MODELING AND OPTIMIZATION OF PROPPANT DISTRIBUTIONS IN MULTI-CLUSTER HYDRAULIC FRACTURE-NATURAL FRACTURE (HF–NF) NETWORKS

A Thesis

by

YIDI WU

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

MASTER OF SCIENCE

Chair of Committee,	George J. Moridis
Co-Chair of Committee,	Thomas A. Blasingame
Committee Members,	Eduardo Gildin
	Dorrin Jarrahbashi
Head of Department,	Jeff Spath

August 2021

Major Subject: Petroleum Engineering

ABSTRACT

One of the foremost challenges with proppant transport in hydraulic and/or natural fractures within unconventional reservoirs is that the proppant particles do not remain in the suspension transport mode, deviating significantly from the suspension assumption shared by most existing models that allow particles to be represented by a continuous pseudo-fluid phase. The advantage of the CFD-DEM simulator used in this study is its proppant particle migration within a Lagrangian frame of reference, which enables the description of proppant transport in any transport mode. This coupled simulator can capture the complex interactions between the proppant particles, the carrier fluid, and the walls of the hydraulic and/or natural fractures. Such a knowledge-based tool can provide new insights that can be used to optimize the stimulation design, leading to better proppant placement and higher fracturing effectiveness — which ultimately leads to higher flowrates and improved hydrocarbon recovery.

The first part of the study involves the development of a robust proppant-bridging criterion at the Hydraulic Fracture-Natural Fracture (HF-NF) interface. The simulation results are used with an image classification algorithm to create a reliable proppant bridging criterion obtained from a well-trained Artificial Neural Network (ANN). The ANN is subjected to a standard K-fold cross-validation and is tested against a large suite of "control" simulation results.

The second part of the study focuses on optimizing the proppant distribution in a multi-cluster horizontal well system. The cumulative proppant distributions at each cluster are computed based on the number of representative particles entering each cluster at each timestep. A sensitivity analysis is performed using the key system parameters (cluster spacing, number of clusters, proppant size, and perforation width) and the results of these analyses are used to propose optimization strategies.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. George J. Moridis, my committee co-chair, Dr. Thomas A. Blasingame, and my committee members, Dr. Eduardo Gildin, and Dr. Dorrin Jarrahbashi, for their guidance and support throughout the course of this research.

I would also like to thank my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

I would like to extend my gratitude and acknowledgement to the Crisman Institute for Petroleum Research at Texas A&M University for funding this research project and to the High-Performance Research Center (HPRC) at Texas A&M University for providing the computational services required for this research project.

Finally, thanks to my mother and father for their encouragement and continuous support throughout my entire time pursuing M.S. degree.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a thesis committee consisting of Professor George J. Moridis (advisor), Thomas A. Blasingame (co-advisor), and Eduardo Gildin of the Department of Petroleum Engineering, and Professor Dorrin Jarrahbashi of the Department of Mechanical Engineering.

The idea of using additional simulations to evaluate the efficiency and accuracy of the trained artificial neural network in Chapter 3 was provided by Dr. George J. Moridis. The sensitivity analysis in Chapter 4 was revised by Dr. Thomas A. Blasingame. The deep machine learning approach in Chapter 3 was proposed by Dr. Eduardo Gildin. The simulator and base cases in the simulation were provided by Dr. Kou Rui.

All the numerical simulations were performed on the Ada cluster of the Texas A&M High Performance Research Computing (HPRC) at Texas A&M University

All other work conducted for the thesis was completed by the student independently.

Funding Sources

This graduate study was supported by the Crisman Institute for Petroleum Research (CIPR) at Texas A&M University.

NOMENCLATURE

Dimensional Variables:

$\boldsymbol{m_i}$	Mass of particle **i**, M [kg]
$\boldsymbol{t}$	Time, T [s]
$\boldsymbol{\mu_f}$	Fluid viscosity, M/LT [Pa·s]
$\boldsymbol{p}$	Pressure, M/LT^2 [Pa]
$\boldsymbol{V_{cell}}$	Volume of the fluid cell, L^3 [m^3]
$\boldsymbol{C_p}$	Proppant particle injection concentration, M/L^3 [lbs/gallon]
$\boldsymbol{k_n}$	Elastic coefficients in the normal direction, M/T^2 [N/m]
$\boldsymbol{k_t}$	Elastic coefficients in the tangential direction, M/T^2 [N/m]
$\boldsymbol{\delta_n}$	Overlap distances in the normal direction, L [m]
$\boldsymbol{\delta_t}$	Overlap distances in the tangential direction, L [m]
$\boldsymbol{\gamma_n}$	Viscoelastic damping coefficients in the normal direction, M/T [N·s/m]
$\boldsymbol{\gamma_t}$	Viscoelastic damping coefficients in the tangential direction, M/T [N·s/m]
$\boldsymbol{v_{rn}}$	Relative velocities in the normal direction, L/T [m/s]
$\boldsymbol{v_{rt}}$	Relative velocities in the tangential direction, L/T [m/s]
$\boldsymbol{\rho_f}$	Fluid density, M/L^3 [kg/m^3]
$\boldsymbol{\overrightarrow{u_f}}$	Fluid velocity, L/T [m/s]
$\boldsymbol{\vec{f}_{ij}^{c}}$	Force from particle **j** to particle **i**, M/T^2 [N/m]
$\boldsymbol{\vec{g}}$	Gravity constant, L/T^2 [m/s^2]

$\vec{f}_i^{f-p}$ Fluid to particle force acting on particle **i**, ML/T^2 [N]

$\vec{f}_{ij}^{c}$ Total contact force, ML/T^2 [N]

$\vec{f}_{ij}^{c}$ Contact force applied in the normal direction, ML/T^2 [N]

$\vec{v}_i$ Velocity of particle **i**, L/T [m/s]

d_p Particle diameter [m]

Dimensionless Variables:

CL_i Collection of particles that surrounds particle **i**

ε_f Volume fraction of fluid

k_v Number of particles in the corresponding fluid cell

R_{fp} Ratio of the fracture aperture to the proppant particle diameter

Re_i Ratio of the fracture aperture to the proppant particle diameter

X, C_d Drag coefficients of Di Felice model

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

1.1. Statement of the Problem

Unconventional resources (*i.e.*, tight oil and shale gas) play an ever-increasing role in the modern energy supply (Leimkuhler *et al.*, 2012). Hydraulic fracturing with low-viscosity fluids has become one of the most common techniques to overcome the difficulty in production from unconventional resources. Low-viscosity fluids (such as slick water) are chosen over high-viscosity fluids for hydraulic fracturing operations in the unconventional reservoir because of the very low permeability of formations, which reduces leakoff and reduces formation damage. Hydraulic fractures created using such low-viscosity fracturing fluids connect to pre-existing natural fractures (Beugelsdijk *et al.*, 2000) and create larger simulated reservoir volumes (Warpinski *et al.*, 2005).

Despite its advantages, hydraulic fracturing with low-viscosity fluids (*e.g.*, slick water) has its limitations. First, proppant particles carried by low-viscosity fluids have the tendency to settle to the bottom of a given hydraulic fracture and form immobile beds. As a result, two proppant transport mechanisms are expected in hydraulic fracturing operations with low-viscosity fluids — suspension transport and bed load transport. First, in the suspension transport regime, proppant particles are suspended in the carrier fluid and, under these conditions, the assumption that proppant particles and carrier fluid can be considered as one continuous phase is valid (Han *et al.*, 2016). In the bed load transport regime, however, most of the proppant particles settle down and form an immobile bed, while only a thin layer of proppant particles on the top of the proppant bed can still be

transported (Kou, 2019). In such cases, the assumption of proppant and carrier fluid as a single continuous phase is no longer valid.

Second, during hydraulic fracturing, hydraulic fractures (HF) tend to propagate towards existing natural fractures (NFs), thus connecting NFs to the horizontal well. Because HFs usually approach NFs at an angle (Mighani *et al.*, 2018), the two-phase flow of solid and fluid at the angular HF-NF intersection are expected to have greater turbulence and stronger collisions between proppant particles and the fracture walls. Therefore, proppant particles are more likely to settle down and eventually block the entry at the HF-NF intersections; this is an event called proppant bridging (blocking). Bridging caused by the settling of proppant will hinder proppant entry into the NFs and result in a non-optimized proppant placement. Because most of the recent models represent proppant particle as a continuous fluid phase, the proppant bridging behavior cannot be captured and described by these simulations.

Third, the optimization of proppant transport during hydraulic fracturing using horizontal wells is very important but is also challenging. Existing proppant transport models assume the simple geometry of a single planar fracture to represent hydraulic fractures along horizontal wells, but the geometry of the subsurface fracture network induced by hydraulic fracturing is far more complex and usually involved multiple clusters of fractures connected to the wellbore. As an example, the spacing between perforation clusters is an important design parameter as it significantly influences the hydraulic fracture geometry, the drainage volume, the production rates, and the estimated ultimate recovery (EUR)

from a reservoir (Chen *et al.*, 2019). The assumption of a single planar fracture connected to the wellbore in most of the existing models cannot appropriately simulate the proppant transport in this situation. In addition to cluster spacing, the number of perforation clusters (Cheng, 2010), proppant size (Han *et al.*, 2016), and perforation width (Van Ketterij, 1999) are also significant parameters in the design of operations of optimized proppant distribution in multi-cluster HF horizontal wellbore system, and none of these can be adequately represented and simulated under the assumption of a single-fracture system.

1.2. Objectives

The objectives of this study are:

- To propose a well-documented, robust proppant bridging criterion using an artificial neural network (or ANN) in order to better understand the proppant bridging effect at the intersection of hydraulic and natural fractures.

- To investigate the proppant distribution in multi-cluster fracture system, and to develop an optimized stimulation approach that involves key design parameters (cluster spacing, number of clusters, proppant size, and perforation width) in order to achieve better proppant distribution(s) among the perforation clusters of a multi-fractured horizontal well.

2. SIMULATION METHODOLOGY

2.1. CFD-DEM Method

2.1.1. Overview

For a realistic representation and modeling of the proppant transport in systems with complex geometries, and to capture all the interactions between the proppant particles, the fracturing fluid and the boundaries (fracture walls and horizontal well walls), the CFD-DEM simulator of Kou (2020) was used for this work. More information on the CFD-DEM simulator is provided in the following sections. As mentioned earlier, this simulator couples the discrete element method (DEM) with a computational fluid dynamics (CFD) code. The resulting CFD-DEM model describing the behavior of coupled fluid-granular particle systems can simulate all the associated flow and transport processes, thus enabling the investigation of proppant transport problems that are currently intractable by most existing models.

The use of the CFD-DEM coupled simulator to model the proppant transport enables the tracking of individual or composite particles in a Lagrangian frame of reference, and can still describe the fluid phase flow in an Eulerian frame of reference. Therefore, it is possible to track the location of each individual proppant particle, the fluid and particle velocities, and all other properties as a function of time. The CFD-DEM model can also better capture the strong interactions (such as collisions and turbulence) between the proppant particles, the fracture fluid, and the boundaries at the HF-NF intersection. The description of various processes in the 3D proppant transport model is based on the

CFDEM®coupling software, which is an open source CFD-DEM engine that has already been validated and calibrated with experimental results (Kou *et al.*, 2019; Kou, 2020).

