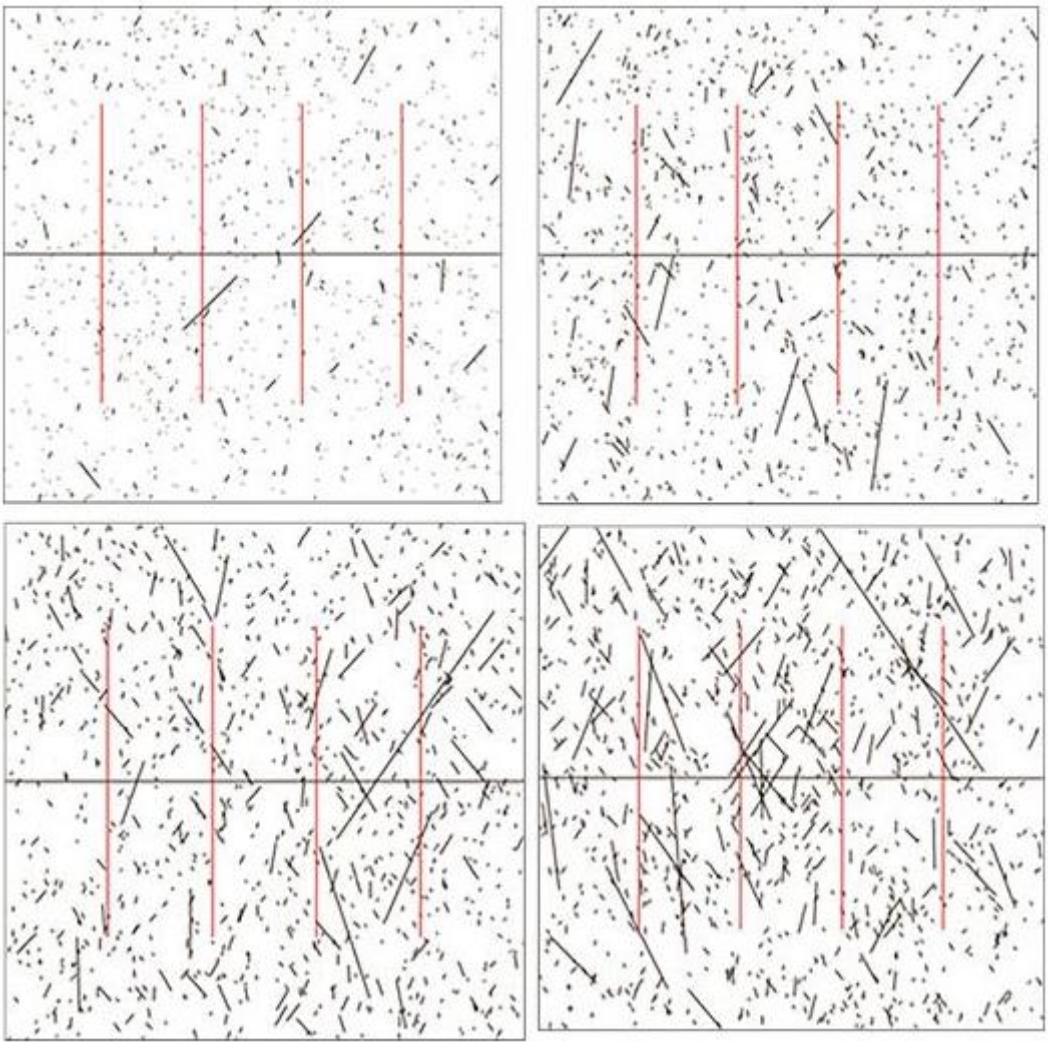

Figure 4.18 Numerical simulation case on natural fracture length. Dt=1.5, Lmin/Lmax = [0.1/10], [0.2/20], [0.3/30], [0.4/40], respectively.

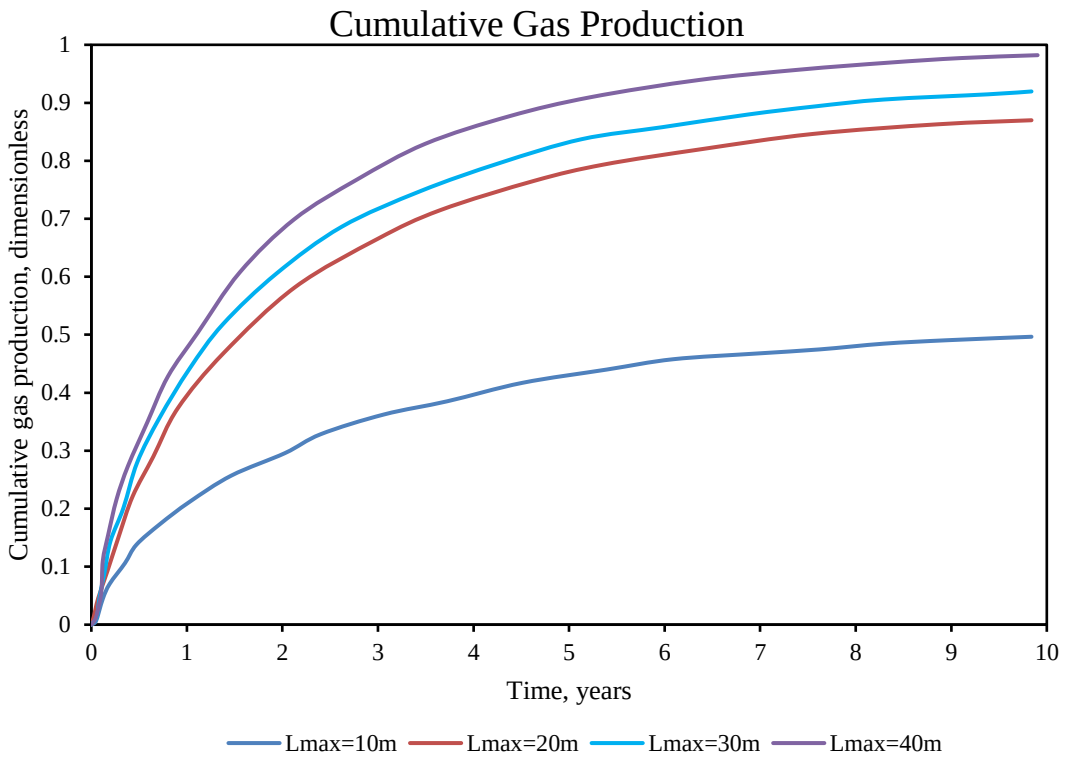

Figure 4.19 Numerical simulation results on natural fracture length sensitivity analysis.

The simulation results show that the fracture length also has a positive impact on global production as expected. However, the relationship between production and fracture length is not linear. When the maximum fracture length setting is increased from 10 to 20 meters, we see a huge jump in total gas production. This results from the fact that fractures with too small lengths cannot form efficient networks with natural fractures, and its drainage area is limited.

4.5. Case study

A single well case from the Permian Basin is demonstrated here to further analyze the performance of our fractured reservoir model. Multistage hydraulic

fracturing has been performed on this well, and the hydraulic fracture network, in this case, is shown in **Figure 4.20**. The generated natural fracture network is shown in **Figure 4.21**.

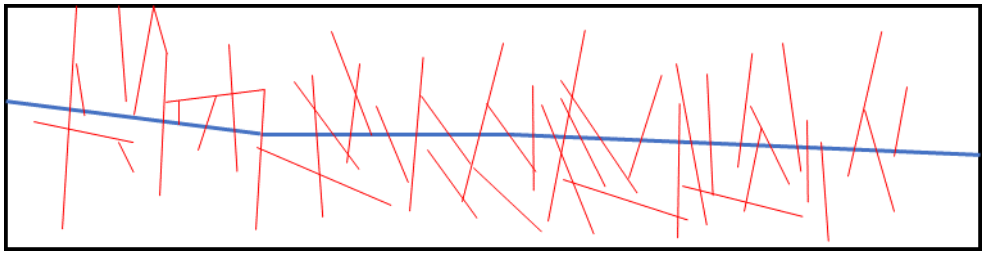

Figure 4.20 Sketch of the hydraulic fracture distribution along this horizontal wellbore in the Permian Basin. The blue line represents the horizontal wellbore, and the purple lines represent the hydraulic fractures.

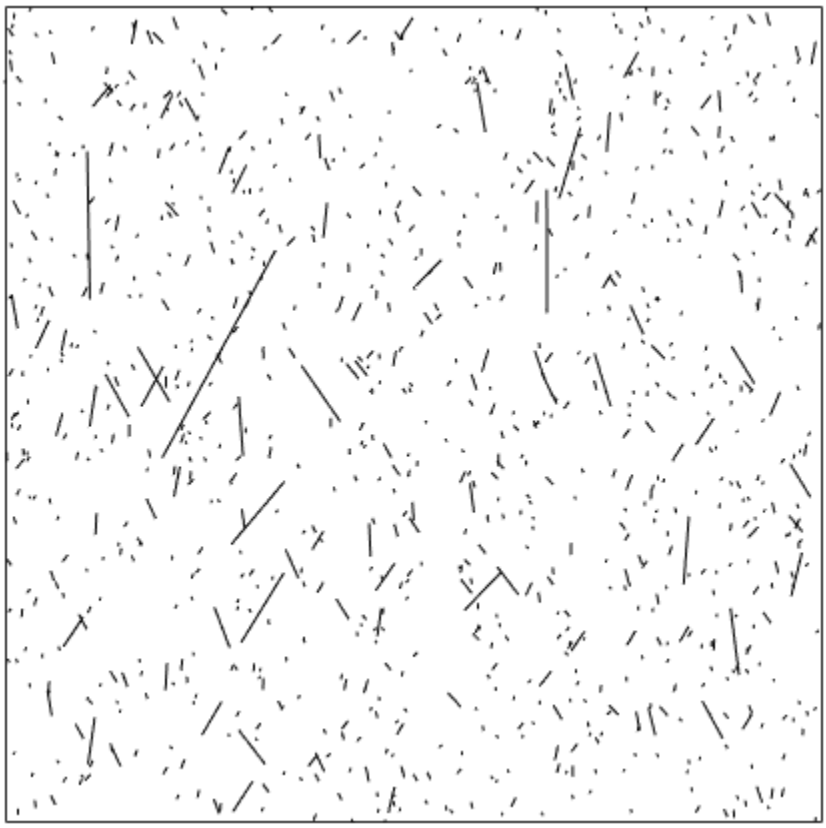

Figure 4.21 The generated natural fracture network for Permian well study.

