June 2019

Transient Flow Analysis of a Closing Blowout Preventer Using Computational Fluid Dynamics (CFD)

Daniel Barreca

TRANSIENT FLOW ANALYSIS OF A CLOSING BLOWOUT PREVENTER USING COMPUTATIONAL FLUID DYNAMICS (CFD)

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science

in

The Craft & Hawkins
Department of Petroleum Engineering

by
Daniel Dominic Barreca
B.S. Louisiana State University, 2017
August 2019

Acknowledgements

I would like to give sincere thanks and appreciation to my primary advisor, Dr. Mayank Tyagi, who has consistently helped and given guidance to me throughout during my academic studies. I would also like to give thanks to my thesis committee members, Dr. Krishnaswamy Nandakumar and Dr. Richard Hughes, who gave much of their time to me during this process.

This research was possible thanks to resources provided by LSU. Software packages such as ANSYS Fluent, Autodesk Inventor, and LSU HPC resources (allocations provided by LSU CCT) were made freely available to me for use of work in this thesis.

Finally, I would like to give special thanks to my family who have given encouragement and support to me throughout this entire process.

Table of Contents

Nomenclature

c	$Sonic\ Velocity, m/s$
D	$Pipe\ or\ Equivalent\ Diameter, m$
f	$Friction\ Factor, dimensionless$
F	$Force, N$
g	$Gravitational\ Acceleration, m/s^2$
H	$Piezometric\ Head, m$
J	$Productivity\ Index, m^3/(s * Pa)$
k	$turbulent\ kinetic\ energy$
k_s	$Spring\ Constant$
P	$Total\ Pressure, Pa$
Q	$Volume\ Flow\ Rate, m^3/s$
t	$Time, seconds$
u_*	$friction\ velocity$
V	$Volume, m^3$
v	$Fluid\ Velocity, m/s$
x	$Mesh\ Position, m$
$\dot{x}$	$Mesh\ Velocity, m/s$
$\ddot{x}$	$Mesh\ Acceleration, m/s^2$
α	$Volume\ Fraction, dimensionless$
ϵ	$Dissipation\ of\ turbulence$
μ	$viscosity$
ρ	$Fluid\ Density, kg/m^3$
τ	$shear\ stress$
$Subscripts$	
BHP	$Bottomhole\ Pressure$
i, j, k	$x, y, z\ coordinate\ directions$
m	$mixture$
p	$p^{th} phase$
q	$q^{th}\ phase$
r	$reservoir$

Abstract

Reliability of blowout preventers (BOPs) is crucial for drilling and production operations. Erosion of BOP components and hydrodynamic forces on rams may cause failure of BOP elements to seal the well. Transient computational fluid dynamics (CFD) simulations of fluids within the wellbore and BOP offer quantitative and qualitative data related to this reliability during the closure of various BOP components. Since limited research has been published in transient CFD simulations of closing BOPs, this thesis discusses challenges and solutions to simulating closing blowout preventers. Single component fluids are simulated through several BOP geometries such as annular preventers, pipe rams, and shear rams. Cavitation, pressure fields, velocity fields, and shear rates along walls are monitored during the simulations. The present work provides a basis to which future directions may be built upon, such as more complex fluid properties.

Chapter 1. Introduction

1.1. Background

During the drilling process, influx of fluid from a formation may occur into the wellbore. Influx of fluid may occur for many reasons, such as an insufficient pressure provided by the drilling fluid. In cases as these, a blowout preventer (BOP) may be closed to control the well. A BOP is essentially a series of valves located at the top of a well that serves as a secondary means of controlling the well. An illustration of a BOP stack with labeled components is provided in Figure 1.1. A BOP consists of various sealing mechanisms such as pipe rams, shear rams, and an annular preventer. Each of these components can be investigated separately through simulation.

During a well shut-in, valves are hydraulically activated to seal the well. However, issues may occur during the shut-in process for various reasons. The most well-known event of such an issue occurring in the last decade is the Macondo Blowout in 2010. Many issues led to the blowout of this well, one of those being the BOP failing to seal the flow path. In the Transocean report (2011), it was documented that various component within the BOP stack eroded, including both drill pipe and blind shear rams. The report from BSEE (2011) also concluded erosion of blind shear rams and stated that the "Agency should consider researching the effects of a flowing well on Subsea BOP to shear Pipe."

1.2. Motivation

CFD software packages have demonstrated over the years to be a powerful tool in various industries. Some common software packages include OpenFOAM, ANSYS Fluent, COMSOL, and Star CCM+. When utilized by a knowledgeable user, CFD tools such as these are a proven means in saving costs and improving work efficiency. CFD can provide an engineer

with several important design related information quantities. John Slater (2008) classifies these results into three different categories, organized in order of required accuracy.

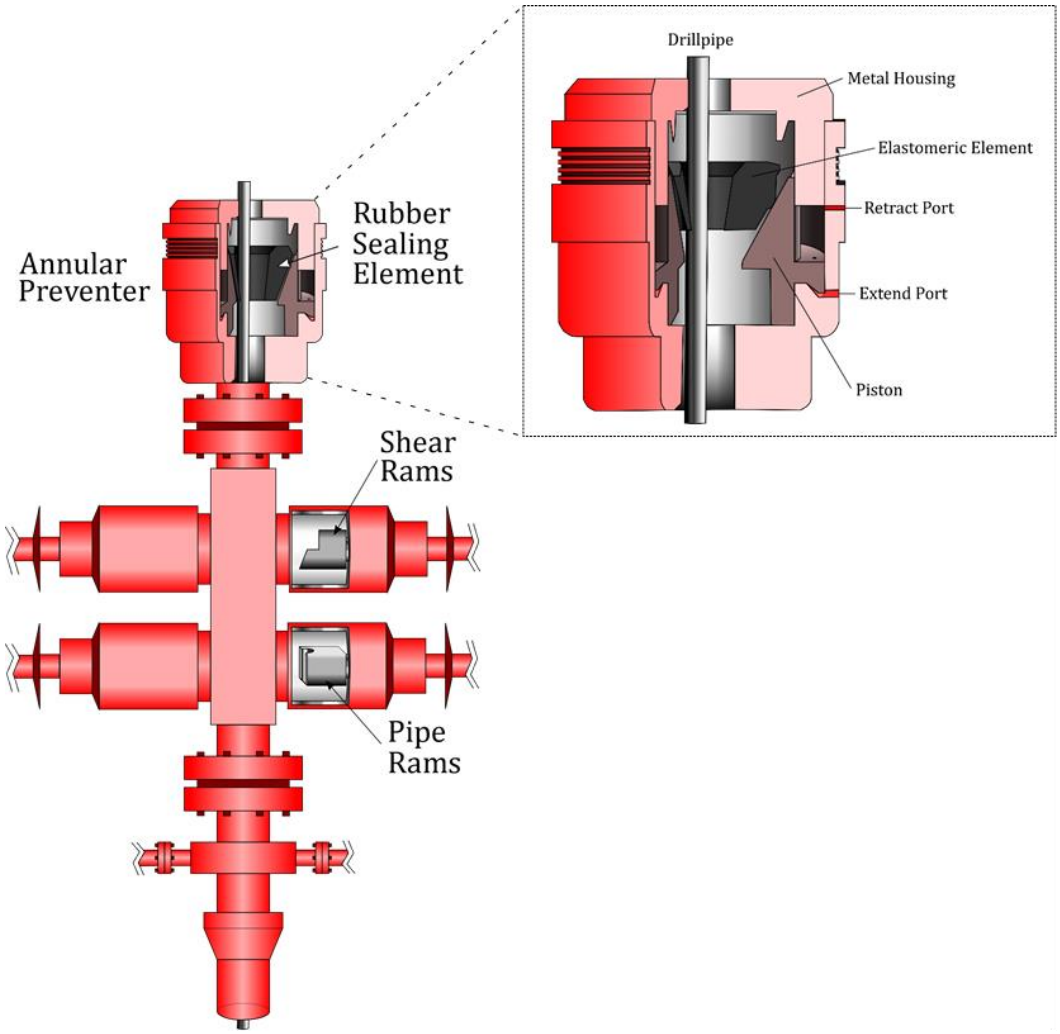


Figure 1.1. Illustration showing typical components of a BOP stack, including subcomponents that facilitate closure and opening of the annular preventer

First, CFD may provide qualitative information of the flow field. Contour plots, streamlines, and other data visualization techniques can be applied to view and analyze solution variables from the simulation. Qualitative information can be used to make basic design choices and to gain further understanding of a problem.

Second, CFD may provide incremental quantities. For example, CFD may be used to analyze how various input parameters affect quantities of interest. This provides useful information to which input parameters should be carefully considered during the design process, since some parameters will have a much larger impact than others.

Third, CFD may provide absolute quantities. Forces, shear rates, mixing processes, and other similar quantities of interest may be needed to make critical design decisions. When properly implemented, CFD can be used to gain this information.

The work in this thesis will demonstrate how CFD can be utilized for simulation of closing blowout preventers. In line with the discussion above, CFD simulation of blowout preventers can provide various avenues of details not otherwise possible. Qualitative information from the flow field (which cannot be easily viewed experimentally) may be analyzed through simulation. Simulation of blowout preventers can also be used to understand and analyze dynamic forces on the rams during shut in. This information provides useful insight for many reasons. Shear rams, for example, require a certain force to be able to shear pipe. In using CFD, this dynamic response of ram motion from hydrodynamic forces may be investigated.

Such simulations also deliver a cost-efficient method of analyzing the problem since various geometries and conditions can be tested before investing in the manufacturing of components. For example, off-center pipe can be investigated, and different ram resigns can be compared. Information related to optimizing ram designs and ensuring reliability of certain BOP components during a shut in can be gained through CFD simulation. Some example designs are shown in Figure 1.2.

Simulation can also provide information related to erosional analysis. Erosion may be significant for several reasons, such as high flowrates, presence of corrosive fluids, and transported particles that collide with walls. During the case of shutting in during the drilling process, drilling mud will be flowing through the annulus. Drilling fluid will be carrying drill cuttings, which may significantly contribute to erosion when the well is shut-in. Even in the case of shutting in on produced fluids, sand will be produced, adding to the erosional effect.

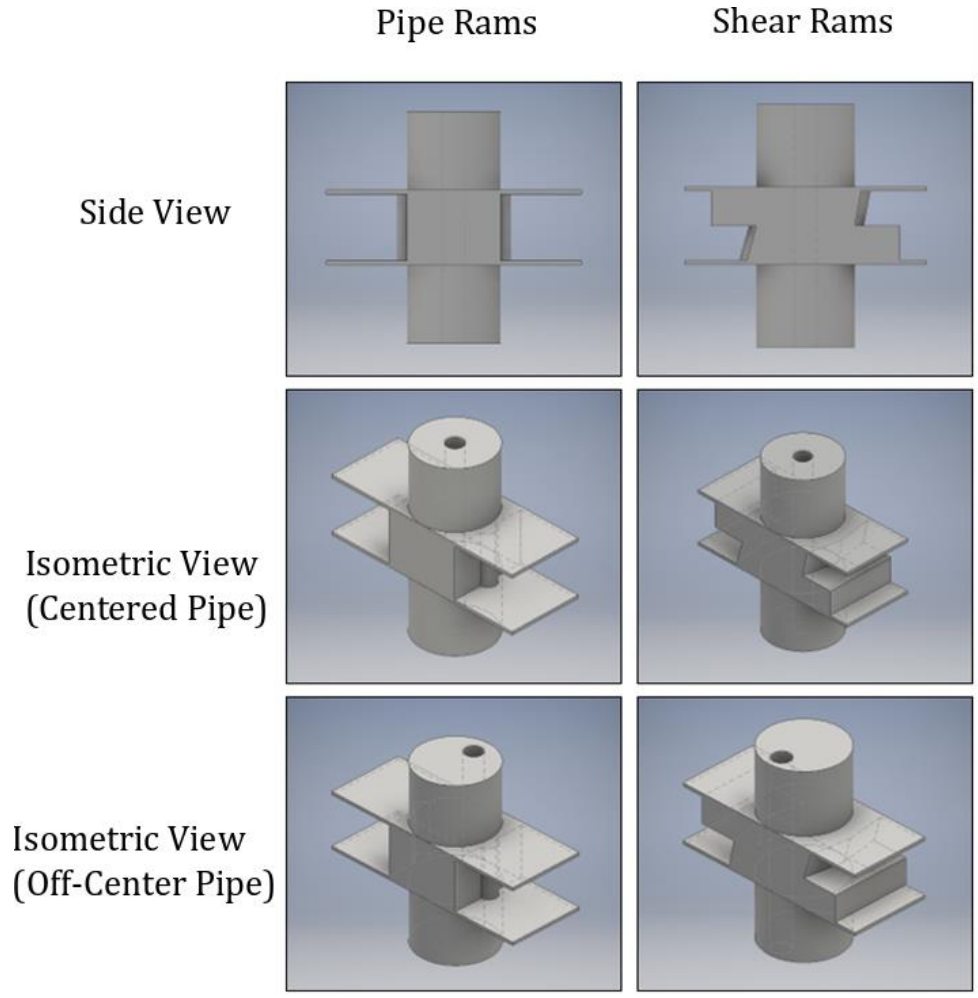


Figure 1.2. Example geometries created in Inventor, demonstrating possibilities in varying geometric features.

Another purpose for this study is to explore and report on dynamic meshing applications to closing valves. The dynamic meshing options in Fluent, as well as other

simulation software packages, were not specifically created for valve closures. Instead, the dynamic meshing options were created for geometry movement problems in general. As a result, dynamic meshing applied to closing blowout preventers has not been extensively discussed.

While similar previous published works on CFD simulation of blowout preventers have been completed, few have included time-dependent simulations to analyze a blowout preventer as it closes. Additionally, none of the published works have been coupled to 1D pipeline simulators. Successful completion of this work improves understanding of flow fields and dynamic characteristics through various geometries in the event of a kick. Additionally, simulation provides a safe method of testing hazardous conditions such as high flowrates and multiphase flow. ANSYS Fluent is capable of approximating multiphase flow through various models, discussed in a later section.

1.3. Problem Statement

A blowout preventer is connected to thousands of feet of wellbore during the drilling process. Because the length of a wellbore may span thousands of feet, 3D CFD simulations of an entire wellbore are practically impossible. While it may technically be possible considering the computational resources available today, using CFD to solve such a problem is simply inefficient. Therefore, only a section of the wellbore should be modelled using CFD, such as a couple of feet above and below the BOP's fluid domain. However, an issue arises when only modelling a portion of the wellbore: inlet boundary conditions must be applied to the CFD domain, such as a specified flowrate. Since the inlet boundary condition depends on information from the unmodeled section of the wellbore, a constant inlet boundary condition should not be applied to the CFD domain.

Two physical phenomena must be captured to provide accurate inlet boundary conditions. The first is that the velocity profile throughout the wellbore and BOP depends on the dynamic pressure gradient throughout the system, such as shown in

Figure 1.3. As the blowout preventer closes, pressure differential across the BOP increases. Since all fluids have some compressibility associated with them, this pressure wave will not instantly travel to bottomhole. There will be some sonic velocity, associated with the fluid properties, at which this pressure wave will travel. A pressure and velocity profile will develop until a new steady state condition is reached. For additional information related to how these profiles develop given a variety of situations of compressible flow transients, refer to Hati, Verma, and Chhabra (2001).

The other physical situation that must be captured is flowrate response from the reservoir. The wellbore is also connected to a constant pressure reservoir. The closure time, sonic velocity, and wellbore length may be such that the pressure wave reaches the reservoir while the BOP is still closing. If this situation occurs, flowrate from the reservoir into the wellbore will reduce. This velocity reduction may travel back up the wellbore and reach the BOP before complete closure. This process is illustrated in Figure 1.4.

To model these two processes, a 1-D transient simulation approach is implemented. The pressure transients within the 1D pipeline simulator are to be attained by numerically solving the 1D conservative form of mass and momentum balance for unsteady, compressible, and single-phase flow. Current work includes the coupling of this 1D pipeline simulator with 3D BOP simulations to provide accurate inlet boundary conditions to the CFD

simulation. Since the current implementation is only for single phase flow, future work would include the extension to multiphase wellbore flow.

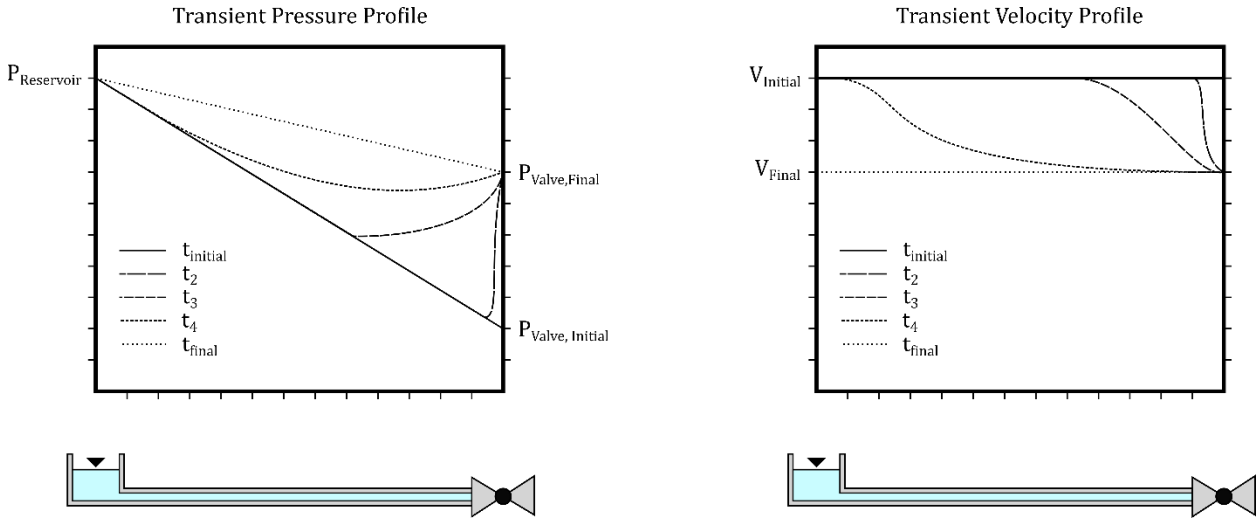


Figure 1.3. Example situation for transient pressure and velocity profiles throughout the wellbore. A valve on one end of the wellbore is slightly closed, increasing the pressure that propagates to the reservoir.

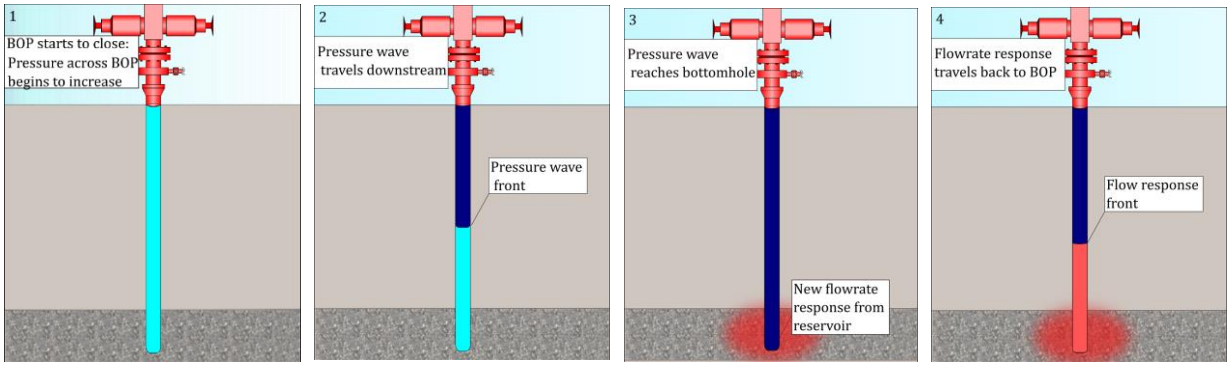


Figure 1.4. Flow response from the reservoir in the event of a kick taken and consequent BOP closure.

1.4. Scope of Problem

The aim of this study is to carry out a comprehensive investigation of simulation for a closing BOP in the event of a kick. This includes:

- Validation and verification study for transient CFD blowout preventer simulations with dynamic meshing. For example, grid independence studies are to be completed for each drastically different geometry. While not much experimental data is available for fluid

flow through closing blowout preventers, experimental data from closing valves is widely available. CFD simulations will be compared to data from these closing valve experiments, including comparisons of information such as hydrodynamic forces, stresses, pressure transients, and flow rates.

- Development of a basic 1D wellbore simulator using the Method of Characteristics (MOC) for handling hydraulic transients within the wellbore. The 1D hydraulic simulator is coded in C to be compiled and used as a user defined function (UDF) within ANSYS Fluent.
- Implementation of explicit MOC(1D)-CFD(2D, 3D) coupling between wellbore and closing BOP.
- Discussing implementation of ram motion dynamics to account for change in velocity of rams due to fluid forces acting on the surfaces.
- Transient analysis of closing a BOP using the coupled simulations for single and multiphase flow.

1.5. Previous Work

1.5.1. Simulation through Valves

A blowout preventer is essentially a large valve. While simulation of blowout preventers is not a research topic that has been fully explored through publications, simulation of valves throughout various industries has been a topic of extensive research. Some relevant literature of simulation through valves is briefly reviewed. Leutwyler and Dalton (2008) demonstrate applicability of CFD for accurately predicting hydrodynamic forces on a butterfly valve for single phase flow. Davis and Stewart (2002) used CFD in order to attempt to predict performance of a control valve. Steady simulations were run at various

valve closures from 0% to 90% closure. Axisymmetric CFD simulation of the valve was able to accurately predict flow characteristics of the valves.

Srikanth and Bhasker (2009) utilize dynamic meshing to analyze the flow in a typical puffer type chamber. Spring-like dynamic meshing, described in a later section, is used here to simulate moving geometry. Similarly, a study by Song et al. (2014) used spring-like dynamic meshing techniques to analyze flow in a safety valve. Here, the grid motion is governed by the force balance on the valve. The grid's acceleration is calculated from the force balance between mechanical spring force, flow force, and gravitational forces from a moving component of the valve. Damper effects were ignored during this study. A sensitivity analysis was performed on various inputs such as spring stiffness, geometric variations, and inlet/outlet boundary conditions. The authors demonstrated dynamic meshing techniques applicability to capture dynamic motion of grid in response to fluid flow.

Beune, Keuren, and Heumen (2012) perform a similar study for fluid structure interaction in high pressure safety valves. However, a multi-grid approach is used instead for this study. Pre-defined meshes are set up for various closure points in the simulation process. The authors take this approach due to the large deformation in the mesh which will cause low quality elements. Further discussion of these issues is presented in a later section which illustrates a similar issue when using dynamic meshing.

1.5.2. Previous Work in BOP CFD Simulation

Springett et al. (2016) develop a patent for a seal based on erosion analysis from CFD simulations through a BOP. Various simulations were performed in this work, including studying cavitation/choking under multiple conditions and studying erosion on variable bore rams. For the base case of their simulations, dynamic meshing capabilities in ANSYS

FLUENT were utilized to simulate single phase flow through closing BOPs. A constant inlet pressure boundary condition was used for the work in the patent, unlike the work in this thesis where a 1D hydraulic transient simulator provides the inlet boundary condition. Another patent work similar to this includes a patent by Khandoker and Hydril (2008).

McCleney et al. (2018) combine FEA and CFD to analyze the fluid structure interaction during the shearing of a drillpipe. Due to computational expenses, complete fluid structure interaction was not modelled and CFD was only used to compute steady state simulations at five different instances. Similarly, Tulimilli et al. (2014) also utilize CFD to simulate the flow field throughout a BOP at several different instances. However, the CFD simulations are performed separately from the structural analysis in this case and is not coupled. Similar studies have also been completed on devices such as subsea surface safety valves. Xu et al. (2014) use CFD to calculate turbulent intensity and velocity fields throughout a subsea surface valve for erosion analysis.

1.5.3. Water Hammer and Coupled Simulations

Using OLGA, Han et. al (2013) study the water hammer effect for multiphase flows in wellbores during start up and shut in. It was determined that water hammer effect was mainly affected by wellbore fluid volume and fluid compressibility. Additionally, they concluded that a small amount of compressible fluid will reduce the effect of water hammer significantly. However, in OLGA, a predetermined flow coefficient is needed to calculate pressure drop across a valve. For the work in this thesis, a flow coefficient will not be needed to determine pressure drop because CFD is utilized to solve for the 3D flow field through a closure. This allows increased flexibility as different valve geometries can be investigated in

combination with the additional input variables such as closure time, wellbore volume, and fluid influx rate.

While coupling BOP CFD simulations with water hammer simulations has not been completed in any of the referenced papers, the concept is not entirely new. A similar process has been accomplished by Feng et al. (2017) where the complex flow field throughout a valve is modelled through CFD and a thermal-hydraulic code handles water hammer calculations throughout a complex pipeline. Wu et al. (2015) coupled a MOC pipeline simulator to a transient, fast-closure pump CFD simulation for more accurate inlet boundary conditions of the CFD domain. Zhang et al. (2016) analyze water hammer during collapse of a draft tube by simulation of a complex geometric portions using CFD and a MOC approach to model the rest of the domain.

Chapter 2. Theory and Methodology

Simulation of a shut-in is carried out through CFD simulation of a BOP coupled to a hydraulic transient simulation of the wellbore. Here, governing equations and discretization methods are described for all simulations included in this study. Additionally, the methodology behind the simulation and coupling process will be described in this section.

2.1. Governing Equations

2.1.1. Fluid Flow through Wellbore

Similar governing equations of fluid flow to the CFD calculations are discretized for solving transient flow throughout the pipeline. The one-dimensional equations of continuity and momentum are presented here in a more convenient form using piezometric head and flowrate:

$$\frac{\partial H}{\partial t} + \frac{c^2}{g}\frac{\partial v}{\partial x} = 0 \tag{1}$$

$$\frac{\partial H}{\partial x} + \frac{1}{g}\frac{\partial v}{\partial t} + \frac{f}{2gD}v|v| = 0 \tag{2}$$

Where piezometric head, H, can be related to pressure through the following relationship:

$$H = \frac{P}{\rho g} \tag{3}$$

2.1.2. Fluid Flow through the BOP

The fluid flow throughout a blowout preventer and the wellbore is again governed by the continuity and Navier Stokes Equations, except now presented in 3D form.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \tag{4}$$

$$\rho \frac{\partial \vec{v}}{\partial t} + \rho(\vec{v} \cdot \nabla)\vec{v} = -\nabla \mathrm{P} + \rho \vec{g} + \nabla \cdot \tau_{ij} \tag{5}$$

2.1.3. Multiphase Flow

As an extension to single phase flow throughout a BOP, it is also important to consider multiphase flows. During shut, fluid will likely not be a single phase. In the case of drilling fluid, cuttings may be flowing with the fluid throughout the BOP stack. When petroleum fluids are flowing through the wellbore, two phases of liquid and gas are probable to exist at pressure and temperature conditions at the BOP. For the work in this thesis, gas-liquid flow will be investigated through simulation.

Through multiphase flow analysis, various scenarios can be investigated. Some examples include the impact of slugs on hydrodynamic forces on shear rams, cavitation of a fluid flowing through small restrictions, choking conditions for worst case discharge estimates, and mass transfer between phases. Various models exist in commercial CFD software packages for simulating multiphase flow. The three main models that will be utilized during this study are the volume of fluid (VOF) approach, the mixture model, and the Eulerian-Eulerian approach. The VOF approach is an interface tracking method with a shared momentum equation between the phases. The phases are tracked through the continuity equation of each phase:

$$\frac{1}{\rho_q}\left[\frac{\partial}{\partial t}\left(\alpha_q \rho_q\right) + \nabla \cdot \left(\alpha_q \rho_q \vec{v}_q\right) = S_{\alpha_q} + \sum_{p=1}^{n}\left(\dot{m}_{pq} - \dot{m}_{qp}\right)\right] \tag{6}$$

Since the momentum is shared between phases and only a single equation is needed, the same momentum equation as in equation 2 is used for the VOF model. Applications of the VOF model include flows where the interface should be tracked, such as in stratified flows.

Similar to VOF, the Eulerian-Eulerian approach calculates volume fractions for each phase using equation 6. However, the interface between fluids is no longer of interest, so a

surface tracking technique is no longer applied. An additional contrast to the VOF model is that the momentum equation is no longer shared between the phases, resulting in a momentum equation for each phase:

$$\frac{\partial}{\partial t}(\alpha_q \rho_q \overrightarrow{v_q}) + \nabla \cdot (\alpha_q \rho_q \vec{v}_q \vec{v}_q) = -\alpha_q \nabla \mathrm{p} + \nabla \cdot \bar{\bar{\tau}}_\mathrm{q} + \alpha_q \rho_q \vec{g} + \sum_{p=1}^{n} (\vec{R}_{pq} + \dot{m}_{pq} \vec{v}_{pq} - \dot{m}_{qp} \vec{v}_{qp}) + \vec{F} \quad (7)$$

The Eulerian-Eulerian model is a popular choice for flows such as bubble columns or fluidized beds.

The final and simplest of the models is the mixture model. Rather than modelling each phase, the properties are averaged. For example, the continuity equation would now consist of terms such as an average density and velocity:

$$\frac{\partial \rho_m}{\partial t} + \nabla \cdot (\rho_m \vec{v}_m) = 0 \quad (8)$$

The same assumption of averaged properties is used for other flow equations. Using this model, even though properties are averages, velocities may slip past one another based on a drift velocity between the two phases. Aside from the multiphase models described above, other multiphase models exist to simulate phenomena such as mass transfer, choking, and cavitation.

2.1.4. Fluid Flow from Reservoir

Flow into the wellbore from the reservoir is assumed to be single phase. Given a productivity index (J), flowrate can be estimated through:

$$Q = J \cdot (P_r - P_{BHP}) \quad (9)$$

Where Q is flowrate, P_r is reservoir pressure and P_{BHP} is bottomhole pressure of the wellbore. A more complicated model (such as reservoir simulation) may be used to simulate fluid flow from the reservoir. However, a simple model is used here to save on computational

resources and because reservoir simulation is not the focus of this thesis. Additionally, it is important to note that fluid response from the reservoir will not affect the solution in many cases. This will be further discussed in a later section of the thesis and demonstrated with simulation results.