2.1.2. Proppant Particles Simulation in the Lagrangian Frame of Reference

The Discrete Element Method (DEM) was used to model the solid particle (proppant) transport in the coupled simulations of this study. The conservation of momentum of the Lagrangian particles is described by the following governing equation:

$$\boldsymbol{m_i}\frac{d\vec{v}_i}{dt} = \boldsymbol{m_i}\vec{\boldsymbol{g}} + \sum_{j\in CL_i}\vec{\boldsymbol{f}}_{ij}^{\,c} + \vec{\boldsymbol{f}}_{i}^{\,f-p} \quad \text{..... (2.1)}$$

where, $\boldsymbol{m_i}$ is the mass of particle $\boldsymbol{i}$ (kg); $\vec{\boldsymbol{v}}_{\boldsymbol{i}}$ is the velocity of particle $\boldsymbol{i}$ (m/s); $\boldsymbol{t}$ is the time (s); $\vec{\boldsymbol{f}}_{\boldsymbol{ij}}^{\,\boldsymbol{c}}$ is the force from particle $\boldsymbol{j}$ to particle $\boldsymbol{i}$ (N/m); $\vec{\boldsymbol{g}}$ is the gravitational constant (m/s^2); $\vec{\boldsymbol{f}}_{\boldsymbol{i}}^{\,\boldsymbol{f-p}}$ is the fluid particle force (N); and $\boldsymbol{CL_i}$ is the collection of particles that surrounds particle $\boldsymbol{i}$.

The three components on the right side of **Eq. 2.1** fully capture the motion of proppant particles. The body force (gravity) is represented by the term $\boldsymbol{m_i}\vec{\boldsymbol{g}}$, the particle-to-particle contact force is represented by the term $\vec{\boldsymbol{f}}_{\boldsymbol{ij}}^{\,\boldsymbol{c}}$, and the fluid-to-particle force is represented by $\vec{\boldsymbol{f}}_{\boldsymbol{i}}^{\,\boldsymbol{f-p}}$. The Hertz-Mindlin-Deresiewicz (H-MD) model that is used to describe the particle-particle contact force (Hertz, 1882; Mindlin and Deresiewicz, 1953 is described by the following equation:

$$\vec{\boldsymbol{f}}_{\boldsymbol{ij}}^{\,\boldsymbol{c}} = \vec{\boldsymbol{f}}_{\boldsymbol{ij}}^{\,\boldsymbol{n}} + \vec{\boldsymbol{f}}_{\boldsymbol{ij}}^{\,\boldsymbol{t}} = \left(\boldsymbol{k_n\delta_n}\vec{\boldsymbol{n}}_{\boldsymbol{ij}} - \boldsymbol{\gamma_n v_{rn}}\vec{\boldsymbol{n}}_{\boldsymbol{ij}}\right) + \left(\boldsymbol{k_t\delta_t}\vec{\boldsymbol{t}}_{\boldsymbol{ij}} - \boldsymbol{\gamma_t v_{rt}}\vec{\boldsymbol{t}}_{\boldsymbol{ij}}\right) \quad \text{..... (2.2)}$$

where $\vec{f}_{ij}^{c}$ is the total contact force (N); $\vec{f}_{ij}^{n}$ is the contact force applied in the normal direction (N); $\vec{f}_{ij}^{t}$ is the contact force applied tangentially (N); k_n, k_t are the elastic coefficients in the normal and the tangential directions, respectively (N/m); δ_n, δ_t are the overlap distances in the normal and the tangential directions, respectively (m); γ_n, γ_t are the viscoelastic damping coefficients in the normal and the tangential directions, respectively (N·s/m); and v_{rn}, v_{rt} are the relative velocities in the normal and the tangential directions, respectively (m/s). In the H-MD contact force model, the elastic coefficients in the normal and the tangential directions (k_n, k_t) conserve the kinetic energy, and the viscoelastic damping coefficients in the normal and the tangential directions (γ_n, γ_t) represented the energy dissipation due to particle collision.

The DEM simulator used in this study is the LIGGGHTS®-PUBLIC code (Goniva *et al.*, 2012). This is an open-source parallel DEM particle simulation software, distributed by DCS Computing GmbH, Linz, Austria. The name LIGGGHTS stands for "**L**AMMPS **I**mproved for **G**eneral **G**ranular and **G**ranular **H**eat **T**ransfer **S**imulations". LAMMPS is an acronym for the Large-scale Atomic/Molecular Massively Parallel Simulator, which is a classical molecular dynamics code developed for materials modeling (Plimpton, 1995). LIGGGHTS®-PUBLIC uses neighbor lists to keep track of nearby particles. The lists are optimized for systems with particles that are repulsive at short distances, so that the local density of particles never becomes too large. On parallel machines, LIGGGHTS®-PUBLIC uses spatial-decomposition techniques to partition the simulation domain into small 3D sub-domains, each assigned to a different processor. Processors communicate

and store "ghost" atom information for atoms that border their sub-domain (Kloss, 2016).

2.1.3. Fracture Fluid Simulation in the Eulerian Frame of Reference

Computational Fluid Dynamics (CFD) are used to model the carrier/fracturing fluid phase (slick water) in the simulations. The governing equations describing the conservation of fluid mass and fluid momentum are, respectively,

$$\frac{\partial(\rho_f \varepsilon_f)}{\partial t} + \nabla \cdot (\rho_f \overrightarrow{u_f}) = 0 \quad \text{..... (2.3)}$$

$$\frac{\partial(\rho_f \varepsilon_f \overrightarrow{u_f})}{\partial t} + \rho_f [\, \overrightarrow{u_f} \nabla \cdot (\varepsilon_f \overrightarrow{u_f}) \,] = -\nabla p + \mu_f \nabla^2 \overrightarrow{u_f} + \varepsilon_f \rho_f \vec{g} - \frac{1}{V_{cell}} \sum_{i=1}^{k_v} \overrightarrow{f_i}^{f-p} \quad \text{..... (2.4)}$$

where ρ_f is the fluid density (kg/m^3); ε_f is the volume fraction of fluid; $\overrightarrow{u_f}$ is the fluid velocity (m/s); t is the time (s); μ_f is the fluid viscosity (Pa.s); p is the pressure, (Pa); $\vec{g}$ is the gravitational constant (m/s^2); $\overrightarrow{f_i}^{f-p}$ is the fluid-to-particle force for particle **i** (N); k_v is the number of particles in the corresponding fluid cell; and V_{cell} is the volume of the fluid cell (m^3). The last term $-\frac{1}{V_{cell}} \sum_{i=1}^{k_v} \overrightarrow{f_i}^{f-p}$ is the summation of the external particle forces as a coupling term. The $\overrightarrow{f_i}^{f-p}$ term contains the fluid drag force, the pressure gradient force and the fluid shear force. This term is then reduced to the fluid drag force ($\overrightarrow{f_i}^{d}$) to minimize the fluid drag force the computational effort and memory required for coupling the CFD and the DEM simulations (Kou 2020). The fluid drag force is calculated by using the Di Felice model (Di Felice, 1994).

$$\overrightarrow{f_i}^{d} = 3\pi \mu_f d_p \hat{f}_i^{d} (\overrightarrow{u_f} - \overrightarrow{v_i}) \quad \text{..... (2.5)}$$

$$\widehat{f}_i^d = \frac{C_d}{24} Re_i \varepsilon_f^{-X} \quad (2.6)$$

$$X = 3.7 - 0.65 e^{-0.5*(1.5-log_{10} Re_i)^2} \quad (2.7)$$

$$C_d = \left(0.63 + 4.8 Re_i^{-0.5}\right)^2 \quad (2.8)$$

where, Re_i is the particle Reynolds number; $\overrightarrow{v_i}$ is the velocity of particle i (m/s); d_p is the particle diameter (m); ε_f is the volume fraction of the fluid; X and C_d is the dimensionless drag coefficients.

The CFD simulator used in this study is based on the OpenFOAM standard solver libraries (Weller *et al.*,1998). The OpenFOAM is C++ open-source toolbox for the development of customized numerical solvers, with pre- and post-processing utilities for the solution of continuum mechanics problems, most prominently problems of CFD (The OpenFOAM Foundation, 2011).

2.2. CFDEM®coupling Software

The CFDEM®coupling Software is used to couple the LIGGGHTS and OpenFOAM packages to enable the two-phase flow (particles and carrier/fracturing liquid) simulation in my study . First, OpenFOAM is compiled to generate a standard solver library and utility executables. Second, LIGGGHTS is compiled to generate the dynamic library. Finally, the CFDEM®coupling is compiled as a modified CFD solver. This process creates a modified OpenFOAM solver, that includes coupling factors and is capable of simulating the transport of proppant particles in a Lagrangian frame of reference and the flow of the fracturing fluid (slick water) in a Eulerian frame of reference.

2.3. 3D Visualization

The Paraview software (Ayachit, 2015) is used to visualize the simulation results from CFDEM®coupling simulator. This is a powerful open-source, multi-platform 3D plotting software. It simplifies the process of calculating the proppant distribution in multi-clusters system and provides a straight-forward way to determine the proppant bridging effect at fracture intersections. An example of 3D visualization result is shown in **Fig. 2.1**.

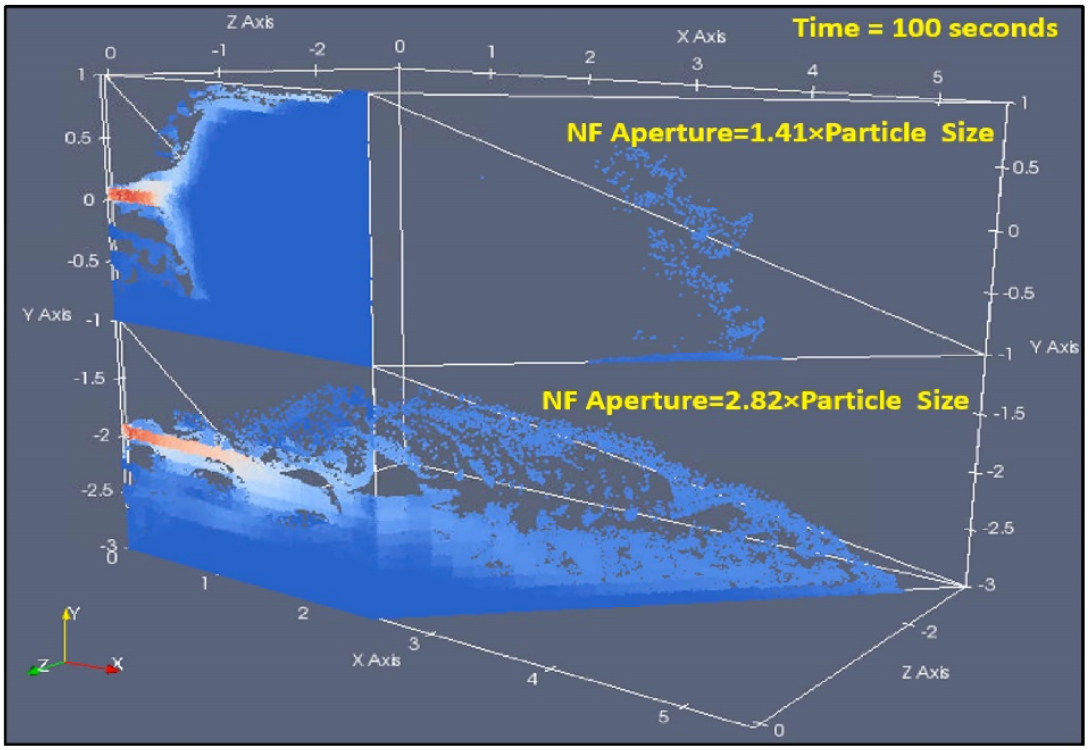

Figure 2.1 3D visualization of the simulation results from a study of field-scale proppant transport across HF-NF interface (Kou, 2020).

2.4. Scaling, Parallel Computing, and High-Performance Computing Resource

The ability to fully capture the proppant particle behavior in HF-NF networks using the CFD-DEM simulator requires significant computational effort in the study of field-scale problems, in which the number of particles tracked by DEM is on the order of millions (Sun, 2018). To reduce the time and complexity of these simulations, the concept of composite particles (Kou *et al.*, 2019; Kou, 2020) was used in this work. The parallel coupled simulations were performed on the Ada high-performance computing platform at the Texas A&M University Supercomputer Center.