Simulation with GURU is performed, and the results are compared with a few other commonly used fracture network modeling methods. The results are shown in **Figure 4.22**.

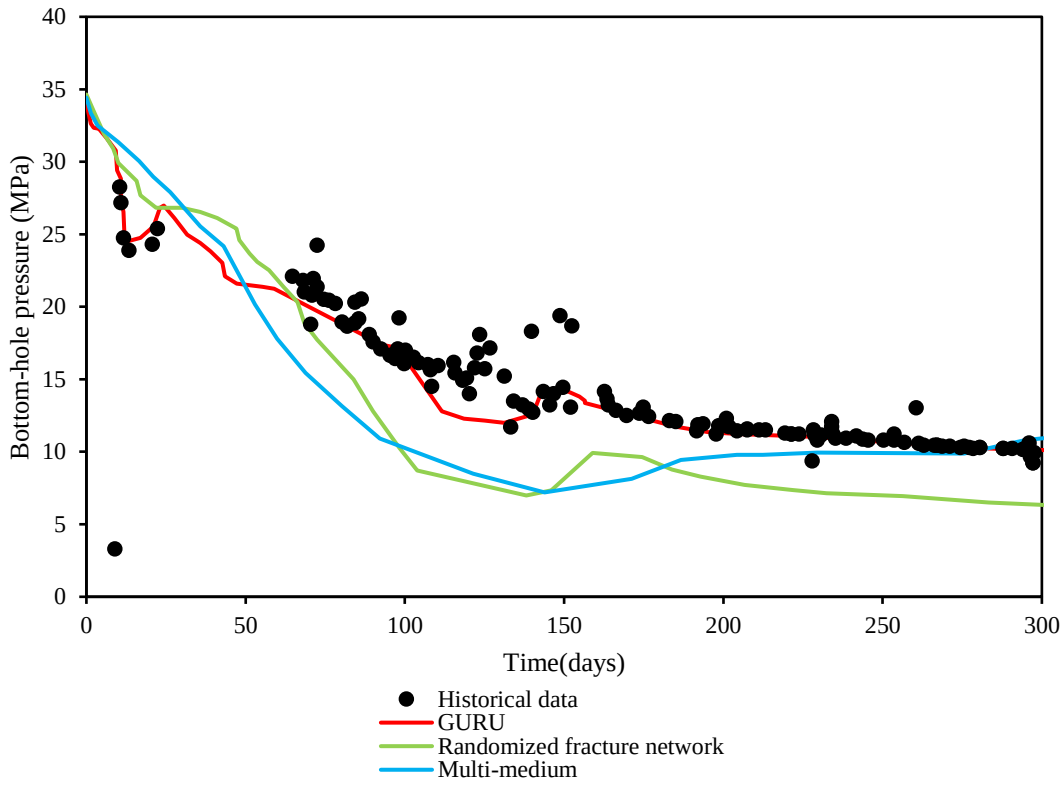

Figure 4.22 Comparison of bottom-hole pressure curves for the Permian well case study.

From the simulation results, the GURU approach developed in this dissertation gives a better estimation of the bottom-hole pressure with respect to historical data. The randomized natural fracture model shows somewhat similar results in the early stage of simulation, but the late stages failed to match with field measurements. The dual permeability model shows the worst agreement with historical production data and even differed in trend, inferring that the multi-medium method does not have enough

capability to handle models with complex fracture conditions in unconventional reservoirs.

5. FRACTURED RESERVOIR SIMULATION TECHNIQUES[1]

5.1. Pre-stage initialization

The initialization methods can be important in unconventional reservoirs with fractures, especially when hydraulic fracturing is performed. Water is injected into the wellbore and enters the formation during fracking, which causes the redistribution of pressure and saturation profiles in the fractured regions and leads to noticeable differences during production. Current methods either use a global setting with equilibrium initialization or just derive a multiplier for the stimulated regions, which may work in conventional cases but fail to preserve enough accuracy in fractured unconventional reservoirs.

A pre-stage initialization is proposed in this section based on the fracture pumping schedule by simulating the fracking fluids as a water injection process and obtaining non-equilibrium initialized pressure and saturation profiles. A numerical test is performed on a conceptual model, which is developed with four hydraulic fractures and a uniform natural fracture spacing of 30 ft. The grid map is shown in **Figure 5.1**.

---

1 Part of this chapter is reprinted with permission from "Jiang, Y., Killough, J., Li, L., Cui, X., & Tang, J. (2021). EDFM-based Multi-Continuum Shale Gas Simulation with Low Velocity Non-Darcy Water Flow Effect. SPE Reservoir Simulation Conference," Copyright [2021] by Society of Petroleum Engineers.

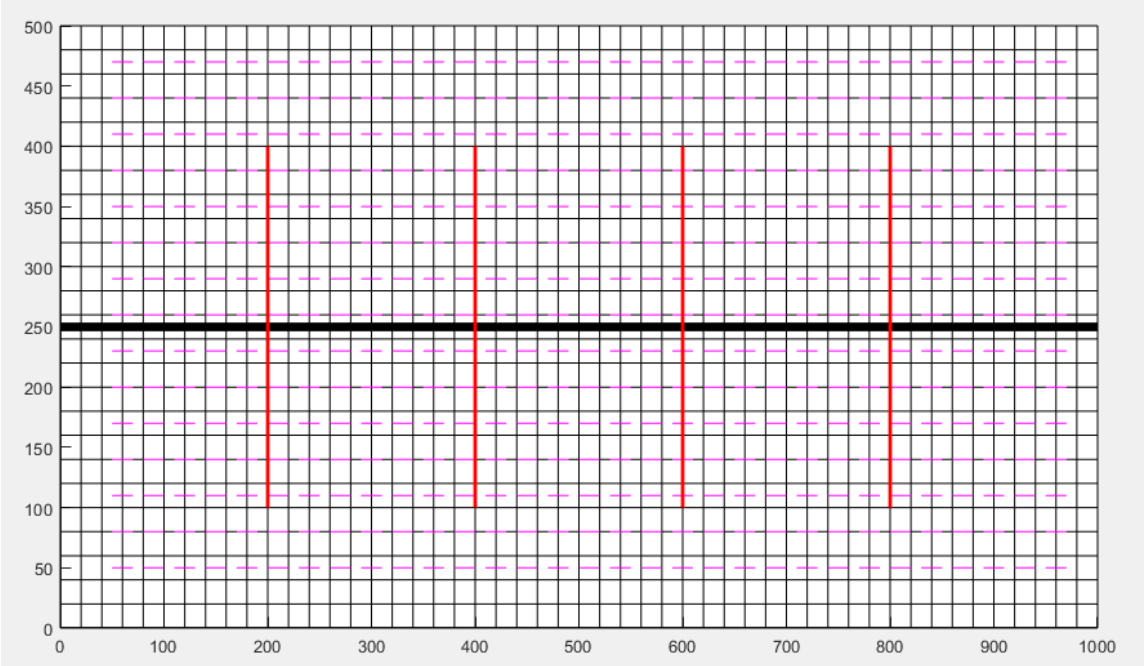

Figure 5.1 Grid and fracture distribution for the test case.

Comparisons are made for the water saturation before and after the water injecting phase. The results are shown in **Figure 5.2**. The water saturation near hydraulic fractures changes from 50% to about 82% after this pre-simulation stage, and saturation distribution in grids along hydraulic fractures has varied shapes and is not uniform along different sides. This is reasoned from the connection between hydraulic fractures and natural fractures, which cannot be preserved directly using multipliers.