2.1.5. Turbulence Modelling

Turbulent flow is a flow field which has fluctuations in variables such as velocity. A turbulent flow regime will begin to develop within the annulus as velocity increases. During various processes in drilling and production, turbulence within the wellbore is a common occurrence. Because of this, it is important that turbulence is accurately modelled. Methods of modelling turbulence include Reynolds-Averaged Navier-Stokes models (RANS), Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS). RANS methods are used in this study because of the computational demands required by LES and DNS.

The concept behind RANS modelling is that flow field variables can be decomposed into two terms: a fluctuating component and a time-averaged component. After these variables are decomposed, new terms are introduced into the governing differential equations, and relationships to solve for the variables are needed for problem closure. Various turbulence models exist for this purpose, such as the k-epsilon and k-omega models. The k-epsilon model (Launder and Spalding 1973) is used widely due to its robustness and accuracy for engineering problems.

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma k}\right)\frac{\partial k}{\partial x_j}\right] + G_k + G_b - \rho\epsilon - Y_M + S_k \tag{10}$$

$$\frac{\partial}{\partial t}(\rho \epsilon) + \frac{\partial}{\partial x_j}(\rho \epsilon u_i) = \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma k}\right)\frac{\partial \epsilon}{\partial x_j}\right] + C_{1\epsilon}\frac{\epsilon}{k}(G_k + C_{3\epsilon}G_b) - C_{2\epsilon}\rho\frac{\epsilon^2}{k} + S_\epsilon \tag{11}$$

The standard k-epsilon model will over-predict the turbulent eddy viscosity for flows with large separation. Known as the realizeable k-epsilon turbulence model and proposed by Shih et al. (1994), new formulations to calculate eddy viscosity and turbulent dissipation rate were developed and tested. Shih et al. (1994) demonstrated the model's superior performance for backwards step-facing flow, boundary free shear flows, and adverse pressure gradients. Such flows are extremely common during simulations of a closing BOP, as shown in Figure 2.1.

Despite improvements to the model, an issue with the k-epsilon model is that several terms in the equation have correlations that are difficult to measure, making problem closure approximations difficult to attain as well (Wilcox 1998). A popular alternative two-equation turbulence model is the k-omega model, initially developed by Kolmogorov (1942) and reassessed by Wilcox (1988). Investigation of this turbulence model is left for a further study.

Another aspect important to turbulence is the boundary layer. This region is the region of fluid nearest the wall, where fluid velocity is transitioning from stationary to free stream velocity. The turbulence models previously described are not applicable in this region because they assume fully developed turbulent flow. In order to accurately model the forces and pressures on the wall, the sub viscous boundary layer should be properly resolved. Turbulence models such as k-epsilon can approximately resolve this region through wall functions. Wall functions (Patankar and Spalding 1967) describe the logarithmic function of the boundary layer, as illustrated in Figure 2.2. Wall functions will severely loosen the requirements of the near wall mesh. However, wall functions are not always applicable. For example, in regions of flow separation, wall functions will not accurately describe the boundary layer velocity profile.

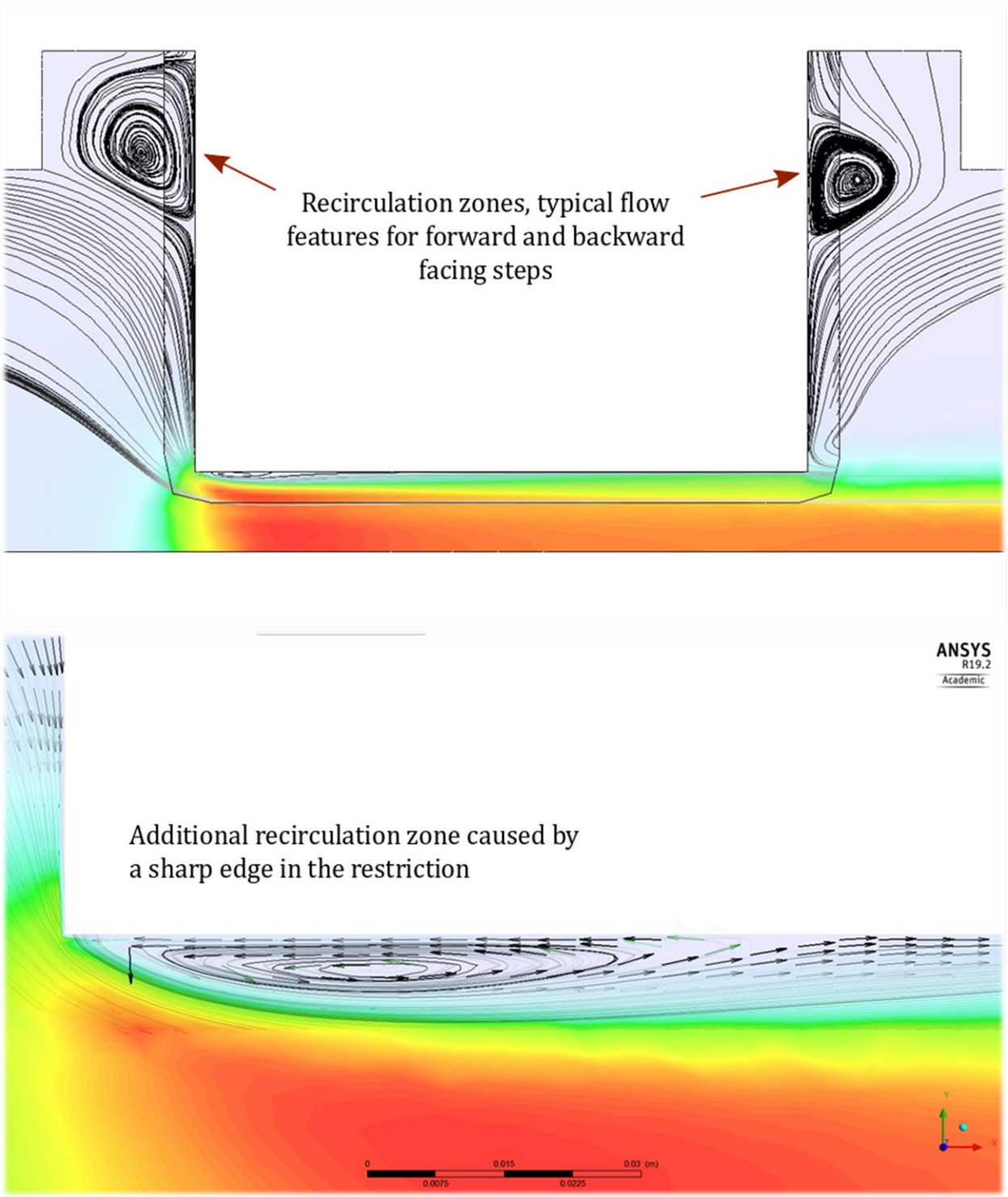


Figure 2.1. Example simulation demonstrating typical causes for recirculation zones in closing valves. The lower image depicts an enlarged view of flow recirculation, enlarged from the upper image.

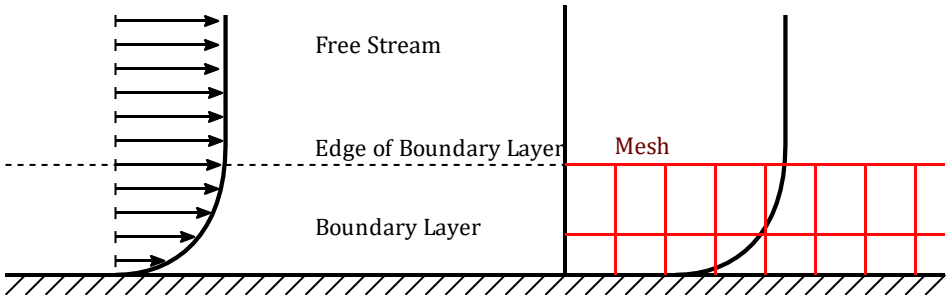


Figure 2.2. Illustration of boundary layer and unresolved meshing of the layer.

For the realizable k-epsilon model, y^+ values for the first cell's height normal to a wall should remain within the range of 30 to 300. The y^+ value can be estimated through the equation, which is important to calculate before mesh creation.

$$y^+ = \frac{u_* y}{\nu} \tag{12}$$

2.2. Fluid and Transport Properties

Fluids flowing through a BOP may include drilling fluids or hydrocarbons. Fluids such as these have more complex properties than single-phase Newtonian fluid properties, which are the default properties of a fluid in ANSYS Fluent. Drilling fluids are typically non-Newtonian fluids, which differ from Newtonian fluids in that the stresses are not linearly proportional to the strain rate for non-Newtonian fluids. An exception to this is the Bingham plastic, which differs from a Newtonian fluid in that Bingham Plastic fluids will not begin to flow until a yield point shear stress is applied. Examples are illustrated in Figure 2.3. ANSYS Fluent has capabilities of modelling non-Newtonian Fluids.

Another property of fluids to mention is the vapor pressure. At constant temperature, when a single component liquid falls below the vapor pressure, formation of gas bubbles will occur. Since large pressure differences develop across the BOP, and since low pressure will exist in areas of flow separation, extremely low pressures will develop during the transient process. Cavitation can be enabled in ANSYS Fluent to model this mass transfer interaction

between gas and liquid. However, hydrocarbons are not single component liquids. Instead, hydrocarbon liquids consist of many different components. As a result, there is no single pressure at which the liquid will completely change to gas. Instead, a bubble point exists at which the first gas bubbles will form. From there, as pressure decreases, increasing amounts of gas will evolve from the liquid. Refer to McCain (2017) for a more comprehensive discussion of petroleum fluid properties. Modelling such fluid properties may require the use of user defined functions to properly implement.

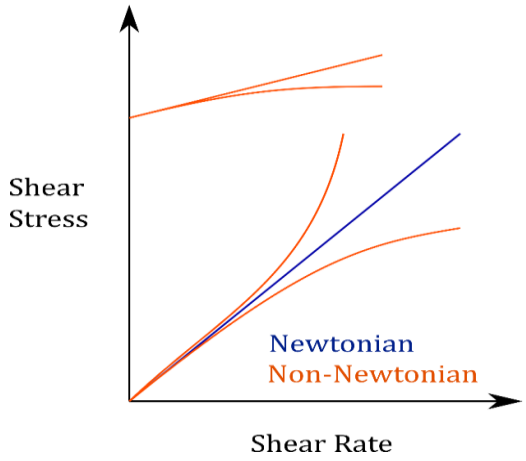


Figure 2.3. Shear stress/rate relationship for Newtonian and non-Newtonian fluids.

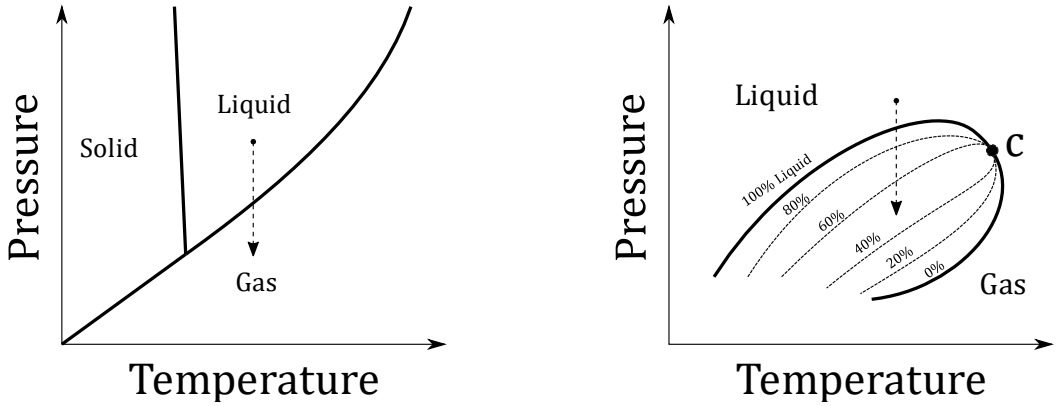


Figure 2.4. Mass transfer for single component (left) and multi-component (right) fluids.

2.3. Dynamic Meshing

To simulate moving rams closing in against the pipe, dynamic meshing methods are utilized. Dynamic meshing is a process through which nodes and volumes move or re-mesh based on a motion description specified by the user. When implementing one of these dynamic meshing options, an additional condition, the geometric conservation law (GCL), must be satisfied:

$$\frac{\partial}{\partial t}\int_V dV = \int_s W_j dn_j \tag{13}$$

The GCL states that the rate of change of each control volume must be equal to the volume swept by its boundaries. Additional information can be found in ANSYS Theory Guide (2001) for FLUENT and Weller et al. (1998) for OpenFOAM.

Several methods of dynamic meshing options exist in ANSYS Fluent, including layering, remeshing, and smoothing. A summary of advantages and disadvantages of each method in ANSYS Fluent from the experiences during this project are presented below. A more in-depth discussion demonstrating these issues related to BOP closure is included in this section. Additionally, more advanced meshing options are discussed in the ANSYS Fluent user manual.

2.3.1 Layering

As a boundary moves, elements will split or merge with other elements in the mesh. The process is illustrated in Figure 2.5. The cell height at which the layering remeshes is specified in ANSYS Fluent as a constant value. Therefore, a constant sized mesh must be used across the direction of ram wall motion.

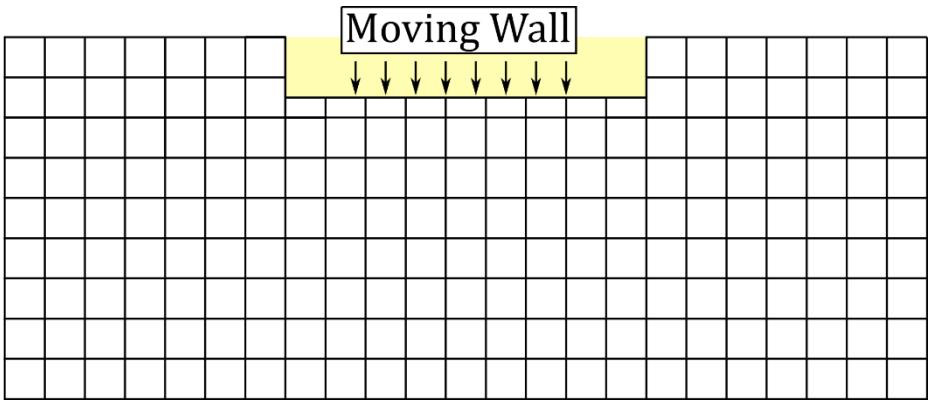


Figure 2.5. Illustration of the layering technique: a moving boundary collapses cells as it closes in on the fluid domain.

- Advantages
    1. Can maintain high quality cells. Most of the cells in domain remain undisturbed, and even the cells that become remeshed still remain hexahedral.
    2. Can be used with hexahedral cells, unlike standard remeshing.
    3. Area of remeshing can be easily controlled, since it is known exactly where and how the remeshing will occur.
- Disadvantages
    1. Must be used with hexahedral meshes, and this may be an issue since structured meshes are not always easy or even possible to create.
    2. Remeshing may occur in areas where remeshing is undesirable. For example, remeshing may occur on a wall where physical quantities of interest are extracted. When using layering, this issue may be difficult to avoid depending on the geometry motion and mesh setup.
    3. If layering is applied along a wall, the first cell height along this wall will be drastically varying throughout the entire process. First cell height must be within a specific range depending on the turbulence model and fluid velocity,

so this may be difficult to control when using layering. Specific setups can be created to avoid this issue, though the process may be too tedious to consider.

4. Depending on the geometry, layering may require a specific setup that involves interfaces. Interfaces between fluid domains will introduce additional error in the solution, since interpolation across the domains is involved in this process.

2.3.2. Spring Based Smoothing

In using the spring-based method, elements will act as a spring. As a boundary moves, the elements of the mesh will stretch or compress depending on the motion of the boundary and specified spring constants. An illustration of this process is shown in Figure 2.6.

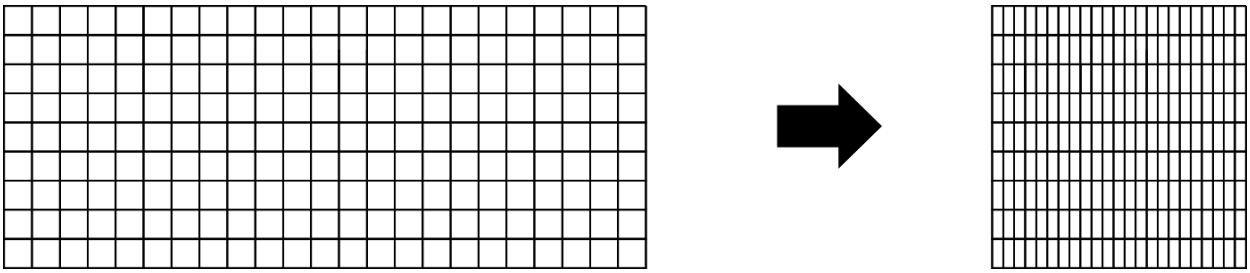


Figure 2.6. Illustration of spring-based smoothing: a boundary moves and compresses a set of cells.

- Advantages

    1. Easiest implementation of the three options listed here.
    2. Can be used with hexahedral cells, though the cells will deform.
    3. Fluid cells will not remesh. This may be advantageous in certain situations, since remeshing involves interpolation of solutions onto the new mesh, introducing some error to the simulation process. There is also no risk of remeshing prism layers.

- Disadvantages

1. In cases of large deformation, cells will extensively stretch and twist, causing extremely low-quality elements. In other cases, these deformations will be so extreme that cells “overlap” each other, causing a negative cell volume error and stopping the simulation.

2.3.3. Smoothing and Remeshing

For this dynamic meshing option, a cell skewness or size criteria can be input by the user. During boundary motion, cells will deform through spring-based smoothing. As cells deform, the cells that do not meet these specified criteria are selected to be remeshed at the next timestep.

- Advantages
    1. Extremely robust in comparison to the other mentioned options. When enabled, this option will allow for a much wider possibility of geometric setups and deformations to occur.
- Disadvantages
    1. Remeshing may be difficult to control. Prism layers may remesh. Fluid cells along walls may remesh. Simulations with a large variance in cell size may also be particularly difficult to control. These issues require specific setups to avoid. This will be explained in a following section.

2.4. Discretization and Solution Methods to Governing Differential Equations

To solve the governing differential equations of fluid flow, numerical methods must be implemented since no analytical solution is available unless major simplifications are made. The governing equations can be discretized to approximate a solution. Various discretization methods exist, such as finite volume (FVM), finite element (FEA), method of

characteristics (MOC), finite difference, and spectral methods. The choice of discretization method is problem specific, depending on advantages and disadvantages of the chosen method. The chosen discretization for each domain is discussed in this section.

2.4.1. Discretization of Wellbore Domain

As described earlier, CFD is only used to model the BOP domain. An additional discretization is needed for the wellbore, where pressure propagates through the system. A comprehensive investigation of water hammer theory and practice was completed by Ghidaoui et al. (2006). It was stated that a common approach to solving propagation problems is the method of characteristics approach. It's accuracy, efficiency, and ease of implementation makes the method of characteristics a widespread choice among many commercial hydraulic transient simulators. The concept of MOC is to reduce the governing differential equations into a set of ordinary differential equations. The first step is linear combination of the governing equations:

$$L_1 = \frac{\partial H}{\partial x} + \frac{1}{g}\frac{\partial V}{\partial t} + \frac{f}{2gD}V|V| \tag{14}$$

$$L_2 = \frac{\partial H}{\partial t} + \frac{c^2}{g}\frac{\partial V}{\partial x} \tag{15}$$

$$L = L_1 + \lambda L_2 = 0 \tag{16}$$

$$L = \frac{\partial H}{\partial x} + \frac{1}{g}\frac{\partial V}{\partial t} + \frac{f}{2gD}V|V| + \lambda\left(\frac{\partial H}{\partial t} + \frac{c^2}{g}\frac{\partial V}{\partial x}\right) = 0 \tag{17}$$

Rearranging gives:

$$\left(\frac{\partial H}{\partial x} + \lambda\left(\frac{\partial H}{\partial t}\right)\right) + \left(\frac{1}{g}\frac{\partial V}{\partial t} + \lambda\left(\frac{c^2}{g}\frac{\partial V}{\partial x}\right)\right) + \frac{f}{2gD}V|V| = 0 \tag{18}$$

From the material derivative:

$$\frac{dV}{dt} = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x}\frac{dx}{dt} \tag{19}$$

$$\frac{dH}{dt} = \frac{\partial H}{\partial t} + \frac{\partial H}{\partial t} + \frac{\partial H}{\partial x}\frac{dx}{dt} \tag{20}$$

By solving the 3 equations above, λ can be solved for as

$$\lambda = \pm\frac{1}{c} = \frac{dx}{dt} \tag{21}$$

Substitution of this solution and the material derivative into the original governing equation results in:

$$\frac{dV}{dt} + \frac{g}{c}\frac{dH}{dt} + \frac{f}{2D}V|V| = 0 \qquad for\ \lambda = +\frac{1}{c} \tag{22}$$

$$\frac{dV}{dt} - \frac{g}{c}\frac{dH}{dt} + \frac{f}{2D}V|V| = 0 \qquad for\ \lambda = -\frac{1}{c} \tag{23}$$

A forward finite difference can be applied to discretize the differential equations, and the equations can be arranged to solve for the next timestep's variable:

$$V_x^{t+1} = \frac{1}{2}\left(V_{x-1} + V_{x+1} + \frac{g}{c}(H_{x-1} - H_{x+1}) - f\frac{\Delta t}{2D}(V_{x-1}|V_{x-1}| + V_{x+1}|V_{x+1}|)\right) \tag{24}$$

$$H_x^{t+1} = \frac{1}{2}\left(H_{x-1} + H_{x+1} + \frac{c}{g}(V_{x-1} - V_{x+1}) - f\frac{\Delta t\ c}{2Dg}(V_{x-1}|V_{x-1}| - V_{x+1}|V_{x+1}|)\right) \tag{25}$$

A standalone python code is provided in the appendix. It is important to note the assumptions made here:

- Single phase flow is assumed. Single phase flow will affect several of the assumptions below. In reality, flow throughout the wellbore will most likely not be single phase. Evolution of gas from the oil is expected as the fluid travels up the wellbore.

- Constant friction factor is assumed in the model. In reality, friction factor would not be constant for two main reasons. As fluid velocity changes within the wellbore, friction factor will also change. Additionally, friction factor will be different in varying regions of multiphase flow.
- Constant wave speed is assumed. Wave speed will vary depending upon two parameters. Varying gas void fractions throughout the entire wellbore will cause wave speed to be different throughout the wellbore. Additionally, flow pattern will influence wave speed.
- Diameter is assumed to be an equivalent diameter expression from annulus to pipe.

Because the focus for this thesis is in simulation of closing BOP and not developing a flexible multiphase hydraulic transient simulator, the assumptions will be left as is. To improve on the simulations, these issues would need to be addressed. Since the wellbore simulator is explicitly coupled, the most practical solution would be to couple ANSYS Fluent to an already existing commercial hydraulic transient simulator or to a wellbore simulator. Coupling of ANSYS Fluent to OLGA (an oil and gas pipeline simulator) has already been accomplished by Xing, Yeung, and Lo (2011).

2.4.2. Discretization of Blowout Preventer Domain

For simulation of fluid flow throughout the blowout preventer, the finite volume method is utilized. Because the finite volume method can handle complex geometries and has an inherent capability of conservation, it is popular choice in the discretization of the mass, momentum, and energy conservation equations for fluid flow. FVM is implemented by discretizing a domain into finite volumes, and the governing equation is integrated over an arbitrary control volume. The gauss-divergence theorem is then used to convert volume

integrals to surface integrals to finally obtain the integral form of the governing equations. Specific algorithms and further discretization of nonlinear terms are still required for solving the set of generated algebraic equations. Rather than develop the code for a finite volume solver, a commercial software package is used to discretize and solve the governing equations. For a more in-depth explanation of the finite volume method, refer to an introductory text such as Patankar (1980), Versteeg and Malasekera (2007), or Anderson and Wendt (1995).

2.5. Boundary Conditions

Along with the discretization of the governing differential equations, boundary conditions must be applied to solve for fluid flow in both domains. The boundary conditions and derivation of the boundary conditions for the wellbore simulator are described here. Boundary conditions are also described here for the CFD BOP domain. Since a commercial package is used for the CFD simulations, the derivation of boundary conditions is not described in this section. Refer to the introductory CFD texts mentioned in the previous section for derivation and implementation of these boundary conditions.

2.5.1 MOC (1D) Wellbore Simulation Boundary Conditions

Boundary conditions are needed at each end of the wellbore. On the reservoir side, a constant pressure throughout the reservoir is assumed since change in fluid volume of the reservoir is assumed to be insignificant. On the blowout preventer side of the wellbore domain, a pressure boundary condition is provided from the CFD simulation during the coupling process. This process will be described in a later section. The velocities on each end of the wellbore domain may also be calculated by rearranging equation 24, as demonstrated in Appendix B.

2.5.2. CFD (2D,3D) Blowout Preventer Boundary Conditions

CFD simulations require boundary conditions to be applied on the surfaces of a computational domain. The blowout preventer geometries contain three different types of faces. The first is the physical walls of the blowout preventer. This includes the pipe walls, annulus walls, and ram component walls. Conditions such as no-slip velocity along the walls are applied on these surfaces. The second is an outlet boundary condition, applied at the top of the domain. In this case, it is assumed that the blowout preventer's outlet is at 0-50,000 gauge psi. More complex boundary conditions may be implemented to take into consideration frictional pressure loss in the pipe above the BOP as flow through the system decreases. A third inlet boundary condition is needed on the lower face of the domain, where generally some flowrate, velocity, or pressure boundary condition is specified. The boundary condition in the case of a closing blowout preventer is more complicated than simply assigning a constant value of flow or pressure. An inlet velocity boundary condition is provided here by coupling the CFD simulation to a 1D wellbore simulation. The wellbore simulation provides this inlet boundary condition and is further explained in following sections

2.6. MOC (1D) – CFD (2D,3D) Coupling Description

The coupling process between MOC and CFD is similar to that in the study by Feng at al. (2015) and is illustrated in Figure 3.. The flow field is initialized at steady state. From this point, the transient simulation starts with a single time step of the BOP CFD simulation. Pressure information from this simulation is recorded by averaging the properties over a selection of cells. This pressure information is sent to MOC code and serves as a pressure boundary condition at the top of the wellbore domain. One time-step of the MOC hydraulic

transient simulator is computed, and the velocity value at the top of the MOC code is recorded. This information is then used as a velocity inlet boundary condition for the BOP CFD simulation.

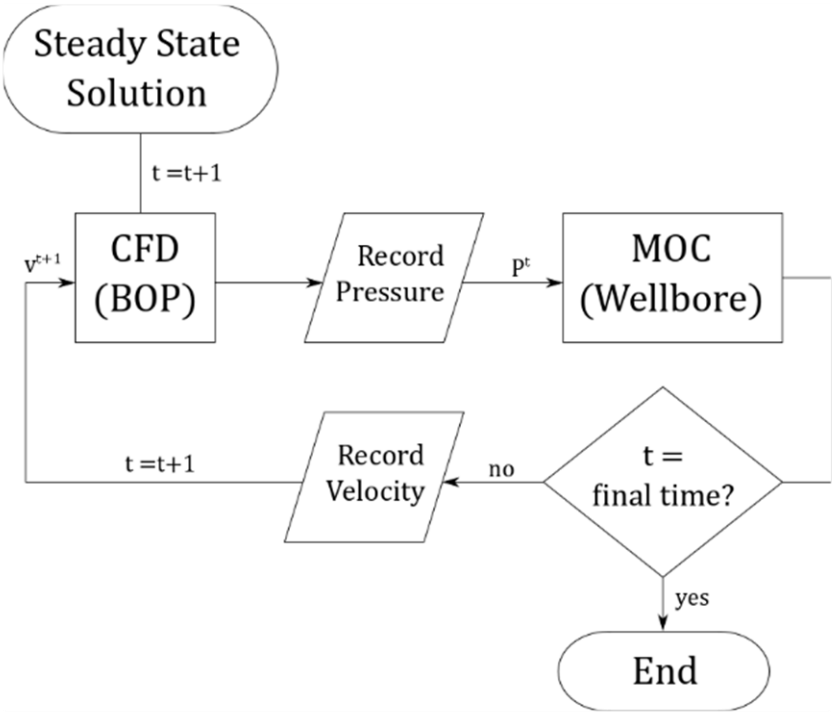


Figure 2.7. Explicit coupling algorithm between MOC and CFD simulation used in this thesis.

2.7. Validation and Verification Process

Verification and validation are important processes to ensuring reliable CFD simulation results. Some basic validation and verification work have been completed to demonstrate accuracy and credibility of current results. Examples are shown here, but all verification and validation examples are not shown for the sake of brevity. Validation is defined as determining if the computed simulation results agree with physical reality. On the other hand, verification is related to checking if the implementation and models are correct. An example of verification is a grid independence study. Grid independence studies can be conducted to determine if a grid has guaranteed spatial convergence. A CFD study generally

starts on a coarse grid. The grid may be refined until the solution has insignificant changes after successive refinements. An example grid independence study is shown in Figure 2.7. Four grid sizes for an axisymmetric closing BOP were simulated. It can be seen that after the grid is refined to 44,000 cells, the change in solution with further gird refinement becomes insignificant.