Composite particles (also known as "coarse-grained" or CG particles) are introduced to replace and represent a large number of the original proppant particles in the simulator without loss of robustness in the physical representation of the system behavior. Thus, CG particles with appropriate scaling factors can accelerate the DEM simulations without compromising the accuracy of results (Kou, 2018). Taking advantage of the built-in parallel computing architecture of OpenFOAM, the project simulations were conducted in parallel on Ada using 10-15 nodes, which provides an execution speedup ranging between 5 and 8 over the execution times in serial computations for the solution of the same problems. The implementation of CG particles and the parallel computing on the Ada high-performance computer cluster enabled the simulation of the field-scale proppant transport problems involved in this study to be performed within a few days.

3. PROPPANT BRIDGING CRITERION AT THE HF-NF INTERFACE

This section focuses on the study of proppant transport at (and through) the Z-shaped intersection of a HF-NF interface. Because the propagation of a HF follows the path of least resistance, a HF typically propagates in the direction of minimum horizontal stress, which often involves advancing through a pre-existing NF. The direction of a pre-existing NF is not always vertical due to a combination of depositional, geodynamic, and tectonic processes. Therefore, there is no predetermined estimate of the magnitude of the interacting angle between a HF and a NF.

3.1. Simulations

Figure 3.1 shows a HF-NF intersection at 45^{o}. The results of the 3D proppant transport simulation in the domain corresponding to this geometry were visualized by ParaView and have been shown in **Fig. 3.1**. As discussed earlier, CFDEM®coupling is used to conduct the associated coupled fluid flow and proppant transport simulations. The fracture fluid and the proppants are injected at the middle of the left boundary (see **Fig. 3.1.b**), and gravitational effects are fully accounted for. Because the overburden stress is generally the largest stress at the depths of most shale oil and gas reservoirs, the HFs there tend to be vertical.

Kou (2020) conducted numerous simulation studies to assess the proppant bridging effect at HF-NF intersections in systems with the geometry shown in **Fig. 3.1**, and proposed a bridging criterion (correlation) based on a relatively limited data set. In this study, the population of the bridging data set was increased by conducting a systematic suite of

simulations that investigated the effects of various proppant sizes, proppant concentrations, fracture widths, and fluid injection velocities on the bridging behavior at the HF-NF intersection.

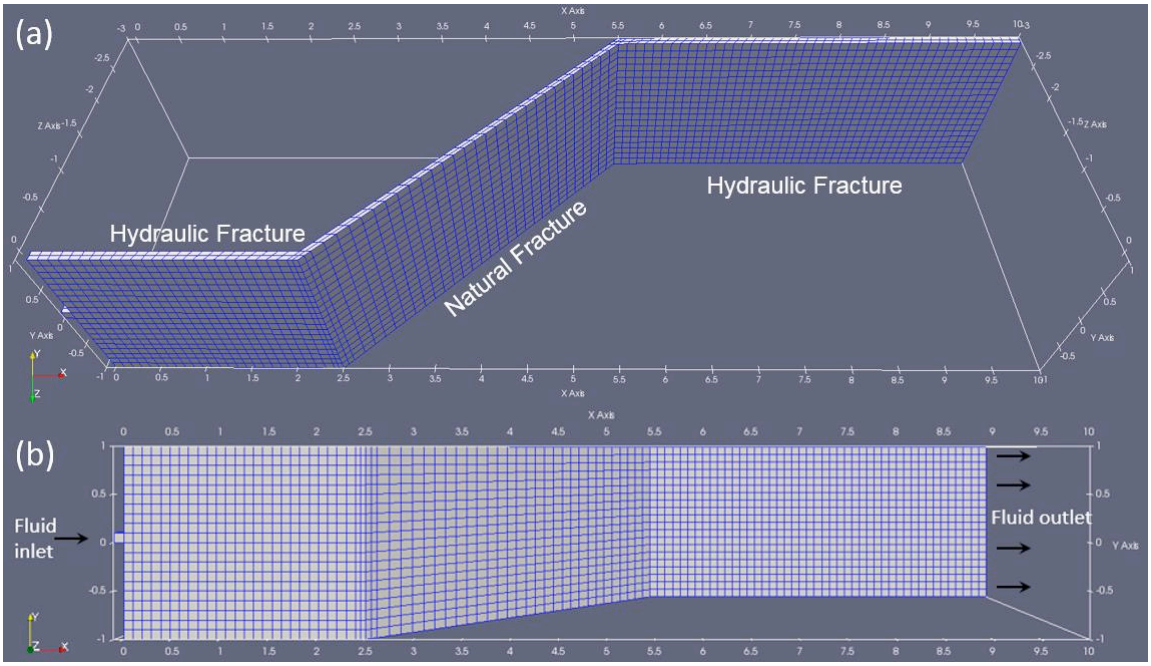

Figure 3.1 The domain used in the HF-NF intersection studies: (a) 3D view, and (b) side view of the Z-shape HF-NF simulation domain with boundary condition. (Kou, 2020).

To better visualize the particles, (a) the fracture walls and the fluid are both represented as "transparent" in the visualization of the system in **Figs. 2.1** and **3.1**, and (b) the proppant particles are colored based on the magnitude of their velocity. The red and blue color represent the limits of the velocity magnitude observed in this study, *i.e.*, a high velocity and low velocity, respectively. The injection of the fracturing fluid and the proppants at the injection box located at the mid-point of the left boundary of the simulation domain is clearly depicted by the red color of the (expected) highest velocities at this point.

In an effort to quantify the significant effects of the ratio of the fracture aperture to the proppant particle diameter R_{fp} and of the proppant concentration in the injection stream C_p on proppant bridging, the C_p vs R_{fp} relationship is plotted in **Fig. 3.2** using the results from a large number of simulation cases that attempted to cover a very wide range of possible C_p and R_{fp} values. This dataset consists of 22 cases from Kou (2020) and 48 new cases from this work that are listed in **Appendix A**. The occurrence of bridging was determined through visual inspection of the results (analogous to the work shown in **Fig. 2.1**). In **Fig. 3.2**, the red circles represent bridging cases, and the blue circles depict continuous proppant transportation through the HF-NF intersection without bridging. Based on these additional results, the bridging criterion function proposed by Kou (2020) is validated as it holds true for the expanded data set. The curve quantifying this bridging criterion is described by the relationship:

$$\boldsymbol{C_{p,max} = \frac{3}{4} R_{fp}^3 - \frac{1}{2}} \quad \text{.....(3.1)}$$

and is plotted using a dashed line in **Fig. 3.2**. The obvious conclusions to be drawn from **Fig. 3.2** and the proposed bridging criterion are that (a) the magnitude of R_{fp} is critical and (b) lowering C_p enhances proppant transport through the HF-NF interface.

3.2. Artificial Neural Network (ANN)

Although the proposed bridging criterion function in **Eq. 3.1** provides a better understanding of the potential for proppant bridging at HF-NF intersections, this correlation is constructed based on the analysis of relatively few model cases. To better capture the nature of this correlation, this work requires a non-parametric correlation to objectively interpret and predict this behavior.

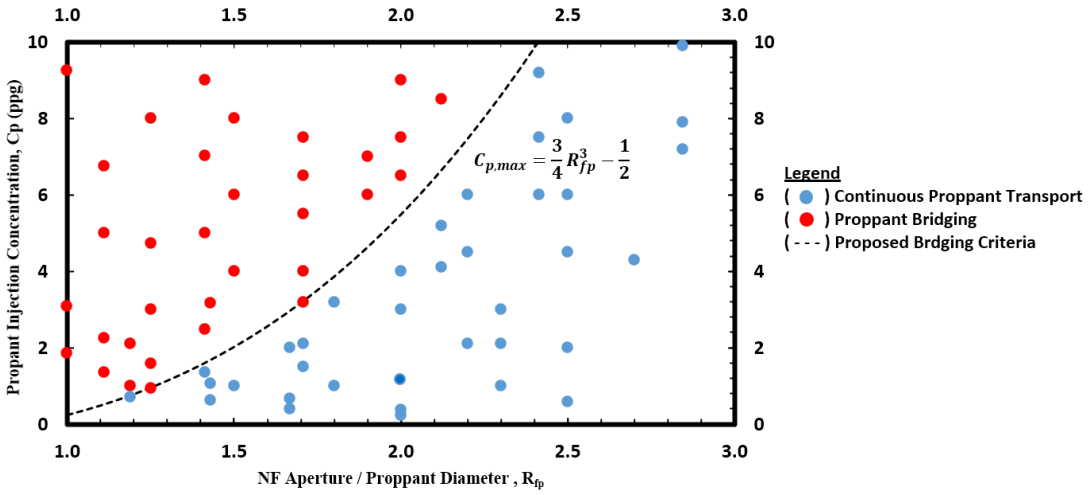

Figure 3.2 Plot of numerical simulation results and the proposed bridging criterion function (Kou, 2020).

The nature and benefit of an artificial neural network (ANN) as a "non-parametric" correlation may be the subject of arguments, but a properly calibrated ANN has the potential to yield sufficiently unbiased estimates, provided its training and validation are performed properly. The full MATLAB code used for the ANN training can be found in **Appendix B**. To reduce the possibility of an overfitting problem, I conducted a standard K-fold cross-validation and an additional 10 simulation results in order to evaluate the effectiveness of the trained ANN.

A 4-layer ANN was constructed by treating proppant transport through a HF-NF interface as an image-classification problem in deep machine learning (Higham, 2019). The ANN approach uses the C_p and R_{fp} values as inputs and provides as an output a prediction of the possibility of bridging. The training process requires completed simulation cases,

including inputs (C_p and R_{fp}) and outputs (bridging or continuous transport). The dataset displayed in **Fig. 3.2** that consists of 70 simulation cases is used to train the ANN, and is listed in **Appendix A**. The outputs of the simulations are represented as vectors in ANN, as will be shown later. Recall that ANN has internal correlations that are not revealed to the user, and the goal of training is to correlate the C_p and R_{fp} values with the bridging criteria. Once the ANN is trained, the expectation is that the ANN can predict the bridging criteria based on an independent (*i.e.*, new) set of inputs (C_p and R_{fp}). The goal of this approach is to provide an objective (unbiased) and reliable correlation.

The artificial neural network is based on the following sigmoid function.

$$\boldsymbol{\sigma}(\vec{\boldsymbol{x}}) = \frac{\mathbf{1}}{\mathbf{1}+e^{-\vec{x}}}, \quad (3.2)$$

where $\boldsymbol{\sigma}$ is the sigmoid function and $\vec{\boldsymbol{x}}$ is the input vector. **Eq. 3.2** is modified by using a weights matrix ($\overrightarrow{\boldsymbol{W}}$) and a bias vector ($\vec{\boldsymbol{b}}$), as shown in **Eq. 3.3**.

$$\boldsymbol{\sigma}(\overrightarrow{\boldsymbol{W}} \bullet \vec{\boldsymbol{x}} + \vec{\boldsymbol{b}}) = \frac{\mathbf{1}}{\mathbf{1}+e^{-\overrightarrow{\boldsymbol{W}}\bullet\vec{\boldsymbol{x}}+\vec{\boldsymbol{b}}}} \in \boldsymbol{R^2} \quad (3.3)$$

The weights and bias vectors become parameters of this predictive model. With 4 layers of networks, the final form of this predictive model is:

$$\boldsymbol{F}(\vec{\boldsymbol{x}}) = \boldsymbol{\sigma}(\overrightarrow{\boldsymbol{W}}^{[4]} \bullet (\boldsymbol{\sigma}(\overrightarrow{\boldsymbol{W}}^{[3]} \bullet (\boldsymbol{\sigma}(\overrightarrow{\boldsymbol{W}}^{[2]} \bullet \vec{\boldsymbol{x}} + \vec{\boldsymbol{b}}^{[2]}) + \vec{\boldsymbol{b}}^{[3]}) + \vec{\boldsymbol{b}}^{[4]}))) \in \boldsymbol{R^2} \quad (3.4)$$

in which $\overrightarrow{\boldsymbol{W}}^{[i]}$ is the weights vector for layer $\boldsymbol{i}$, and $\vec{\boldsymbol{b}}^{[i]}$ is the bias vector for layer $\boldsymbol{i}$. The dimensions of all the inputs, outputs, and parameters are shown in **Table 3.1**.