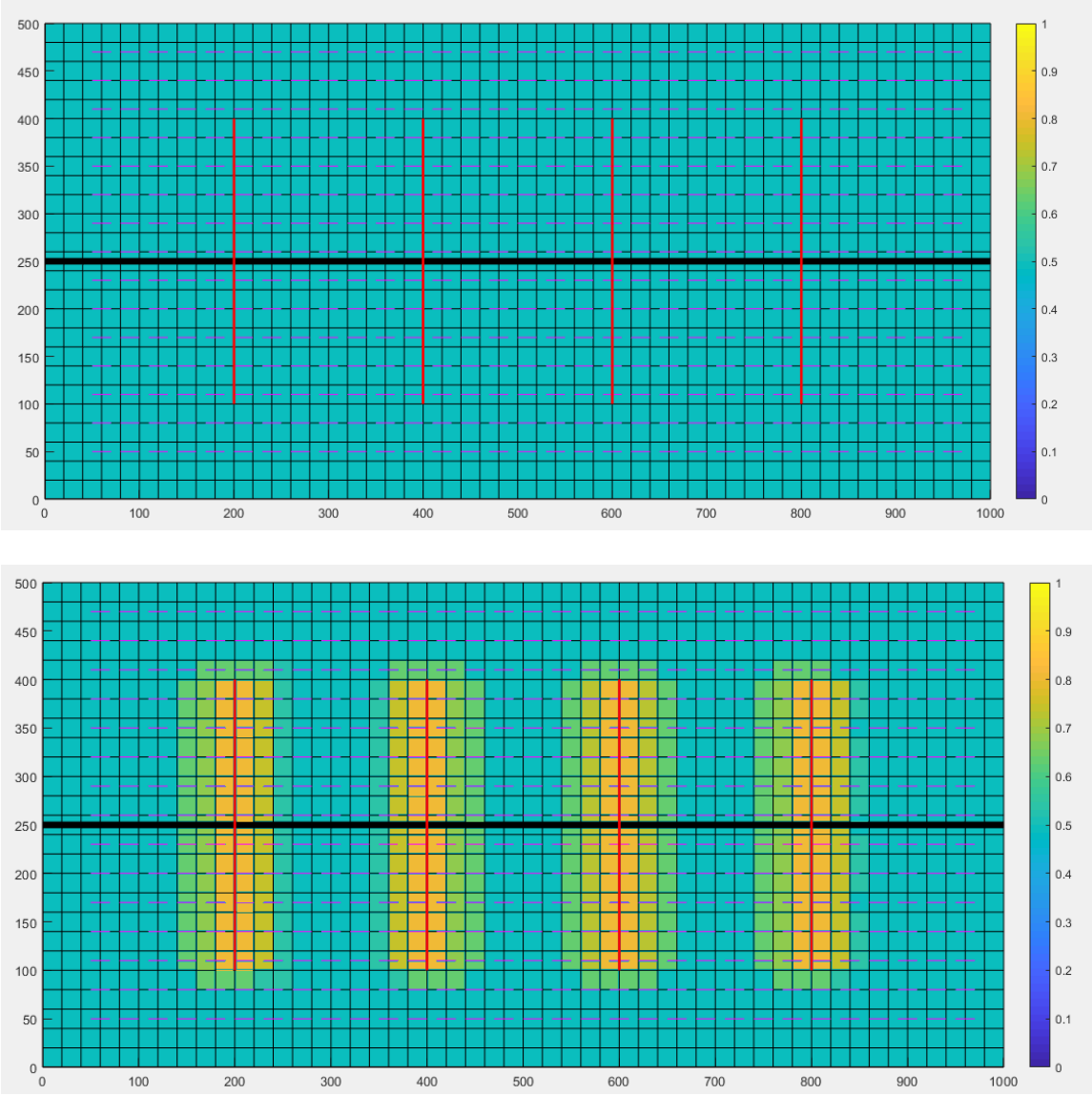

Figure 5.2 Water saturation profile before (top) and after (bottom) pre-stage water injection process.

Studies have also been made to evaluate the effectiveness of this pre-stage initialization. Different cases are run for the initialization methods and results are compared in **Figure 5.3**. The blue curve represents simulation which is initialized using non-equilibrium pressure and saturation profiles from pre-stage simulation. The red

curve represents a simulation case initialized using multipliers for pressure and saturation inside SRV regions. The yellow curve represents a model initialized with only the initial pressure and saturation conditions without any modification. The results of the blue and red curves have reached somewhat a decent agreement while the multiplier initialization method turns out to have higher production in early-stage and lower production later. This is because the SRV evaluation methods it uses do not handle natural fracture distributions accurately enough. On the other hand, the yellow curve ignores fracking fluid imbibition, which results in a much larger difference in production profiles and shall not be considered as a valid initialization method.

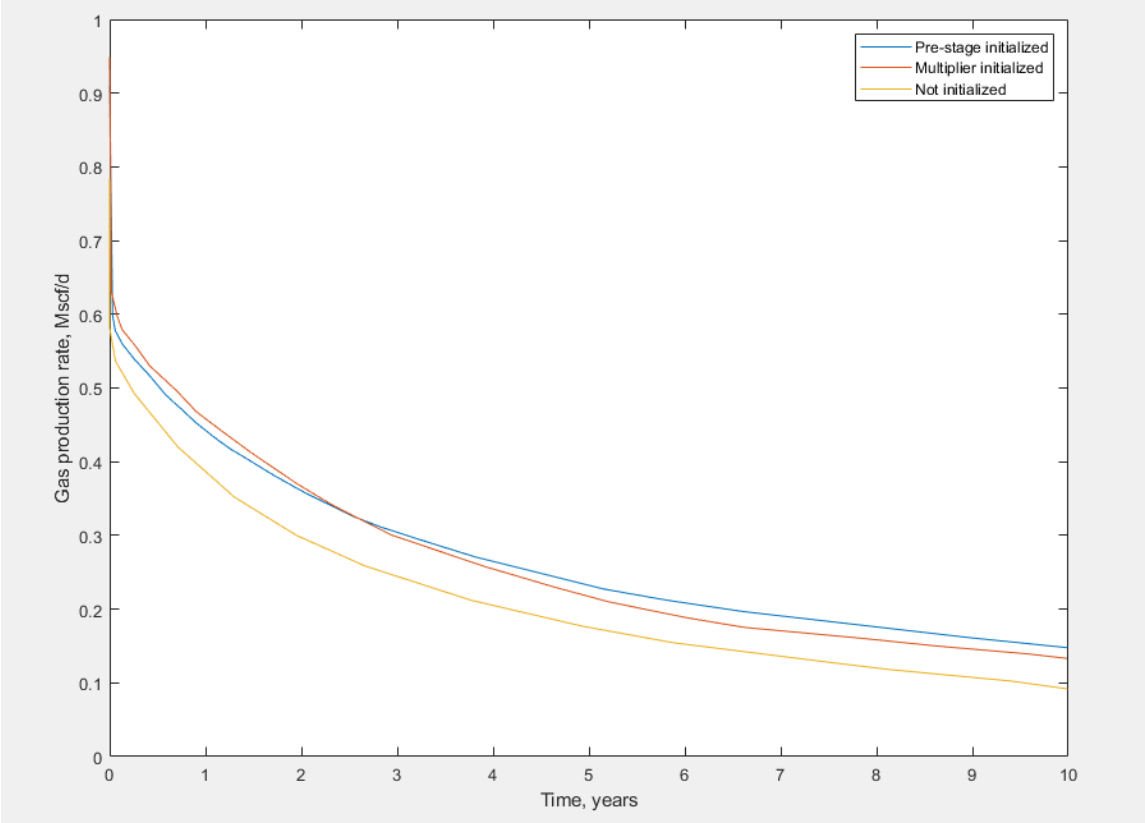

Figure 5.3 Gas production rate using different initialization methods.

5.2. Water flowback

Water flowback is an important factor in evaluating the fracking process for unconventional reservoirs. When water is pumped into the wellbore and breaks the formations, some enter the pores and flow back during the early production stage. Current knowledge holds the opinion that the higher the flowback, the better the fracking process is. However, some field practices observe the opposite performance.

A numerical test is proposed to analyze the flowback effect on gas production. Three cases with different flowback percentages varying from 40% to 80% are proposed, and the results are provided in **Figure 5.4**.

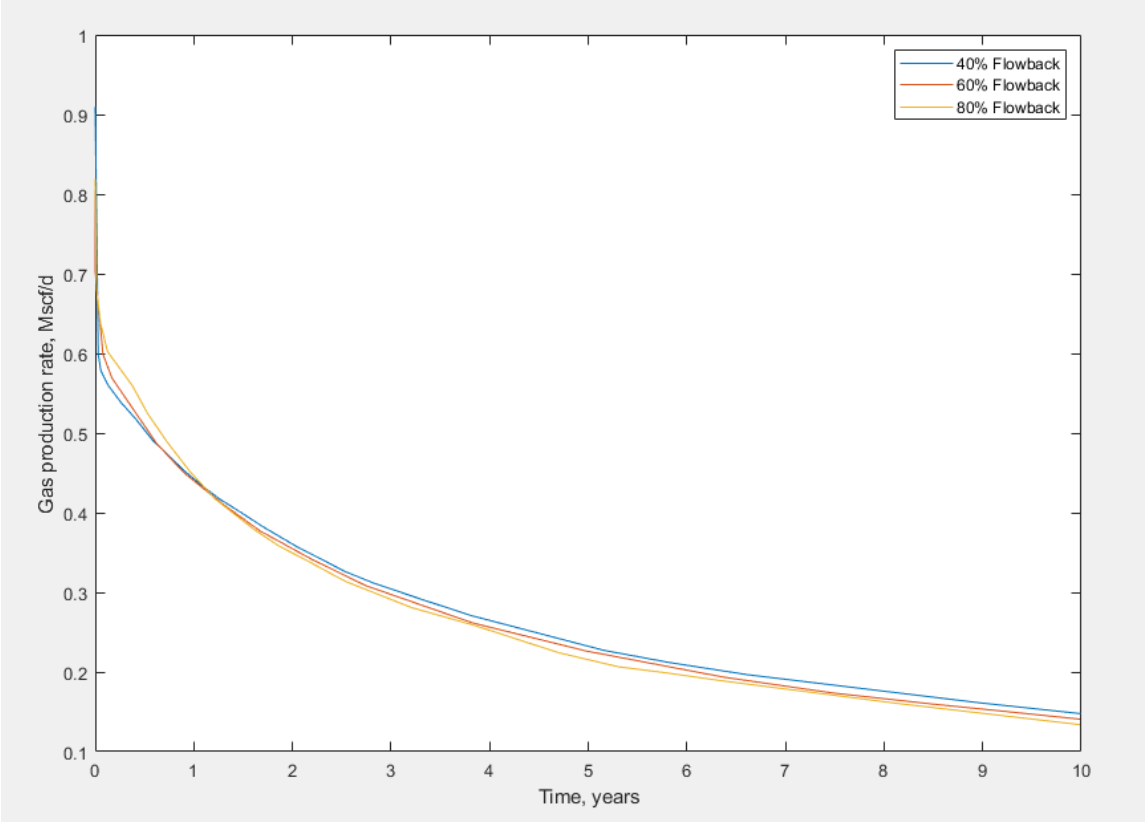

Figure 5.4 Gas production rate using different flowback data.

As seen from the results, the early-stage production is enhanced when we have a higher flowback percentage. This is reasoned from less water imbibition and less formation damage as higher relative gas permeability gives out higher rates. The later production, on the other hand, shows a different trend that lower flowback profiles have slightly higher cumulative production. This matches the observation for a few shale gas field practices, but currently, no widely accepted explanation has been given out yet for this behavior. One possible explanation can be that the fracking fluid imbibition gives out extra pressure support inside reservoirs, which does not decline as fast as normal reservoir pressure during gas production.

5.3. Hydrocarbon transportation mechanism analysis

There are multiple hydrocarbon transport mechanisms which were discussed in Chapter 2. A numerical test is performed to measure how different mechanisms like adsorption/desorption and diffusion affect the overall production by removing the diffusion and absorption terms in governing equations. The results are shown in **Figure 5.5**.