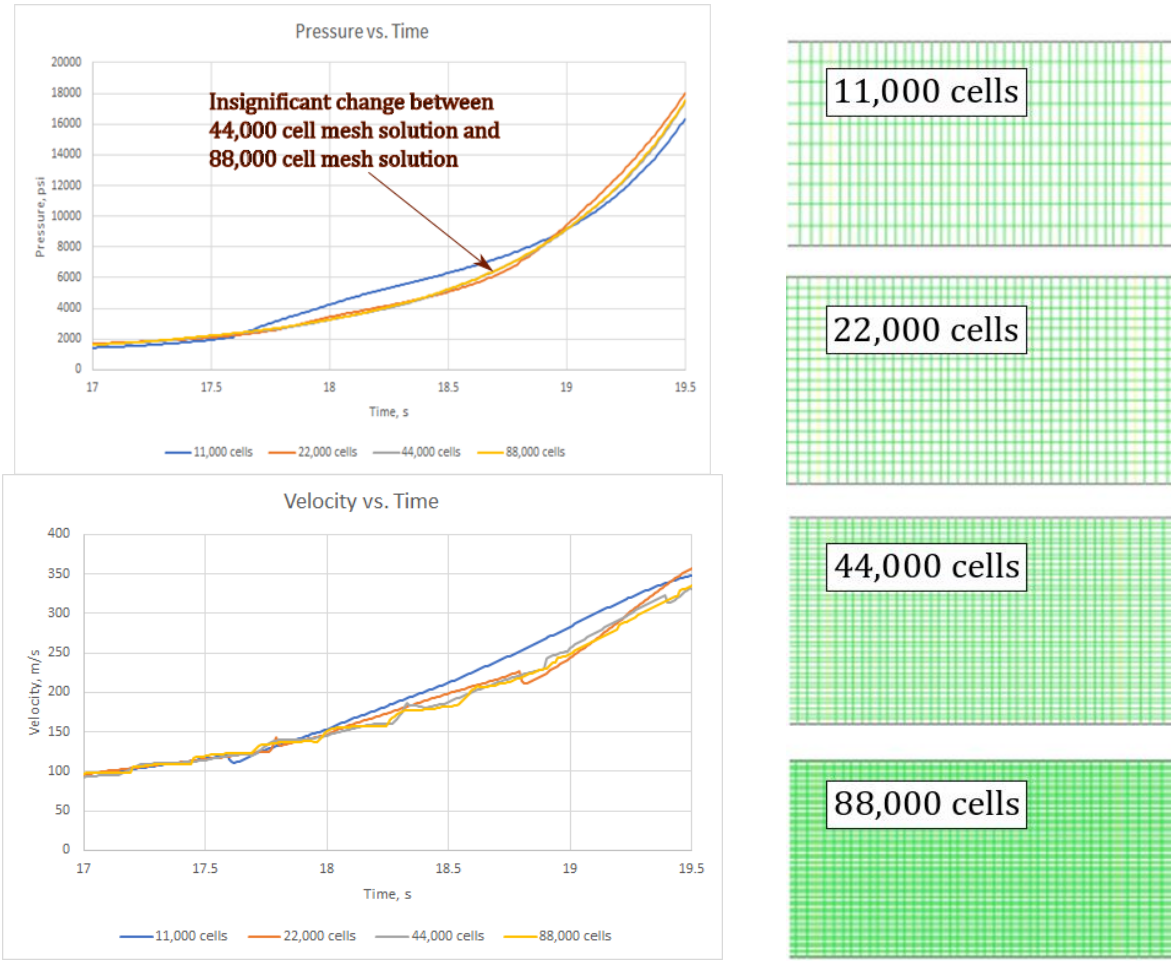


Figure 2.7. Grid independence study example using four different mesh sizes for closing valve with a closing time of 20 seconds.

Other verification processes are also completed to build confidence in the solution. Residuals, computed from each equation solved in ANSYS Fluent, are monitored over the entire process. Physical quantities of interest are also monitored to ensure that convergence is achieved. Wall y+ values are also monitored to maintain appropriate mesh sizing along

walls during the dynamic meshing process. Validation against experimental results are shown in later sections. For the work in this thesis, comparison to pressure drops across valves is included.

2.8. General Workflow Process and Tools Utilized

2.8.1. Software

All geometries are created in Autodesk Inventor, a commercial CAD software available on the virtual engineering desktop through LSU's resources. After the CAD file is created, it is saved as a .step file and imported into ANSYS ICEM CFD, a meshing software also available through LSU's resources. This mesh file is saved and imported into ANSYS FLUENT. After simulation is completed, postprocessing is completed through ParaView or CFD Post. In addition to the above-mentioned software, some coding is implemented for User Defined Functions (UDFs) in ANSYS Fluent. UDFs are used in fluent to enhance the standard features of Fluent, such as implementing advanced boundary conditions. These UDFs are written in C. Python is used for initial development and testing of codes. The Python codes are then translated into C to be compiled as a UDF in ANSYS Fluent.

2.8.2. Computational Resources

Most of the preliminary and testing simulations are run on a standard desktop. However, as grids become refined, computational demands increase. Particularly, multiphase flow simulations on the 3D grids require HPC resources. LSU's Center for Computation and Technology (CCT) maintains a high-performance computing environment with several different clusters. Of those resources, SuperMike-II is used when necessary for the simulations in this study. SuperMike-II is a 146 TFlops Peak Performance 440 compute

node cluster, with nodes that contain two 8-Core Sandy Bridge Xeon 64-bit processors that function at a core frequency of 2.6 GHz.

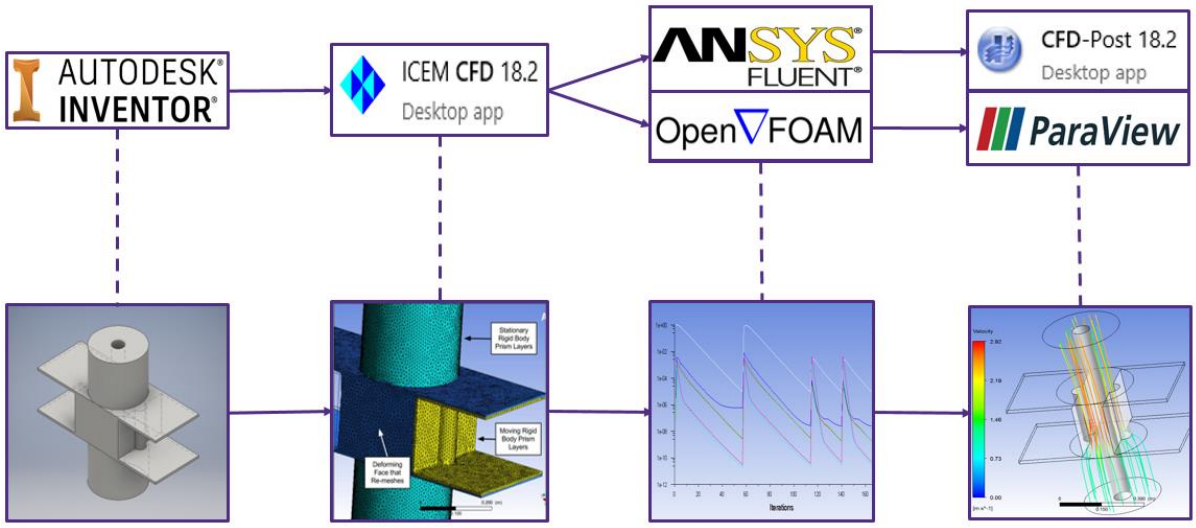


Figure 2.8. Software package workflow for this thesis from geometry creation to solution analysis.

Chapter 3. Numerical Setup and Implementation

3.1. MOC (1D) Wellbore Simulation Implementation

The hydraulic simulator was initially created in python for testing and development purposes. A standalone python code is included in Appendix B. This python code is eventually converted to C and then included as a UDF in ANSYS Fluent.

The code begins with initialization of important variables such as gravitational acceleration, initial fluid velocity, wellbore length, wave speed, and timestep. Because of the relationship between timestep and discretization length (as described in the theory section for MOC applied to water hammer), discretization length is not predefined. Instead, discretization length is calculated from timestep and wavespeed.

$$dx = dt * c \tag{26}$$

After the variables are created, [N x T] arrays are created for storage of piezometric head and velocity values, where N is the number of length discretizations and T is the number of time discretizations. This is modified in the later C code, where the spatial discretization values are stored only for the previous and current timestep, being constantly overwritten during simulation progress. Flow is then initialized by assuming constant velocity at all locations in the wellbore. Piezometric head is initialized by assuming constant reservoir pressure, then calculating the frictional head loss based on friction factor, velocity, and length.

$$\Delta p_{friction} = \frac{f\ L(V_{init})^2}{2gD} \tag{27}$$

3.2. General Setup for CFD (2D, 3D) BOP Simulations

While many input variables will vary across the simulation results shown in this thesis, several setups remain constant. The consistent setups are described here. Any simulation-specific properties and setups will be described in the relevant sections.

3.2.1. Implemented Fluid and Transport Properties

All fluids in the simulations within this thesis are specified as single component fluids. Multi-component fluids, such as hydrocarbons, are left for future work. Hydrostatic pressure conditions may vary across simulations, though an outlet pressure of 50,000 Pascals was general applied. Fluid properties such as density and viscosity may have varied across simulations, but values in the range of 700-1000 kg per cubic meter and 0.5-1 cp were chosen. Other than for validation purposes, viscosity and density values are not significant for this thesis, and were chosen to be similar properties to that of oil or water. Turbulence properties for the models chosen were typically left as the default values.

3.2.2. Solution Methods

Solution methods describe the spatial and transient discretization process. Lower order methods are generally less accurate, but easier to converge than higher order methods. Simulations in this thesis are initially run using the default settings, which generally consist of first order discretization methods. After convergence is achieved, first-order methods are switched to second-order methods. Solution stability was not a major issue for any of the simulations in this thesis.

An algorithm is needed to solve the set of discretized equations. This algorithm may be specified in the solution methods section. The Semi Implicit Method of Pressure Linked Equations (SIMPLE) algorithm is used for the simulations in this thesis. This algorithm, along

with other available algorithms, has been discussed extensively in various CFD texts. Refer to Patankar (1980) for description of the SIMPLE algorithm.

3.2.3. Solution Controls

Solution controls are used to specify under-relaxation factors for various variables and equations during the solution process. Higher factors will speed up the convergence, but may affect stability. Lower factors will slow down the convergence process, but may ensure convergence of unstable problems. Solution controls were left as default for most simulations, since convergence was not an issue during most runs. Default values of relaxation values range from 0.7 to 1. For multiphase simulations, some setups may have been difficult to converge. Default values were cut by one-third to assist in convergence.

3.2.4. Solution Convergence Criteria

Unlike the MOC (1D) wellbore simulations, variables in the CFD simulations cannot be directly solved for. Iterations must be completed until convergence is achieved. Convergence is generally monitored through residuals, which are quantifications of error. For the simulations in this thesis, residuals convergence criteria is set to 10^{-5} for all equations. Convergence is also judged by monitoring physical quantities. Pressures and shear rates along the ram walls, as well as pressure drops across the BOP are monitored during the simulation to judge convergence.

While not particularly difficult to attain for most of the simulations in this work, convergence may require some special attention, particularly for multiphase flows and cavitation of single-phase flow. Transient problems generally can have improved convergence by reducing the timestep, improving mesh quality, or changing solution controls. However, some steady state simulations can be difficult to converge depending on

initial conditions provided. For steady-state simulation cases, convergence was easily obtained for all simulations using the procedure below.

1. Set turbulence model to standard k-epsilon. This initial simulation will provide an initial guess for the realizable k-epsilon simulation.
2. Initialize the solution from outlet or inlet.
3. Run the simulation in transient at small time steps (e.g. 1e-5) for 10-100 iterations.
4. Gradually increase the time step until flow seems somewhat adequately developed.
5. Attempt to run the simulation in steady state until convergence is achieved.
6. Switch the turbulence model to realizable k-epsilon and again run the simulation in steady state until convergence is achieved.
7. If modelling cavitation, switch on the appropriate mass transfer model and repeat steps 2-4.

3.3. Dynamic Meshing Considerations for CFD domain

Specific setups are required for both layering and smoothing/remeshing. Some valve geometries are simple enough to approximate as an axisymmetric design, such as in Figure 3.1. In geometries such as these, hexahedral meshes are not time consuming to setup, allowing the possibility of layering. Application of layering requires a specific setup using interfaces. Necessity of interfaces is purely problem and software dependent. Other problem setups or software packages may not require interfaces as described here. Figure 3.2 illustrates the general setup for the simulations. Three separate fluid domains are imported into ANSYS Fluent and they are linked through the interface boundary condition. Standard wall, inlet velocity, and pressure outlet boundary conditions are then applied where relevant.

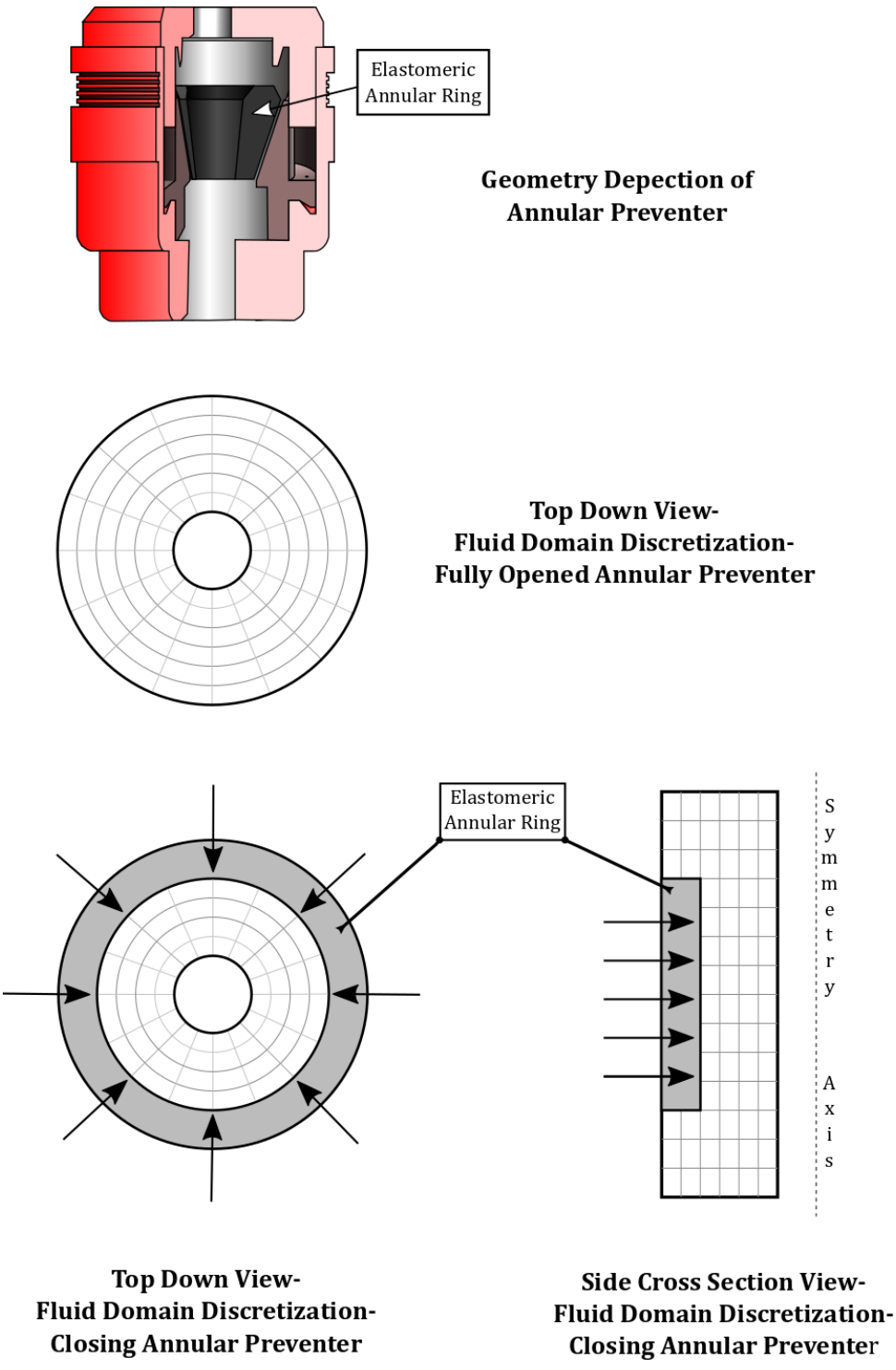
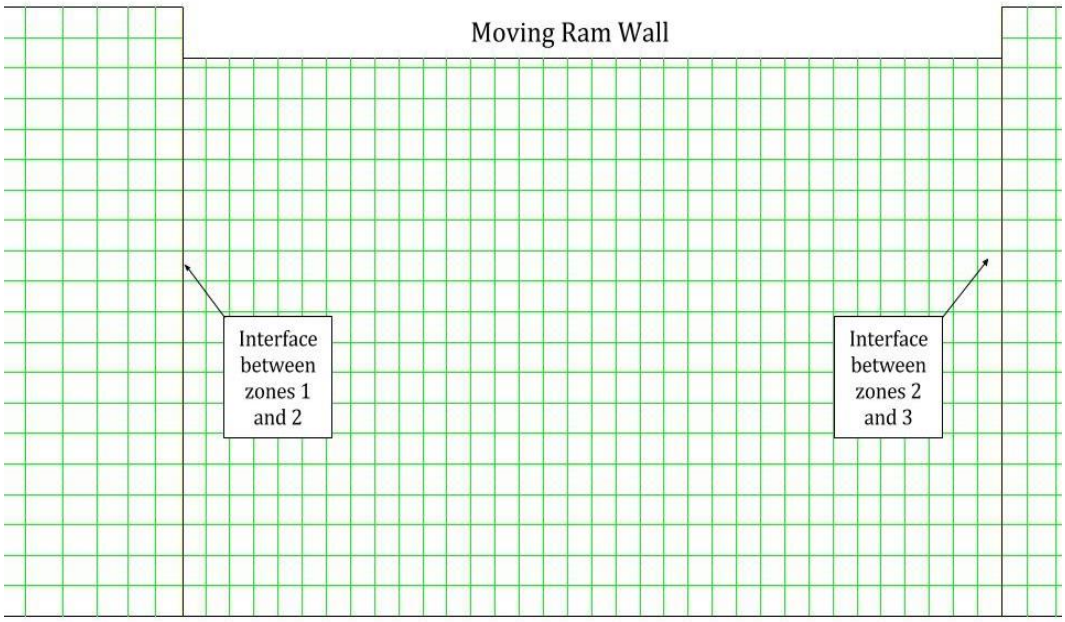
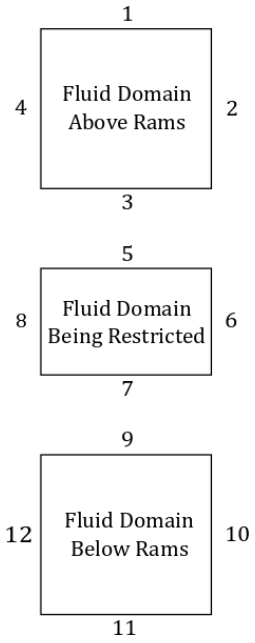


Figure 3.1. Axisymmetric representation of an annular preventer.





Number	Boundary Condition
1	Pressure Outlet
2	Stationary Wall
3	Interface
4	Stationary Wall
5	Interface
6	Stationary Wall
7	Interface
8	Moving Wall
9	Interface
10	Stationary Wall
11	Velocity Inlet
12	Stationary Wall

Figure 3.2. Description of boundary conditions for a closing annular preventer simulation using layering. Upper image shows the mesh and interfaces. Lower left image illustrates the boundary condition setup, labelled according to the table.

In cases where remeshing and smoothing are used, only a single fluid domain is needed. However, to apply boundary conditions in different regions, the fluid domain and surfaces are broken into several different parts during the meshing process. Moving prism layers must be separated from the main domain to apply a motion boundary condition to that section of the fluid domain. Stationary prism layers should also be separated to apply a rigid body boundary condition, ensuring that prism layers do not remesh. Additionally, deforming faces should be separated from the main body since a dynamic mesh boundary condition (deforming face) will need to be specified here.

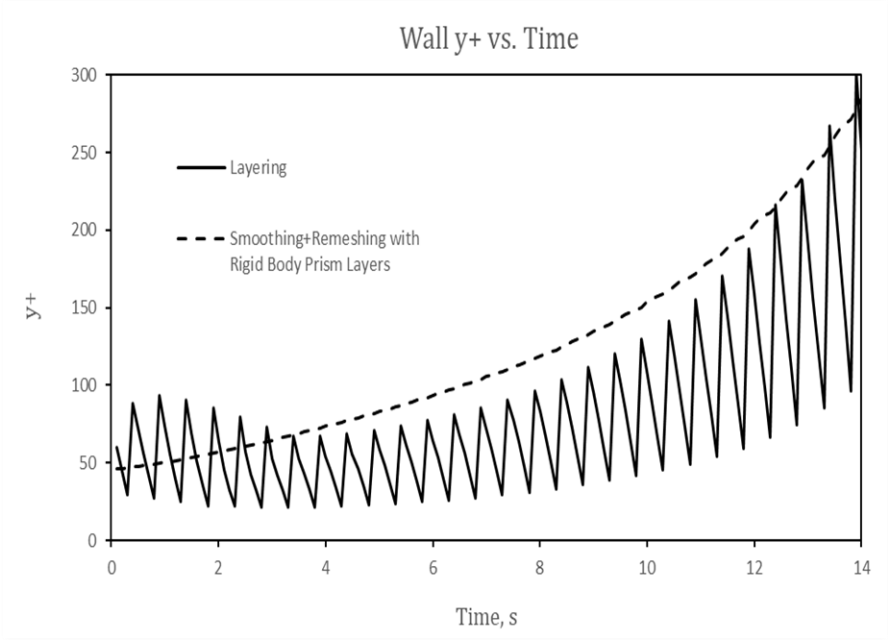


Figure 3.3. y+ Comparison between layering at a wall and rigid body elements at a wall.

The importance of rigid body prism layers is demonstrated partially in Figure 3.. When cells near the wall are allowed to deform (such as in the mesh setup shown in Figure 3.2), the y+ value becomes difficult to control since the cell height is always changing. More importantly, cells in critical regions are being remeshed when using a mesh setup as in Figure 3.2. Therefore, it is recommended that rigid body elements are used, such as in Figure 3. and Figure 3.5 meshes.

Another important meshing consideration to discuss is the idea of using multiple pre-defined meshes throughout the simulation process. As discussed in the turbulence modelling section, a y+ value between 30 and 300 should be used for the turbulence models used in this thesis. The simulation data in Figure 3. comes from a 20 second BOP closure simulation. It can be seen here that the y+ value reaches 300 at 14 seconds, which is long before the simulation ends. This can be managed by setting up a series of meshes that have successively refined prism layers, as demonstrated in Figure 3.3. An example is provided in the results section of this thesis.

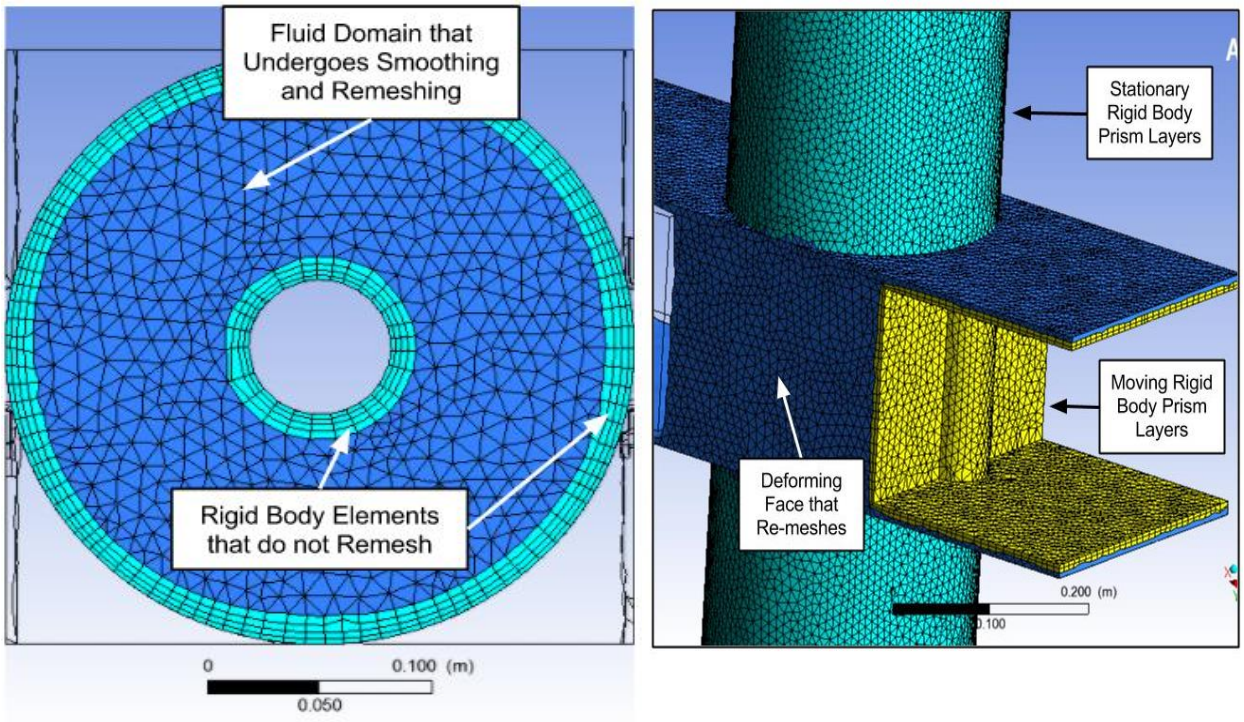


Figure 3.4. Case setup for smoothing and Remeshing with rigid body prism layers.

Another reason why multiple pre-defined meshes are needed is because the rigid body prism layers will eventually collide with the edge of the domain, as shown in Figure 3.. The domain cannot close beyond the thickness of these prism layers. However, when

multiple pre-defined meshes are set up, a thinner set of prism layers may be used. This would allow for further closure of the BOP.

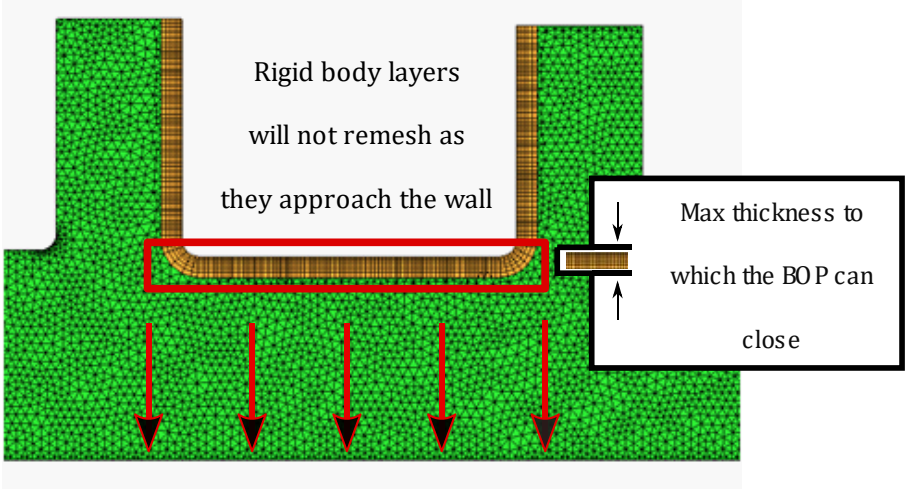


Figure 3.5. Limitation of mesh closure with rigid body prism layers.

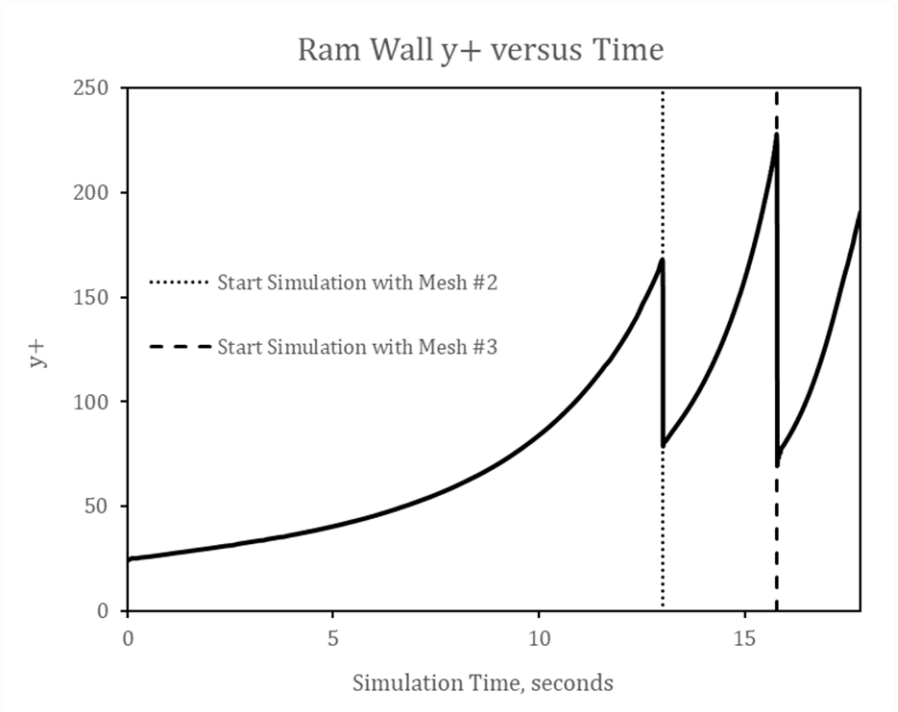


Figure 3.3. Multi-stage meshing strategy to manage wall y+ values.

3.4. Implementation of Boundary Motion and Important Considerations

As explained in chapter 2, the boundary motion of the rams may be described in three different ways. The implementation of those three methods in ANSYS Fluent are described

in this section. Example codes to be compiled are found in Appendix B. In general, a CG_MOTION UDF may be used to specify the motion of a dynamic zone. That motion may be applied to nodes, lines, or entire regions of cells. Only details relevant to the work in this thesis are provided here. Comprehensive documentation of this UDF may be found in the relevant ANSYS Fluent user manual.

For a constant velocity case, implementing a UDF is as simple as specifying a constant value and direction. For example, "vel[1] = 0.05" would apply a constant velocity of 0.05 m/s in the y-direction to all nodes, lines, or cells. A time-varying motion may be specified just as easily, by including a time multiplier. For example, "vel[0] = -0.05*time" would apply a linearly changing time-varying velocity in the negative x-direction to all nodes, cells, or lines.

The dynamic ram motion case is slightly more involved, since forces on the rams must first be calculated. The forces on rams may be calculated from the dot product of two relevant vectors: the vector pointing in the direction of ram motion, and a vector pointing in the direction normal to the face, as illustrated in Figure 3.4. A dot product is necessary to calculate the force only in the direction of ram motion.

$$F_{fluid} = -P * A * n_1 \cdot n_2 \tag{28}$$

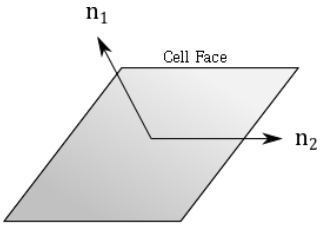


Figure 3.4. Illustration of Fluid Force and Cell Motion Vector Directions

P is the pressure on the cell face, A is cell face area, n_1 is the direction vector normal to the face area, and n_2 is the direction vector of unit motion. A negative sign is necessary because the normal direction of the face always points in the opposite direction that the force

is acting on. In the provided UDF in Appendix B, the direction vector is always equal to ram motion vector, so this dot product calculation is not included.