Table 3.1 Dimensionality of inputs, outputs, and parameters of neural network.

Dimension	$\mathbb{R}^{2x1}$	$\mathbb{R}^{2x2}$	$\mathbb{R}^{3x2}$	$\mathbb{R}^{3x1}$	$\mathbb{R}^{2x3}$
Variables	$\vec{\boldsymbol{x}}, \vec{\boldsymbol{F}}, \vec{\boldsymbol{b}}^{[2]}, \vec{\boldsymbol{b}}^{[4]}$	$\vec{\boldsymbol{W}}^{[2]}$	$\vec{\boldsymbol{W}}^{[3]}$	$\vec{\boldsymbol{b}}^{[3]}$	$\vec{\boldsymbol{W}}^{[4]}$

In **Table 3.1**, $\vec{\boldsymbol{x}}$ is the input vector with 2 entries: C_p and R_{fp}. We divide the outputs into two categories: proppant bridging and continuous transport. Recalling that the network is based on a sigmoid function, the outputs are then represented in ANN as

$$y(\vec{\boldsymbol{x}}^i) = \begin{cases} \begin{bmatrix}1\\0\end{bmatrix} & if\ in \text{ proppant bridging category} \\ \begin{bmatrix}0\\1\end{bmatrix} & if\ in \text{ continuous transport category} \end{cases} \quad \text{..... (3.5)}$$

The cost function,

$$Cost = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\left\|y(\vec{\boldsymbol{x}}^i) - \boldsymbol{F}(\vec{\boldsymbol{x}}^i)\right\|_2^2 \quad \text{..... (3.6)}$$

measures the discrepancy between ANN outputs and target outputs by averaging the squared Euclidean norm over the data points, which is often referred to as a quadratic cost function. Recall that only the weights ($\vec{\boldsymbol{W}}$) and the bias ($\vec{\boldsymbol{b}}$) are modified, but not the data inputs ($\vec{\boldsymbol{x}}$). For reference, there are 23 parameters in this ANN: entries in the weight matrices and bias vectors (recall that these are internal to the ANN). In the terminology of optimization, the cost function in **Eq. 3.6** is the objective function.

To fully train the artificial neural network, the parameters are modified to minimize the cost function. To perform this optimization, the stochastic gradient descent and back

propagation methods are implemented by using the nonlinear least-squares solver *lsqnonlin* in the MATLAB optimization toolbox.

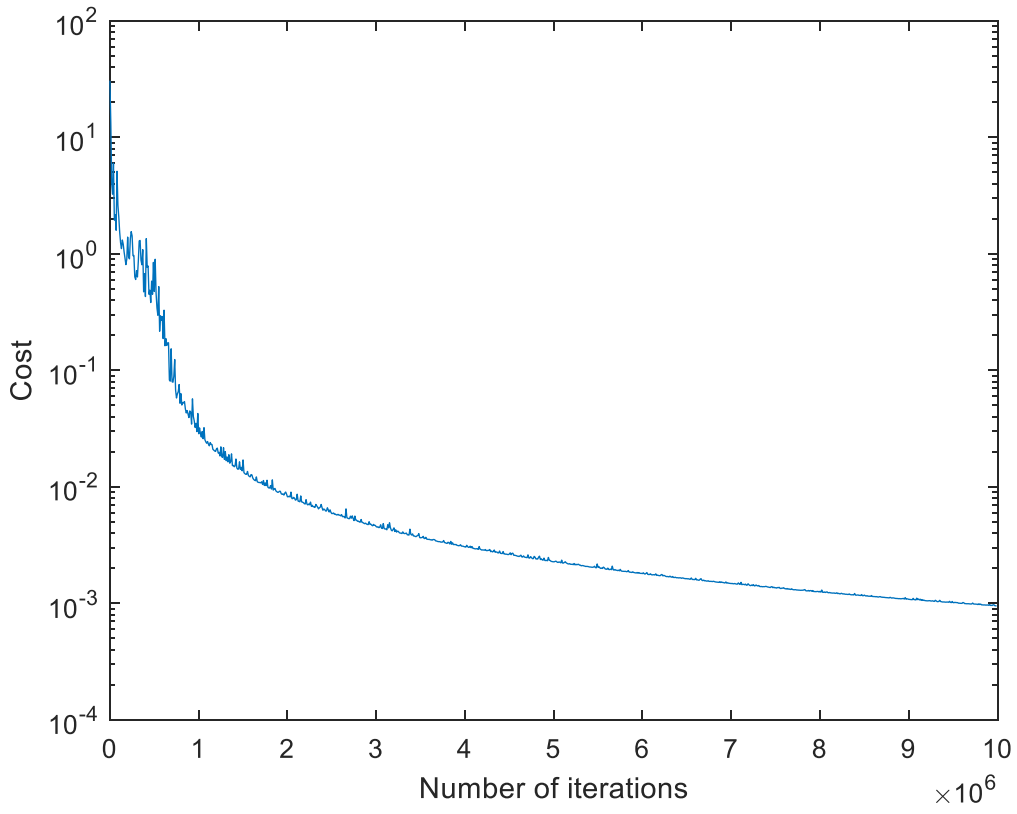

Figure 3.3 Plot of cost function value vs the number of iterations.

In the ANN optimization process, the cost function value decreases exponentially with an increasing number of iterations, as shown in **Fig. 3.3**. At the end of training (10^7 iterations for this work), the cost function value is below the defined tolerance of 10^{-3}, which means that the network is reasonably-well trained and appears to match the simulation results quite well.

As mentioned earlier, to reduce the possibility of "overfitting," a K-fold cross-validation

was implemented and an additional 10 simulation results were obtained to evaluate the effectiveness of the trained ANN. In this study, a K value of 10 is selected as a common choice that usually results in an ANN skill estimate with low bias and a modest variance. In **Fig. 3.4**, the steps to perform 10-fold cross-validation of the ANN are illustrated.

Figure 3.4 Illustration of 10-fold cross-validation of the ANN.

The original dataset consists of 70 cases: 33 proppant bridging cases and 37 continuous proppant transport cases. The dataset was first shuffled randomly. The shuffled dataset was subsequently divided into 10 equally sized subsets. Then, for each ANN, 10 subsets were separated into a training dataset (involving a majority fraction of the data — *i.e.*, 9 of the 10 subsets in our case) that is used to train an ANN, and a test dataset (the remaining data— *i.e.*, 1 of the 10 subsets in our case) that is used to evaluate the performance of the trained ANN. This process is repeated 10 times, so that all the data have been included once in the test dataset.

The skill/effectiveness of the ANNs is summarized in **Table 3.2**, which lists the cost function value of the ANN (Cost), the average and the maximum difference between the ANN predictions and the test dataset (Test_Error_Avg and Test_Error_Max, respectively), as well as the standard deviation of the difference between the ANN predictions and the test dataset (Test_Error_StdDev). All the ANNs fit their training dataset very well, as all the Cost values are on the order of magnitude of -3 or smaller. ANNs 3, 6, and 8 have a problem matching their corresponding test datasets, resulting in large Test_Error_Avg, Test_Error_Max, and Test_Error_StdDev. Therefore, based on the performance of these trained ANNs in matching their corresponding test datasets, ANNs #3, #6, and #8 were eliminated from further evaluation.

Table 3.2 Evaluation of the ANNs from K-fold cross-validation with K=10 using the corresponding test dataset.

ANN#	Cost	Test_Error_ Avg	Test_Error_ Max	Test_Error_ StdDev
1	1.49E-03	5.13E-05	1.18E-04	6.16E-05
2	1.05E-03	2.46E-06	5.37E-06	1.91E-06
3	9.35E-04	2.02E-01	1.41E+00	5.35E-01
4	1.07E-03	1.32E-04	4.59E-04	1.98E-04
5	1.07E-03	6.81E-04	1.51E-03	6.24E-04
6	5.46E-04	2.07E-01	1.32E+00	4.92E-01
7	1.05E-03	1.65E-04	3.84E-04	1.84E-04
8	4.25E-04	4.50E-01	1.41E+00	6.54E-01
9	1.20E-03	1.25E-04	2.90E-04	8.58E-05
10	1.33E-03	2.46E-04	3.51E-04	1.68E-04

In the second part of the ANN evaluation, the remaining 7 ANNs were ranked based on the Test_Error_Avg values. Then, 10 additional proppant simulations were conducted and grouped into a new dataset. The specifics of these 10 simulations are listed in **Appendix A**. In a manner analogous to the previous evaluation of the ANNs in **Table 3.2**, the skill of the remaining 7 ANN is summarized in **Table 3.3**, using as evaluation criteria the average and maximum difference between ANN predictions and the new dataset (New_Error_Avg and New_Error_Max, respectively), in addition to the standard deviation of the difference between the ANN predictions and the new dataset (New_Error_StdDev). All the remaining ANNs match the new dataset very well, with the order of magnitude of the New_Error_Avg estimates being equal to, or smaller than, -4. Among the remaining ANNs, ANN#9 is deemed to be the best ANN, as it is associated with the lowest values of New_Error_Avg, New_Error_Max, and New_Error_StdDev.

Table 3.3 Evaluation of the ANNs from K-fold cross-validation with K=10 using the new dataset.

ANN#	New_Error_ Avg	New_Error_ Max	New_Error_ StdDev
2	9.98E-05	4.08E-04	1.66E-04
1	3.28E-04	1.14E-03	4.14E-04
9	8.60E-05	1.55E-04	5.00E-05
4	1.02E-04	4.16E-04	1.69E-04
7	1.00E-04	4.12E-04	1.66E-04
10	5.84E-04	1.55E-03	5.51E-04
5	3.28E-04	1.14E-03	4.14E-04

Table 3.4 Parameters of ANN#9 from K-fold cross-validation with K=10

$\vec{W}^{[2]}$	$\vec{W}^{[3]}$	$\vec{W}^{[4]}$	$\vec{b}^{[2]}$	$\vec{b}^{[3]}$	$\vec{b}^{[4]}$
$\begin{pmatrix} -20.21 & 8.16 \\ -11.81 & 1.18 \end{pmatrix}$	$\begin{pmatrix} -10.74 & -9.17 \\ 10.33 & 9.17 \\ 0.019 & -0.71 \end{pmatrix}$	$\begin{pmatrix} -8.74 & 12.13 & 1.21 \\ 8.73 & -12.14 & -1.39 \end{pmatrix}$	$\begin{pmatrix} 17.67 \\ 15.90 \end{pmatrix}$	$\begin{pmatrix} 13.04 \\ -12.78 \\ -4.68 \end{pmatrix}$	$\begin{pmatrix} -2.40 \\ 2.41 \end{pmatrix}$

Finally, to compare the best ANN predictions with the 70 simulation data points, **Fig. 3.5** is shaded based on the predictions from ANN#9. The red shading denotes ANN#9 predictions of proppant bridging, while the blue shading indicates ANN#9 predictions of continuous proppant transport. The validity of the ANN#9 predictions is confirmed by the delineation between the two differently-shaded regions, which is in good agreement with the parametric curve of **Eq. 3.1** (see **Fig. 3.2**). The parameters for ANN#9 are listed in **Table 3.4** and in **Fig. 3.5** the validity of the ANN#9 predictions is confirmed by the delineation between the two differently-shaded regions, which is in good agreement with the parametric curve of **Eq. 3.1**. The proposed bridging criterion by Kou (2020) is not

only able to capture all the bridging cases in the expanded populated dataset that comprises of 70 simulation cases, but is also confirmed by a well-trained artificial neural network. Therefore, the proposed bridging criterion (**Eq. 3.1)** can provide a robust and reliable proppant bridging criterion at the HF-NF interface