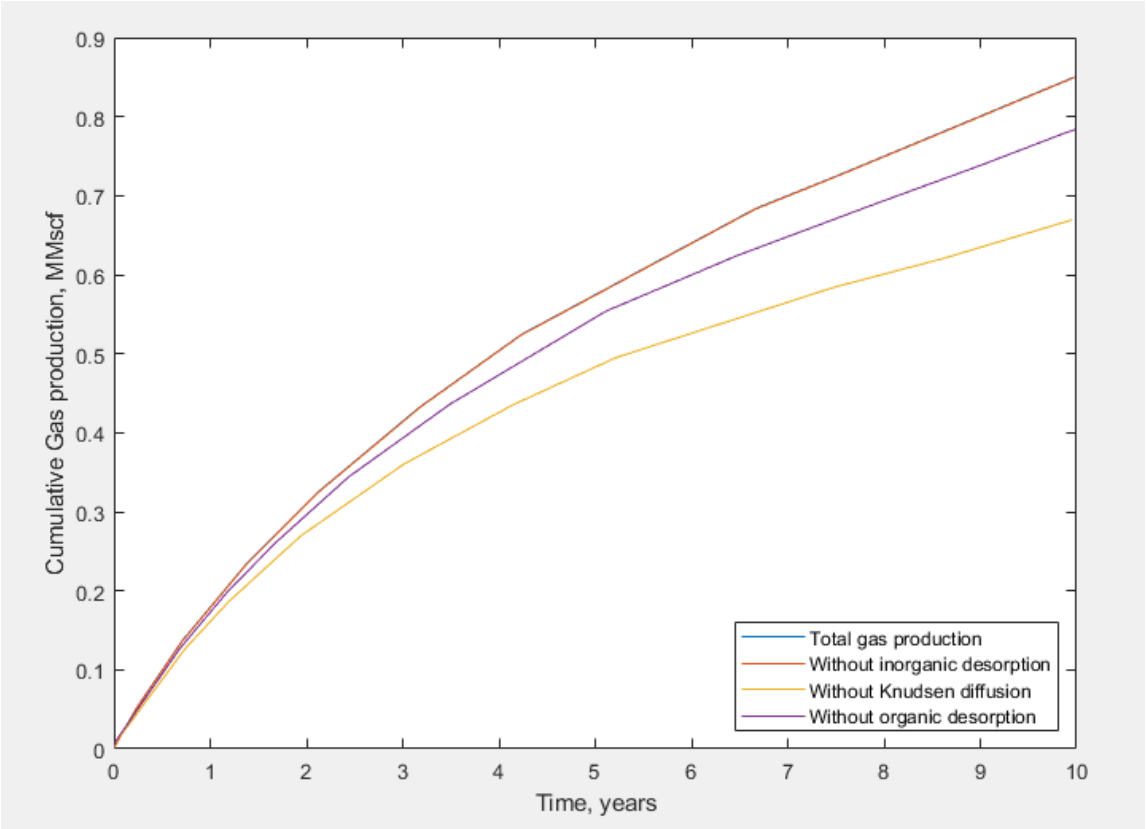

Figure 5.5 Hydrocarbon transportation mechanisms.

A 10-year cumulative gas production simulation is performed. The Knudsen diffusion has shown a massive impact on total gas production, which weights the highest. The Langmuir absorption-desorption of gas in organic matters has also shown a large impact, while the inorganic desorption is negligible. This can be used as a guide when performance weights more than accuracy that a few mechanisms can be safely removed without much hassle.

5.4. Non-Darcy water flux

While the Darcy flow model simplifies the calculation and saves computational time, the water flux in nano-pores experimentally shows strong non-Darcy behavior.

Field experiences show that using the Darcy model tends to overestimate water production.

A non-Darcy water model is proposed based on the boundary layer theory. The non-Darcy correlation for water can be derived as a multiplier to the potential gradient, which has the following form:

$$\alpha = \nabla\varphi_{w,m}\left(1-\frac{\delta}{r_0}\right)^4\left(1-\frac{8\tau_0}{3r_0\left(1-\frac{\delta}{r_0}\right)\nabla\varphi_{w,m}}\right) \tag{5.1}$$

Whereas $\frac{\delta}{r_0}$ and $\frac{8\tau_0}{3r_0}$ are fitting parameters from lab test data (Yu, 2012).

A numerical experiment is performed to evaluate the effect of non-Darcy water flux by comparing the results with a simulation case done using a normal potential gradient. The result is shown in **Figure 5.6**. Compared with the Darcy flow, non-Darcy tend to give out much less water flux in the early stage. This agrees with field practice, while the Darcy flux commonly overestimate the water flux amount.

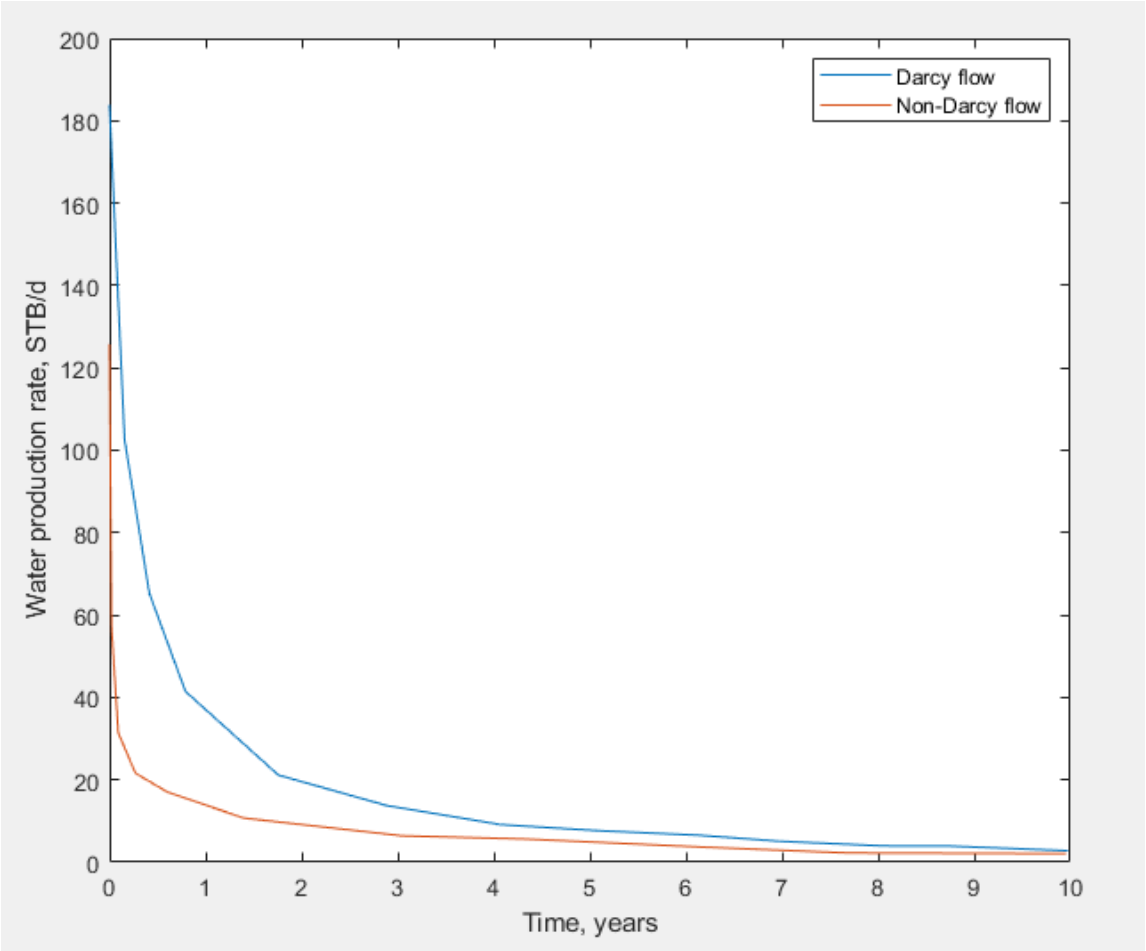

Figure 5.6 Hydrocarbon transportation mechanisms.

5.5. Domain decomposition solver for complex fracture conditions

The flow in unconventional reservoirs is dominated by the fracture network, which is smaller but rather difficult to converge due to the flux between matrix and fracture surfaces. On the other hand, the reservoir bulk matrix is usually large in problem size, but the condition is usually much better. Thus, a domain decomposition solver is designed to take the benefit from both parts.

The EDFM fracture model couples the fracture network with corresponding matrix grids with non-neighbor connections (NNC). Thus, a grid domain decomposition can be directly made to split all grids with fractures from the bulk reservoir as well as further splitting the rest parts of the bulk matrix into colored connection graphs. The reservoir problem is then divided into smaller models which are connected to each other via the boundary conditions (**Figure 5.7**).

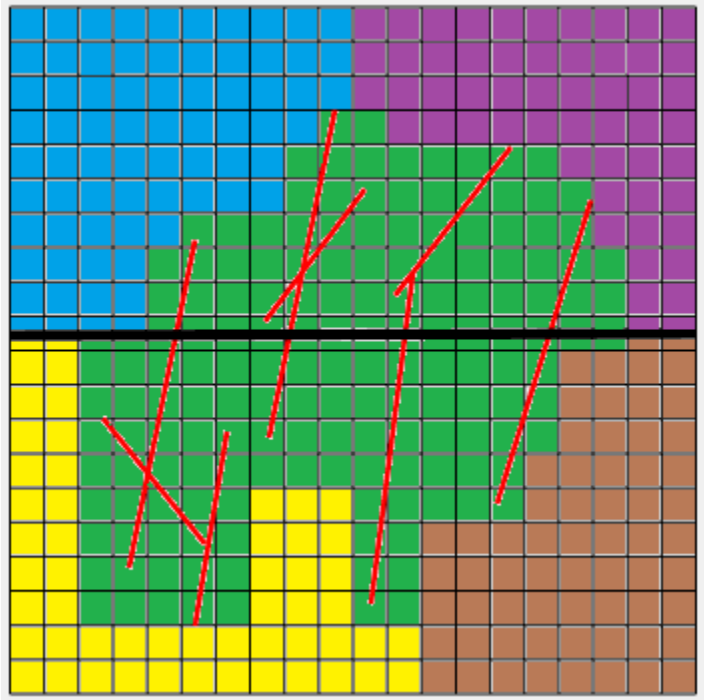

Figure 5.7 Colored graph of grid domain decomposition. The red lines represent the fracture network, the green region is marked as the fracture sub-domain, while the other colors are matrix sub-domains. Note that this decomposition will request all matrix sub-domains to communicate with the fracture sub-domain, which can create unexpected performance barriers.