Additionally, using the information for calculating acceleration in Chapter 2 regarding hydro pneumatic systems, a velocity at each time step must still be approximated. The velocity can be approximated through a first order forward finite difference.

$$\ddot{x}^t = \frac{\dot{x}^t - \dot{x}^{t-1}}{\Delta t} \tag{29}$$

The new x position may also be calculated from the old position and velocity.

$$x^t = x^{t-1} + \dot{x}^{t-1}\,\Delta t \tag{30}$$

These two equations may be substituted into the original equation from the section in chapter 2.

$$m_{ram}\,\ddot{x}^t = F_{fluid} - kx^t - b\dot{x}^t \tag{31}$$

$$m_{ram}\,\frac{\dot{x}^t - \dot{x}^{t-1}}{\Delta t} = F_{fluid} - k(x^{t-1} + \dot{x}^{t-1}\,\Delta t) - b\dot{x}^t \tag{32}$$

Moving unknowns to the right hand side, and known values to the left hand side, the equation may be rearranged as:

$$\dot{x}^t = \frac{\left(\dot{x}^{t-1} + \frac{\Delta t\,\left(F_{fluid} - k(x^{t-1} + \dot{x}^{t-1}\,\Delta t) - b\dot{x}^t\right)}{m_{ram}}\right)}{\left(1 + \frac{b\Delta t}{m_{ram}}\right)} \tag{33}$$

When damper forces are assumed to be negligible, the equation is reduced to:

$$\dot{x}^t = \dot{x}^{t-1} + \frac{\Delta t\,\left(F_{fluid} - k(x^{t-1} + \dot{x}^{t-1}\,\Delta t)\right)}{m_{ram}} \tag{34}$$

The spring rate, "k," can be calculated as discussed in Appendix C. This implementation is left for future work.

3.5. MOC(1D) – CFD(3D) Coupling Implementation

A UDF to be compiled in ANSYS Fluent is provided in Appendix B. This UDF is used as an inlet velocity boundary condition in ANSYS Fluent. Implementation of this coupling process involves the code described in section 3.1, which is the section that describes the implementation of the MOC solution to hydraulic transients throughout the wellbore domain. A simple pseudocode is provided in Figure 3.5, which provides details of the coupling process. An important aspect to note is that all wellbore calculations are only calculated on the first iteration of the CFD solution at each time step. Successive iteration between simulations is not done in this coupling process. Maintaining stability of the solution is achieved through another manner, described in a later section.

3.6. Averaging Pressure in the CFD Simulation

A selection of cells needs to be looped over to attain an average pressure value. This is implemented my splitting the fluid domain into separate zones (as was done with rigid body prism layers). An example setup is shown in Figure 3.6. In ANSYS Fluent, a separate zone will be associated with its own ID which can be accessed and looped over (see the UDF in the appendix for reference on how this is done). Ideally, the pressure should be averaged by:

$$\frac{\sum(Pressure\ of\ Cell_i * Cell_i\ Volume)}{\sum(Cell_i\ Volumes)} \tag{35}$$

However, since the pressure is fairly uniform in this region, this is approximated through the sum of cell pressures divided by the number of cells.

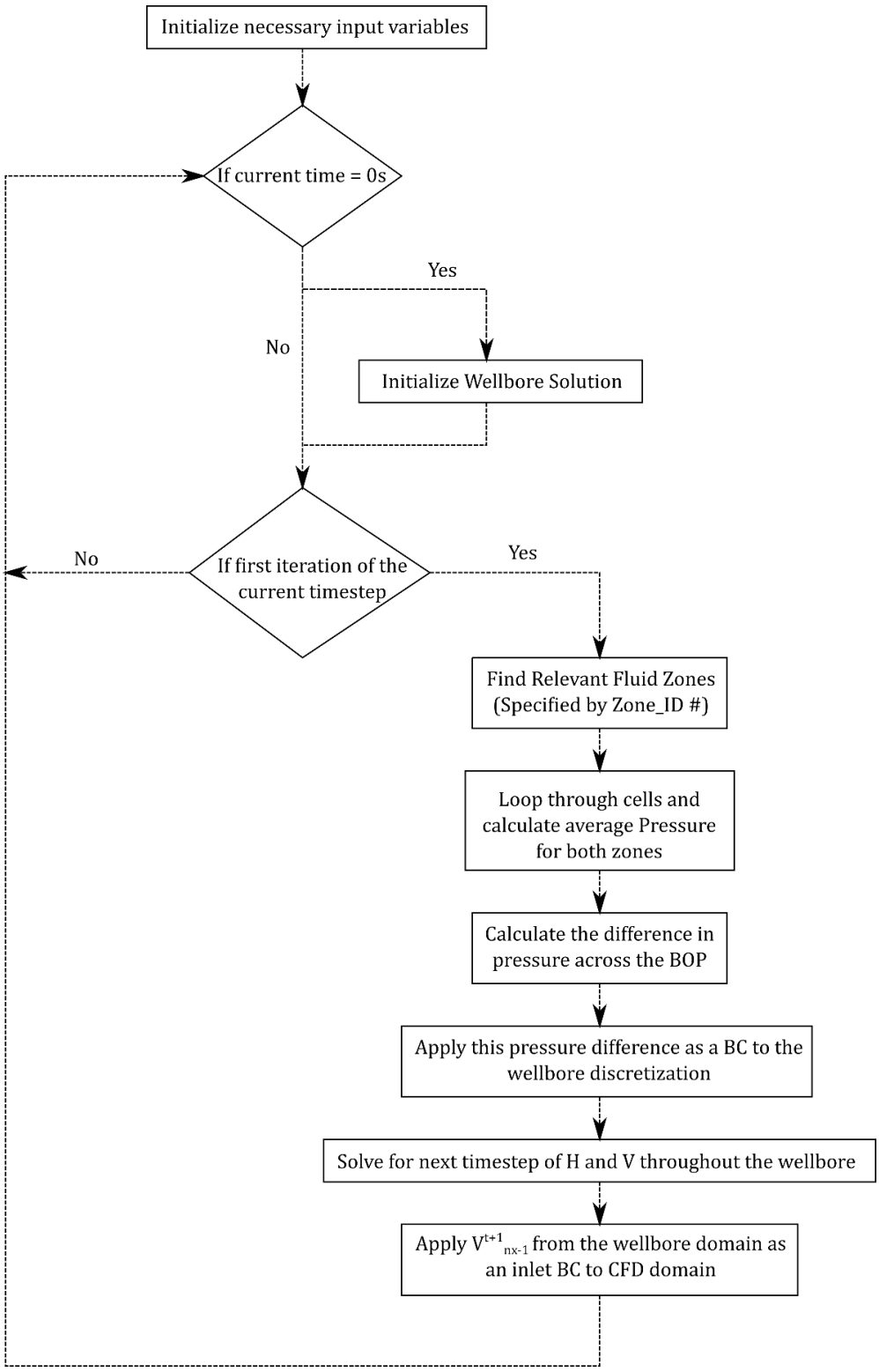


Figure 3.5. Algorithm of explicit coupling UDF for ANSYS Fluent boundary condition. The main loop containing pressure averaging continues until the final timestep is reached.

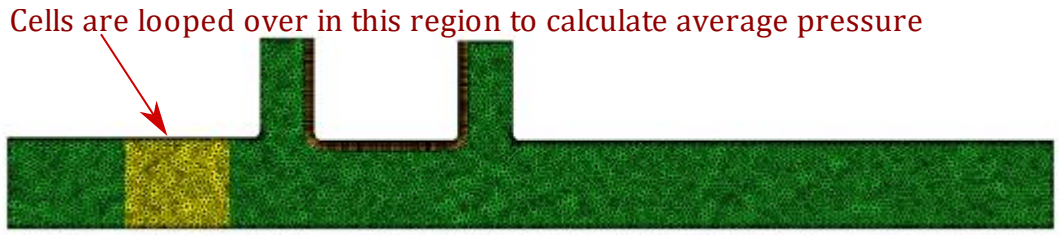


Figure 3.6. Averaged pressure region during simulations.

3.7. Dealing with Numerical Instability During the Coupled Simulations

Because the simulations are explicitly coupled, there may be stability issues. In most of the coupled simulations shown in this thesis, these stability issues caused large errors in the solution, particularly because the oscillations were amplified as the simulation progressed. An example illustration of these oscillations is provided in Figure 3.7.

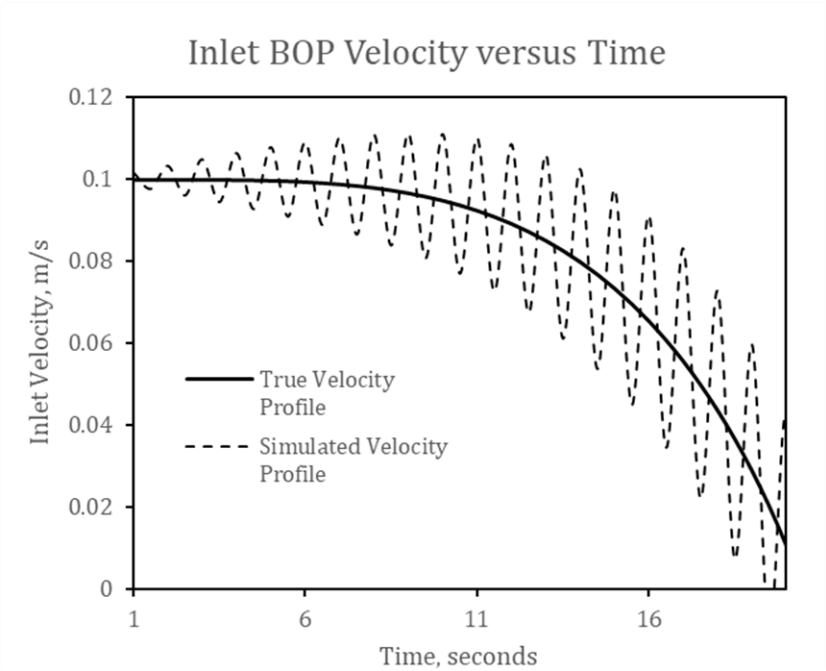


Figure 3.7. Oscillating inlet profile of velocity during ram closure.

Figure 3.7 is simply an illustration for visualization purposes, not data from one of the simulations. As shown in a later section of the thesis, the actual simulation results' oscillations are much more drastic, causing the simulations to diverge at early stages of the

simulation. Oscillations also amplify much more rapidly than shown here, causing the simulation to seem stable until sudden divergence.

The cause of these oscillations is most likely related to the explanation and illustration that follows. It is important to note that these dynamic meshing grid movements are done in timesteps. For example, a simulation may have a timestep of 0.01 seconds. Between 0.01 and 0.02 seconds, the grid will instantly move from one location to another. This will cause the simulation to recalculate a new higher pressure to which the wellbore will then respond with a new lower velocity. This oscillation between responses will continue to grow until the solution diverges.

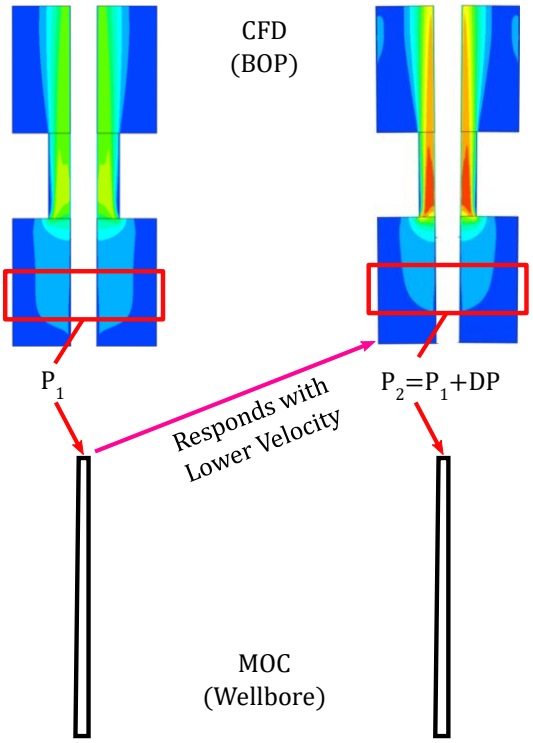


Figure 3.8. Oscillating nature of the explicit coupling procedure.

3.7.1. Some Recommended Approaches for Dealing with Numerical Instability

The following discussion is related to recommended approaches, which come from other researchers and published works. However, none of these were successfully

implemented in this thesis, either due to time constraints or failure of the method to dampen oscillations. Despite failure, the assumptions made in these approaches are more valid than this thesis works' current implementation, which will be discussed in the next section.

The first recommended attempt was to reduce $\Delta t/\Delta x$. While this may help stability in some numerical cases, reducing or increasing this ratio had no significant effect on damping oscillations. Another approach is to successively iterate until convergence is achieved between the two simulation domains. Again, no improvement in solution stability was achieved, though this may have been attributed to improper implementation of successive iteration. A final approach that may be considered is to include a 3rd simulation domain, as done in Dobroserdova et al. (2016). A 0-dimensional domain lies between the 1D and 3D domain. This domain consists of an elastic sphere that will deform and appropriately absorb pressure oscillations. Whether or not this approach is applicable to the closing BOP simulations still needs to be investigated.

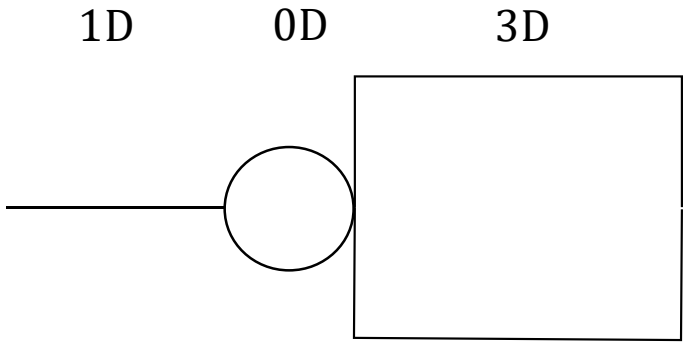


Figure 3.9. Illustration of a 1D-0D-3D coupling procedure that can be used to dampen oscillations.

3.7.2. Implemented Approaches for dealing with Numerical Instability

To dampen numerical oscillations, the work in this thesis utilizes the idea of the moving average. The approach is not entirely accurate but is straightforward in implementation. A simple moving average (SMA) can be calculated through:

$$V^t_{moving\ average} = \frac{V^t + V^{t-1} + V^{t-2} + \cdots + V^{t-n-1}}{n} \tag{36}$$

where n is the number of points chosen to include in the average. A larger number of points will increase the dampening effect for a selection of points. However, a larger number of points will also cause the moving average to move further from the true curve, as shown in Figure 3.10. Since all points are equally weighted, this SMA does not take into consideration that the true value should be heavily weighted towards V^t.

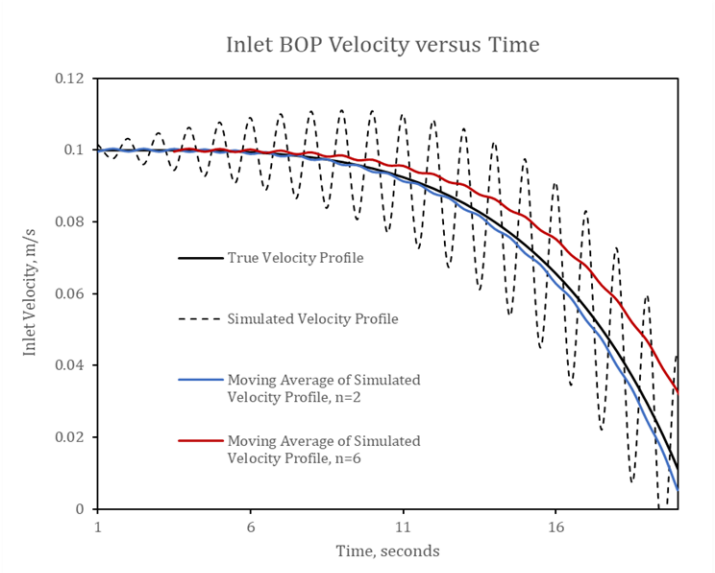


Figure 3.10. Demonstration of moving average in dampening oscillations for an inlet velocity profile.

3.8. Dealing with Changing Timesteps

There may be different desired timesteps throughout the entire simulation. For example, during the initial stages of the BOP shut-in process, pressures and velocities are changing slowly. A larger timestep may be used in these times to save on computation time. However, as the BOP rams near complete closure, pressures and velocities begin to rapidly change. Lower timesteps are needed here for stability of the CFD solution. The current implementation of the UDF requires that the 1D-MOC simulator runs at the same timestep

as the CFD simulation. When the 1D-MOC simulation changes in timestep, the grid size must also change due to the pre-defined relationship: $dx = dt * c$.

To deal with any changing grid size during the process, values are simply interpolated onto the new grid. Either spline interpolation or polynomial regression have been used, depending on if oscillations need to be dampened. An example of this is shown in Figure 3.11 and Figure 3.12.

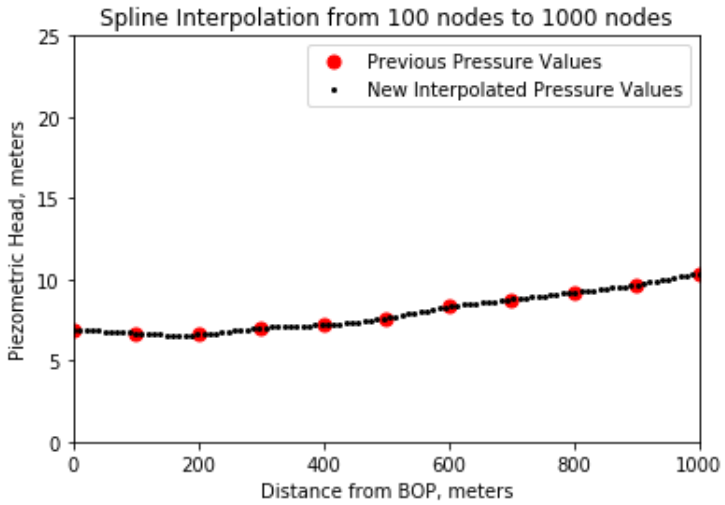


Figure 3.11. Interpolation to calculate a new array for piezometric head values (magnified view from 0-1000 meters).

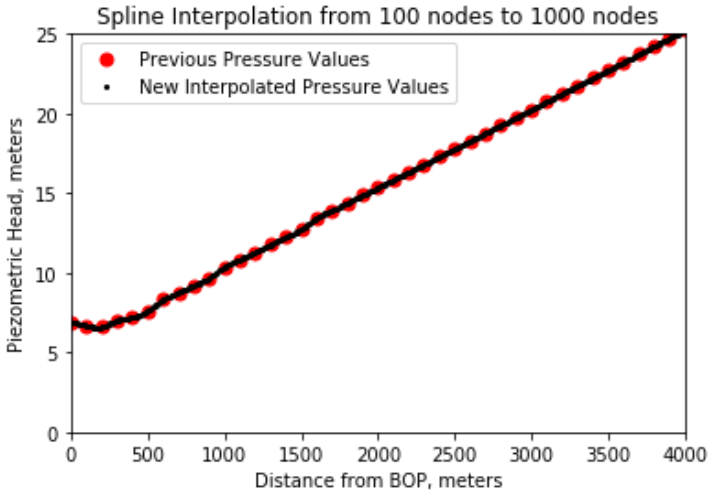


Figure 3.12. Interpolation to calculate a new array for piezometric head values (total view from 0-4000 meters).

Chapter 4. Simulation Validation, Results, and Discussion

4.1. Wellbore Simulation Validation and Sensitivity Analysis

To ensure that the water hammer simulation is giving reasonable results, basic validation and sensitivity analysis is demonstrated here. The water hammer effect is an increase in pressure when fluid flow is abruptly halted. This effect depends on several factors such as fluid velocity, closing valve time, and fluid properties. For an instant closure and perfectly smooth pipe the equation for change in pressure can be approximated as

$$\Delta P = \rho \Delta v C \tag{37}$$

where P is the change in pressure, v is the velocity of the fluid, is the density of the fluid, and C is the sonic velocity. Converted to piezometric head, this would be represented as:

$$\Delta H = \frac{\Delta v C}{g} \tag{38}$$

The simulation is set up to match this theoretical expectation. An instant closure can be simulated by setting the boundary condition at the valve inlet to 0 during the first timestep. Input parameters are described in Table 4.1.

Table 4.1. Input Values for Smooth Pipe Simulation

Sonic Velocity, m/s	1000
Pipe Length, m	100
Friction Factor	0
Initial velocity, m/s	10
Number of length subdivisions	100
Simulation Time, s	1
Pipe Diameter, m	0.1
Gravitational acceleration, m/s^2	9.81
Initial Piezometric Head at Valve, m	1000

From the information above, it can be calculated that theoretical expectations for the piezometric head increase from the water hammer effect should be 10*1000/9.81 m, or 1019 m. It is also apparent that the wave should reflect 5 times across the entire domain,

based on the given properties of sonic velocity, system length, and simulation time. A theoretical curve can be created from this known information. The results, shown in Figure 4.1, demonstrate that the simulation can accurately predict hydraulic transients throughout a smooth pipeline.

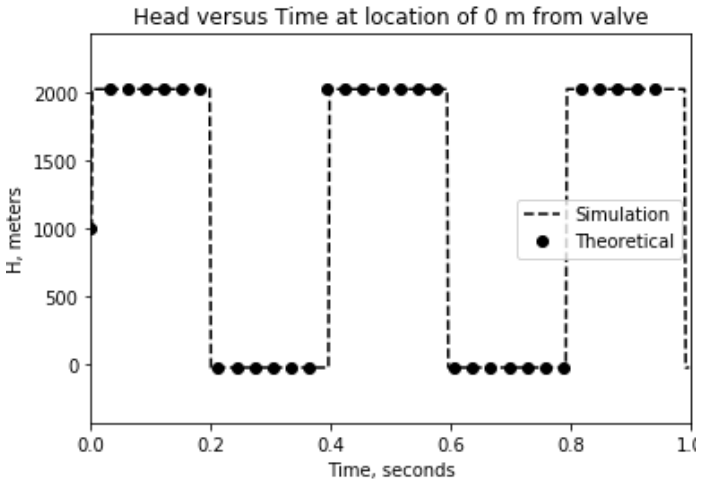


Figure 4.1. Simulation and theoretical simulation data for pressures at the valve location, using instant valve closure and smooth pipe

A sensitivity analysis is now performed to analyze parameters that may heavily affect fluid velocity during the shut-in process. A table of properties is given in Table 4.2.

Table 4.2. Input Values for Sensitivity Analysis

Sonic Velocity, m/s	Varies
Pipe Length, m	Varies
Friction Factor	0.01
Initial velocity, m/s	0.1
Simulation Time, s	40
Pipe Diameter, m	0.1
Gravitational acceleration, m/s^2	9.81
Initial Piezometric Head at Valve, m	1000

This wellbore simulation is not yet coupled to the CFD BOP domain. Therefore, a basic Piezometric Head profile is provided, shown in Figure 4.2. The function is simply representative of what the hydraulic profile should look like when coupled to the CFD

simulation. A simple exponential profile of piezometric head versus time is applied as the boundary condition in these cases.

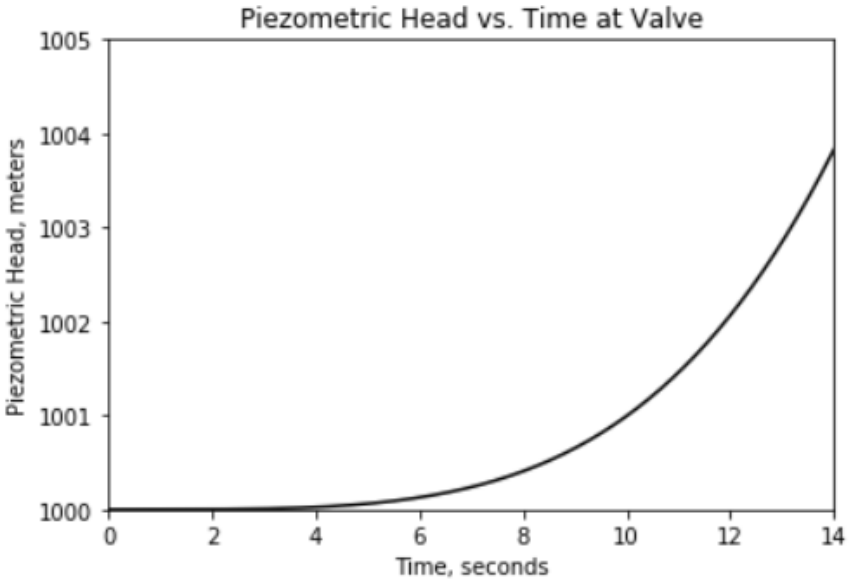


Figure 4.2. Hydraulic head boundary condition profile applied to simulations in Figure 4.3 and Figure 4.4

The effect of wellbore length is first investigated. Five different lengths are tested: 500, 1000, 2500, 5000, and 10000-meter wellbore lengths. A sonic velocity of 1000 meters per second is applied to all lengths. The results are shown in Figure 4.3. As expected, decrease in wellbore length results in a faster reduction in velocity. This reduction is due to the explanation provided previously in Figure 1.4. As the domain shortens, the pressure wave reaches upstream constant pressure reservoir at a shorter time. The constant reservoir pressure responds with a decrease in fluid velocity. This explanation is also consistent with the reason why no noticeable solution difference is observed between lengths of 5000 and 10000 meters. The sonic velocity in relation to total length is such that it will take either 5 or 10 seconds to even reach the upstream reservoir pressure. From this, it can be concluded that if sonic velocity is small in relation to the entire wellbore length, the change in length of

wellbore will have no impact on the solution. Figure 4.5 illustrates this, showing that the pressure wave will only propagate throughout a certain distance of the domain

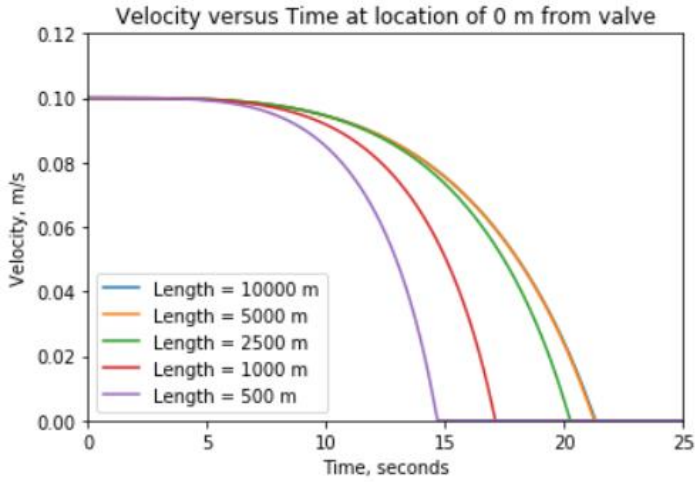


Figure 4.3. Length Sensitivity Analysis for Wellbore Simulation, c= 1000m/s

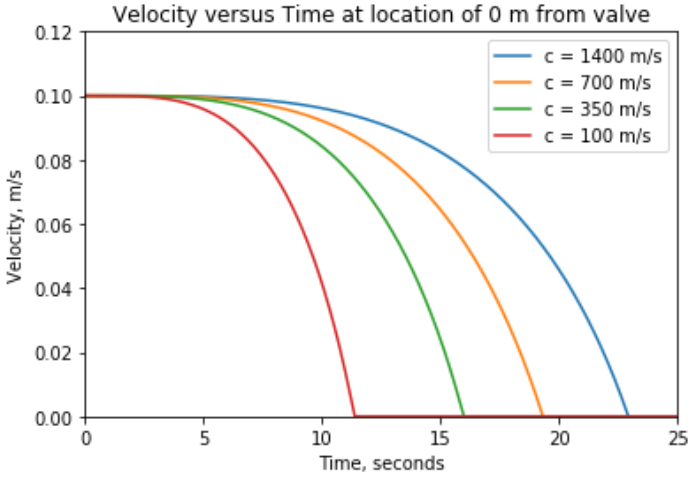


Figure 4.4. Sonic Velocity Sensitivity Analysis for 10,000 ft Wellbore Simulation

The effect of sonic velocity is then investigated. Four different sonic velocities are tested: 1400, 700, 350, and 100 meters per second. The wellbore length is chosen to be 10000 meters, such that sonic velocity will always be small in relation to total wellbore length. The results for varying sonic velocity are shown in Figure 4.4. As expected, a decrease in sonic velocity will result in more rapid reduction of fluid velocity, given identical boundary

conditions. This is consistent with the governing equations of fluid flow, where it can be seen that dV/dt is inversely related to c, sonic velocity.

While the water hammer effect is not large for slow closure of valves, it is important to note that these variables will also have an effect on the hydraulic head. As shown in Figure 4.6, these simulations can also be used to determine severity of water hammer throughout a system. A comprehensive study of water hammer is not completed in this thesis, but results can be easily stored from the UDFs and wellbore simulator described in this thesis.

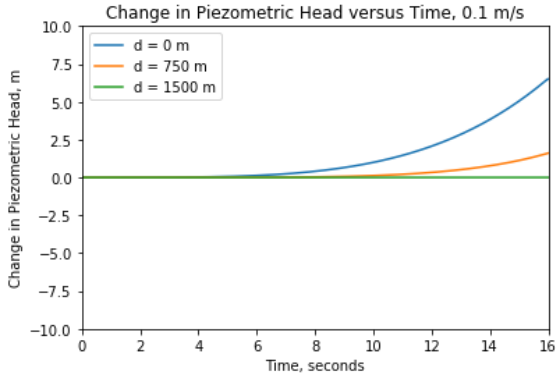


Figure 4.5. Visualization of pressure wave propagation throughout the wellbore. At 1500 meters, the pressure remains constant throughout the entire simulation.