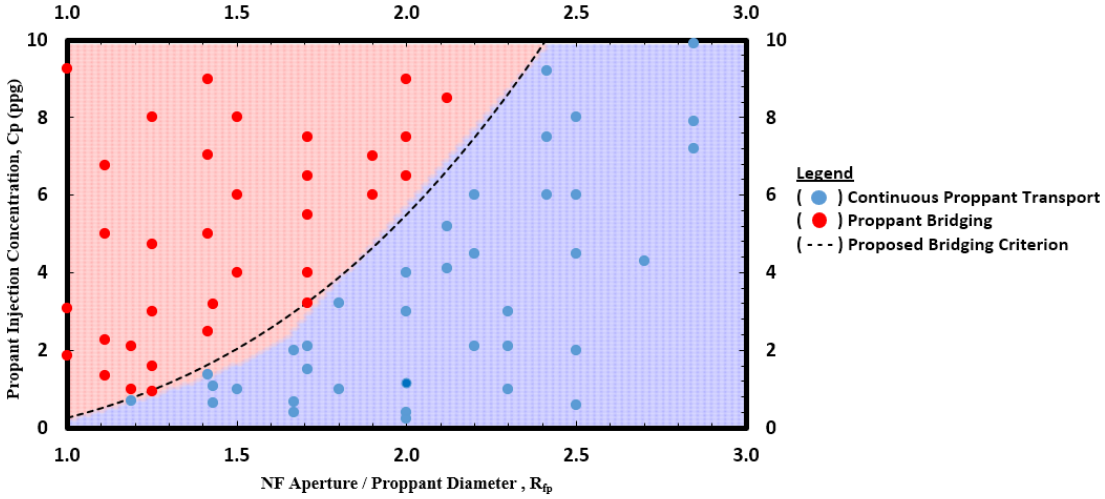

Figure 3.5 Plot of the ANN results, simulation results, and proposed bridging criterion.

4. OPTIMIZATION OF THE PROPPANT DISTRIBUTION IN A MULTI-CLUSTER HORIZONTAL WELL SYSTEM

The previous work of Kou (2020) studied the effect of injection rate on the proppant distribution; the present study focuses on the analysis of the sensitivity of proppant transport and placement to the four most important parameters for the design of fracturing operation — which are: cluster spacing, number of clusters, proppant size, and perforation width.

The base case investigated involves a 40m-long multi-cluster horizontal well system with a casing plug in the right end, an injection point in the left end, an injection velocity of 19m/s, a cluster spacing of 10m, 3 vertical planar fracture clusters, a proppant size/diameter of 0.5mm, and a perforation width 0.1m. This base case was selected because it shows the most even distribution of the proppant among the three clusters proppant distribution investigated in the Kou (2020) study. **Fig. 4.1** shows a visualization of the base case. There are three clusters: the left cluster colored in red, the middle cluster colored in blue, and the right cluster colored in grey. In **Appendix C**, the mesh files are provided for the 5-cluster case that were used to build the corresponding model for the simulation.

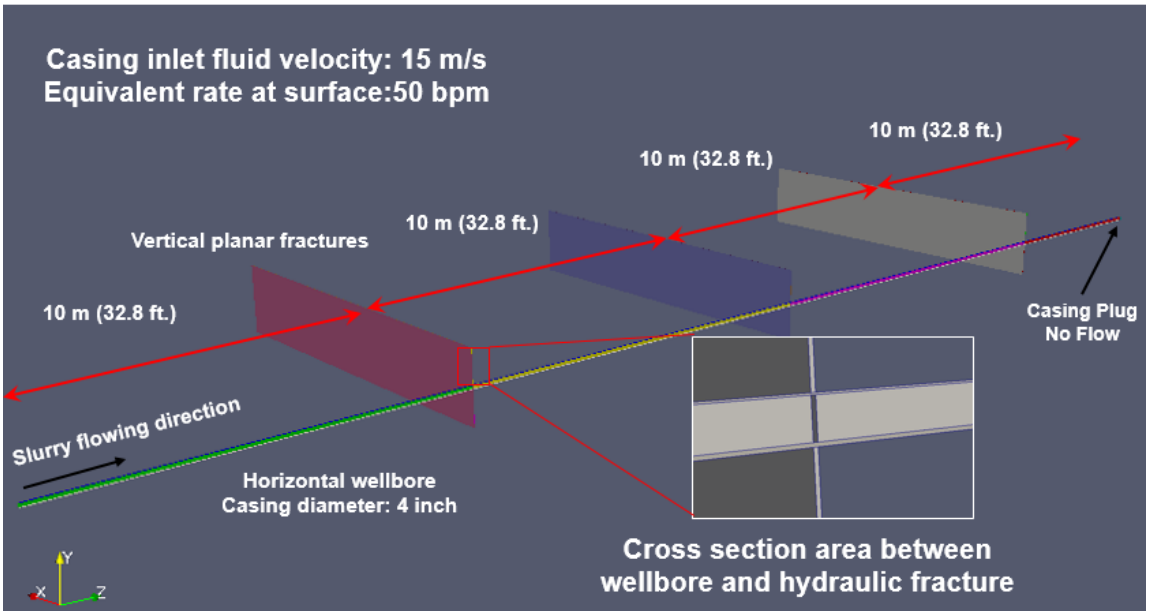

Figure 4.1 Visualization of the multi-cluster horizontal wellbore-fracture model, with the cross-section area illustrated. (Kou, 2020)

4.1. Cluster Spacing

To begin, a sensitivity analysis was conducted to relate proppant transport and placement to cluster spacing. In addition to the base case that involves a 10m cluster spacing, two more cases were investigated: a 5m cluster spacing and a 15m cluster spacing. To evaluate the effect of cluster spacing on the proppant distribution, the percentage of proppant entering the various clusters is plotted in **Fig. 4.2** against the distance from the injection point along the horizontal wellbore.

Because the cluster spacing changes about the location of the middle cluster, the clusters

to the left and right of the middle cluster in the new cases are located differently from those in the base case and, therefore, lead to the horizontal shifting of the proppant distribution as shown in **Fig. 4.2**. The increase in the cluster spacing (5m-10m-15m) significantly increases the percentage of the proppant entering the left and middle clusters (the first ones encountered by the injected fracturing fluid and proppant), but decreases the percentage of the proppant entering the right cluster.

This behavior is explained by the interference among the flows into each cluster. As the cluster spacing increases, the interference effect is reduced, thus allowing more fluid and proppant particles to enter each cluster. However, with the total amount of the proppant particles being fixed, the amount of fluid and proppant particles that reaches the right (last) cluster decreases, which results (as expected) in a decrease in the percentage of the proppant particles entering the right cluster.

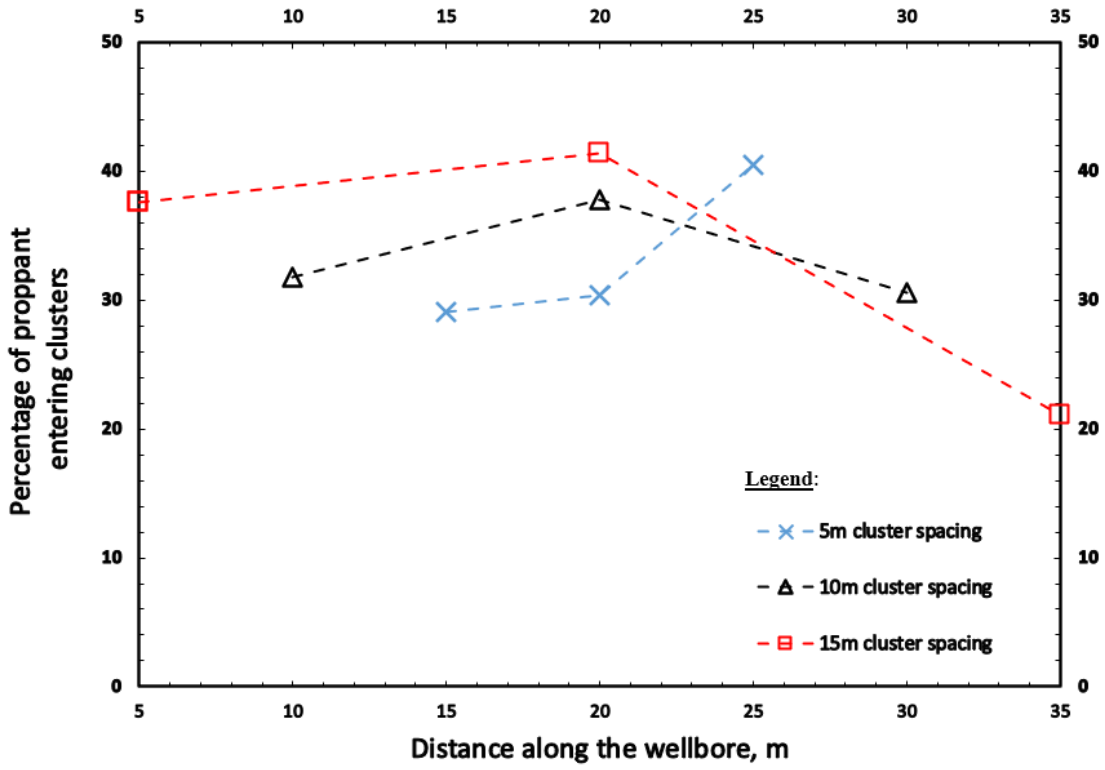

Figure 4.2 Effect of cluster spacing on the proppant distribution in a multi-cluster horizontal well system.

4.2. Number of Clusters

The second part of this investigation involved an analysis of the sensitivity of proppant transport and placement to the number of clusters. In addition to the base case that involved a total of 3 clusters, two more cases involving 2 and 5 clusters were investigated. The 2-cluster case was created by removing the middle cluster from the base case; the 5-cluster case was created by adding a new identical fracture cluster between the clusters in the base case (*i.e.*, the 3-cluster case). This configuration ensures that the distance between

(a) the injection point to the fracture clusters and (b) the casing plug (no-flow boundary) to the fracture clusters remains unchanged, which serves to minimize the influence of changes in the number of clusters on the other parameters.

The percentage of proppant entering the clusters is plotted in **Fig. 4.3** against the distance from the injection point along the horizontal wellbore. Compared to the base case, the decrease in the cluster number in the 2-cluster case leads to (a) a small decrease in the percentage of proppant entering the left cluster, but also (b) to a substantial increase in the percentage of the proppant entering the right (last in the direction of flow) cluster, *i.e.*, from 30.5% to 74.1%. This is caused by the removal of the middle cluster, which appears to enable the proppant particles that would have entered the middle cluster now available to enter the right cluster, thus resulting in the significant increase in the percentage of the proppant that enters the right cluster.

Compared to the base case, the percentage of the proppant entering each cluster in the 5-cluster case is consistently lower. This is attributed to the increase in the total fracture volume and to the decrease in the cluster spacing in the 5-cluster system. Because of the larger total fracture volume and the fixed fluid/proppant particle injection rate, each cluster receives less fluid and fewer proppant particles, as **Fig. 4.3** clearly shows. Another observation is that the proppant distribution among clusters exhibits significant differences in the 2-cluster case, but becomes almost uniform in 5-cluster case. This indicates that, the larger the number of clusters, the more uniform the proppant distribution.