Once the grid domain decomposition is finished, it becomes possible to solve all the sub-domains using a Schwarz method (Smith, 1997; Toselli & Widlund, 2004). Addictive Schwarz is preferred in this case due to its nature of being highly

parallelizable. Boundary values are saved as ghost cells and passed between the connected sub-domains at every simulation timestep.

However, this method must be used with enough care. The first issue a user may face is that the decomposition of sub-domain grids can have a significant impact on solver performance, as they must wait for communication from every other connected sub-domain to exchange their solved boundary values for the current timestep. Thus, it is preferred to have every sub-domain connected to the least-possible number of other subdomains with the least-possible number of boundary ghost cells to reduce the communication cost.

The second issue comes from the nature that the fracture sub-domain is much harder to solve compared with the other sub-domains. This may result in waiting time cost of computational resources, for which a load balancing control is required to both balance the static load by decomposing different sub-domain with adequate sizes and balance the dynamic load by automatically assigning tasks to corresponding computational resources.

A CUDA implementation of this domain decomposition solver is proposed with an MPI-based load balancing program. Performance has been tested and comparisons are made with respect to the global solver. Results are shown in **Figure 5.8**.

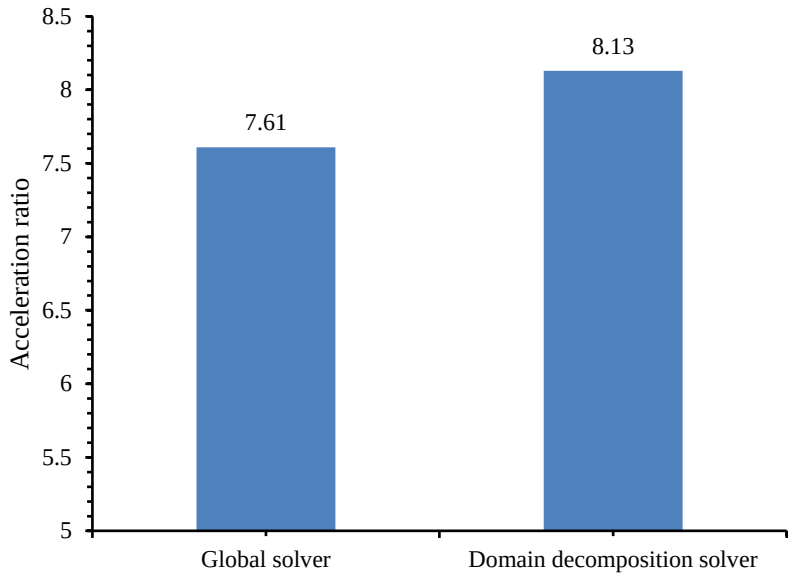

Figure 5.8 Performance test for the domain decomposition solver.

The performance of this domain decomposition solver is only slightly ahead of the global solver on a single GPU. However, when the case becomes extremely large and the usage of distributed memory architectures becomes a must, the domain decomposition solver is expected to perform much better.

5.6. Fracture closure effect

Fractures are not constant in length and width during fracturing and production. The fractures are monitored to close in practices when the reservoir is being produced and the pressure depletes inside fractures. This led to the shrinking of fracture conductivity as well as a reduction in stimulated rock volume (**Figure 5.9**).

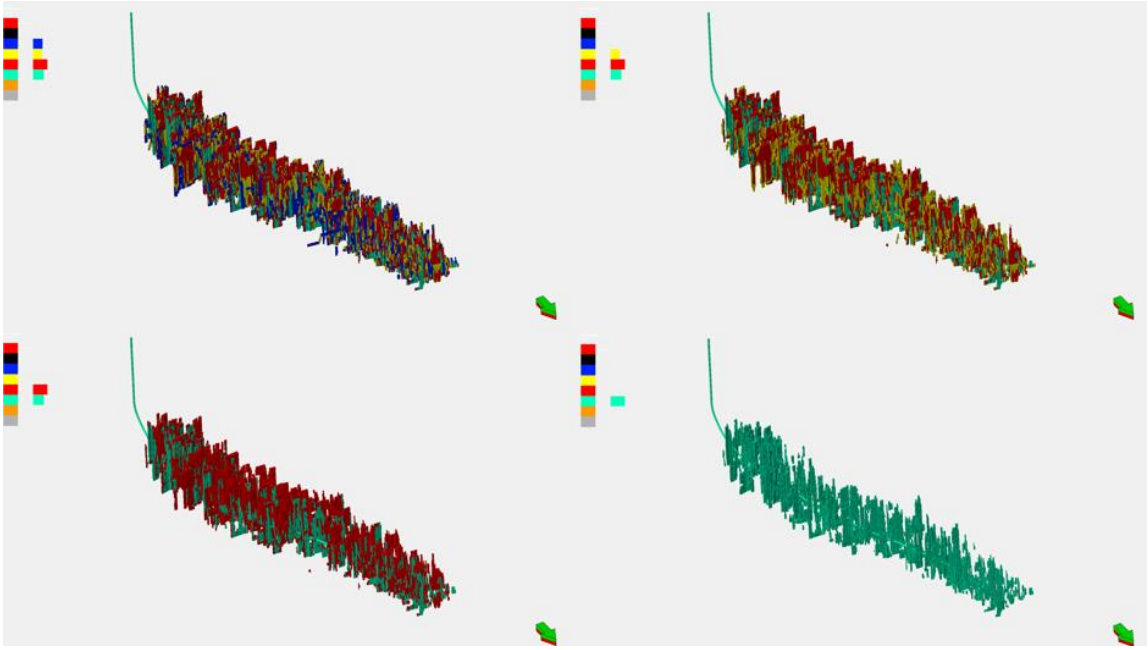

Figure 5.9 Fracture closure effect during pressure depletion.

The current approach to model this behavior usually goes to a transmissibility curve method (Xiong et al., 2021), which creates transmissibility multiplier curves with respect to the rock pressure for different regions and reduces the fracture conductivity directly. However, this method does not consider that the closure of fractures can lead to disconnection from the fracture network. This dissertation provides an approach to model this process by using a dynamic EDFM gridding technique (Jiang, Killough, Li, et al., 2021), which uses the pressure and temperature computed at every simulation timestep to explicitly recalibrate the fracture length with a propagation model (Jiang, Killough, Wu, et al., 2021) and update the fracture-fracture and fracture-matrix connections. A workflow of this process is shown in **Figure 5.10**.

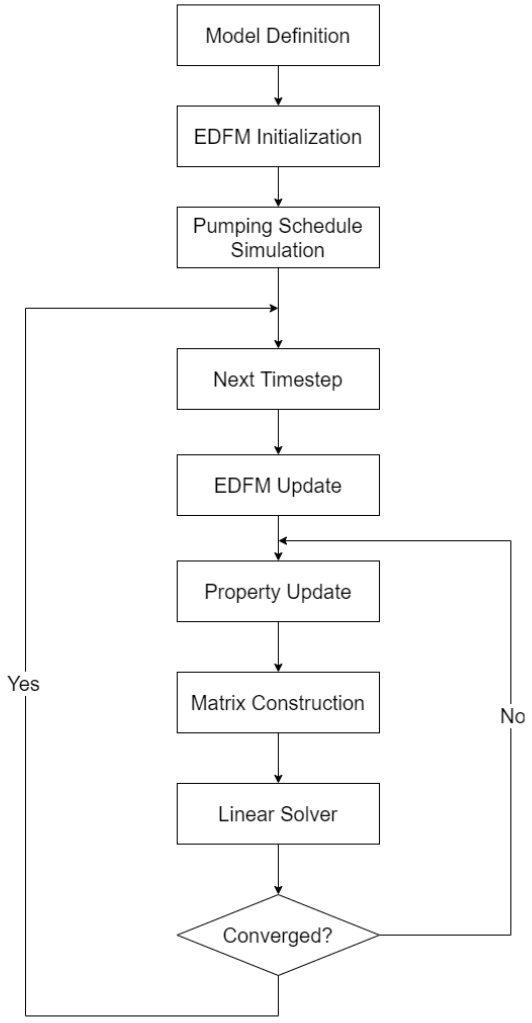

Figure 5.10 Workflow of dynamic EDFM method.

Simulations have been performed to further test this fracture closure behavior using this dynamic gridding method. For the test case, one can monitor that the fracture closure removes the hydraulic fracture from the outermost natural fractures (**Figure 5.11**), which causes a noticeable difference in the production curve (**Figure 5.12**).