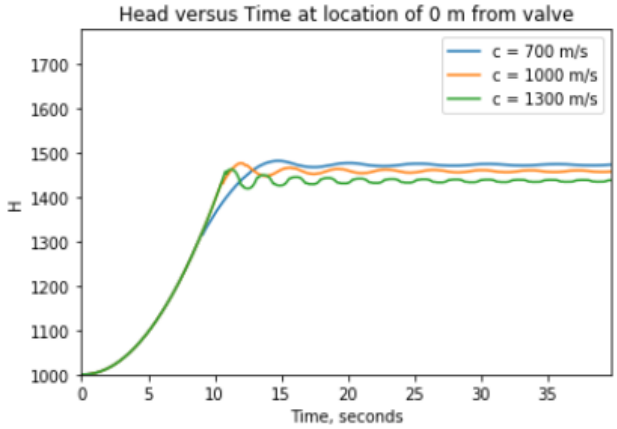


Figure 4.6. Impact of sonic velocity on the water hammer effect.

4.2. CFD (2D, 3D) Simulations Through Chokes and Valves Using Static Geometries

An important aspect of closing BOPs to model is simulation through narrow restrictions. Throughout the entire transient process, there is a point in time where velocity

throughout the BOP will be greatest. This will likely occur during a point in time that the BOP rams are nearing complete closure. Accurate modelling of this process is important for several reasons:

- Large fluid velocities during this time will cause high shear rates at the ram walls. This information provides useful insight towards erosional analysis.
- Conditions at which the geometry chokes can be analyzed if the process is accurately modelled. For example, the conditions at which multiphase flows choke through various BOP geometries can be analyzed.
- Cavitation may occur for single phase fluids depending on fluid velocity and BOP geometry. Cavitation will contribute to erosion and may choke flow.
- Significant mass transfer between two fluid phases may occur during this process, depending on the fluid properties and pressure conditions at the BOP entrance.

For simulation validation, CFD simulations are compared against experimental data of flow through chokes from Omana (1969). The researchers ran numerous experiments of gas and water flow through various choke sizings. A pressure is specified upstream and downstream the choke, then liquid and gas flowrates are measured. Pressures upstream and downstream the choke were varied from 400 to 1000 psi. Liquid flowrate was varied from 0 to 800 barrels per day. Experiments were run for single phase water, single phase gas, and multiphase water/gas simulations at various gas/oil ratios. Choke sizes varied from 4/64" to 14/64". Pipe diameter of the upstream and downstream the choke was 2". Gas gravity for this particular experiment was 0.611. Other gas related properties can be found in the paper.

A geometry is created in ICEM CFD for comparison against flow through 10/64" chokes. The geometry shown in Figure 4.7 illustrates the setup. The design is assumed to be

axisymmetric, and flow is assumed to be steady state. Fluid properties for that of the experimental studies are used in the CFD simulation. The boundary conditions used for the simulation are velocity inlet and a specified gauge pressure outlet. For turbulence modelling, the realizable k-epsilon turbulence model is used. The importance of this choice is demonstrated later through results. The mesh, shown in Figure 4.7, is designed such that y+ values will be within the range of 30-100 at both the ram and pipe walls. A finer mesh is used downstream the valve to capture jet flow features.

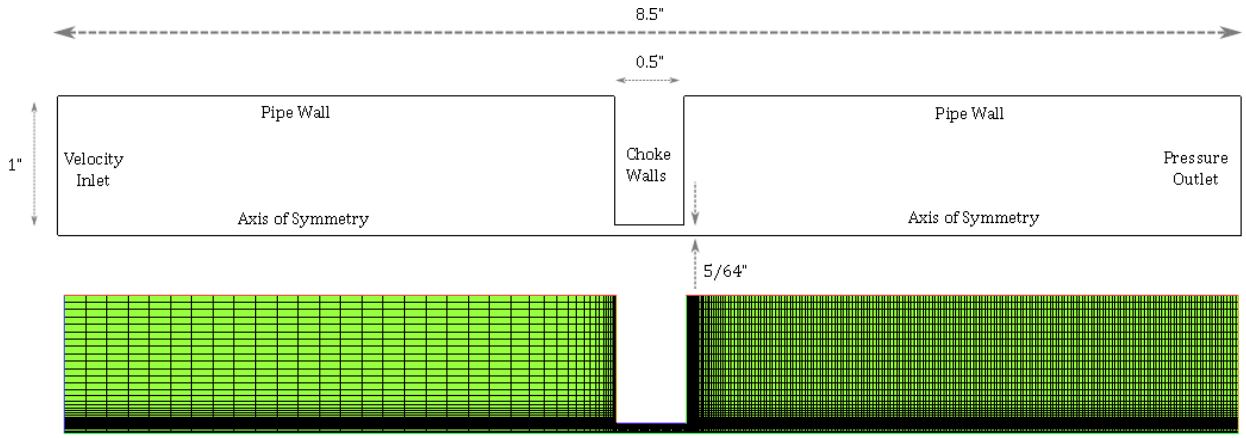


Figure 4.7. Grid and Geometry for Narrow Restriction Simulations

Table 4.3. Input velocities and simulated pressure ratios for experimental comparison where P_d is downstream pressure and P_u is upstream pressure.

Liquid Flowrate, bbl/day	Equivalent Velocity, m/s	P_d/P_u
226	0.205	0.718
383	0.347	0.488
490	0.444	0.368
572	0.519	0.286

Simulation results against single phase water experiments are initially compared. 4 data points are recorded for the 10/64" choke for single phase flow in the study by Omana (1969). The experimental results are summarized in Table 4.3. Further details of the validation are shown in Table 4.4.

Comparison of simulation to measured values are plotted in

Figure 4.8. All simulation points lie within a 10% error range, demonstrating strong similarity to experimental results. Exact match of simulation results is not expected due to geometry differences. Velocity contours and pressure contours are shown in Figure 4.9 for a simulation of inlet condition 572 bopd. Patterns for the contours demonstrate nearly identical structures, thus only one of the contour maps is shown here. The primary difference between these contours lies in the magnitudes of values.

Table 4.4. Input values with associated calculated max velocity, max pressure, and minimum pressure.

Liquid Flowrate, bbl/day	Max Pressure (Pa)	Min Pressure (Pa)	Max Velocity (m/s)
226	2.99e+6	1.386e+6	52
383	4.68 e+6	6.237e+4	88
490	6.32 e+6	-1.222e+6	112
572	7.91 e+6	-2.460e+6	125

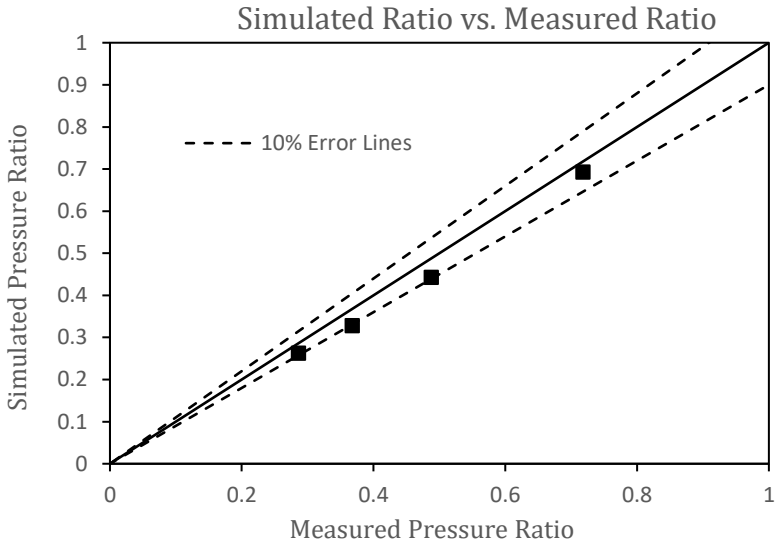


Figure 4.8. Simulation results for pressure ratio, solid line representing exact results from experimental work.

Geometry of the chokes in the study by Omana (1969) were not completely described. For example, choke width is not specified. A value of 0.5” is assumed for the the simulation domain. Additionally, exact geometry of the choke is not specified either. For example, the

chance that the experimental geometry is purely axisymmetric is unlikely. These geometry differences will cause a difference in valve coefficient, C_v, meaning that pressure drop across the choke will slightly differ for varying flowrates. Unless the valve coefficient is matched through pure chance, differences between simulation and experimental results are expected.

An important aspect of the pressure contours to highlight is the negative pressure values displayed for flowrates of 490 and 572 barrels per day. The pressure contours represent gauge pressure; however, even in adding atmospheric pressure to either of these solutions, it is clear that total pressures will still be negative. These low-pressure regions are caused by flow separating from the sharp-edged orifice, as can be seen in the velocity contours. Despite this, negative absolute pressures are physically unrealistic. Thus, if simulation results are accurate, it is implied that cavitation must be occurring.

A simulation is set up with mass transfer between water and water-vapor. A vapor pressure of 3000 psi is used for the simulations, and flowrate is specified for 570 barrels per day. Volume fractions are displayed in Figure 4.11, demonstrating that cavitation exists for this scenario, as expected. While it is possible to determine the existence of cavitation simply by viewing pressures, shear rates along walls will not be accurately modelled without incorporating a cavitation simulation model.

Another important aspect to highlight is choice of turbulence model. Differences between standard and realizable k-epsilon are described in a previous section. The impact of that choice is demonstrated here. Two simulations are run at the exact same boundary conditions. The only difference between these two simulations is the turbulence model chosen. One of the simulations is run with standard k-epsilon and the other is run with the realizable k-epsilon model. Again, looking at Figure 4.9 and Figure 4.10, the comparison

between these two models can be observed. The difference in overall pressure distribution is minimal. However, from the results, it is clear that the standard k-epsilon model will severely under predict the degree of flow separation from the wall. While this impact may not seem significant at first, it is important to analyze the physical variable that this will affect. Shear stress at the walls are plotted for three different models, shown in Figure 4.12. Between the three models, a major impact in shear rate is demonstrated. Therefore, when modelling something where shear rates have an impact, careful selection of multiphase and turbulence models should be employed. Turbulence model choice does not have as significant of an impact on the pressure values shown in Table 4.4 and

Figure 4.8, so simply comparing simulation results to pressure values may seem misleading. Unfortunately, pressure values are the most commonly reported data for these experiments, as shear rates along walls are not as simple to measure.

Additional analysis of turbulence models needs to be completed. While it has been demonstrated that realizable and standard k-epsilon models will produce different shear stresses, the results shown here do not imply which (if either) model will produce accurate results. Turbulence models that do not require the use of wall functions in ANSYS Fluent, such as the k-omega model, may be more applicable in these scenarios. However, in absence of wall functions, the strict meshing requirements near the walls are not practical for use in the dynamic meshing scenarios. Using a more appropriate turbulence model that does not require wall functions, some steady state simulations may be run at different time snapshots during the transient processes to determine shear rates along the walls.

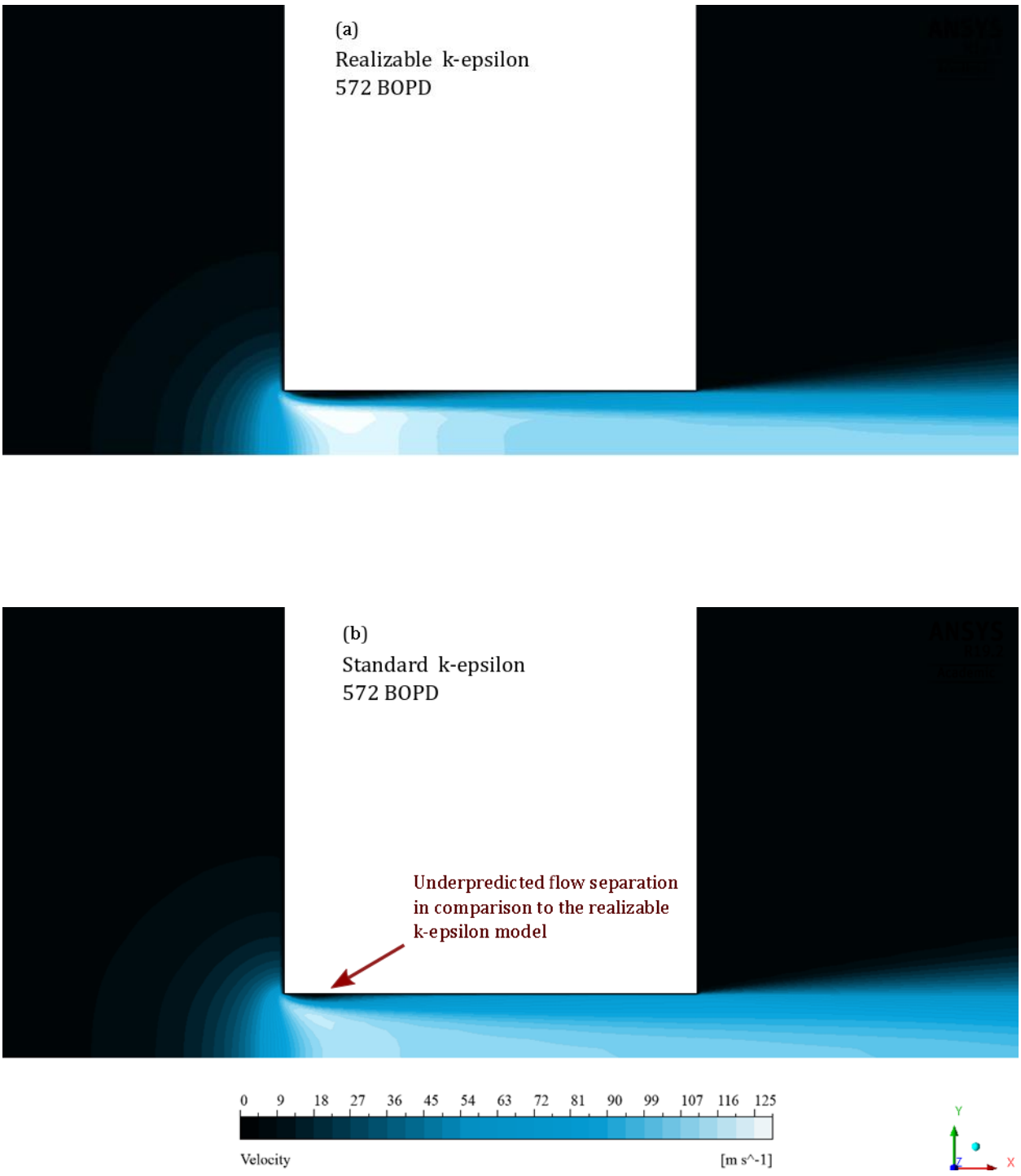


Figure 4.9. Velocity contours for inlet velocity of 0.519 m/s at restriction of the choke using: (a) realizable k-epsilon model (b) standard k-epsilon model.

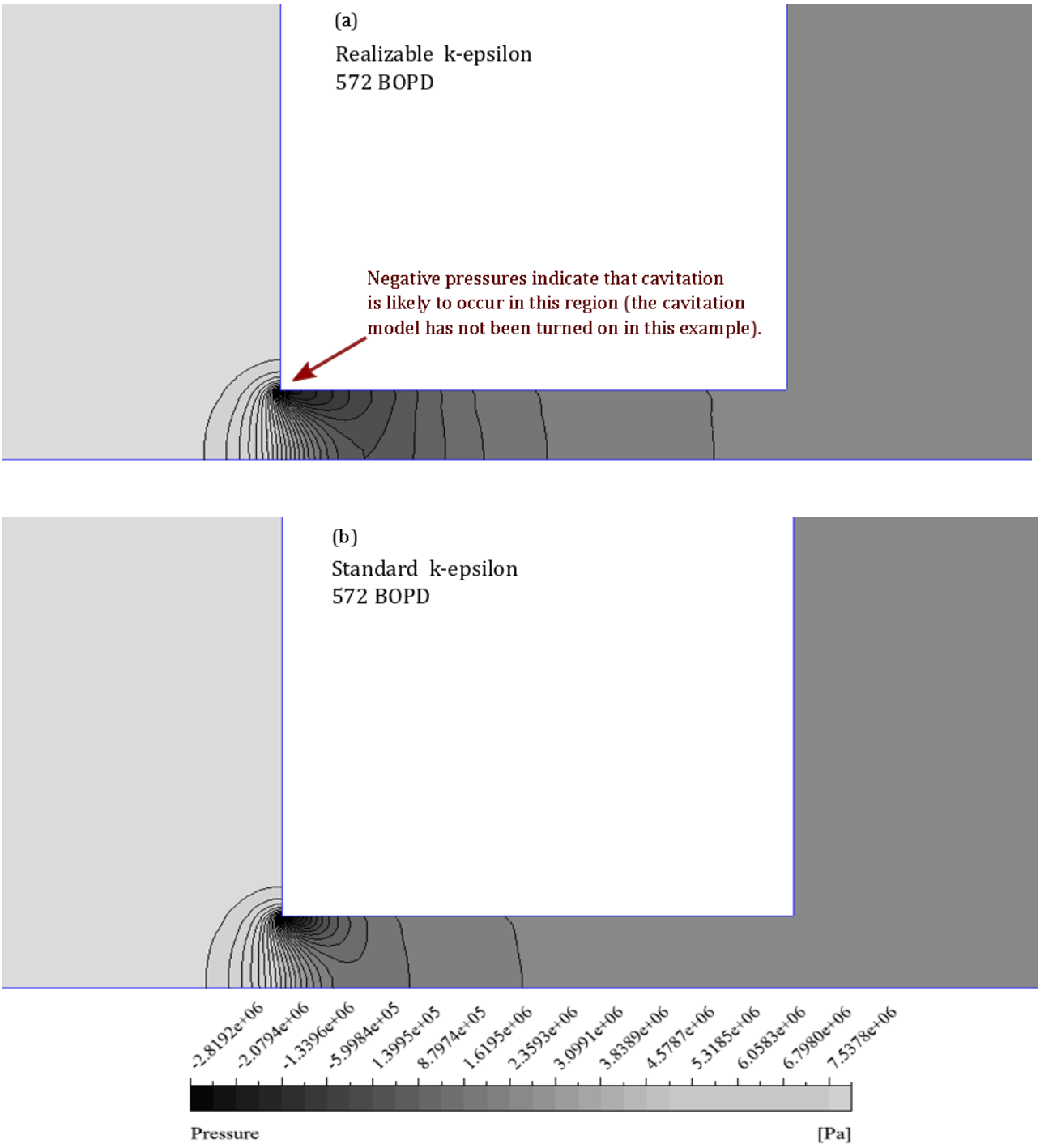


Figure 4.10. Pressure contours for inlet velocity of 0.519 m/s at restriction of the choke using: (a) realizable k-epsilon model (b) standard k-epsilon model.

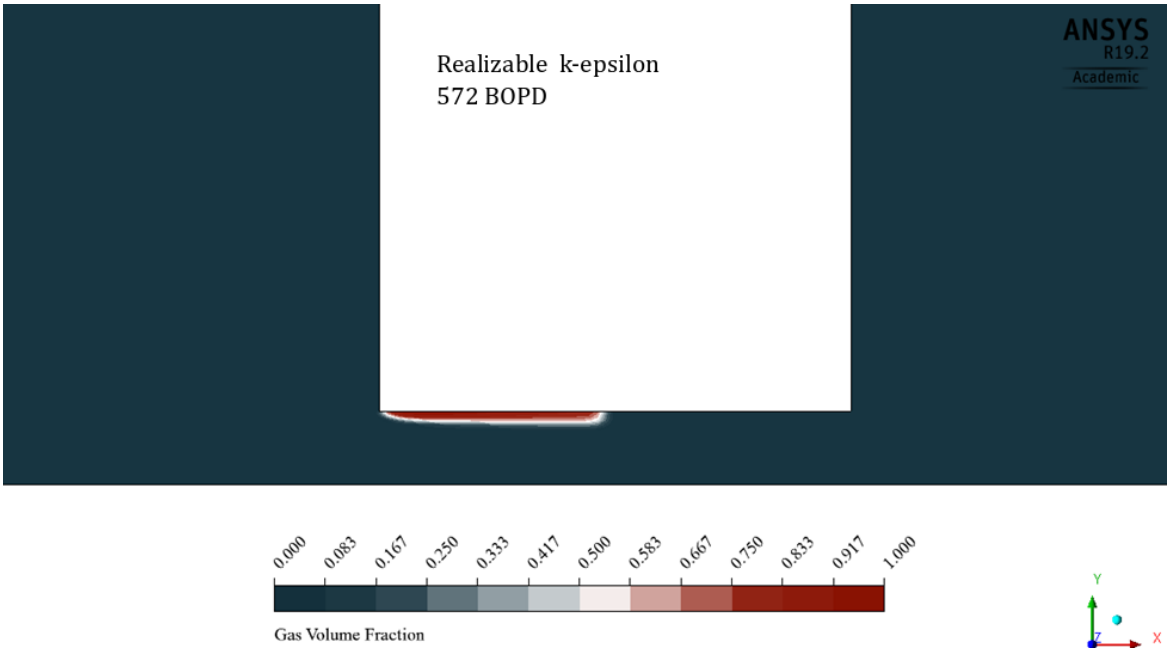


Figure 4.11. Volume fraction contours for inlet velocity of 0.519 m/s at restriction of the choke using the realizable k-epsilon model and the Schnerr-Sauer cavitation model for mass transfer between water-liquid and water-vapor.

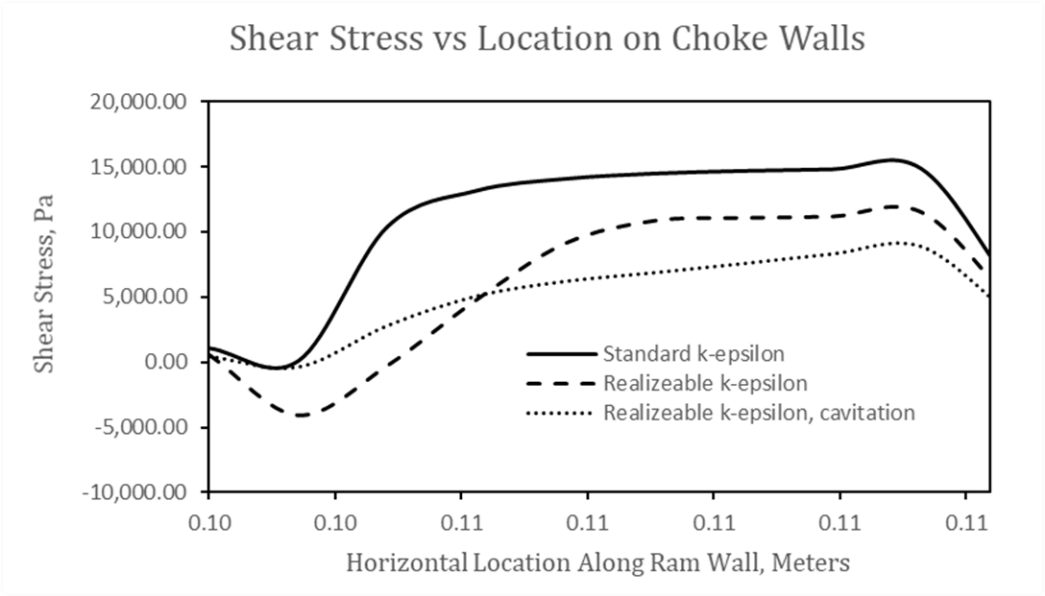


Figure 4.12. Shear stress along restriction walls with 0.519 m/s inlet, comparison of 3 modelling assumptions: single phase standard k-epsilon, single phase realizable k-epsilon, and realizable k-epsilon with mass transfer between water-liquid and water-vapor.

4.3. CFD Simulations Through Closing Valves without a Coupled Boundary Condition

4.3.1. CFD Simulation of Fast Valve Closure

While the coupled MOC-CFD simulations provide benefits in case of extremely long domains, this coupling is not always necessary. In the case of shorter domains, closure of valve may be simulated with a single CFD domain. An example validation study is provided to demonstrate this. Bergant et. al (2001) conducted various experiments to study water hammer through pipes. A CFD simulation is compared against these experimental results. Since the length is significantly longer than the diameter, the domain and mesh are not shown here due to the difficulty in visually presenting the geometry.

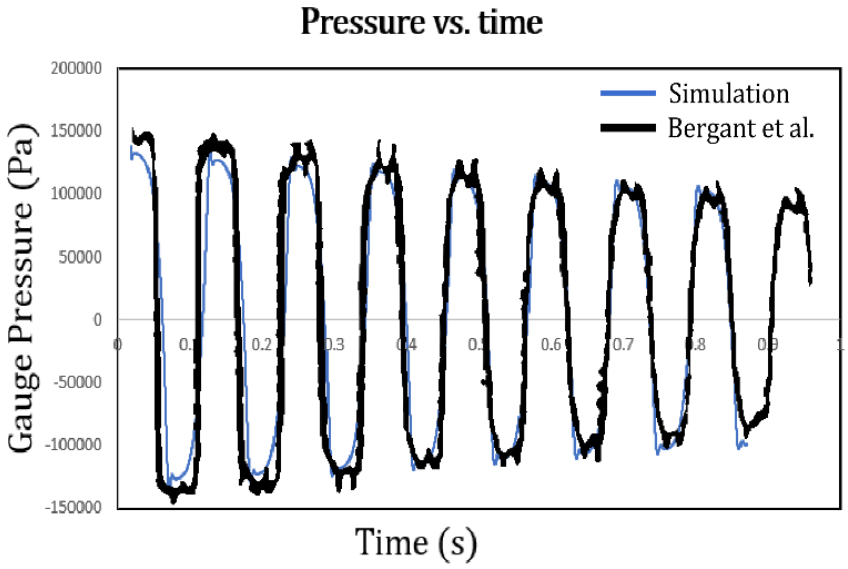


Figure 4.13. Closing valve simulation comparison to experimental data for fast valve closure, with entire pipe and valve modelled as axisymmetric 2D domain for the simulation.

The geometry is a 37.23-meter pipe with a diameter of 22.1 millimeters. Water is flowing at 0.1 meters per second throughout this pipe until steady state is achieved. Afterwards, the valve is closed within 0.007 seconds. The wave speed is approximated to be 1319 meters per second. For the CFD simulations, the domain is approximated by an axisymmetric assumption. The valve is closed using the layering technique, and the pressure

below the closure is recorded and compared to an experiment from Bergant et al. (2001). As shown in Figure 4.13, pressure values match reasonably well.

While these single domain simulations cannot be replicated for lengths of wellbore, they demonstrate accurate modelling capabilities of dynamic meshing techniques. Additionally, other closing valve applications throughout the oil and gas industry may be able to take this approach if the length of pipe is short enough. Using a single domain avoids issues arising in coupling of solutions, such as numerical instability, though it will always be more computationally expensive than coupling CFD with a 1D simulation.

4.3.1. Examples of Closing BOP Simulations with Constant Inlet Boundary Condition

While some sort of coupled simulation (one-way, explicit, or fully implicit) is needed to provide accurate inlet boundary conditions, valuable results can still be gained from "incorrect" boundary conditions. For example, simulations can be run with constant pressure or velocity inlet boundary conditions to analyze the flow field and compare designs. Areas of flow recirculation, pressure distributions, and forces on rams can be qualitatively investigated throughout the entire simulation. For any particular time instance, snapshots of the solution may be also be analyzed.

Some example simulations using a constant inlet velocity boundary condition are shown in Figure 4.14. One design is a rough representation of pipe rams, and the other is a rough representation of shear rams. A snapshot at 66% closure is shown for each. Areas of recirculation, maximum velocity, and maximum pressure can be viewed, giving insight to features relevant to the BOP ram design process.

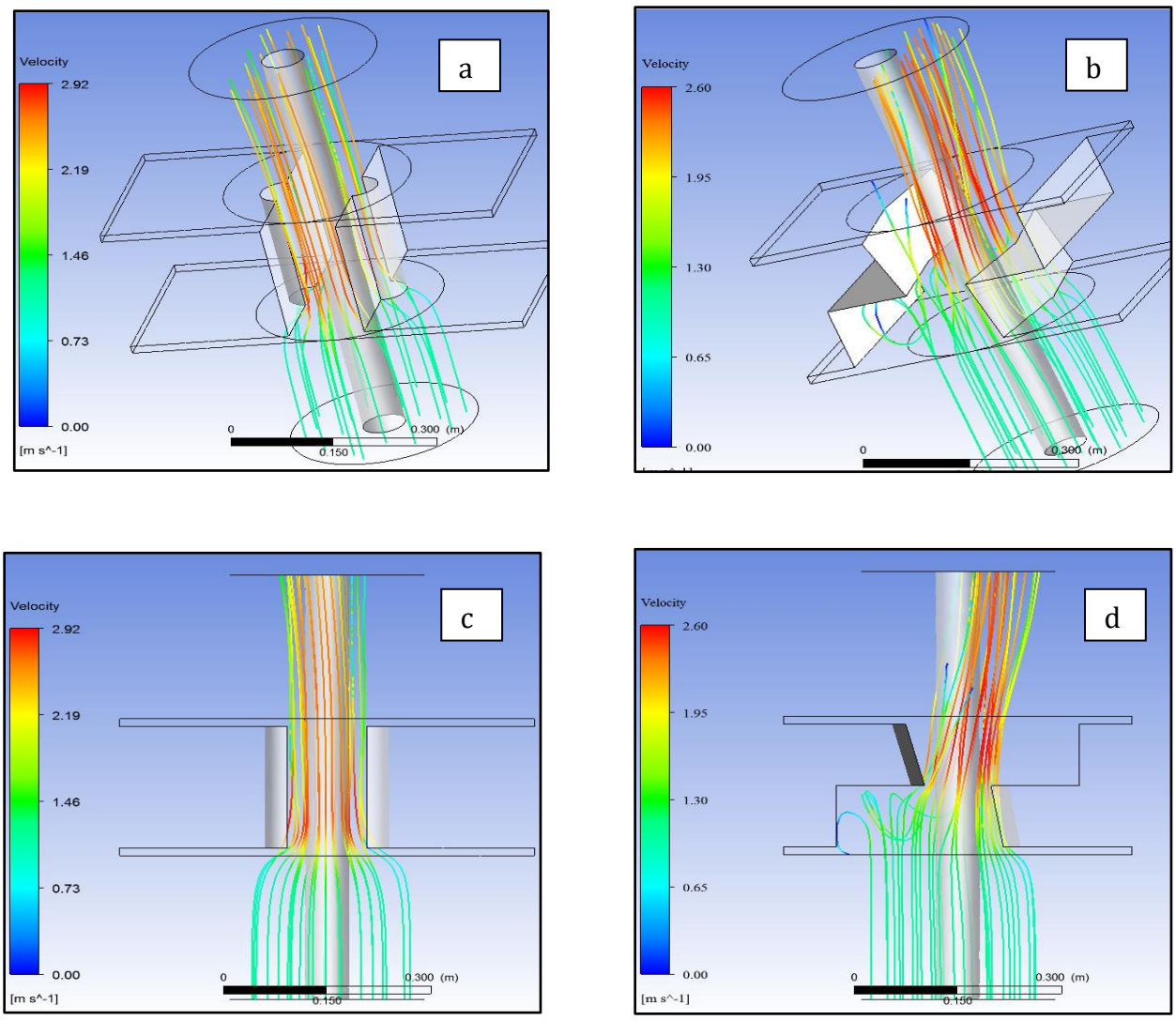


Figure 4.14. Streamlines at 66% closure with a constant inlet velocity of 1 m/s for (a) isometric view of pipe rams (b) isometric view of shear rams (c) side view of pipe rams (d) side view of shear rams.