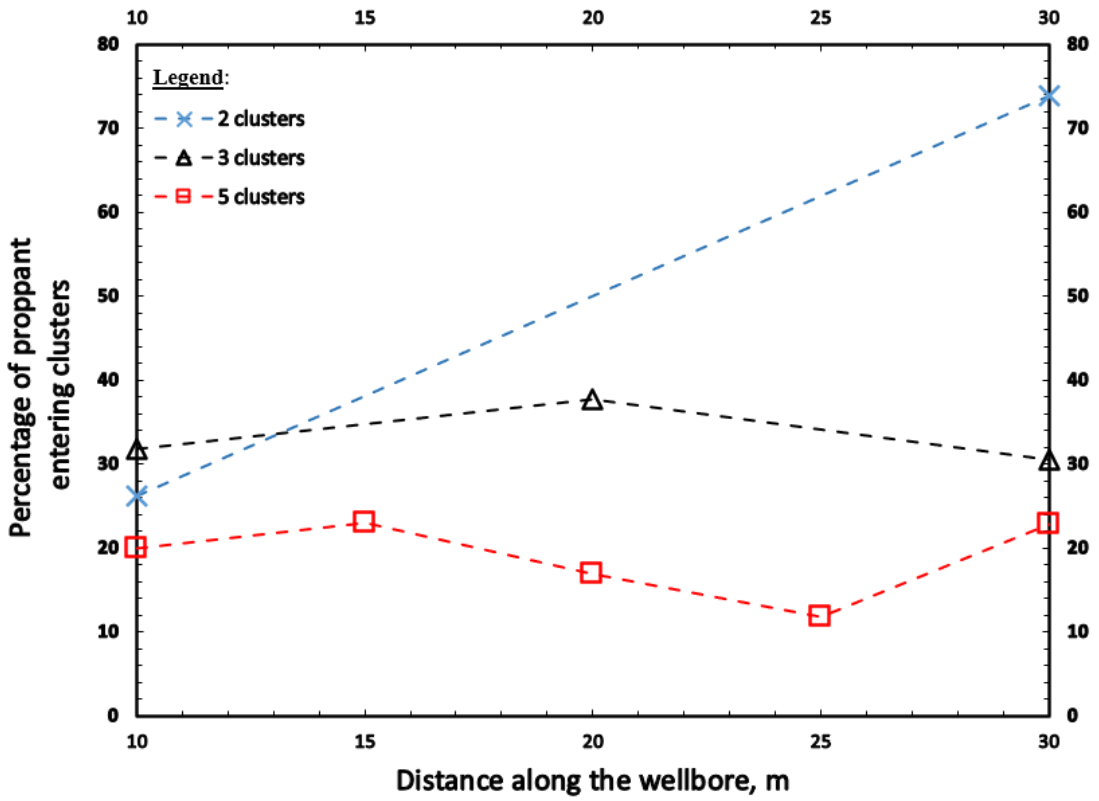

Figure 4.3 Proppant distribution in the multi-cluster horizontal well system with different numbers of clusters per stage.

4.3. Proppant Size

The sensitivity of proppant transport and placement to the proppant size was also investigated. In addition to the 0.5mm proppant size in the base case, two more cases involving proppant sizes of 0.4mm and 0.6mm were studied.

Fig. 4.4 shows the effect of the proppant size on the percentage of proppant entering the various clusters as a function of the cluster distance from the injection point. A larger proppant size (0.4mm-0.5mm-0.6mm) is associated with (a) larger proppant percentages

entering the first two (left and middle) clusters that the injectates encounter, and (b) a decrease in the amount of the proppant entering the right cluster.

This behavior is attributed to the decrease in the "proppant inertia" at the last cluster. The inertia for a uniform solid sphere is described by the following equation

$$Inertia = \frac{8}{15}\rho\pi R^5 \quad \text{.....(4.1)}$$

where ρ is the density of the sphere, and R is the radius of the sphere. **Eq. 4.1** clearly shows the strong dependence of inertia on the particle size (particle size is raised to the fifth power) and indicates that, for a given particle density, a smaller proppant size is associated with a significantly reduced inertia. This significantly facilitates the ability of a proppant particle to change the direction of its motion and to enter the fracture clusters, which intersect the horizontal well at an angle that is often as sharp as 90 degrees Consequently, more proppant particles can enter the first cluster(s) encountered (*i.e.*, the left and middle clusters), leaving fewer proppant particles available to enter the last (right) cluster.

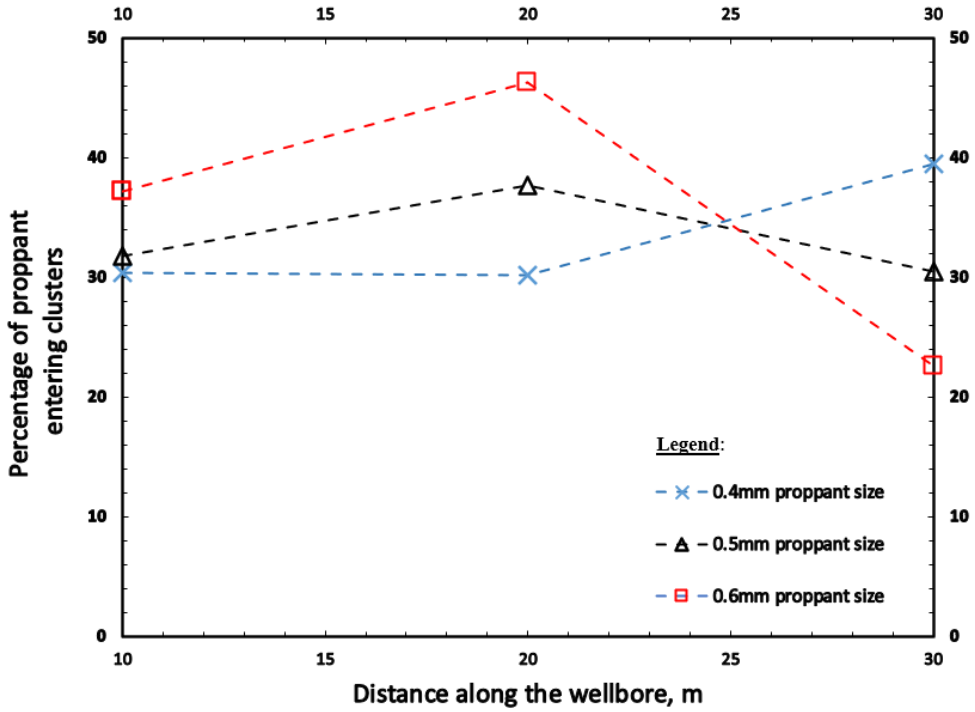

Figure 4.4 Proppant distribution in the multi-cluster horizontal well system with different proppant sizes.

4.4. Perforation Width

Finally, the sensitivity of proppant transport to a measure of the Cumulative Perforation Width (CPW) (*i.e.*, diameter) of a cluster was considered. Recall that, because of the complexity of the near-wellbore conditions, the simulation model assumes that one single hydraulic fracture will grow for each fracture cluster in order to reduce the computational cost to an acceptable level (Kou, 2020). Thus, the CPW quantity used in this study is the summation of the perforation diameters of all the potential perforations that could be available for the cluster. This study included the 0.1 m CPW as the base case and two

more cases with CPWs of 0.05 m and 0.2 m.

Fig. 4.5 shows the effect of the CPW on proppant transport in plots describing the percentage of proppant entering the various clusters as a function of their distance from the injection point. **Fig. 4.5** indicates that an increase in the CPW from 0.05m to 0.1m to 0.2m leads to a corresponding increase in the fraction of the proppant entering the left cluster (first cluster in the injectate pathway) and the middle cluster, and a decrease in the amount of proppant entering the right (last) cluster. This behavior is caused by the changes at the perforation openings, where the proppant particles enter the clusters. With the injection velocity remaining unchanged, an increase in the perforation width leads to an increase in the amount of the proppant particles entering the cluster. This results in an increase in the fraction of the proppant entering the left (first) and middle (second) clusters in the direction of flow. Because the total amount of proppant injection is fixed, the larger the number of proppant particles entering the first two clusters, the less the amount of proppant particles remaining to enter the right cluster, yielding the relative distributions of **Fig. 4.5**.

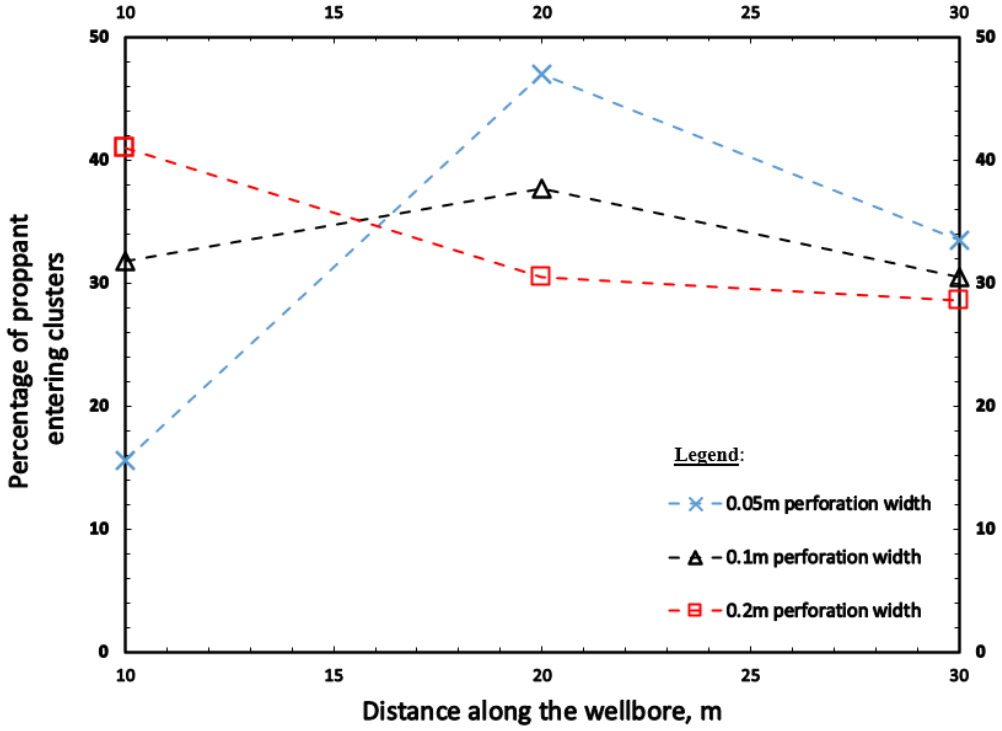

Figure 4.5 Proppant distribution in the multi-cluster horizontal well system with different perforation widths (diameters).

5. SUMMARY

This study focuses on the modeling and analysis of the proppant distributions in multi-cluster hydraulic fracture (HF) and natural fracture (NF) networks. The CFDEM®coupling simulator is used to couple (a) the transport of the proppant particles described by the Discrete Element Method (DEM) with (b) the flow behavior of the fracturing fluid modeled by a Computational Fluid Dynamics (CFD) code. Compared to most currently available simulators, this coupled simulator can track individual or composite particles in a Lagrangian frame of reference and still describe the fluid phase flow in a Eulerian frame of reference. This formulation makes possible to fully capture the proppant particle activities and interactions in the narrow and complex HF-NF network(s).

The first part of the study involves the development of a robust proppant bridging criterion at the HF-NF interface. The earlier data set of Kou (2020) that had been used for the development of a correlation (equation) based bridging criterion was significantly augmented with many additional simulation cases. The older results, combined with the newer simulation results from this study, were used with an image classification algorithm to yield a reliable proppant bridging criterion obtained from a well-trained artificial neural network (ANN).