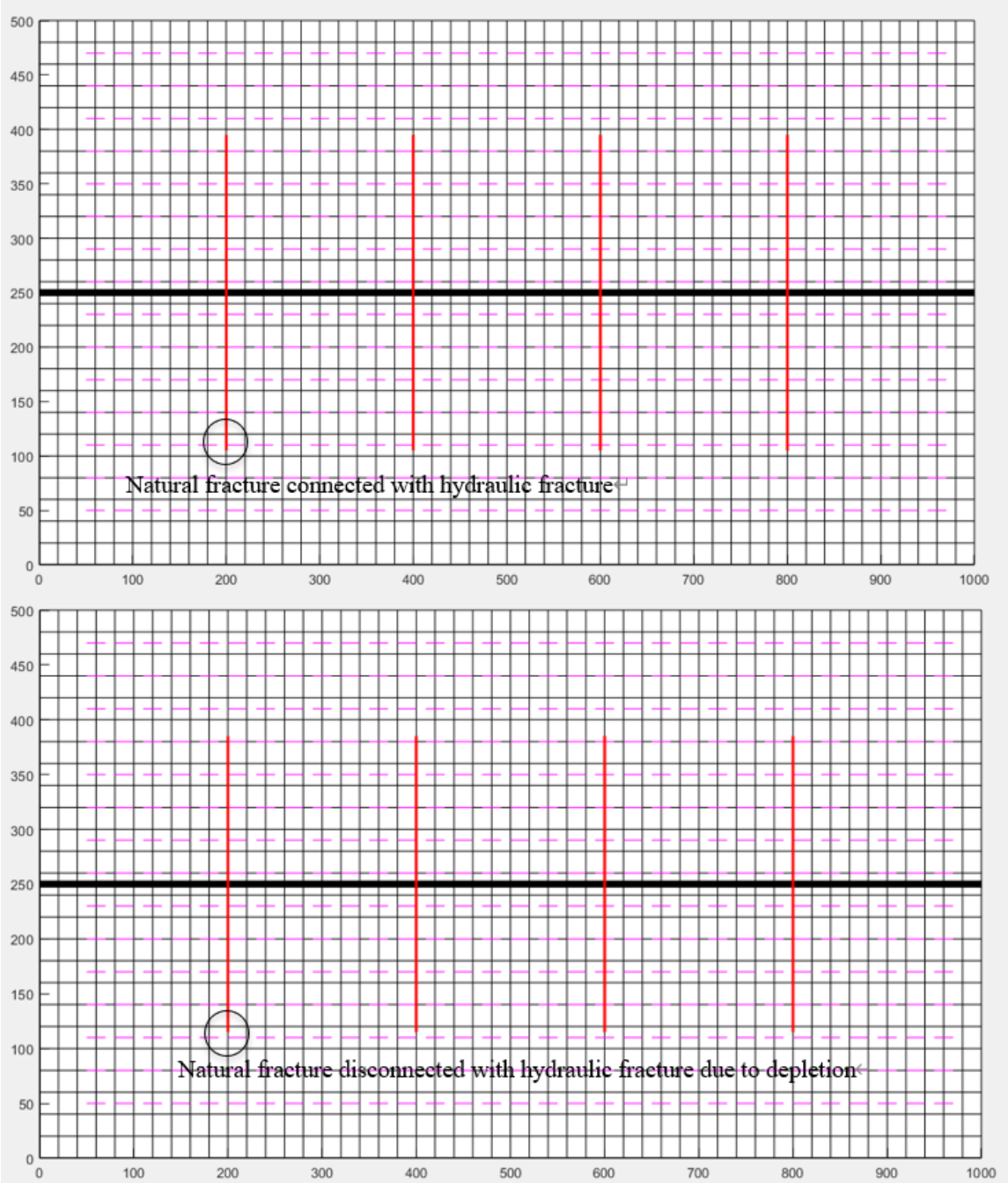

Figure 5.11 Case study of dynamic fracture closure.

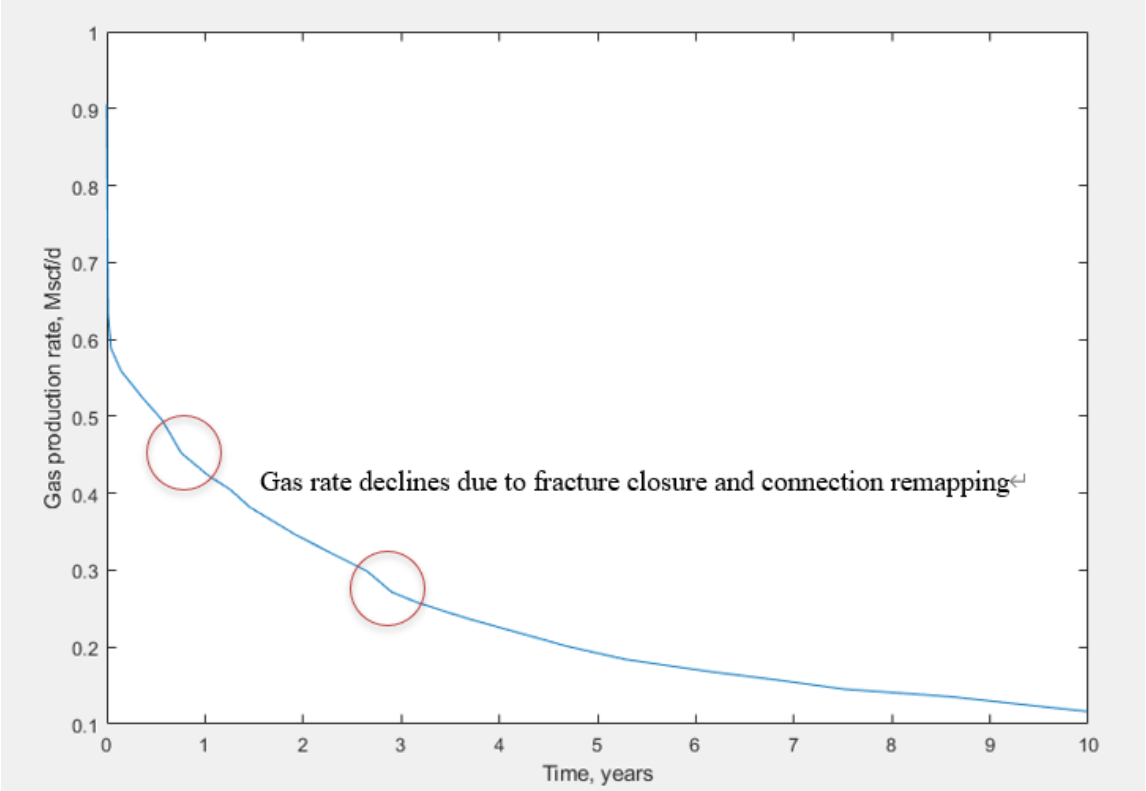

Figure 5.12 Case study of dynamic fracture closure.

6. CONCLUSIONS

Simulation of unconventional reservoirs with complex fracture conditions has long been a challenging task due to the high complexity and uncertainty of the reservoir. The difficulty comes from both the modeling and the computing. In this dissertation, a complete workflow has been proposed for this problem, including:

1) A reservoir simulator GURU with the multi-component, multi-physics, multi-porosity-permeability setting. The fully implicit Jacobian is created for both matrix and fractures as well as wells.
2) The performance of GURU is further optimized to enhance its capability to solve larger and more complex problems. A parallel linear solver is developed as well as a CPR-type multi-stage parallel preconditioner with adaptive settings. Implementations on both CPU and GPU have been finished.
3) An accurate and efficient fractured unconventional modeling method is proposed. This includes a CVT-type point-based matrix gridding method as well as an LGR enhanced EDFM fracture gridding method.
4) New methods have been proposed for fracture network characterization. Hydraulic fractures are characterized using a micro-seismic data-driven imaging method while the natural fractures are captured using a fractal model with empirical risk minimization calibration.

5) Specific topics related to unconventional reservoir simulation with complex fracture conditions have been discussed, including the initialization methods, water flowback effect, hydrocarbon transportation mechanisms, non-Darcy water flow effect, and fracture closure effect.
6) A domain decomposition solver is developed based on the nature of fractured reservoir performance and EDFM modeling. CUDA and MPI implementations are finished which pushes GURU's capability to a new scale.

REFERENCES

Alharthy, N. S., Teklu, T. W., Nguyen, T. N., Kazemi, H., & Graves, R. M. (2016). Nanopore Compositional Modeling in Unconventional Shale Reservoirs. *SPE Reservoir Evaluation & Engineering*, *19*(03), 415-428. https://doi.org/10.2118/166306-pa

Ambrose, R. J., Hartman, R. C., Diaz-Campos, M., Akkutlu, I. Y., & Sondergeld, C. H. (2010). New Pore-scale Considerations for Shale Gas in Place Calculations. SPE Unconventional Gas Conference,

Ambrose, R. J., Hartman, R. C., Diaz-Campos, M., Akkutlu, I. Y., & Sondergeld, C. H. (2011). Shale Gas-in-Place Calculations Part I: New Pore-Scale Considerations. *SPE Journal*, *17*(01), 219-229. https://doi.org/10.2118/131772-pa

Anbalagan, H. P. (2019). *Case Study: Analysis of Stimulated Reservoir Volume Properties Through Application of Embedded Discrete Fracture Modelling*

Azim, R. A., & Abdelmoneim, S. S. (2013). Modeling hydraulic fractures in finite difference simulators using amalgam local grid refinement (LGR). *Journal of Petroleum Exploration and Production Technology*, *3*(1), 21-35. https://doi.org/10.1007/s13202-012-0038-6

Balberg, I., Berkowitz, B., & Drachsler, G. E. (1991). Application of a percolation model to flow in fractured hard rocks. *Journal of Geophysical Research: Solid Earth*, *96*(B6), 10015-10021. https://doi.org/https://doi.org/10.1029/91JB00681

Bolles, R. C., & Fischler, M. A. (1981). A RANSAC-based approach to model fitting and its application to finding cylinders in range data. IJCAI,

Brandt, A. (1984). Algebraic multigrid (AMG) for sparse matrix eqations. *Sparsity and its Applications*, 257-284.

Cao, H. (2002). *Development of techniques for general purpose simulators*. Stanford University.

Cao, Y., Yan, B., Alfi, M., & Killough, J. E. (2015). A Novel Compositional Model of Simulating Fluid Flow in Shale Resrvoirs - Some Priliminary Tests and Results. SPE Reservoir Characterisation and Simulation Conference and Exhibition,

Chai, Z., Yan, B., Killough, J. E., & Wang, Y. (2016). Dynamic Embedded Discrete Fracture Multi-Continuum Model for the Simulation of Fractured Shale Reservoirs. International Petroleum Technology Conference,

Cipolla, C. L., Fitzpatrick, T., Williams, M. J., & Ganguly, U. K. (2011). Seismic-to-Simulation for Unconventional Reservoir Development. SPE Reservoir Characterisation and Simulation Conference and Exhibition,

Civan, F. (2010). Effective Correlation of Apparent Gas Permeability in Tight Porous Media. *Transport in Porous Media*, *82*(2), 375-384. https://doi.org/10.1007/s11242-009-9432-z

Concus, P., Golub, G. H., & Meurant, G. (1985). Block Preconditioning for the Conjugate Gradient Method. *SIAM Journal on Scientific and Statistical Computing*, *6*(1), 220-252. https://doi.org/10.1137/0906018

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms, Third Edition*. The MIT Press.