4.4. MOC(1D) – CFD(3D) Coupled Simulation Results

All previous results shown in this thesis were from simulations that had not yet been coupled. In this section, demonstration of simulation coupling will be shown. Additionally, example cases related to simulation and mesh design are demonstrated. A 2D and 3D case are demonstrated here. The 2D case is meant to be an axisymmetric representation of a closing annular preventer. The 3D case is presented to demonstrate the same process for a

more geometrically complex case, being a shear ram for this work. The same process could ideally be applied to any geometry.

4.4.1. Axisymmetric 2D Coupled Simulation

An 8” thick closing annular preventer is simulated using smoothing and remeshing. The geometry and mesh are shown in Figure 4.15. The region of orange prism layers moves towards the axis of symmetry to close on the pipe.

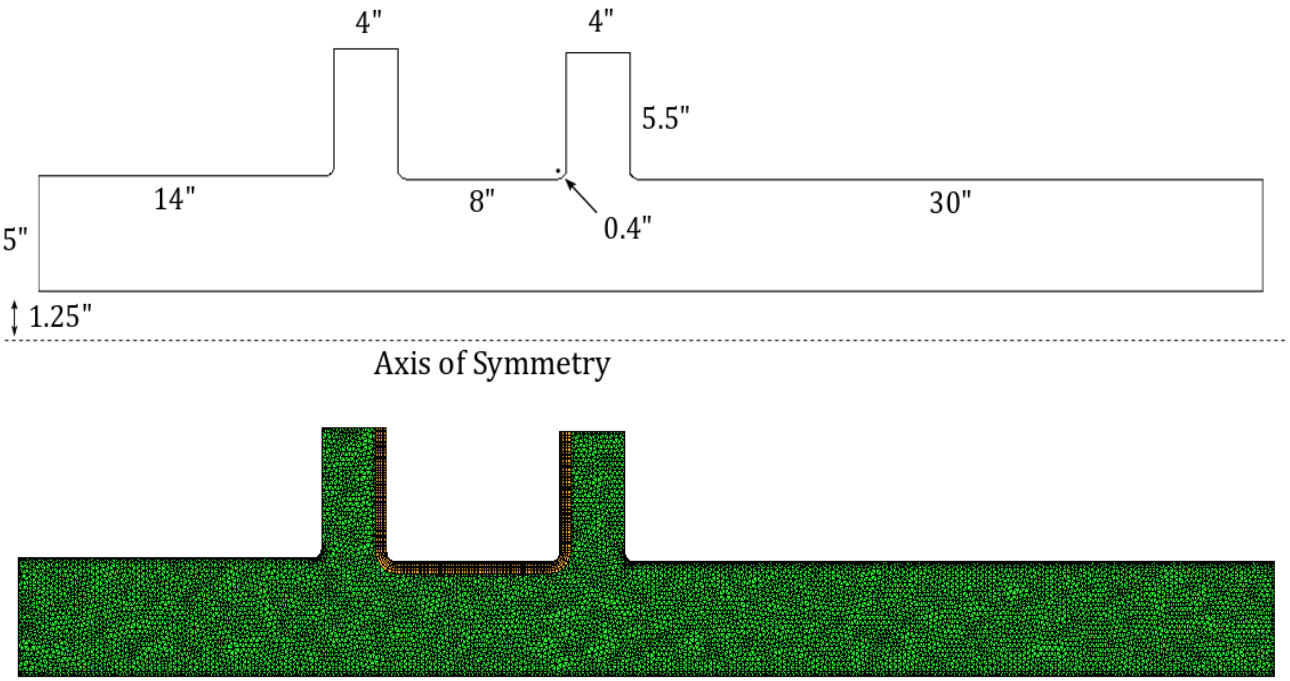


Figure 4.15. Geometry and mesh for axisymmetric simulations.

To ensure high mesh quality and proper near wall refinement as described by y+ values in a previous section, three successive meshes are set up for three different stages of the simulation. The total BOP closure time is 18.2 seconds, with a constant ram velocity. The first mesh simulates to 13.00 seconds, the second mesh is used until 15.77 seconds, and the third mesh is used until 17.98 seconds. This corresponds to around 98.8% closure. The input properties are shown in Table 4.5.

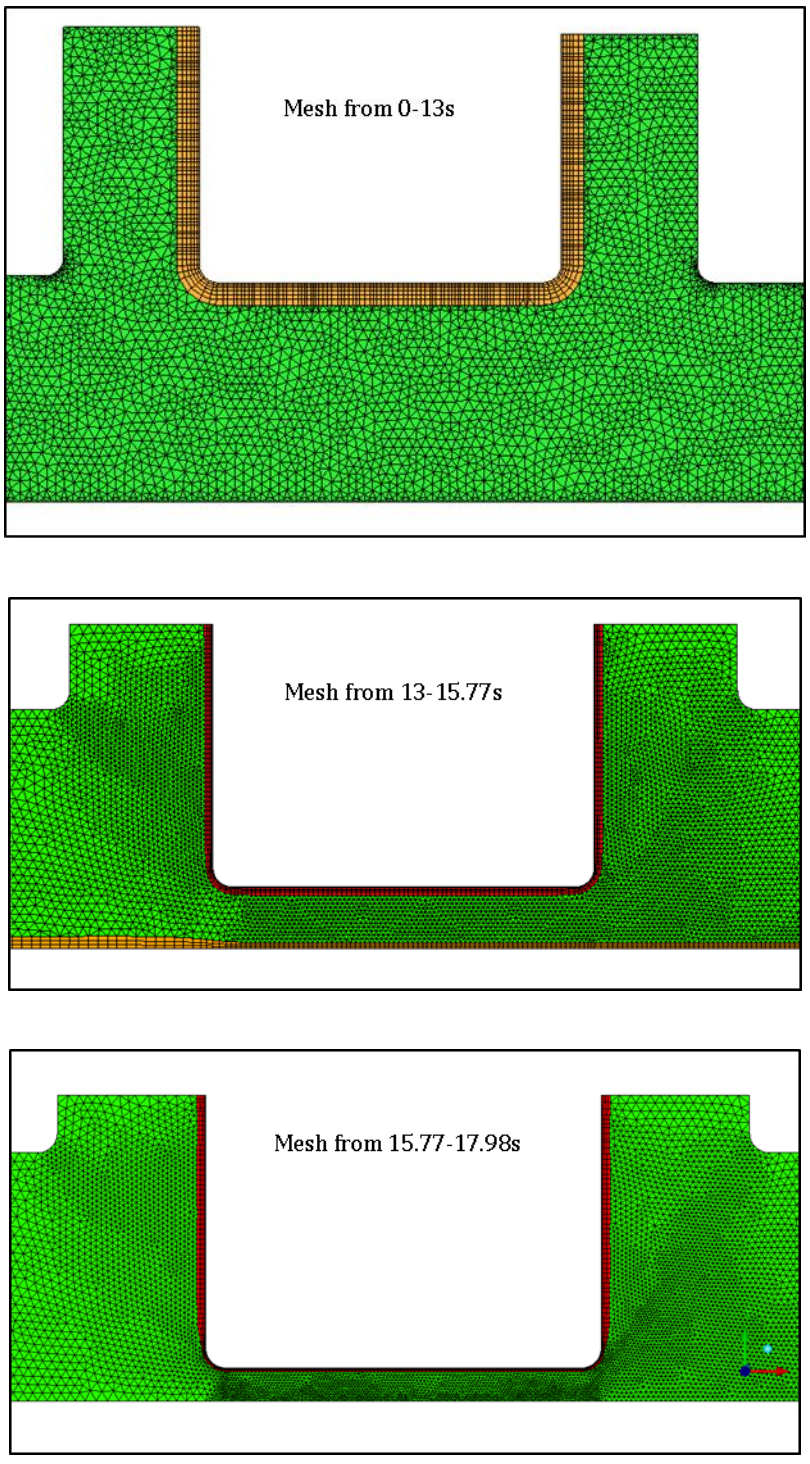


Figure 4.16. Multi-stage meshes at different times throughout the simulation.

Figure 4.17 shows the inlet velocity profile over the transient simulation up to 98.3% closure. Some minor discontinuities exist between simulation runs. This may be caused by a few issues in the problem setup. For example, solution values may have been incorrectly interpolated across each mesh. Additionally, successive geometries may not have been perfectly aligned. Issues such as these need to be investigated before proceeding with these simulations.

Table 4.5. Input Values for Axisymmetric Simulation.

Property	Value/Description
Initial Fluid Velocity, m/s	1.00
Fluid Density, kg/m^3	800.00
Fluid Viscosity, $kg/(m*s)$	0.01
Valve Closure Time, seconds	18.20
Residuals (all)	10^{-5}
Initial Cell Count	~100,000
Initial Timestep (Mesh#1), seconds	0.1
Final Timestep (Mesh#2, Mesh#3), seconds	0.01
Turbulence Model	Realizeable $k-\epsilon$
Wall Function	Standard
Wellbore Length, meters	10,000
Sonic Velocity (Wellbore), m/s	1000
Equivalent Wellbore Diameter, m	0.1
Wellbore Friction Factor, dimensionless	0.01
Moving Average Number Applied for Stability	7

The concept of a moving average is applied in this simulation to ensure stability of the solution. A simulation is initially run without using any moving average. As can be seen in Figure 4.18, the solution will not maintain stability, diverging before complete closure. Another simulation is run with moving average, n=2:

$$v^t_{averaged} = (v^t - v^{t-1})/2 \quad (38)$$

It can be seen that the solution maintains stability for 0.5 seconds longer, but again diverges before complete closure. A final simulation is run with moving average, n=7, and solution stability is achieved throughout the entire simulation.

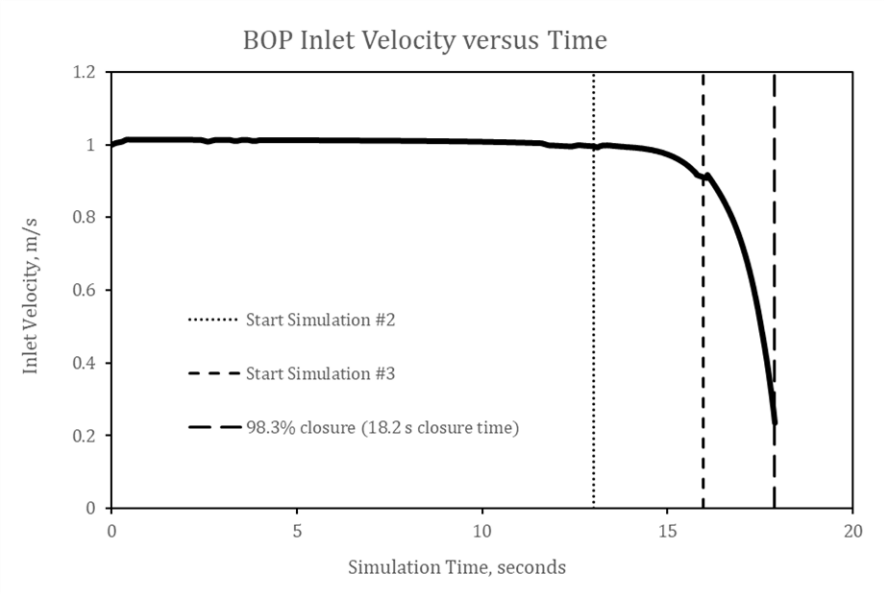


Figure 4.17. Calculated inlet velocity profile from a coupled simulation over the multi-stage simulation process

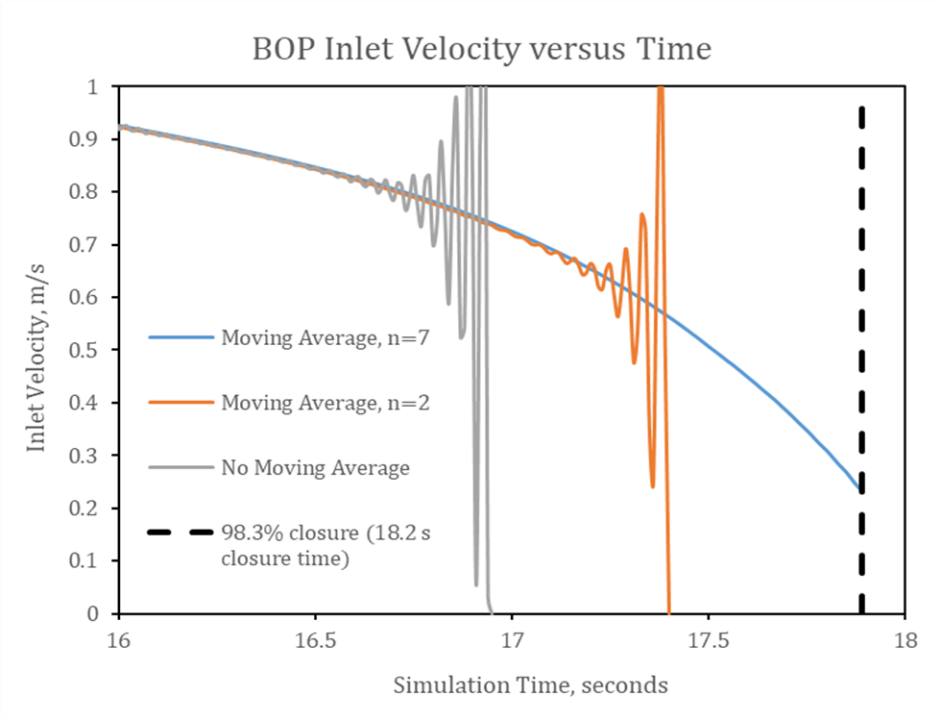


Figure 4.18. Demonstration of moving average effectiveness in dampening oscillations. Time scale is from 16 to 18 seconds.

The impact of boundary condition choice can also be analyzed by comparing the coupled simulation to constant velocity and pressure boundary conditions. Two more simulations are run. The first has a constant inlet velocity of 1 m/s. The second has a constant inlet pressure boundary condition that provides an initial 1 m/s velocity. As is expected, the constant velocity boundary condition will drastically overpredict rates, and the constant pressure boundary condition will drastically underpredict rates.

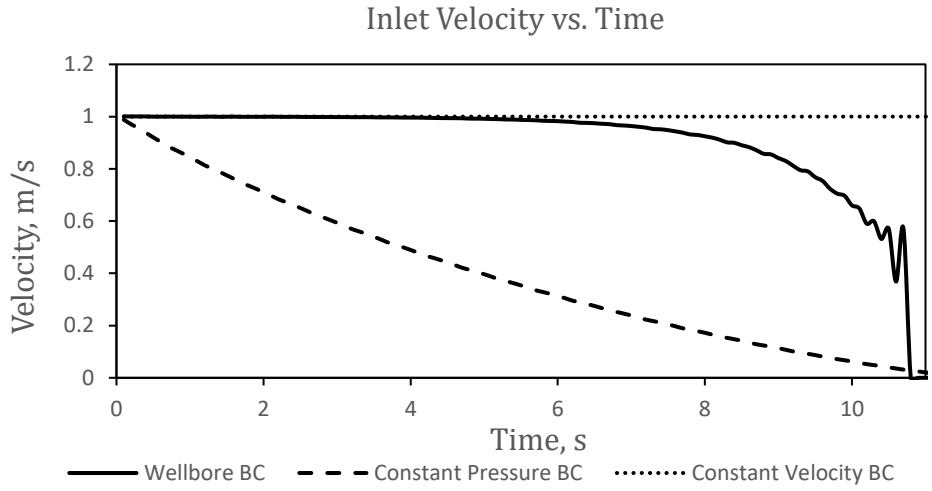


Figure 4.19. Comparison of inlet velocity for different inlet boundary conditions.

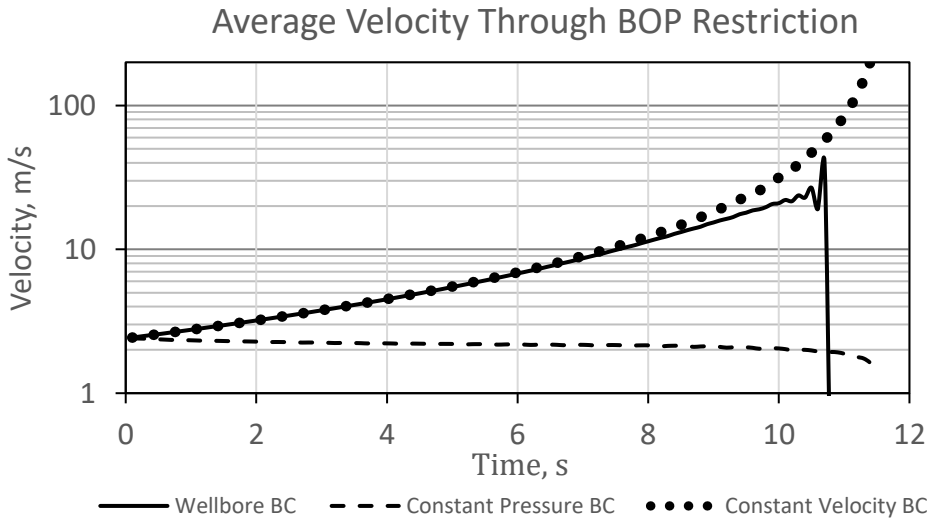


Figure 4.20. Comparison of average BOP velocity for different inlet boundary conditions.

Information that is not extensively discussed in this thesis is the pressure and velocity values throughout the wellbore. While not particularly interesting for these simulations, pressure profiles throughout the wellbore are simple to record and export as well, shown in Figure 4.21. Information such as this could be valuable in determining pressure head increases during more sudden closures.

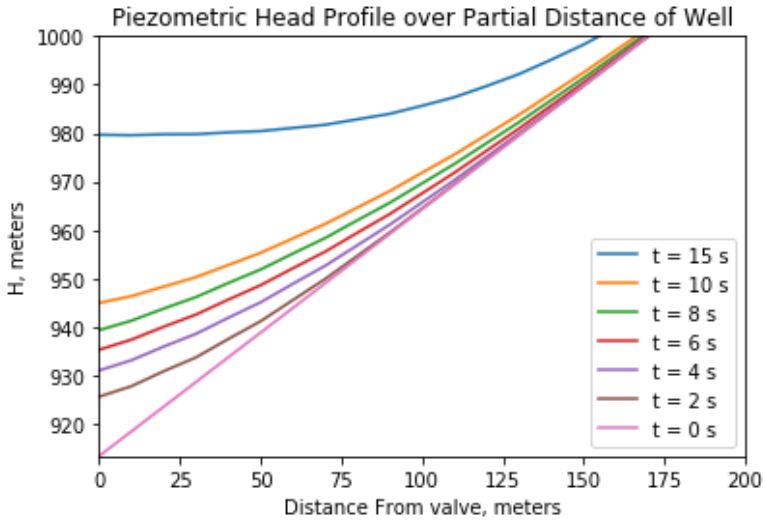


Figure 4.21. Piezometric head profile at different times during the transient simulation.

While the idea of successive meshes may seem tedious to complete, the influence that a proper mesh has on results is significant enough to justify the extra effort. Another simulation for a single mesh is run and compared against the multi-stage simulation. This unresolved mesh consists of the exact same geometry and mesh as in Figure 4.15 except with prism layers removed. Despite the single mesh's residuals indicating convergence (10^{-5}), results will still differ significantly from a properly resolved mesh. Four variables are compared between the simulations: Average y+ along the ram wall, average velocity through the BOP, average shear stress, and average pressure below the BOP. These values are shown in Figure 4.22, Figure 4.23, Figure 4.24, and Figure 4.25. It is also important to note that these single-mesh simulations are also run with the dampened SMA boundary condition.

Therefore, the undampened irregularities in the single mesh results will be even more drastic than the results shown. These irregularities in the single-mesh simulations are not due to unstable numerical solution, but instead are due to undesirable remeshing in critical areas.

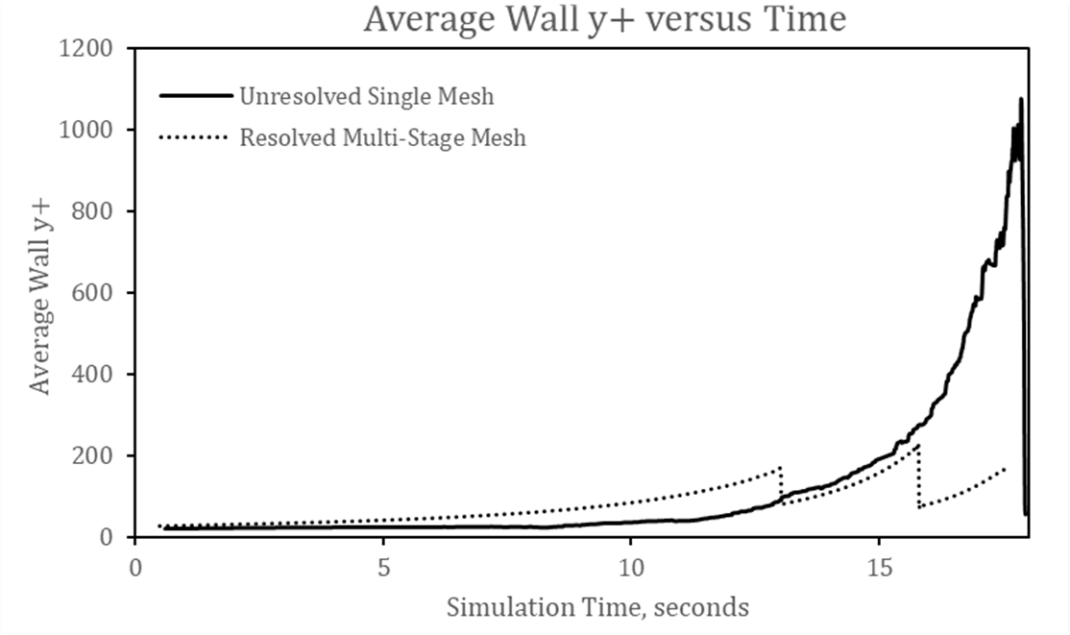


Figure 4.22. Demonstration of y+ management using a multi-stage mesh process.

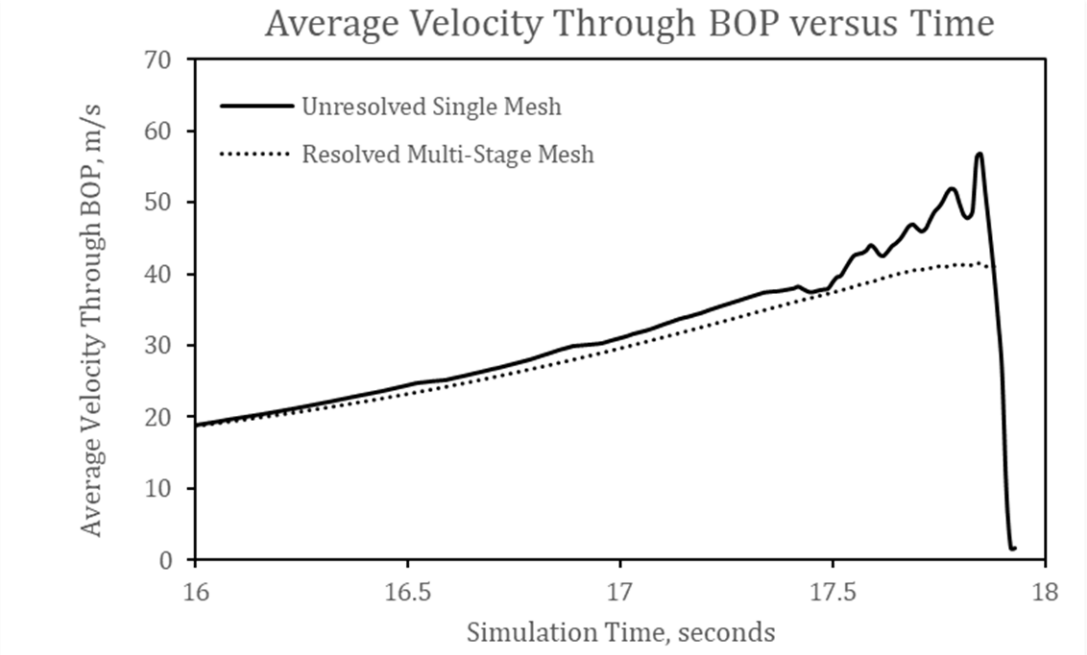


Figure 4.23. Demonstration of erroneous velocity values through BOP for poor quality mesh.

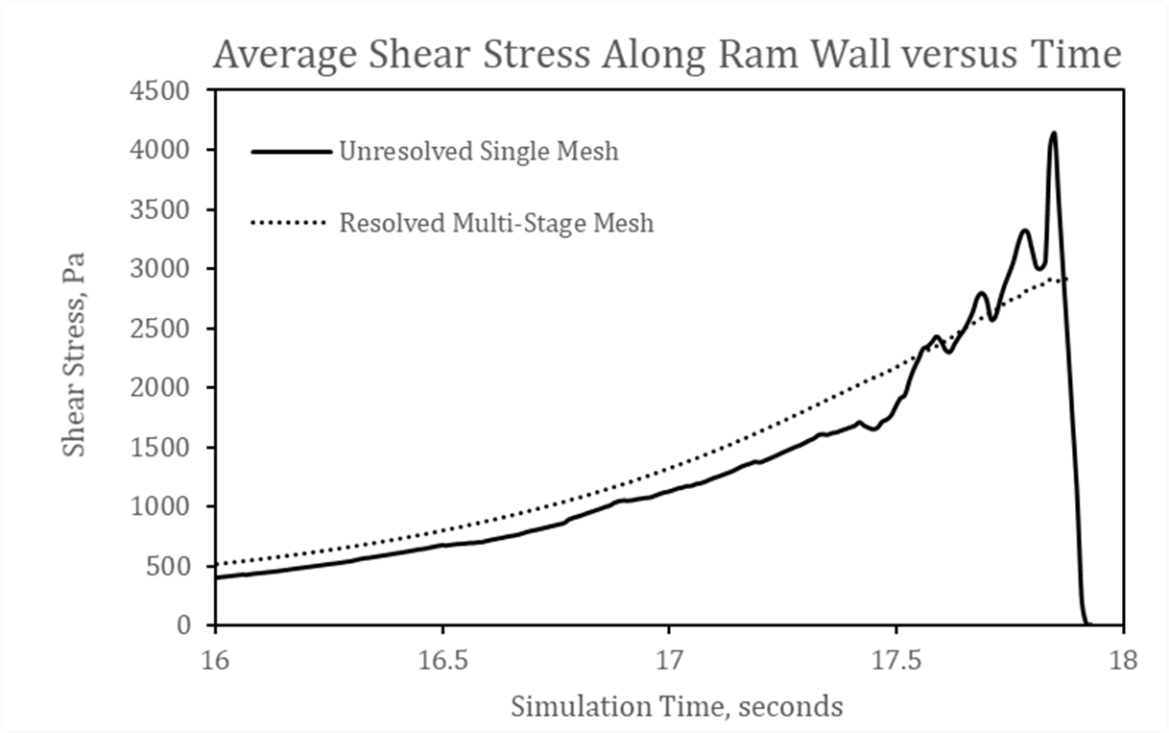


Figure 4.24. Demonstration of erroneous wall shear values for poor quality mesh.

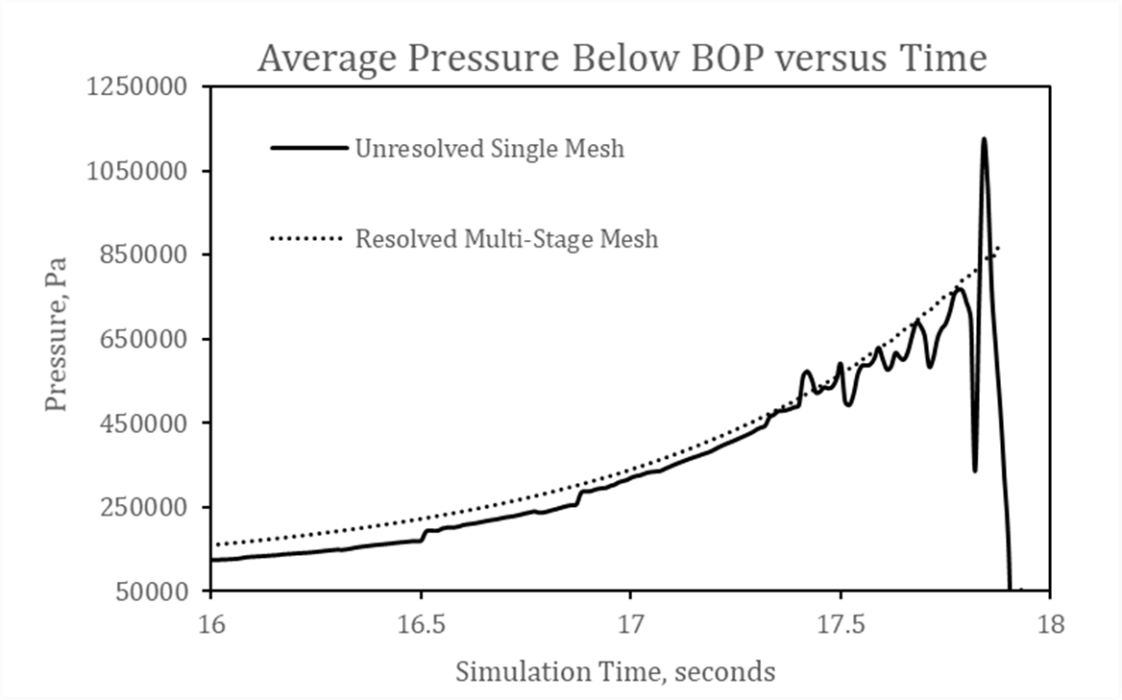


Figure 4.25. Demonstration of erroneous pressure values below BOP for poor quality mesh.