The ANN was subjected to a standard K-fold cross-validation based on a dataset consisting of 70 simulation cases. The proposed ANN was subsequently tested against 10

new simulation cases to further evaluate its accuracy and efficiency against proppant transport cases outside of the initial dataset that had been used for the ANN training. The well-trained ANN confirmed the proppant bridging criterion originally proposed by Kou (2020) with a high overall performance score, and showed that it can be used as an alternate, robust correlation approach. The simulation results and the ANN predictions shown in **Fig. 3.5** illustrate the high performance of the ANN in the prediction of proppant bridging at the HF-NF interface and its consistency with the Kou (2020) predictions shown in **Fig. 4.2**.

The second part of the study focuses on the analysis of the proppant distribution in a multi-cluster horizontal well system, the results of which can be used to optimize proppant placement and to provide increased flowrates and improved recovery. The cumulative proppant distributions for each cluster were computed based on the number of representative particles entering each cluster during each timestep. We then evaluated the sensitivity of proppant transport and placement to the following critical system (and fracturing operation design) parameters — the cluster spacing, number of clusters, proppant size, and the cumulative perforation width (CPW).

6. CONCLUSIONS

Based on the results of this work, the following conclusions can be drawn regarding completion optimization strategies for horizontal well with multiple hydraulic fracture stimulation treatments:

- An increase in the cluster spacing, the proppant size, and/or the CPW increases the percentage of the proppant entering the clusters closest to the injection point. Inevitably, this also leads to a decrease in the fraction of the proppant entering the clusters further away from the injection point along the horizontal well.
- The effect of the number of clusters on the proppant distribution is more complicated. In general, the increase in the number of clusters is associated with a decrease in the average percentage of proppant entering each cluster because of the larger total fracture volume and the fixed proppant inject rate. However, with more clusters in each horizontal wellbore stage, proppant distribution becomes more uniform, as shown in **Fig. 4.3**.

REFERENCES

Ayachit, U. 2015. The ParaView Guide: A Parallel Visualization Application. Kitware, New York. ISBN:978-1-930934-30-6.

Beugelsdijk, L.J.L., de Pater, C.J., and K. Sato. 2000. Experimental Hydraulic Fracture Propagation in a Multi-Fractured Medium. Paper presented at the SPE Asia Pacific Conference on Integrated Modelling for Asset Management, Yokohama, Japan, April. SPE-59419-MS. https://doi.org/10.2118/59419-MS.

Chen, R., Xue, X., Datta-Gupta, A., Yu, H., and Kalyanaraman, N. 2019. The Impact of Cluster Spacing on Multi-Fractured Well Performance. Paper presented at the SPE Liquids-Rich Basins Conference - North America, Odessa, Texas, USA, November. SPE-197103-MS. https://doi.org/10.2118/197103-MS.

Cheng, Y. 2010. Impacts of the Number of Perforation Clusters and Cluster Spacing on Production Performance of Horizontal Shale Gas Wells. Paper presented at the SPE Eastern Regional Meeting, Morgantown, West Virginia, USA, October. SPE-138843-MS. https://doi.org/10.2118/138843-MS.

Goniva, C., Kloss, C., Deen, N.G., Kuipers, J.A.M. and Pirker, S. 2012. Influence of rolling friction on single spout fluidized bed simulation. *Particuology*, **10** (5): 582-591. ISSN 1674-2001. https://doi.org/10.1016/j.partic.2012.05.002.

Han, J., Yuan, P., Huang, X. *et al.* 2016. Numerical Study of Proppant Transport in Complex Fracture Geometry. Paper presented at the SPE Low Perm Symposium,

Denver, Colorado, USA, May. SPE-180243-MS. https://doi.org/10.2118/180243-MS.

Hertz, H. 1882. On the contact of elastic solids. *J.reine und angewandte Mathematik* **92** (1882): 156-171.

Higham, C.F., and Higham, D. 2019. Deep learning: An introduction for applied mathematicians. *SIAM Rev.* **61** (4), 860–891. https://doi.org/10.1137/18M1165748.

Kloss, C. 2016. LIGGGHTS(R)-PUBLIC Documentation, Version 3.X, https://www.cfdem.com/media/DEM/docu/Manual.html (accessed 15 March 2020).

Kloss, C., Goniva, C., Hager, A., Amberger, S. *et al.* 2012. Models, algorithms and validation for opensource DEM and CFD-DEM. *Progress in Computational Fluid Dynamics* **12** (2/3): 140–152. https://doi.org/10.1504/PCFD.2012.047457.

Kou, R. 2020. Simulation of proppant transport in hydraulic and natural fractures using a coupled CFD-DEM Method. Ph.D. dissertation, Texas A&M University, College Station, Texas (July 2020). https://hdl.handle.net/1969.1/192592.

Kou, R., Moridis, G. J., and Blasingame, T. A. 2018. Analysis and Modeling of Proppant Transport in Inclined Hydraulic Fractures. Paper presented at the SPE Hydraulic Fracturing Technology Conference and Exhibition, The Woodlands, Texas, USA, January. SPE-189856-MS. https://doi.org/10.2118/189856-MS.

Kou, R., Moridis, G. J., and Blasingame, T. A. 2019. Bridging Criteria and Distribution Correlation for Proppant Transport in Primary and Secondary Fracture. Paper presented at the SPE Hydraulic Fracturing Technology Conference and Exhibition,

The Woodlands, Texas, USA, February. SPE-194319-MS. https://doi.org/10.2118/194319-MS.

Leimkuhler, J., and Leveille, G. 2012. Unconventional Resources. *The Way Ahead* **08** (2012): 27–28. SPE-0112-026-TWA. https://doi.org/10.2118/0112-026-TWA.

Mighani, S., Lockner, D. A., Kilgore, B. D. *et al.* 2018. Interaction Between Hydraulic Fracture and a Pre-existing Fracture under Triaxial Stress Paper presented at the SPE Hydraulic Fracturing Technology Conference and Exhibition, The Woodlands, Texas, USA, January 2018. SPE-19901-MS. https://doi.org/10.2118/189901-MS.

Mindlin, R.D., Deresiewicz, H. 1953. Elastic Spheres in Contact Under Varying Oblique Forces. In *Transactions of ASME*, Series E. Journal of Applied Mechanics **20**: 327–344.

Plimpton, S. 1995. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *Journal of Computational Physics* **117** (1): 1-19. ISSN 0021-9991. https://doi.org/10.1006/jcph.1995.1039.

Sun, Z., Tang, H., Espinoza, D.N., *et al.* Discrete element modeling of grain crushing and implications on reservoir compaction. *Journal of Petroleum Science and Engineering* **171** (2018): 431-439. ISSN 0920-410. https://doi.org/10.1016/j.petrol.2018.07.046.

The OpenFOAM Foundation. 2011. OpenFOAM v8 User Guide, https://cfd.direct/openfoam/user-guide/ (accessed 18 March 2020).

Van Ketterij, R.B., and de Pater, C.J. 1999. Impact of Perforations on Hydraulic Fracture

Tortuosity. *SPE Prod & Fac* **14** (1999): 117–130. SPE-56193-PA. https://doi.org/10.2118/56193-PA.

Warpinski, N., Kramm, R. C., Heinze, J. R. *et al.* 2005. Comparison of Single-and Dual-Array Microseismic Mapping Techniques in the Barnett Shale. Paper presented at the SPE Annual Technical Conference and Exhibition, Dallas, Texas, October. SPE-95568-MS. https://doi.org/10.2118/95568-MS.

Weller, H. G., Tabor, G., Jasak, H. *et al.* 1998. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics* **12** (1998): 620-632. https://doi.org/10.1063/1.168744.

APPENDIX A - PROPPANT BRIDGING SIMULATION CASES

Table A.1 R_{fp} and C_p, and of proppant transport bridging status.

(Original) 70 Simulation Cases						(New) 10 Simulation Cases		
R_{fp}	C_p	Proppant Bridging	R_{fp}	C_p	Proppant Bridging	R_{fp}	C_p	Proppant Bridging
2.30	1.00	NO	1.25	0.95	YES	2.00	7.00	YES
1.25	8.00	YES	1.80	3.20	NO	1.20	3.10	YES
1.11	6.75	YES	1.50	4.00	YES	1.40	2.10	YES
1.41	5.00	YES	1.80	1.00	NO	1.60	4.50	YES
1.67	2.00	NO	1.67	0.40	NO	1.80	6.20	YES
1.71	1.50	NO	2.84	7.90	NO	1.50	0.50	NO
1.43	0.64	NO	2.00	0.23	NO	2.20	5.00	NO
1.11	1.35	YES	1.11	5.00	YES	2.40	4.00	NO
1.00	9.26	YES	2.20	4.50	NO	2.60	7.00	NO
1.11	2.25	YES	2.50	6.00	NO	2.00	2.00	NO
1.71	4.00	YES	2.70	4.30	NO			
1.19	1.00	YES	1.25	1.58	YES			
1.50	1.00	NO	2.50	4.50	NO			
1.25	4.74	YES	1.90	7.00	YES			
1.71	2.10	NO	1.25	3.00	YES			
2.41	9.20	NO	1.41	1.36	NO			
2.00	0.39	NO	1.00	1.85	YES			
2.12	8.50	YES	2.50	8.00	NO			
1.43	1.06	NO	1.19	0.70	NO			
1.41	7.03	YES	1.71	3.20	YES			
2.00	4.00	NO	2.50	0.59	NO			
2.50	2.00	NO	1.50	6.00	YES			
1.41	9.00	YES	2.30	3.00	NO			
2.84	7.20	NO	1.50	8.00	YES			
2.00	7.50	YES	2.00	9.00	YES			
2.41	7.50	NO	2.20	6.00	NO			
1.19	2.10	YES	1.90	6.00	YES			
2.00	3.00	NO	2.12	4.10	NO			
1.67	0.67	NO	2.30	2.10	NO			
1.00	3.09	YES	1.71	7.50	YES			
2.20	2.10	NO	1.71	6.50	YES			
2.12	5.20	NO	1.41	2.49	YES			
2.41	6.00	NO	1.43	3.18	YES			
2.84	9.90	NO	2.00	1.16	NO			
2.00	6.50	YES	1.71	5.50	YES			

APPENDIX B - ANN MATLAB CODE

```
function netbp

%NETBP Uses backpropagation to train a network

%%%% DATA %%%%

% Manually input

N_blocked = 23;

N_continuous = 27;

N_cases = N_blocked + N_continuous;

% Read data set

Simulation_results = readtable('Simulation_results.txt');

% Store data in array

x1 = Simulation_results(1,:);

x2 = Simulation_results(2,:);

x1 = table2array(x1);

x2 = table2array(x2);

y = [ones(1,N_blocked) zeros(1,N_continuous); zeros(1,N_blocked)

ones(1,N_continuous)];

%%%% END %%%%

% Initialize weights and biases

rng(5000);

W2 = 0.5*randn(2,2); W3 = 0.5*randn(3,2); W4 = 0.5*randn(2,3);
```

```
b2 = 0.5*randn(2,1); b3 = 0.5*randn(3,1); b4 = 0.5*randn(2,1);

% Forward and Back propagate
eta = 0.05; % learning rate
Niter = 1e7; % number of SG iterations
savecost = zeros(Niter,1); % value of cost function at each iteration

for counter = 1:Niter
    k = randi(N_cases); % choose a training point at random
    x = [x1(k); x2(k)];
    % Forward pass
    a2 = activate(x,W2,b2);
    a3 = activate(a2,W3,b3);
    a4 = activate(a3,W4,b4);
    % Backward pass
    delta4 = a4.*(1-a4).*(a4-y(:,k));
    delta3 = a3.*(1-a3).*(W4'*delta4);
    delta2 = a2.*(1-a2).*(W3'*delta3);
    % Gradient step
    W2 = W2 - eta*delta2*x';
    W3 = W3 - eta*delta3*a2';
    W4 = W4 - eta*delta4*a3';
    b2 = b2 - eta*delta2;
```

```
    b3 = b3 - eta*delta3;
    b4 = b4 - eta*delta4;
    % Monitor progress
    newcost = cost(W2,W3,W4,b2,b3,b4); % display cost to screen
    savecost(counter) = newcost;
end

% Show decay of cost function
save costvec
figure
semilogy([1:1e4:Niter],savecost(1:1e4:Niter))

    function costval = cost(W2,W3,W4,b2,b3,b4)
        costvec = zeros(N_cases,1);
            for i = 1:N_cases
            x =[x1(i);x2(i)];
            a2 = activate(x,W2,b2);
            a3 = activate(a2,W3,b3);
            a4 = activate(a3,W4,b4);
            costvec(i) = norm(y(:,i) - a4,2);
            end
        costval = norm(costvec,2)^2;
    end % of nested function
```

```
% Plot the model prediction
   colorMap = [zeros(N_cases, 1), zeros(N_cases, 1), ones(N_cases,1)];

   for j = 1:N_cases
      x =[x1(j);x2(j)];
      a2 = activate(x,W2,b2);
      a3 = activate(a2,W3,b3);
      a4 = activate(a3,W4,b4);
      if norm([1;0]-a4,2) < 0.01
         colorMap(j, :) = [1,0,0]; % Red, blocked
      else
         colorMap(j, :) = [0,0,1]; % Blue, continuous
      end
   end

   figure
   scatter(x1,x2,24, colorMap, 'filled');

% Save Weights and Bias
   save('parameters.mat','b2','b3','b4','W2','W3','W4')
end

function y = activate(x,W,b)
```

```
%ACTIVATE Evaluates sigmoid function.
%
% x is the input vector, y is the output vector
% W contains the weights, b contains the shifts
%
% The ith component of y is activate((Wx+b)_i)
% where activate(z) = 1/(1+exp(-z))

y = 1./(1+exp(-(W*x+b)));

end
```