Courant, R., Isaacson, E., & Rees, M. (1952). On the solution of nonlinear hyperbolic differential equations by finite differences. *Communications on Pure and Applied Mathematics*, *5*(3), 243-255. https://doi.org/https://doi.org/10.1002/cpa.3160050303

Cusini, M., Lukyanov, A. A., Natvig, J., & Hajibeygi, H. (2015). Constrained pressure residual multiscale (CPR-MS) method for fully implicit simulation of multiphase flow in porous media. *Journal of Computational Physics*, *299*, 472-486. https://doi.org/https://doi.org/10.1016/j.jcp.2015.07.019

Dagum, L., & Menon, R. (1998). OpenMP: an industry standard API for shared-memory programming. *IEEE Computational Science and Engineering*, *5*(1), 46-55. https://doi.org/10.1109/99.660313

Douglas, J., Jr., Peaceman, D. W., & Rachford, H. H., Jr. (1959). A Method for Calculating Multi-Dimensional Immiscible Displacement. *Transactions of the AIME*, *216*(01), 297-308. https://doi.org/10.2118/1327-g

Du, Q., Faber, V., & Gunzburger, M. (1999). Centroidal Voronoi tessellations: Applications and algorithms. *SIAM review*, *41*(4), 637-676.

Ebel, D. S., Ghiorso, M. S., Sack, R. O., & Grossman, L. (2000). Gibbs energy minimization in gas+ liquid+ solid systems. *Journal of Computational Chemistry*, *21*(4), 247-256.

EIA. (2021). Annual Energy Outlook 2021. [www.eia.gov/aeo].

Falgout, R. D. (2006). *An introduction to algebraic multigrid*.

Fathi, E., & Akkutlu, I. Y. (2012). Lattice Boltzmann Method for Simulation of Shale Gas Transport in Kerogen. *SPE Journal*, *18*(01), 27-37. https://doi.org/10.2118/146821-pa

Feast, G., Boosari, S. S. H., Wu, K., Walton, J., Cheng, Z., & Chen, B. (2015). Modeling and Simulation of Natural Gas Production from Unconventional Shale Reservoirs. *International Journal of Clean Coal and Energy*, *Vol.04No.02*, 10, Article 54685. https://doi.org/10.4236/ijcce.2015.42003

Forum, M. P. (1994). *MPI: A Message-Passing Interface Standard*.

Gale, J. F. W., Laubach, S. E., Olson, J. E., Eichhubl, P., & Fall, A. (2014). Natural fractures in shale: A review and new observations. *AAPG Bulletin*, *98*(11), 2165-2216. https://doi.org/10.1306/08121413151

Goyat, S., & Sahoo, A. (2019). Scheduling algorithm for CPU-GPU based heterogeneous clustered environment using map-reduce data processing. *ARPN Journal of Engineering and Applied Sciences*, *14*, 213-221.

Grama, A., Gupta, A., Karypis, G., & Kumar, V. Introduction to Parallel Computing.

Helnemann, Z. E., Brand, C. W., Munka, M., & Chen, Y. M. (1991). Modeling Reservoir Geometry With Irregular Grids. *SPE Reservoir Engineering*, *6*(02), 225-232. https://doi.org/10.2118/18412-pa

Holditch, S. A. (2003). The Increasing Role of Unconventional Reservoirs in the Future of the Oil and Gas Business. *Journal of Petroleum Technology*, *55*(11), 34-79. https://doi.org/10.2118/1103-0034-jpt

Hu, B., Wang, J., & Ma, Z. (2020). A Fractal Discrete Fracture Network Based Model for Gas Production from Fractured Shale Reservoirs. *Energies*, *13*(7), 1857. https://www.mdpi.com/1996-1073/13/7/1857

Ji, L., Zhang, T., Milliken, K. L., Qu, J., & Zhang, X. (2012). Experimental investigation of main controls to methane adsorption in clay-rich rocks. *Applied Geochemistry*, *27*(12), 2533-2545. https://doi.org/https://doi.org/10.1016/j.apgeochem.2012.08.027

Jiang, Y., Killough, J., Li, L., Cui, X., & Tang, J. (2021). EDFM-based Multi-Continuum Shale Gas Simulation with Low Velocity Non-Darcy Water Flow Effect. SPE Reservoir Simulation Conference,

Jiang, Y., Killough, J. E., Wu, X., & Cui, Y. (2021). Transient Temperature Impact on Deep Reservoir Fracturing. *Geofluids*, *2021*, 6653442. https://doi.org/10.1155/2021/6653442

Kazemi, H., Merrill, L. S., Jr., Porterfield, K. L., & Zeman, P. R. (1976). Numerical Simulation of Water-Oil Flow in Naturally Fractured Reservoirs. *Society of Petroleum Engineers Journal*, *16*(06), 317-326. https://doi.org/10.2118/5719-pa

Kestin, J., Khalifa, H. E., & Correia, R. J. (1981). Tables of the dynamic and kinematic viscosity of aqueous NaCl solutions in the temperature range 20–150 °C and the pressure range 0.1–35 MPa. *Journal of Physical and Chemical Reference Data*, *10*(1), 71-88. https://doi.org/10.1063/1.555641

Koester, D. P., Ranka, S., & Fox, G. C. (1994, 14-18 Nov. 1994). A parallel Gauss-Seidel algorithm for sparse power system matrices. Supercomputing '94:Proceedings of the 1994 ACM/IEEE Conference on Supercomputing,

Le Calvez, J. H., Craven, M. E., Klem, R. C., Baihly, J. D., Bennett, L. A., & Brook, K. (2007). Real-Time Microseismic Monitoring of Hydraulic Fracture Treatment: A Tool To Improve Completion and Reservoir Management. SPE Hydraulic Fracturing Technology Conference,

Lee, S., Jensen, C., & Lough, M. (2000). Efficient finite-difference model for flow in a reservoir with multiple length-scale fractures. *SPE Journal*, *5*(03), 268-275.

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, *28*(2), 129-137. https://doi.org/10.1109/TIT.1982.1056489

Lohrenz, J., Bray, B. G., & Clark, C. R. (1964). Calculating Viscosities of Reservoir Fluids From Their Compositions. *Journal of Petroleum Technology*, *16*(10), 1171-1176. https://doi.org/10.2118/915-pa

Maxwell, S. C., Urbancic, T. I., Steinsberger, N., & Zinno, R. (2002). Microseismic Imaging of Hydraulic Fracture Complexity in the Barnett Shale. SPE Annual Technical Conference and Exhibition,

Merland, R., Levy, B., & Caumon, G. (2011). *Building PEBI Grids Conforming To 3D Geological Features Using Centroidal Voronoi Tessellations*. https://doi.org/10.5242/iamg.2011.0064

Mesbah, M., Vatani, A., Siavashi, M., & Doranehgard, M. H. (2019). Parallel processing of numerical simulation of two-phase flow in fractured reservoirs considering the effect of natural flow barriers using the streamline simulation method. *International Journal of Heat and Mass Transfer*, *131*, 574-583. https://doi.org/https://doi.org/10.1016/j.ijheatmasstransfer.2018.11.097

Meyerink, J. A. (1983). Iterative Methods for the Solution of Linear Equations Based on Incomplete Block Factorization of the Matrix. SPE Reservoir Simulation Symposium,

Miao, T., Yu, B., Duan, Y., & Fang, Q. (2015). A fractal analysis of permeability for fractured rocks. *International Journal of Heat and Mass Transfer*, *81*, 75-80.

Moinfar, A., Varavei, A., Sepehrnoori, K., & Johns, R. T. (2013). Development of a Coupled Dual Continuum and Discrete Fracture Model for the Simulation of Unconventional Reservoirs. SPE Reservoir Simulation Symposium,

Molenaar, M. M. M., Hill, D. J. J., Webster, P., Fidan, E., & Birch, B. (2012). First Downhole Application of Distributed Acoustic Sensing for Hydraulic-Fracturing Monitoring and Diagnostics. *SPE Drilling & Completion*, *27*(01), 32-38. https://doi.org/10.2118/140561-pa

Monteagudo, J. E. P., & Firoozabadi, A. (2007). Comparison of fully implicit and IMPES formulations for simulation of water injection in fractured and unfractured media. *International Journal for Numerical Methods in Engineering*, *69*(4), 698-728. https://doi.org/https://doi.org/10.1002/nme.1783

Nickolls, J., Buck, I., Garland, M., & Skadron, K. (2008). Scalable Parallel Programming with CUDA: Is CUDA the parallel programming model that application developers have been waiting for? *Queue*, *6*(2), 40–53. https://doi.org/10.1145/1365490.1365500

Niven, E. B., & Deutsch, C. V. (2012). Non-random Discrete Fracture Network Modeling. In P. Abrahamsen, R. Hauge, & O. Kolbjørnsen (Eds.), *Geostatistics Oslo 2012* (pp. 275-286). Springer Netherlands. https://doi.org/10.1007/978-94-007-4153-9_22

Norbeck, J. H., Huang, H., Podgorney, R., & Horne, R. N. (2014). An Integrated Discrete Fracture Model for Description of Dynamic Behavior in Fractured Reservoirs.

Notay, Y. (2010). An aggregation-based algebraic multigrid method. *Electronic transactions on numerical analysis*, *37*(6), 123-146.

Odusina, E., Sondergeld, C., & Rai, C. (2011). An NMR Study on Shale Wettability. Canadian Unconventional Resources Conference,

Ouillon, G., Castaing, C., & Sornette, D. (1996). Hierarchical geometry of faulting. *Journal of Geophysical Research: Solid Earth*, *101*(B3), 5477-5487.

Ozkan, E., Brown, M., Raghavan, R., & Kazemi, H. (2011). Comparison of Fractured-Horizontal-Well Performance in Tight Sand and Shale Reservoirs. *SPE Reservoir Evaluation & Engineering*, *14*(02), 248-259. https://doi.org/10.2118/121290-pa

Passey, Q. R., Bohacs, K. M., Esch, W. L., Klimentidis, R., & Sinha, S. (2010). From Oil-Prone Source Rock to Gas-Producing Shale Reservoir – Geologic and Petrophysical Characterization of Unconventional Shale-Gas Reservoirs. International Oil and Gas Conference and Exhibition in China,

Peaceman, D. W. (1990). Interpretation of Wellblock Pressures in Numerical Reservoir Simulations Part 3—Off-Center and Multiple Wells Within a Wellblock. *SPE Reservoir Engineering*, *5*(02), 227-232. https://doi.org/10.2118/16976-pa

Peng, D.-Y., & Robinson, D. B. (1976). A New Two-Constant Equation of State. *Industrial & Engineering Chemistry Fundamentals*, *15*(1), 59-64. https://doi.org/10.1021/i160057a011

Priest, S. D. (1993). *Discontinuity analysis for rock engineering*. Springer Science & Business Media.