4.4.2. 3D Simulations

A second set of runs are set up to demonstrate the transient closure simulation process on more realistic geometry. Top, side, and isometric views of the rams are shown in Figure 4.26. The design does not come from any company and is simply an approximation and simplification of shear rams shown in various Macondo reports. After designing the rams in Autodesk Inventor, the ram solid bodies are subtracted from a fluid domain to get the final fluid domain shown in Figure 4.27.

A coarse mesh of around 500,000 elements is then created in ICEM CFD. The mesh setup and boundary conditions are shown in Figure 4.28. A coarse mesh is used here because the UDF boundary condition is not yet designed for parallel computations. Issues in the UDF implementation cause erroneous calculations in the 1D wellbore simulation. The cause of these issues has not been identified, but the issue may be a simple fix. Ideally, these meshes would have prism layers, be more refined, and then submit as a job for simulation through HPC resources as was done for the simulations previously shown in Figure 4.14 (which used a constant inlet boundary condition). For now, the UDF simulations must be run with a single processor, and consequently, coarser meshes are used to simulate the following results in this section.

In setting up the boundary conditions, identical boundary conditions to the axisymmetric simulations are used, shown in Table 4.5. Results for streamlines, velocities, pressures, and shear rates are plotted and discussed in a subsequent section. The simulations in Figure 4.30, Figure 4.31, and Figure 4.32 have the pipe removed. Since these simulations do not incorporate fluid structure interaction (FSI), closure beyond distance of the pipe would be impossible without removing it.

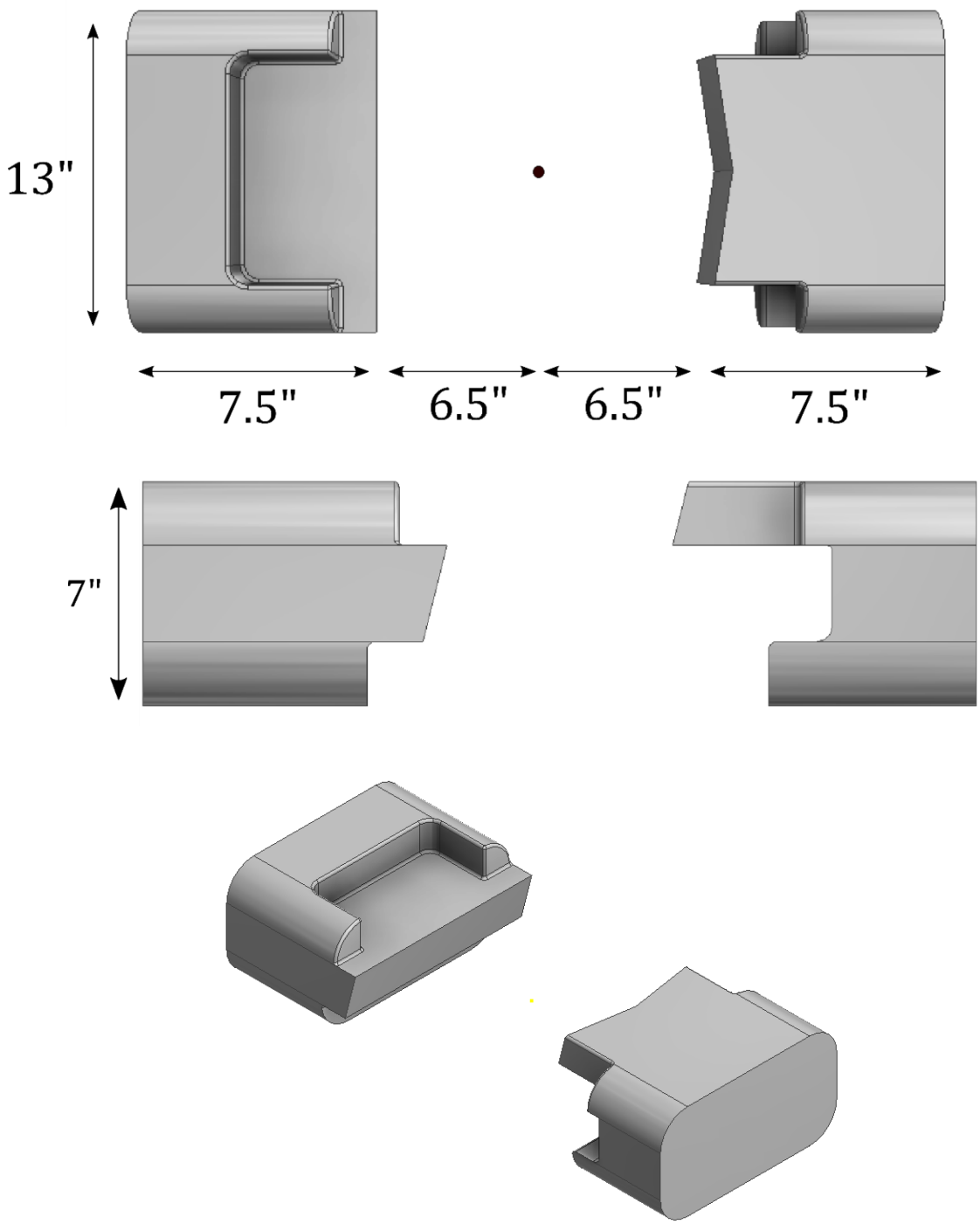


Figure 4.26. Top, side, and isometric views of shear rams.

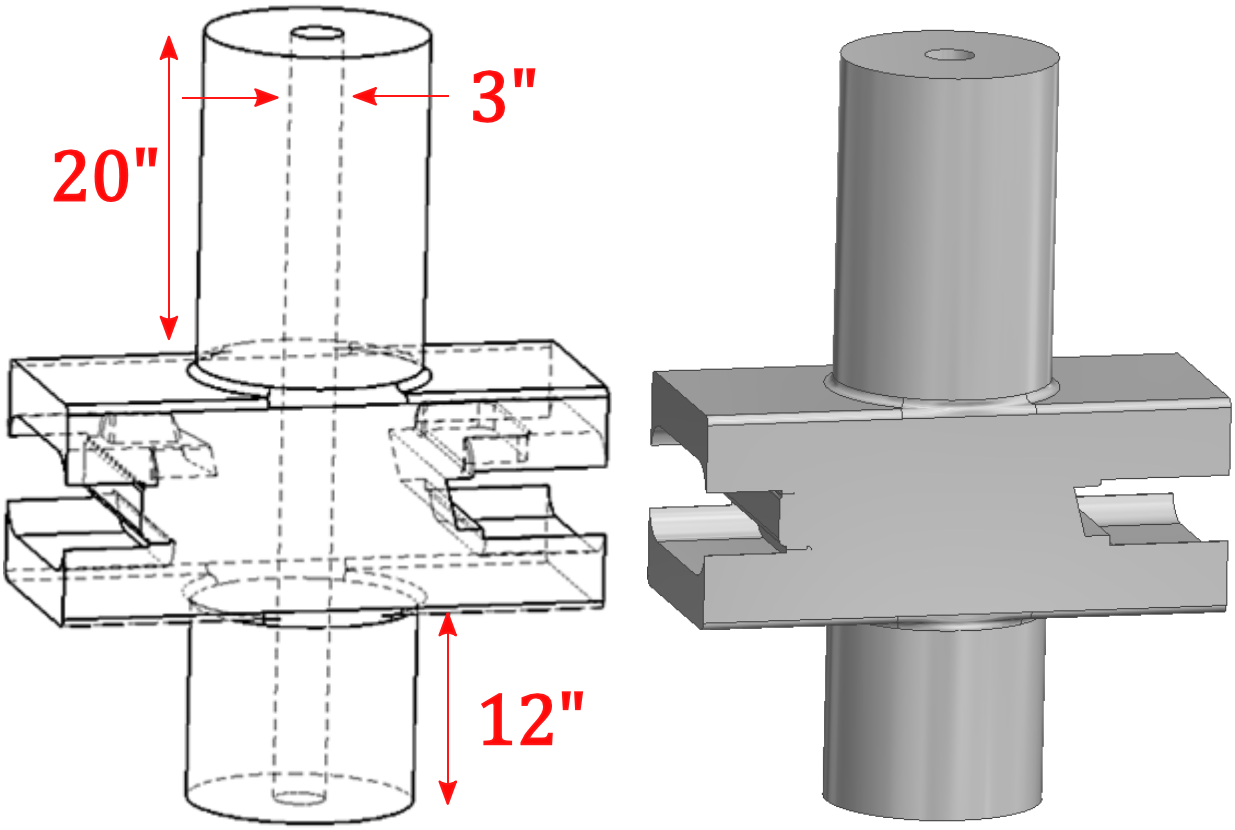


Figure 4.27. Isometric view of fluid domain when solid volume of rams is subtracted out.

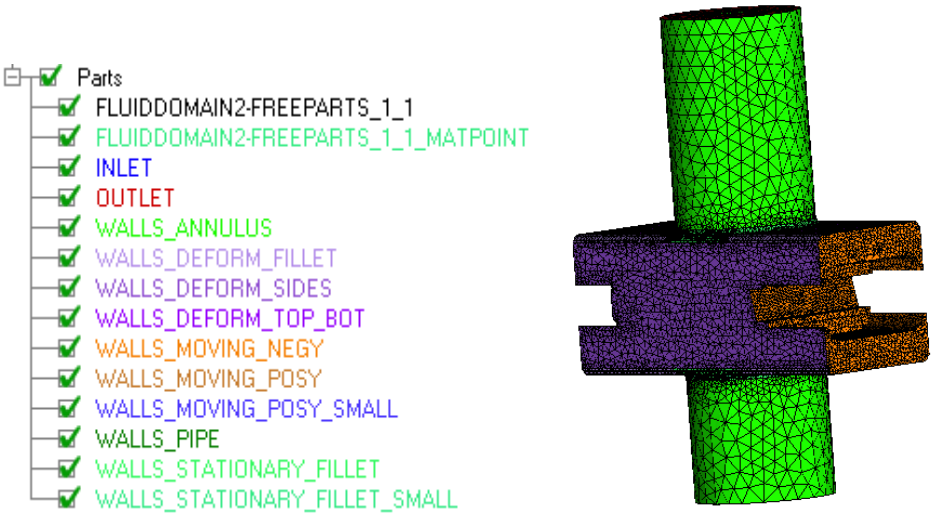


Figure 4.28. Coarse mesh of the fluid domain.

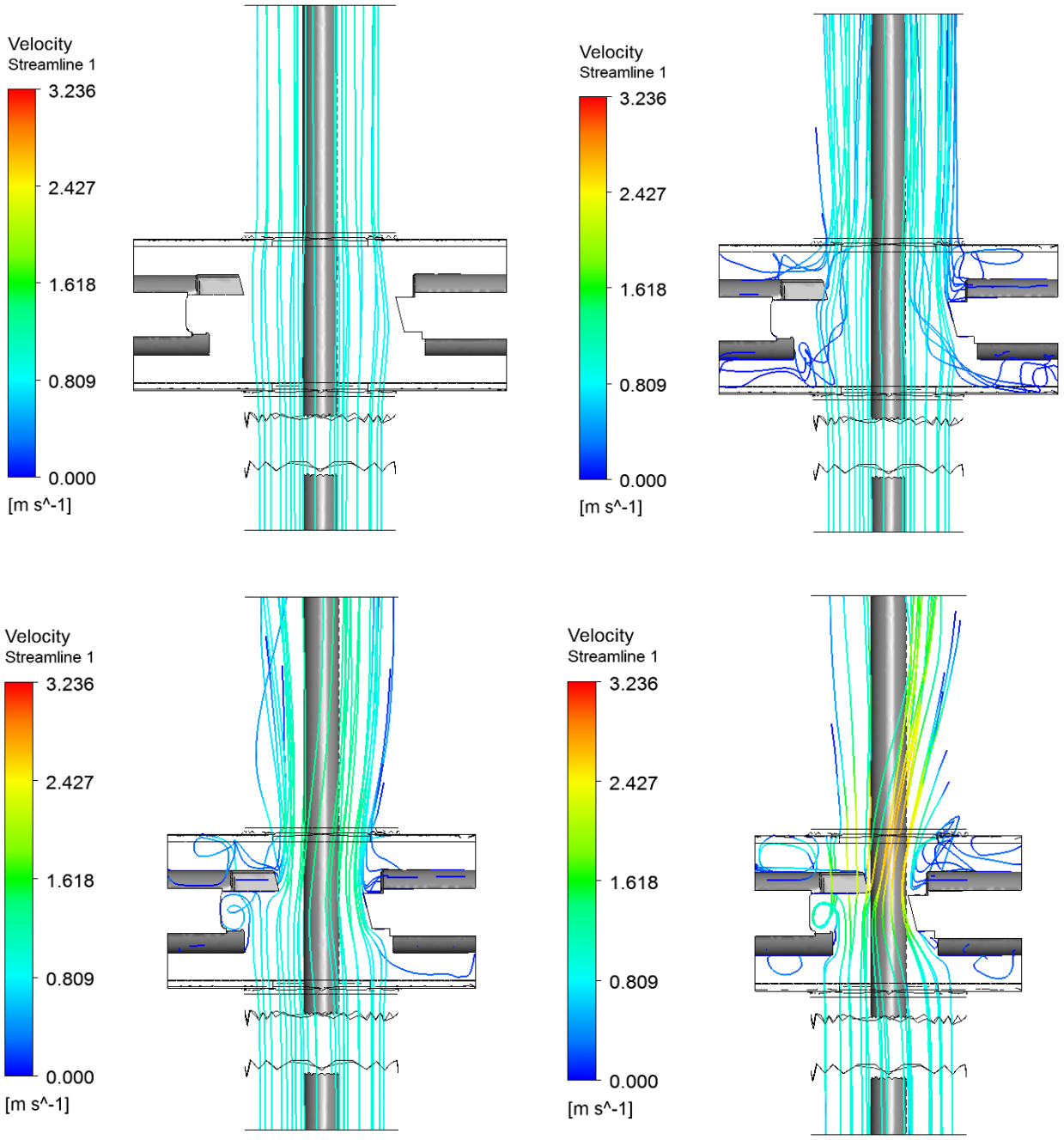


Figure 4.29. Streamlines at four different time snapshots during the simulation.

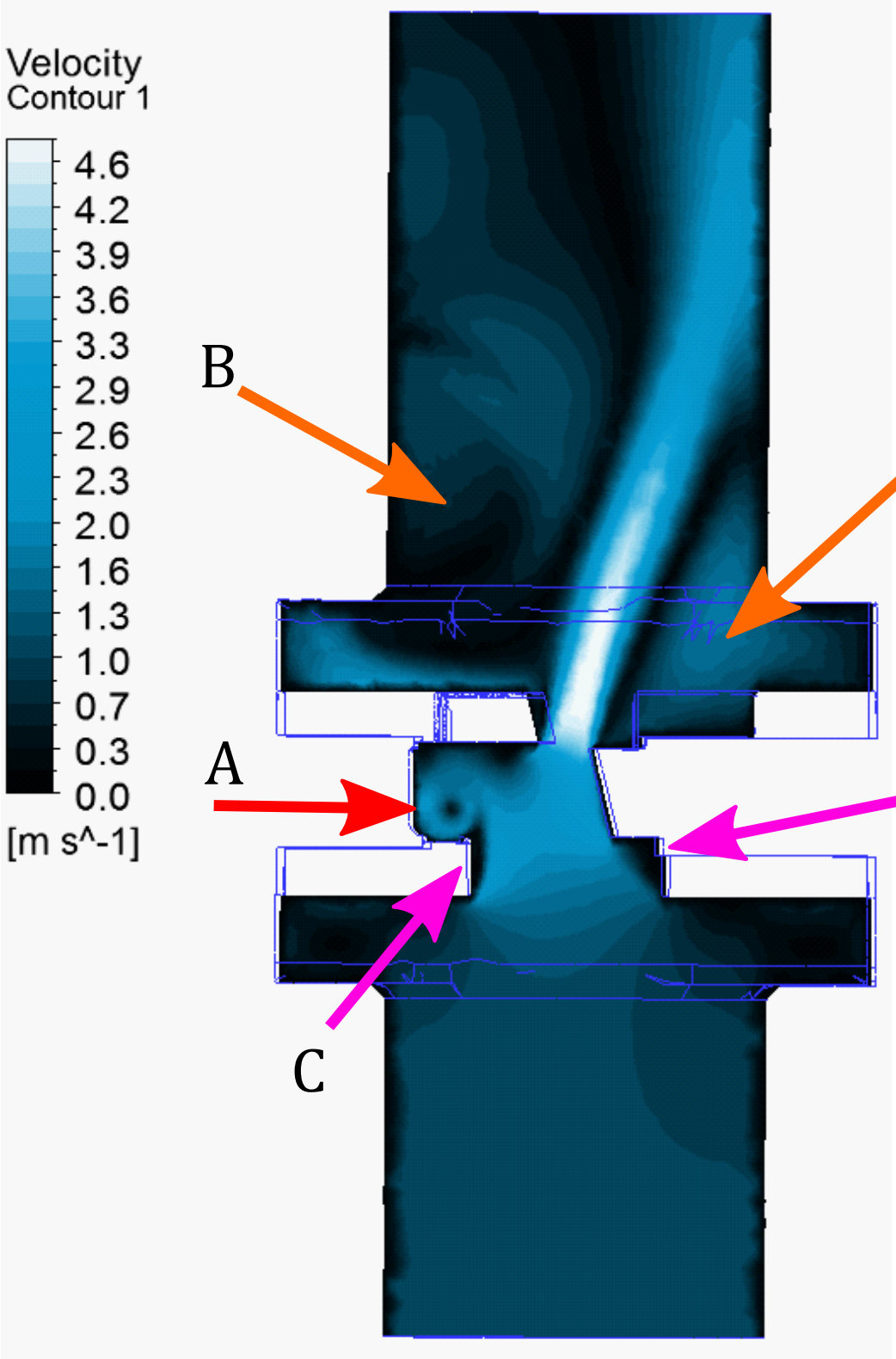


Figure 4.30. Velocity contours for z=0, xy-plane at 10s. The simulation, separate from those in Figure 4.29, has pipe removed.

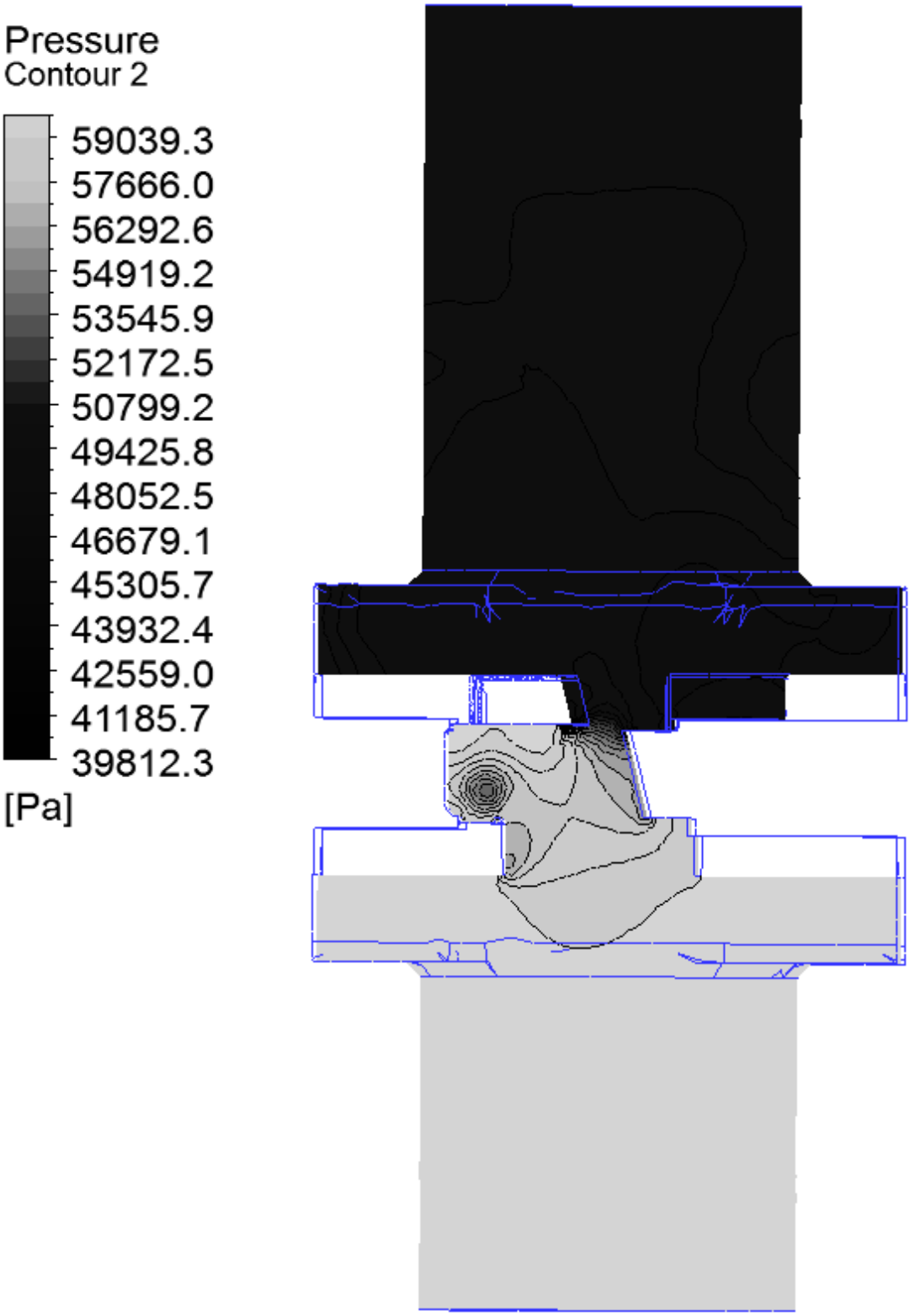
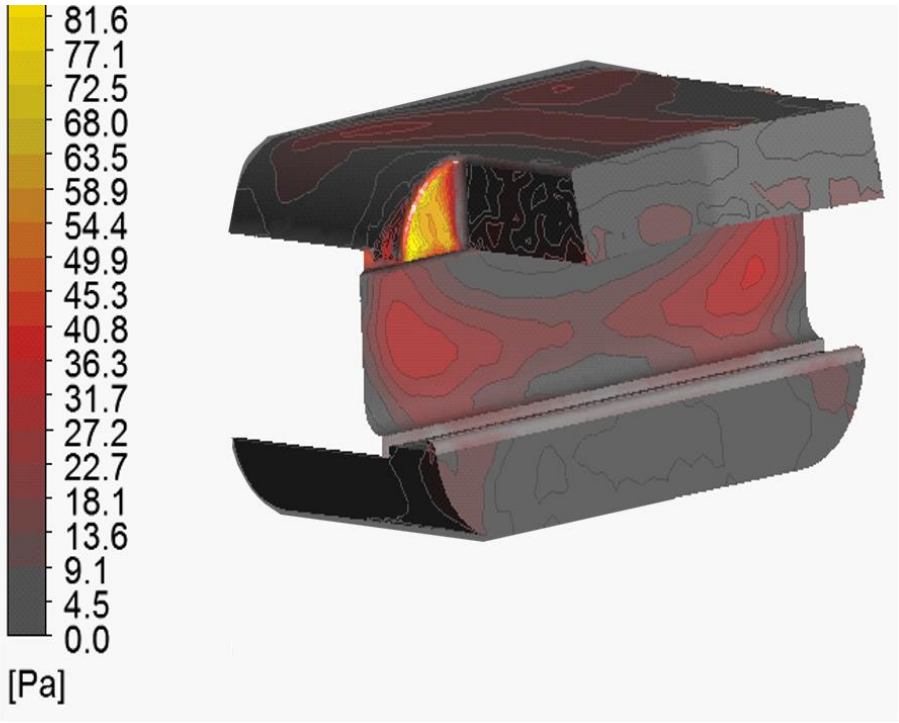


Figure 4.31. Static gauge pressure contours for z=0, xy-plane at 10s with pipe removed.



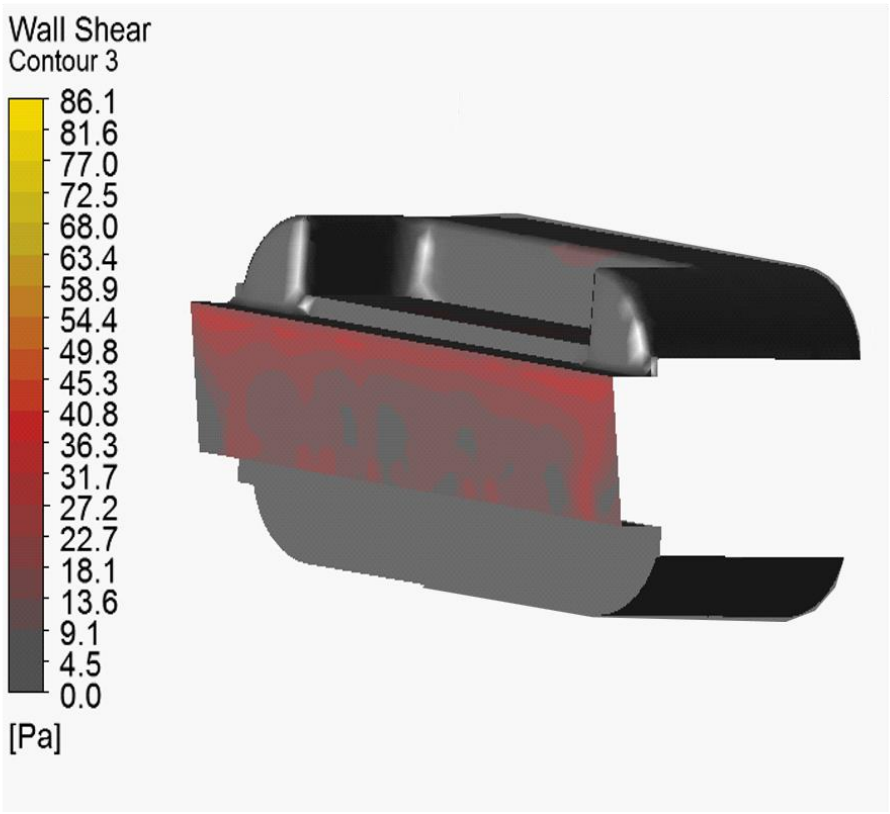


Figure 4.32. Shear rate contours on ram faces at 10s.

Figure 4.29 includes streamlines at time snapshots of at 10%, 20%, 30% and 40% cloure. These streamlines clearly demonstrate the increasingly complex flow field as the shear rams close. A velocity contour plane is displayed in Figure 4.30 to show critical flow features. The major flow recirculation area labelled "A," is something that might normally be investigated during the design process. This region of recirculation could cause several issues during the shearing process. Shear rates against the wall in this region would be large in comparison to other areas of the ram. This is also demonstrated in Figure 4.32, where shear rates are relatively large on the middle shear ram's face. Additionally, this recirculation area may trap solid particles such as sand or drill cuttings. Particles in this recirculation region may cause enough erosion to prevent complete sealing in this ram design. It may be best to consider a different ram design that does not result with such a large degree of recirculation if large flowrates are expected. Region "B" in Figure 4.30 also includes some recirculation areas. However, these regions may always exist regardless of ram design, as recirculation across any sort of backward facing step is expected.

Region "C" in Figure 4.30 shows some areas of flow separation. While this may not be critical at the time snapshot shown here, these areas will become critical as the shear rams near complete closure. Since these regions involve flow separation from the wall, low pressures will develop in this region. Near closure, cavitation is likely to occur in these regions. Erosion in this region would increase due to cavitation, making it another region that should be analyzed during the design process.

Another critical component of the design is hydrodynamic forces on the rams, which can be calculated from static pressures and shear rates on walls. It can be seen in Figure 4.31 that the middle and lower ram faces are exposed to a relatively high-pressure region. This is

due to a restriction and consequent pressure gradient at the top of the ram design. Forces were calculated to be around 1000 lbf on each ram. As stated in a report by McNutt et al. (2012), the flowrate through the BOP at closure time was around 50,000 BOPD, which would correspond to an initial inlet velocity of around 1.13 m/s for this design. For this snapshot at 10s, which used an initial inlet velocity of 1 m/s, the max hydrodynamic force was approximately 4000 N on each ram. This is insignificant in comparison to a typical shearing force. For example, in simulations by Tekin (2010), it was demonstrated that shearing force was on the order of magnitude of 1000 kN. Another simulation was run with a significantly higher (and unrealistic) flowrate of around 500,000 BOPD. Even in this simulation, the max hydrodynamic force on rams only ended up being under 100 kN. To more accurately determine the impact of hydrodynamic forces on shearing force and ram motion, additional simulations need to be run with a dynamic ram motion boundary condition that takes into consideration fluid force.

A final note and additional critical area can be clearly seen by observing Figure 4.32. While shear rates are relatively high in the main flow recirculation area, the highest shear is along walls on the upper face, highlighted in yellow. This is partly due to flow recirculation in this region, but primarily because it is the main flow path when the rams are nearing complete closure. This region could be identified as a region to be made of a more resistant material than other components of the shear ram.

Chapter 5. Conclusions

- Dynamic meshing has been used to simulate closing valves. Successive meshing is accomplished to demonstrate the ability to maintain high quality mesh and near wall refinement throughout even extreme fluid domain deformations. Importance of high-quality mesh on desirable quantities such as shear rates is demonstrated.
- CFD has been used to estimate features such as flow separation, cavitation, flow recirculation, and pressure fields during the transient process of a closing blowout preventer. Features such as these, particularly flow fields, would not be as easy to analyze experimentally. An example case of a 3D shear ram geometry demonstrates how CFD can be used to identify critical flow regions during the transient flow process. CFD could also be used to analyze designs that utilize the flow direction to provide additional force in sealing the well, as already done in some designs by Hydril (Springett et al. 2011).
- Coupling of CFD (2D, 3D) with MOC (1D) has been accomplished to provide more realistic boundary conditions. Constant inlet boundary condition will provide extreme overestimation of velocities, and constant inlet pressure boundary condition will provide severed underestimation of fluid velocities. Some stability issues may arise when using explicit coupling; however, these can be dampened using filtering or time-averaging methods.
- Hydrodynamic forces for single phase flows are not significant for expected flowrates, as also reported in other BOP simulation studies.