APPENDIX C - MESH FILES OF 5-CLUSTER CASE FOR THE SIMULATOR

C.1 Mesh File for CFD Simulator OpenFOAM Version 3.0+

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                      |                                            |
| \\      /  F ield              | OpenFOAM: The Open Source CFD Toolbox      |
|  \\    /   O peration          | Version:  v3.0+                            |
|   \\  /    A nd                | Web:      www.OpenFOAM.com                 |
|    \\/     M anipulation       |                                            |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

convertToMeters 1;

vertices
(
    (0 0 0)   //0
    (10 0 0)
    (10 0.1 0)
    (0 0.1 0)
    (0 0 0.1) //4
    (10 0 0.1)
    (10 0.1 0.1)
    (0 0.1 0.1)
    //-----------------
    (10.1 0 0)   //8
    (10.1 0.1 0)
    (10.1 0.1 0.1)
    (10.1 0 0.1)
    //-----------------
    (20 0 0)   //12
    (20 0.1 0)
    (20 0.1 0.1)
    (20 0 0.1)
    //-----------------
    (20.1 0 0)   //16
    (20.1 0.1 0)
    (20.1 0.1 0.1)
    (20.1 0 0.1)
    //-----------------
    (30 0 0)   //20
    (30 0.1 0)
    (30 0.1 0.1)
    (30 0 0.1)
    //-----------------
    (-0.1 0 0)  //24
    (-0.1 0.1 0)
    (-0.1 0.1 0.1)
    (-0.1 0 0.1)
    //-----------------
    (30.1 0 0)   //28
    (30.1 0.1 0)
    (30.1 0.1 0.1)
    (30.1 0 0.1)
    //-----------------
```

```
    (40 0 0)   //32
    (40 0.1 0)
    (40 0.1 0.1)
    (40 0 0.1)

);

blocks
(
    hex (0 1 2 3 4 5 6 7) (100 2 2) simpleGrading (1 1 1)
    hex (1 8 9 2 5 11 10 6) (1 2 2) simpleGrading (1 1 1)
    hex (8 12 13 9 11 15 14 10) (99 2 2) simpleGrading (1 1 1)
    hex (12 16 17 13 15 19 18 14) (1 2 2) simpleGrading (1 1 1)
    hex (16 20 21 17 19 23 22 18) (99 2 2) simpleGrading (1 1 1)
    hex (20 28 29 21 23 31 30 22) (1 2 2) simpleGrading (1 1 1)
    hex (28 32 33 29 31 35 34 30) (99 2 2) simpleGrading (1 1 1)
    hex (24 0 3 25 27 4 7 26) (1 2 2) simpleGrading (1 1 1)

);

edges
(
);

boundary
(
    inlet
    {
        type patch;
        faces
        (
            (27 26 25 24)

        );
    }


    outlet
    {
        type patch;
        faces
        (
            (2 9 8 1)
            (13 17 16 12)
            (21 29 28 20)
        );
    }


    walls
    {
        type wall;
        faces
        (
            (3 25 26 7)
            (2 3 7 6)
            (9 2 6 10)
            (13 9 10 14)
            (17 13 14 18)
```

```
            (21 17 18 22)
            (29 21 22 30)
            (33 29 30 34)
            //-----------------
            (24 0 4 27)
            (0 1 5 4)
            (1 8 11 5)
            (8 12 15 11)
            (12 16 19 15)
            (16 20 23 19)
            (20 28 31 23)
            (28 32 35 31)
            //-----------------
            (7 26 27 4)
            (6 7 4 5)
        (10 6 5 11)
        (14 10 11 15)
        (18 14 15 19)
        (22 18 19 23)
            (30 22 23 31)
            (34 30 31 35)
            //-----------------
        (25 3 0 24)
        (3 2 1 0)
        (9 13 12 8)
        (17 21 20 16)
        (29 33 32 28)
            //-----------------
        (32 33 34 35)
        );
    }

);

mergePatchPairs
(
);

// ************************************************************************* //
```

C.2 DEM Simulator LIGGGHTS Input File

```
echo            both
log             ../DEM/log.liggghts
thermo_log      ../DEM/post/thermo.txt

atom_style      granular
atom_modify     map array sort 0 0
communicate     single vel yes

boundary        f f f
newton          off

units           si

region          reg block 0 40 -5 5 -5 5 units box
create_box      1 reg

neighbor        0.003 bin
neigh_modify    delay 0 binsize 0.01


# Material properties required for granular pair styles
fix         m1 all property/global youngsModulus peratomtype 5.e6
fix         m2 all property/global poissonsRatio peratomtype 0.45
fix         m3 all property/global coefficientRestitution peratomtypepair 1 0.3
fix         m4 all property/global coefficientFriction peratomtypepair 1 0.5
#fix         m5 all property/global characteristicVelocity scalar 2.0

# pair style
pair_style  gran model hertz tangential history # hertz without cohesion
pair_coeff  * *

# timestep, gravity
timestep    0.00001
fix         gravi all gravity 9.8 vector 0.0 -1.0 0.0

# read walls

fix     sec_10 all mesh/surface file ../DEM/sec_10.stl type 1
fix     sec_11 all mesh/surface file ../DEM/sec_11.stl type 1
fix     sec_12 all mesh/surface file ../DEM/sec_12.stl type 1
fix     sec_13 all mesh/surface file ../DEM/sec_13.stl type 1

fix     sec_20 all mesh/surface file ../DEM/sec_20.stl type 1
fix     sec_21 all mesh/surface file ../DEM/sec_21.stl type 1
fix     sec_22 all mesh/surface file ../DEM/sec_22.stl type 1
fix     sec_23 all mesh/surface file ../DEM/sec_23.stl type 1

fix     sec_30 all mesh/surface file ../DEM/sec_30.stl type 1
fix     sec_31 all mesh/surface file ../DEM/sec_31.stl type 1
fix     sec_32 all mesh/surface file ../DEM/sec_32.stl type 1
fix     sec_33 all mesh/surface file ../DEM/sec_33.stl type 1

fix     sec_40 all mesh/surface file ../DEM/sec_40.stl type 1
fix     sec_41 all mesh/surface file ../DEM/sec_41.stl type 1
fix     sec_42 all mesh/surface file ../DEM/sec_42.stl type 1
fix     sec_43 all mesh/surface file ../DEM/sec_43.stl type 1

fix     opn_10 all mesh/surface file ../DEM/opn_10.stl type 1
fix     opn_11 all mesh/surface file ../DEM/opn_11.stl type 1
fix     opn_12 all mesh/surface file ../DEM/opn_12.stl type 1
```

```
fix     opn_20 all mesh/surface file ../DEM/opn_20.stl type 1
fix     opn_21 all mesh/surface file ../DEM/opn_21.stl type 1
fix     opn_22 all mesh/surface file ../DEM/opn_22.stl type 1

fix     opn_30 all mesh/surface file ../DEM/opn_30.stl type 1
fix     opn_31 all mesh/surface file ../DEM/opn_31.stl type 1
fix     opn_32 all mesh/surface file ../DEM/opn_32.stl type 1

fix     frac_10 all mesh/surface file ../DEM/frac_10.stl type 1
fix     frac_11 all mesh/surface file ../DEM/frac_11.stl type 1

fix     frac_20 all mesh/surface file ../DEM/frac_20.stl type 1
fix     frac_21 all mesh/surface file ../DEM/frac_21.stl type 1

fix     frac_30 all mesh/surface file ../DEM/frac_20.stl type 1
fix     frac_31 all mesh/surface file ../DEM/frac_21.stl type 1


# walls
fix     xwalls1 all wall/gran model hertz tangential history primitive type 1 xplane 0
fix     xwalls2 all wall/gran model hertz tangential history primitive type 1 xplane 40
fix     ywalls1 all wall/gran model hertz tangential history primitive type 1 yplane -5.0
fix     ywalls2 all wall/gran model hertz tangential history primitive type 1 yplane 5.0
fix     zwalls1 all wall/gran model hertz tangential history primitive type 1 zplane -5.0
fix     zwalls2 all wall/gran model hertz tangential history primitive type 1 zplane 5.0
fix     otherwalls all wall/gran model hertz tangential history mesh n_meshes 31 meshes
        sec_10 sec_11 sec_12 sec_13 sec_20 sec_21 sec_22 sec_23 sec_30 sec_31 sec_32
        sec_33 sec_40 sec_41 sec_42 sec_43
        opn_10 opn_11 opn_12 opn_20 opn_21 opn_22 opn_30 opn_31 opn_32
        frac_10 frac_11 frac_20 frac_21 frac_30 frac_31

# create single partciles
region      bc block 0 1 0 0.1 0 0.1 units box
fix         pts1 all particletemplate/sphere 15485863 atom_type 1 density
            constant 2650 radius constant 0.005
fix         pdd1 all particledistribution/discrete 15485867  1 pts1 1.0

fix     ins all insert/rate/region seed 32452843 distributiontemplate pdd1 vel
        constant 15 0 0 nparticles 1500000 particlerate 15000 insert_every 100
        overlapcheck yes region bc



#create_atoms    1 single 0.00733 0.0517 0.6 units box
#set             group all diameter 0.003 density 3000

# cfd coupling
fix     cfd all couple/cfd couple_every 100 mpi
fix     cfd2 all couple/cfd/force

#variable    vx equal vx[1]
#variable    vy equal vy[1]
#variable    vz equal vz[1]
#variable    time equal step*dt
#fix         extra all print 10000 "${time} ${vx} ${vy} ${vz}"
            file ../DEM/post/velocity.txt title "#" screen no
```

```
# apply nve integration to all particles that are inserted as single particles
fix         integr all nve/sphere

# screen output
compute         rke all erotate/sphere
thermo_style    custom step atoms ke c_rke vol
thermo          1000
thermo_modify   lost ignore norm no
compute_modify  thermo_temp dynamic yes

# insert the first particles so that dump is not empty
run     0
dump    dmp all custom 10000 ../DEM/post/dump.liggghts_run id
        type x y z ix iy iz vx vy vz fx fy fz omegax omegay omegaz radius

run     0 upto
```