Rahim, Z., Al-Anazi, H., Al-Kanaan, A., Habbtar, A., Al-Omair, A., Senturk, N., & Kalinin, D. (2012). Productivity Increase Using Hydraulic Fracturing in Conventional and Tight Gas Reservoirs – Expectation vs. Reality. SPE Middle East Unconventional Gas Conference and Exhibition,

Ren, L., Su, Y., Zhan, S., Hao, Y., Meng, F., & Sheng, G. (2016). Modeling and simulation of complex fracture network propagation with SRV fracturing in unconventional shale reservoirs. *Journal of Natural Gas Science and Engineering*, *28*, 132-141. https://doi.org/https://doi.org/10.1016/j.jngse.2015.11.042

Roy, T., Jönsthövel, T., Lemon, C., & Wathen, A. (2019). *A constrained pressure-temperature residual (CPTR) method for non-isothermal multiphase flow in porous media*.

Saad, Y. (2003). *Iterative methods for sparse linear systems*. SIAM.

Saad, Y. (2011). *Numerical methods for large eigenvalue problems: revised edition*. SIAM.

Seibel, M., Baig, A., & Urbancic, T. (2010). Single Versus Multiwell Microseismic Recording: What Effect Monitoring Configuration Has On Interpretation. 2010 SEG Annual Meeting,

Shakiba, M., & Sepehrnoori, K. (2015). Using Embedded Discrete Fracture Model (EDFM) and Microseismic Monitoring Data to Characterize the Complex Hydraulic Fracture Networks. SPE Annual Technical Conference and Exhibition,

Sierra, L. (2016). Is Induced Fracture Complexity Always Required in Unconventional Reservoir Stimulation? SPE Argentina Exploration and Production of Unconventional Resources Symposium,

Smith, B. F. (1997). Domain Decomposition Methods for Partial Differential Equations. In D. E. Keyes, A. Sameh, & V. Venkatakrishnan (Eds.), *Parallel Numerical Algorithms* (pp. 225-243). Springer Netherlands. https://doi.org/10.1007/978-94-011-5412-3_8

Sondergeld, C. H., Ambrose, R. J., Rai, C. S., & Moncrieff, J. (2010). Micro-Structural Studies of Gas Shales. SPE Unconventional Gas Conference,

Sookprasong, P. A., Gill, C. C., & Hurt, R. S. (2014). Lessons Learned from DAS and DTS in MultiCluster, MultiStage Horizontal Well Fracturing: Interpretation of Hydraulic Fracture Initiation and Propagation through Diagnostics. IADC/SPE Asia Pacific Drilling Technology Conference,

Sun, J., & Schechter, D. (2015). Optimization-Based Unstructured Meshing Algorithms for Simulation of Hydraulically and Naturally Fractured Reservoirs With Variable Distribution of Fracture Aperture, Spacing, Length, and Strike. *SPE Reservoir Evaluation & Engineering*, *18*(04), 463-480. https://doi.org/10.2118/170703-pa

Ţene, M., Bosma, S. B. M., Al Kobaisi, M. S., & Hajibeygi, H. (2017). Projection-based Embedded Discrete Fracture Model (pEDFM). *Advances in Water Resources*, *105*, 205-216. https://doi.org/https://doi.org/10.1016/j.advwatres.2017.05.009

Teng, W., Jiang, R., Teng, L., Qiao, X., Jiang, Y., He, J., & Gao, Y. (2016). Production performance analysis of multiple fractured horizontal wells with finite-conductivity fractures in shale gas reservoirs. *Journal of Natural Gas Science and Engineering*, *36*, 747-759. https://doi.org/https://doi.org/10.1016/j.jngse.2016.10.030

Tian, S., Ren, W., Li, G., Yang, R., & Wang, T. (2017). A Theoretical Analysis of Pore Size Distribution Effects on Shale Apparent Permeability. *Geofluids*, *2017*, 7492328. https://doi.org/10.1155/2017/7492328

Toselli, A., & Widlund, O. (2004). *Domain decomposition methods-algorithms and theory* (Vol. 34). Springer Science & Business Media.

Vaněk, P., Mandel, J., & Brezina, M. (1996). Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, *56*(3), 179-196. https://doi.org/10.1007/BF02238511

Wallis, J. R. (1983). Incomplete Gaussian Elimination as a Preconditioning for Generalized Conjugate Gradient Acceleration. SPE Reservoir Simulation Symposium,

Wallis, J. R., Kendall, R. P., & Little, T. E. (1985). Constrained Residual Acceleration of Conjugate Residual Methods. SPE Reservoir Simulation Symposium,

Wang, F. P., & Reed, R. M. (2009). Pore Networks and Fluid Flow in Gas Shales. SPE Annual Technical Conference and Exhibition,

Wang, T., Tian, S., Li, G., Zhang, L., Sheng, M., & Ren, W. (2021). Molecular simulation of gas adsorption in shale nanopores: A critical review. *Renewable and Sustainable Energy Reviews*, *149*, 111391. https://doi.org/https://doi.org/10.1016/j.rser.2021.111391

Warpinski, N. R., Branagan, P. T., Peterson, R. E., Wolhart, S. L., & Uhl, J. E. (1998). Mapping Hydraulic Fracture Growth and Geometry Using Microseismic Events Detected by a Wireline Retrievable Accelerometer Array. SPE Gas Technology Symposium,

Warren, J. E., & Root, P. J. (1963). The Behavior of Naturally Fractured Reservoirs. *Society of Petroleum Engineers Journal*, *3*(03), 245-255. https://doi.org/10.2118/426-pa

Werneck, L. F., de Freitas, M. M., de Souza, G., Jatobá, L. F. C., & Amaral Souto, H. P. (2019). An OpenMP parallel implementation using a coprocessor for numerical simulation of oil reservoirs. *Computational and Applied Mathematics*, *38*(2), 33. https://doi.org/10.1007/s40314-019-0788-6

Xiao, G., Jingjing, M., Yiwei, C., & Weixin, L. (2010). An Integration Approach for Evaluating Well Deliverability in Ultra Deep Sands. SPE Production and Operations Conference and Exhibition,

Xiaogang, L., Changyin, L., Liangping, Y., Zhiyu, S., & Zhaozhong, Y. (2017, 2017/03). A Review and Prospect of Numerical Simulation of Complex Hydraulic Fracture Propagation in Unconventional Reservoirs. Proceedings of the 2017 2nd International Conference on Modelling, Simulation and Applied Mathematics (MSAM2017),

Xiong, H., Yoon, S., & Jiang, Y. (2021). A Novel Method to Speedup Calibrating Horizontal Well Performance Model with Multi-Stage Fracturing Treatments and Its Applications in Delaware Basin. SPE Reservoir Simulation Conference,

Xu, J. (1992). Iterative methods by space decomposition and subspace correction. *SIAM review*, *34*(4), 581-613.

Yamazaki, I., Li, X., S., Rouet, F.-H., & Uçar, B. (2011). *Partitioning, Ordering, and Load Balancing in a Hierarchically Parallel Hybrid Linear Solver*. https://hal.inria.fr/hal-00797207

Yan, B. (2017). *Development of general unstructured reservoir utility and fractured reservoir modeling*

Yan, B., Wang, Y., & Killough, J. E. (2013). Beyond Dual-Porosity Modeling for the Simulation of Complex Flow Mechanisms in Shale Reservoirs. SPE Reservoir Simulation Symposium,

Yang, D., Xue, X., & Chen, J. (2018). High Resolution Hydraulic Fracture Network Modeling Using Flexible Dual Porosity Dual Permeability Framework. SPE Western Regional Meeting,

Yang, H., Sun, S., Li, Y., & Yang, C. (2019). Parallel reservoir simulators for fully implicit complementarity formulation of multicomponent compressible flows. *Computer Physics Communications*, *244*, 2-12.

Yu, J., Byun, J., & Seol, S. J. (2021). Imaging discrete fracture networks using the location and moment tensors of microseismic events. *Exploration Geophysics*, *52*(1), 42-53. https://doi.org/10.1080/08123985.2020.1761760

Yu, J. R. L. L. X. J. Y. R. Z. (2012). A nonlinear mathematical model for low-permeability reservoirs and well-testing analysis. *Acta Petrolei Sinica*, *33*(2), 264-268. https://doi.org/10.7623/syxb201202011

ZHANG, L., CUI, C., MA, X., SUN, Z., LIU, F., & ZHANG, K. (2019). A FRACTAL DISCRETE FRACTURE NETWORK MODEL FOR HISTORY MATCHING OF NATURALLY FRACTURED RESERVOIRS. *Fractals*, *27*(01), 1940008. https://doi.org/10.1142/s0218348x19400085

Zhang, R.-h., Zhang, L.-h., Tang, H.-y., Chen, S.-n., Zhao, Y.-l., Wu, J.-f., & Wang, K.-r. (2019). A simulator for production prediction of multistage fractured horizontal well in shale gas reservoir considering complex fracture geometry. *Journal of Natural Gas Science and Engineering*, *67*, 14-29. https://doi.org/https://doi.org/10.1016/j.jngse.2019.04.011

Zhang, T., Ellis, G. S., Ruppel, S. C., Milliken, K., & Yang, R. (2012). Effect of organic-matter type and thermal maturity on methane adsorption in shale-gas systems. *Organic Geochemistry*, *47*, 120-131. https://doi.org/https://doi.org/10.1016/j.orggeochem.2012.03.012

Zhang, Y., Yu, W., Sepehrnoori, K., & Di, Y. (2017). A Comprehensive Numerical Model for Simulating Fluid Transport in Nanopores. *Scientific reports*, *7*, 40507-40507. https://doi.org/10.1038/srep40507

Zhao, X., Huang, B., & Grasselli, G. (2021). Numerical Investigation of the Fracturing Effect Induced by Disturbing Stress of Hydrofracturing [Original Research]. *Frontiers in Earth Science*, *9*. https://doi.org/10.3389/feart.2021.751626