Chapter 6. Future Work

- For the multiphase simulations, an assumption is made that most of the gas has evolved from the solution. Depending on the fluid properties and pressure conditions below the BOP, this may not be the case. In some cases, significant mass transfer may occur between the liquid and gas phases. The work in this thesis has not explored mass transfer options outside of cavitation for single component fluid.
- The coupled simulation could be improved to take into consideration multiphase flow throughout the wellbore. This can be improved by either coupling the CFD domain to a more advanced commercial simulator, such as that done in Xing, Yeung, and Lo (2011). Otherwise, the current wellbore simulator may be improved upon.
- Ram motion dynamics can be further investigated. Current implementation ignores several key parameters such as damper forces during the process, due to lack of knowledge.
- Simulation of multiphase flow through narrow restrictions did not match experimental results. Further study should be conducted on this to understand the reason for error and how to accurately simulate the process.
- The CFD simulations take a significant amount of effort in both simulation time and case setup. Reduced order modelling of the process may be investigated for real-time simulation of a closing BOP. Directions to look at are those such as simple regression analysis or method of snapshots (Sirovich 1987).
- Work in this thesis does not consider the scenario where drilling fluid is flowing through the wellbore. Drilling fluid is generally a non-Newtonian fluid and will also

carry drill cuttings (solid particles) to the surface. Studies related to this may be carried out, such as how drill cuttings impact erosion.

- The current study in this thesis includes a basic simulation of slugging through a static ram to demonstrate how flow patterns may impact the solution results. A further investigation can be carried out for transient simulation. For example, the impact of slugging on ram motion dynamics can be explored.
- Stability issues in the explicitly coupled solution nearing closure time exist with the current implementation. A simple moving average, applied in this work, introduces error. More advanced coupling algorithms may be applied to achieve stability in this area.
- Off-center pipe geometries are shown, but not fully investigated. Studies can be carried out to demonstrate how off-center pipe will impact ability of shear rams to cut pipe.
- Current work does not consider heat transfer to BOP in the event of a hot fluid flowing through the equipment. Delong et. Al (2014) discuss possible issues with hot fluid travelling through well control equipment. Conjugate heat transfer CFD simulations may be run to further investigate this issue.

Appendix A. Link to CAD files, CAS files, compiled UDFs

https://github.com/dbarre5/Thesis

Appendix B. Codes and Scripts for Simulation

B.1 Standalone Code for Wellbore Simulation in Python

```
import numpy as np
from __future__ import division

#sonic Velocity, m/s
a = 1000;
#gravitational acceleration, m/s^2
g = 9.81;
#Timestep: From Fluent normally: Set as a constant here
dt = 0.001
#total time simulation, seconds
EndT = 20;
#total length of pipe, meters
Length = 1000;

#number of subdivisions in time: This eventually will not be needed because fluent runs until completion
nt = int(EndT/dt);

#Determined from the relationship derived
dx = dt*a

#Calculatte the number of subsections needed
nx = int(Length/dx);

#friction factor
f = 0.01;
#pipe diameter, m
D = 0.1;
#Cross sectional Area, m2
A = 3.14*D**2/4;

#Create empty arrays for velocity and pressure head
#Velocity
V = np.zeros((nx,nt))
#Pressure head
H = np.zeros((nx,nt))


#Initial Conditions and Boundary
#initial Velocity, m/s (This will normally be gained from the fluent simulation)
Vinit = 1;
#initial hydrostatic head at the bottom of the blowout breventer, Pa
PBOP = 10;

#initialize velocity and Head (from the frictional pressure drop)
for x in range(0,nx):
    V[x,0] = Vinit;
    H[x,0] = PBOP + ((nx-x-1)/nx*Length)*(Vinit)**2*f/2/g/D;

#Store the Reservoir Pressure
Phead = H[0,0]


print('dt is %i' %dt)
print('dx is %i' %dx)
```

```
print('nt is %i' %nt)
print('nx is %i' %nx)

#initialize variables
test = 0
closingTime = 20
for t in range(0,nt-1):

    # Simplifying some expressions into single variable
    K = f*dt/2/D
    C = g/a

    #Hydraulic head increases until velocity through the section is 0
    #Test is initially 0 (when the valve is open)
    #When the valve is closed, test = test + 1, and the "else statement" boundary conditions are then applied
    if test < 0.5:

        #Boundary Conditions at constant reservoir pressure on one end of the pipe
        V[0,t+1] = V[1,t]+C*(H[0,t]-H[1,t])-K*(V[1,t]*abs(V[1,t]))
        H[0,t+1] = Phead

        #At the other end of pipe, constant increasing pressure head
        #This could also be set to constant decreasing velocity profile or something else
        H[nx-1,t+1] = H[nx-1,0]+(t*dt)**2/100
        V[nx-1,t+1] = V[nx-2,t]-C*(H[nx-1,t]-H[nx-2,t])-K*(V[nx-2,t]*abs(V[nx-2,t]))

    #switch boundary condition to velocity 0 boundary condition when seal is closed
    else:
        V[0,t+1] = V[1,t]+C*(H[0,t]-H[1,t])-K*(V[1,t]*abs(V[1,t]))
        H[0,t+1] = Phead

        V[nx-1,t+1] = 0
        H[nx-1,t+1] = (V[nx-2,t]-V[nx-1,t]+K*V[nx-2,t]*abs(V[nx-2,t]))/C+H[nx-2,t]

    #Checks if velocity reaches 0: At this point, boundary condition switches to constant 0 velocity
    if V[nx-1,t+1] < 0:
        test = test+1
        V[0,t+1] = V[1,t]+C*(H[0,t]-H[1,t])-K*(V[1,t]*abs(V[1,t]))
        H[0,t+1] = Phead

        V[nx-1,t+1] = 0
        H[nx-1,t+1] = (V[nx-2,t]-V[nx-1,t]+K*V[nx-2,t]*abs(V[nx-2,t]))/C+H[nx-2,t]


    #Solve for the inner nodes at next timestep
    for x in range(1,nx-1):
        #solve for new velocity
        Coef1 = V[x-1,t]+V[x+1,t]
        Coef2 = g/a*(H[x-1,t]-H[x+1,t])
        Coef3 = f*dt/2/D*(V[x-1,t]*abs(V[x-1,t])+V[x+1,t]*abs(V[x+1,t]))
        V[x,t+1] = 1/2*(Coef1+Coef2-Coef3)

        #solve for new Pressure Head
        Coef1 = a/g*(V[x-1,t]-V[x+1,t])
        Coef2 = (H[x-1,t]+H[x+1,t])
        Coef3 = a/g*f*dt/2/D*(V[x-1,t]*abs(V[x-1,t])-V[x+1,t]*abs(V[x+1,t]))
        H[x,t+1] = 1/2*(Coef1+Coef2-Coef3)
```

B.2. Code for Wellbore Coupling to be Compiled as a UDF in ANSYS Fluent

```
#include "udf.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>


int n_time;           /*create variable to store integer number of timesteps*/
real PressureAbove;    /*Variable that stores average pressure above BOP*/
real PressureBelow;    /*Variable that stores average pressure below BOP*/
real PressDiff;        /*Variable that stores piezometric head across BOP*/
int TS;
real nfaces;          /*Variable to store the number of faces that are being looped over - will be used to average pressure values*/
int counter;          /*Something strange is going on for if(first_iteration)... it's looping 3 times through first iteration. This counter will prevent that--->after testing it... it doesn't. Still runs 3 times*/
real A,PBOP, Phead, Coef1, Coef2, Coef3,K,C; /*variables for hydraulic transient simulation*/
int x,t; /*variables for hydraulic transient simulation*/

real g = 9.81; /*gravity*/

/*****/
      real a; /*Sonic Velocity*/
      real Length; /*length of pipe*/
      real dt; /*timestep*/
      real f; /*friction factor, constant*/
      real D;  /*equivalent diameter of pipe/annulus*/
      real Vinit;  /*initial velocity*/
      int nx; /*number of discretizations in length-----hard coded because I can't figure out how to convert to int-----MUST BE CHANGED APPROPRIATELY*/
      FILE *inFile;
      char array[256];
      real ValIDM;
      real numberFromFile;
      int Zone_ID_Below;
/******/

real dx; /*discretization length*/
real V[10000],Vnew[10000],Hold[10000],Hnew[10000]; /*Again hard coded... because I need to fix this. current issue is that nx isn't known at compile time*/

real test = 0; /*Test for negative  at valve(representing shut in)*/
real fraction; /*fractional frictional pressure drop per unit discretization length*/


DEFINE_PROFILE(VelBC, thread, position)
{



   face_t face;          /*Create variable for the face to loop through and assign BC*/
   Thread *tf;        /*Create variable for the thread*/
   cell_t c;         /*Create variable for the cells to loop through to grab pressure*/
   int Zone_ID;       /*Create variable for the zone ID which is where pressure is grabbed from*/

   Domain *domain;         /* domain is declared as a variable   */
```

```
54.    domain = Get_Domain(1);  /* returns fluid domain pointer      */
55.
56.    /*Calculate and initialize some things when the initialization button is pressed in Fluent*/
57.    if ( CURRENT_TIME == 0 )
58.    {
59.
60.       n_time = 0;
61.       Message("Initializing variables and n_time \n");
62.       /*find file and open initial velocity reading---should also get well parameters, zone IDS, etc*/
63.
64.
65.
66.
67.
68.       inFile = fopen("inputDeck.dat","r");
69.       ValIDM = 0;
70.       if (inFile == NULL)
71.       {
72.          Message("Can't find input file");
73.       }
74.       if (inFile != NULL)
75.       {
76.          Message("Found File");
77.       }
78.
79.       /*Read values from text file*/
80.       fscanf(inFile, "%lg", &a);
81.       Message("a %lg \n",a);
82.       fscanf(inFile, "%lg", &Length);
83.       Message("Length %lg \n",Length);
84.       fscanf(inFile, "%lg", &dt);
85.       Message("dt %lg \n",dt);
86.       fscanf(inFile, "%lg", &f);
87.       Message("f %lg \n",f);
88.       fscanf(inFile, "%lg", &D);
89.       Message("D %lg \n",D);
90.       fscanf(inFile, "%lg", &Vinit);
91.       Message("Vinit %lg \n",Vinit);
92.       fscanf(inFile, "%d", &nx);
93.       Message("nx %d \n",nx);
94.       fscanf(inFile, "%d", &Zone_ID_Below);
95.       Message("nx %d \n",Zone_ID_Below);
96.
97.       fclose(inFile);
98.
99.
100.
101.
102.
103.
104.      A = 3.14*pow(D,2)/4; /*area of pipe*/
105.      dx = dt*a; /*discretization length*/
106.      K = f*dt/2./D;
107.      C = g/a;
108.
109.
110.
111.
112.      /* ---------------------------------------------------------------------------------------------------- */
113.      /* Grab pressure from Bottom for top of domain. We are initializing the piezometric head BC for the top of th
      e pipe domain*/
```

```
    /* ---------------------------------------------------------------------------------------------------------------- */
    Zone_ID = Zone_ID_Below;
    tf = Lookup_Thread(domain, Zone_ID);

    nfaces = 0; /*initialize the nfaces variable back to 0*/
    PressureBelow = 0; /*Initialize the pressure value*/

    /* loop over all cells within the cell thread where c is just an */
    /* integer number, enumerating the cells. If your thread holds for */
    /* instance 100 cells the loop wil be perfomed 100 times, each time */
    /* c is increased by one starting from zero --- from https://www.cfd-online.com/Forums/fluent-udf/38375-udf-problem.html */
    begin_c_loop(c,tf)
      {
      nfaces = nfaces + 1;
      PressureBelow  = PressureBelow + C_P(c, tf); /*Sum up all pressure values across the cell faces*/
      }
    end_c_loop(c, tf)
    PressureBelow = PressureBelow/nfaces;
    PressureBelow = (PressureBelow)/9.81/1000;     /*Average the sum of all pressures*/
    PBOP = PressureBelow;                          /*initial hydraulic head*/

    Message("C %lf \n",C); /*Check some values to make sure they are calculated correctly/make sense*/
    Message("a %lf \n",a);
    Message("f %lf \n",f);
    Message("dt %lf \n",dt);
    Message("D %lf \n",D);
    Message("A %lf \n",A);
    Message("dx %lf \n",dx);
    Message("K %lf \n",K);
    Message("PBOP %lf \n",PBOP);


    Message("Piezometric Head Values \n");
    for(x = 0; x < nx; x++ ){
      V[x] = Vinit;
      Hold[x] = PBOP + dx*pow(Vinit,2.0)*f/2.0/g/D*(nx-x-1);/*((nx/1.-x/1.)/nx*Length)*pow(Vinit,2)*f/2/g/D;*/
      Message("Hold[x] =  %lf \n",Hold[x]);
    }
    Phead = Hold[0]; /*constant reservoir boundary condition*/

    fraction = dx*pow(Vinit,2)*f/2.0/g/D;
    Message("Hold diff =  %lf \n",Hold[55]-Hold[54]);
    Message("fraction =  %lf \n",fraction);


  }

  counter = 0; /*reason for creation is stated above in declaration of counter*/

  /*Only perform these calculations during the first iteration*/
  if ( first_iteration )
    {
      if (counter == 0)
      {
      counter = 1; /*set counter to 1, so that only one set of calculations occur*/

      Message("PressureAbove %lf \n",PressureAbove);

      /* ------------------------------------------- */
```

```
173.        /* Grab pressure from Zone for bottom of domain */
174.        /* ---------------------------------------- */
175.
176.        Zone_ID = Zone_ID_Below;
177.        tf = Lookup_Thread(domain, Zone_ID);
178.        nfaces = 0; /*initialize the nfaces variable back to 0*/
179.        PressureBelow = 0; /*Initialize the pressure value*/
180.
181.        /*Again, loop through faces, but for a different zone*/
182.        begin_c_loop(c,tf)
183.          {
184.          nfaces = nfaces + 1;
185.          PressureBelow  = PressureBelow + C_P(c, tf); /*Sum up all pressure values across the cell faces*/
186.          }
187.        end_c_loop(c, tf)
188.        PressureBelow = PressureBelow/nfaces; /*Average the sum of all pressures*/
189.
190.
191.
192.        PressureBelow = (PressureBelow)/9.81/1000; /*Calculate the piezometric head below the valve*/
193.        Message("PressureBelow %lf \n",PressureBelow);
194.
195.
196.        /* ---------------------------------------------------------------------------------------------------*/
197.        /* ----------------------------------SECTION OF CODE FOR WATER HAMMER BELOW----------------------------------
      ----------*/
198.        /* ---------------------------------------------------------------------------------------------------*/
199.
200.        if (test < 0.5){
201.          Vnew[0] = V[1]+C*(Hold[0]-Hold[1])-K*(V[1]*abs(V[1]));
202.          Hnew[0] = Phead;
203.
204.          Hnew[nx-1] = PressureBelow;
205.          Vnew[nx-1] = V[nx-2]-C*(Hold[nx-1]-Hold[nx-2])+K*(V[nx-2]*abs(V[nx-2]));
206.          Message("VnewBoundary[x] =  %lf \n",C*(Hold[nx-1]-Hold[nx-2])+K*(V[nx-2]*abs(V[nx-2])));
207.          Message("VnewBoundary[x] =  %lf \n",C*(Hold[nx-1]-Hold[nx-2])+K*(V[nx-2]*abs(V[nx-2])));
208.          Message("VnewBoundary[x] =  %lf \n",C*(Hold[nx-1]-Hold[nx-2]+fraction));
209.          Message("VnewBoundary[x] =  %lf \n",C*(Hold[nx-1]-Hold[nx-2]-fraction));
210.        }
211.
212.        else{
213.          Vnew[0] = V[1]+C*(Hold[0]-Hold[1])-K*(V[1]*abs(V[1]));
214.          Hnew[0] = Phead;
215.
216.          Vnew[nx-1] = 0;
217.          Hnew[nx-1] = (V[nx-2]-V[nx-1]+K*V[nx-2]*abs(V[nx-2]))/C+Hold[nx-2];
218.        }
219.
220.        if (Vnew[nx-1] < 0){
221.          test = test+1;
222.          Vnew[0] = V[1]+C*(Hold[0]-Hold[1])-K*(V[1]*abs(V[1]));
223.          Hnew[0] = Phead;
224.
225.          Vnew[nx-1] = 0;
226.          Hnew[nx-1] = (V[nx-2]-V[nx-1]+K*V[nx-2]*abs(V[nx-2]))/C+Hold[nx-2];
227.        }
228.
229.        for(x = 1; x < nx-1; x++){
230.          Coef1 = V[x-1]+V[x+1];
231.          Coef2 = g/a*(Hold[x-1]-Hold[x+1]);
232.          Coef3 = f*dt/2/D*(V[x-1]*abs(V[x-1])+V[x+1]*abs(V[x+1]));
```

```
            Vnew[x] = 1./2.*(Coef1+Coef2-Coef3);


            Coef1 = a/g*(V[x-1]-V[x+1]);
            Coef2 = (Hold[x-1]+Hold[x+1]);
            Coef3 = a/g*f*dt/2/D*(V[x-1]*abs(V[x-1])-V[x+1]*abs(V[x+1]));
            Hnew[x] = 1./2.*(Coef1+Coef2-Coef3);
        }
        for(x = 0; x < nx; x++){
            Hold[x] = Hnew[x];
            V[x] = Vnew[x];
            Message("Hold[x] =  %lf \n",Hold[x]);
        }

        /* ------------------------------------------------------------------------------------------------------*/
        /* ----------------------------------SECTION OF CODE FOR WATER HAMMER ABOVE----------------------------------
    ---------*/
        /* ------------------------------------------------------------------------------------------------------*/

        } /* end if_counter*/
    } /*end if_iter*/









/*Loop through and assign velocity to the faces of the cells*/
begin_f_loop(face, thread)
{
   F_PROFILE(face, thread, position) = V[nx-1];
}
end_f_loop(face, thread)


/*Output Relevant Variables in text file*/
inFile = fopen("OldPressureValues.dat", "w+");
for(x = 0; x < nx; x++ ){
      fprintf(inFile, "%lg \n", Hold[x]);
   }
fclose(inFile);

inFile = fopen("NewPressureValues.dat", "w+");
for(x = 0; x < nx; x++ ){
      fprintf(inFile, "%lg \n", Hnew[x]);
   }
fclose(inFile);


}
```

B.3. Input File, InputDeck.dat

```
1000
10000
```

```
0.1
0.01
0.1
1
100
9


VALUES ABOVE ARE IN ORDER

SONIC_VELOCITY          - a
WELL_LENGTH             - Length
TIMESTEP                - dt
FRICTION_FACTOR         - f
DIAMETER                - D
INITIAL_VELOCITY        - Vinit
SPATIAL_DISCRETIZATION  - nx
Zone_ID_BELOW           - Zone_ID
```

B.4. Line to Replace lines 147-151 in Appendix B

For the cases where flow is initialized from a previous simulation's Hydraulic Head values, the following code is used. The text file from which is grabs values should be titled "PressureValuesPrevious.dat" and the file's data should be in the same format as the output files from lines 271-282 in Appendix B.

```
/* */
    /* Storing old Head Values as initialization */
    /*  */
    /* */
    Message("Piezometric Head Values \n");
    inFile = fopen("PressureValuesPrevious.dat","r");
    if (inFile == NULL)
    {
      Message("Can't find input file");
    }
    if (inFile != NULL)
    {
      Message("Found File");
    }
    for(x = 0; x < nx; x++ ){
      ValIDM = 0;
      /*Read values from text file*/
      fscanf(inFile, "%lg", &Hold[x]);
      Message("Hold =  %lg \n",Hold[x]);
    }
    fclose(inFile);
```

B.5. Example Simple Interpolation Script

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import interpolate

# Original dataset
y = [1,4,9,16,25,36,49,64,81,100,121]

# Assigning an x-array which is basically just the x-integer count
x = np.zeros(len(y))
for i in range (0,len(y)):
    x[i] = i

#using scipy intepolation function
s = interpolate.InterpolatedUnivariateSpline(x, y)

#new array for which data will be interpolated
xnew = np.linspace(0,10,101)
ynew = s(xnew)

#To visualize the results
plt.figure()
plt.plot(x, y, "ro", markersize=7)
plt.plot(xnew, ynew, "ko", markersize=2)
plt.legend(['Previous Pressure Values', 'New Interpolated Pressure Values'])
plt.title('Spline Interpolation Example')
plt.show()

#Writing results to python output
for i in range (0,len(ynew)):
    print(ynew[i])

#Writing results to text file which will be read as BC for next simulation
text_file = open("oldPressureValues.txt", "w")
for i in range (0,len(ynew)):
    text_file.write("%f \n" % ynew[i])
text_file.close()
```

B.6. Code for Constant Velocity Ram BC to be Compiled as a UDF

```
#include "udf.h"

DEFINE_CG_MOTION(test, dt, vel, omega, time, dtime)
{
    vel[1]=-0.006986789;
}
```

B.7. Code for Time-Varying Velocity Ram BC to be Compiled as a UDF

```
#include "udf.h"

DEFINE_CG_MOTION(test, dt, vel, omega, time, dtime)
{
```

```
5.      Real Time = CURRENT_TIME
6.      vel[1]=-0.006986789*Time;
7.  }
```

Appendix C. Additional Information and Considerations

C.1. Ram Motion

The motion of a BOP ram can be described in multiple ways, such as:

1. Setting ram motion as a constant velocity
3. Setting ram motion as a time-varying velocity that decreases over time
4. Modelling the kinematics of the hydropneumatic accumulator-ram system and calculating the time-varying velocity at each timestep.

When a BOP ram closes, fluid velocity throughout the restricted areas will increase. This also increases the forces that act on the rams. When forces push against the rams, the rams' motion will decline in response to the increasing fluid forces. The first two options described above are approximations of ram motion that will not consider these forces: the velocity is simply assumed based on commonly reported closure times. To understand accurate estimation of ram motion, details regarding the accumulator-ram system are provided.

Blowout preventer components are activated through pressurized systems known as accumulator units. Accumulator bottles (Fig 1.) are used to store energy that will be used to close the rams. Accumulator bottles are charged by compressing fluids such as nitrogen. This may be done by forcing a hydraulic fluid into the accumulator with the nitrogen which is separated from the hydraulic fluid by a bladder. When the energy is required for closing the rams, the energy is released, and the hydraulic fluid is pushed against the piston which is attached to the BOP ram. This causes the BOP rams to close against the pipe.

Assuming Isothermal state of change (due to relatively slow changes in gas volume), calculation of accumulator capacity can be calculated simply through Boyle's Law:

$$P_1V_1 = P_2V_2$$

At lower pressures, the gas may be assumed to be ideal. At subsea depths, high pressures are needed within the accumulator units, causing nitrogen to trend away from an ideal gas (Curtiss and Buckley 2003, Good and McAdams 2001, Sattler 2002). Calculations using Boyle's Law will give an estimate of usable fluid volume. This calculation is related to the fluid volume required to close a BOP ram. However, these calculations alone are insufficient to calculate dynamic motion of a BOP ram. To calculate this motion, the kinematics of the system must be modelled.

This may be modelled by considering an accumulator unit with compressible fluid as a spring-like system. The dynamics of a spring system can be estimated through an analysis of forces acting on an object. A typical spring system consists of four components: mass, external forces, a spring, and a damper. The force balance for a system is as follows:

$$m\ddot{x} = F - F_k - F_b$$

A simple example is provided in figure 2. The equation above for this system can be expressed as:

$$m\ddot{x} = F - k_s x - b\dot{x}$$

An accumulator-ram system is a hydropneumatic system and contains the four elements described above. The mass of interest is the mass of the ram block. External forces are provided by fluid flowing through the annulus, pushing against the ram block. The compressible fluid acts as a spring system, and frictional forces throughout the system act as a viscous damper.

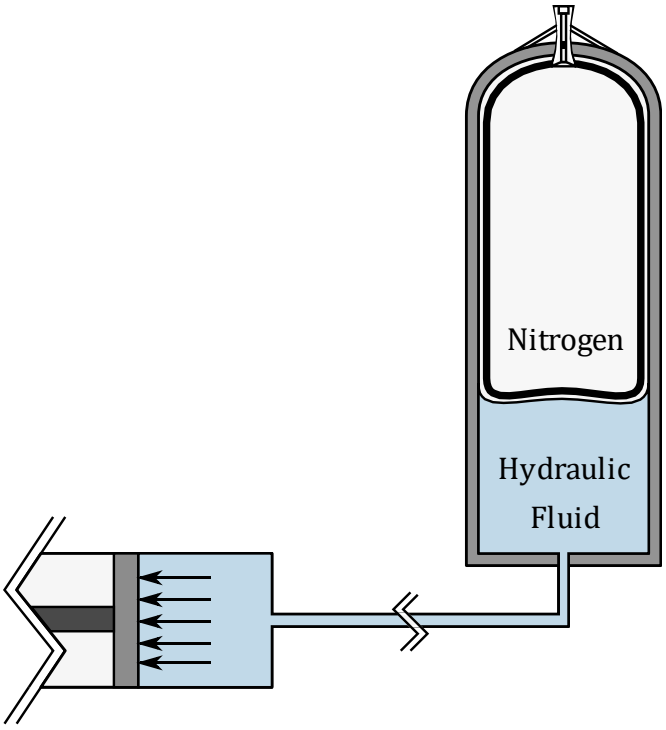


Figure C.1. Illustration of Hydropneumatic Accumulator System

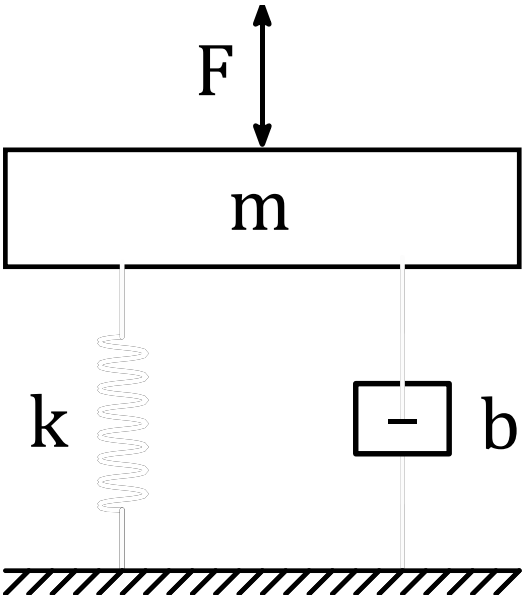


Figure C.2. Example Force Balance of a Simple Spring System

A hydropneumatic system is not entirely identical to the simple mechanical spring system as illustrated in Fig. 2. While the nitrogen bladder may act as a spring, the "spring constant," k, is not constant for a hydropneumatic system. The spring rate changes due to polytropic changes in the state of a gas (Bauer 2010). This is unlike a linear wound mechanical spring where the spring rate will remain constant regardless of spring load.

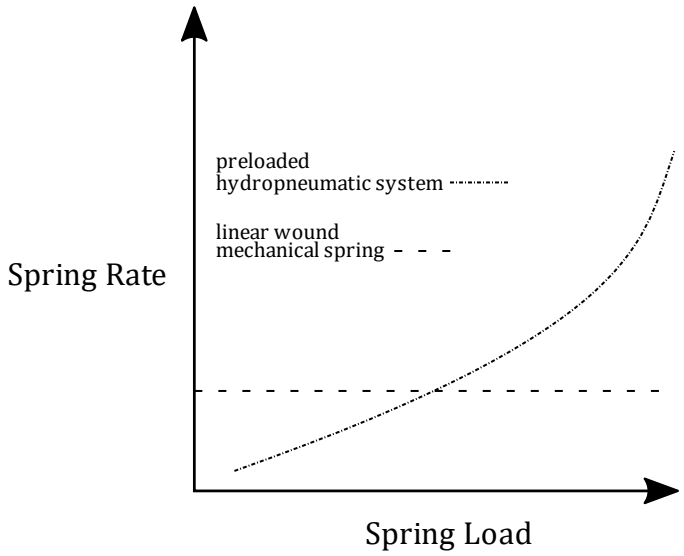


Figure C.3. Spring Rates for Mechanical and Hydropneumatic Systems

Spring rate must be calculated depending on piston position. Full details behind calculation of spring forces for hydropneumatic systems are included in Bauer 2010. A summarization is provided here. Considering the state equation for polytropic changes of state and the general mathematical expression for spring rate, the following spring rate can be derived as:

$$c(s) = F_{F1}\left(\frac{p_0 A_K V_0}{F_{F1}}\right)^{n}(-n)\left(\frac{p_0 A_K V_0}{F_{F1}} - A_K s\right)^{-n-1}(-A_K)$$

where p_0 and V_0 are pre-charge pressure and volume, n is the polytropic constant, and F_{F1} is the force ($p_1 A_k$) acting on the area (A_k) of the piston at state 1, a state other than initial conditions.

In addition to spring-like forces from accumulator, damper forces must also be calculated. In general, three different types of damper forces will exist in this hydropneumatic system. The first is frictional forces from contact of physical forces. If the ram block or components of the piston system come in contact with surrounding walls, a friction will be generated along these walls. Friction will also be generated from fluid velocity along the walls of the piston system. Finally, any sort of throttling within the system will cause an additional damping on the system. Unfortunately, not much is information known about key details (such as geometric details) throughout this system, so estimating this is difficult without further investigation of the system.

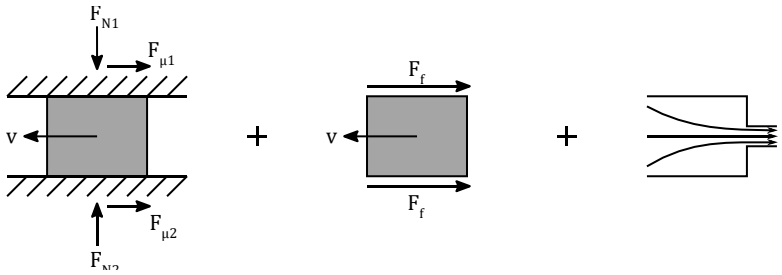


Figure C.4:Illustration of Possible Damper Forces in the Accumulator-Ram System

C.2. Solids Simulation

Another important property of the fluid to discuss is solids content. In the case of hydrocarbons flowing, sand may be carried with the liquid. Additionally, drilling fluid will generally be carrying solid drill cuttings. Discrete Phase Modelling (DPM) may be used, which simulates a second phase of spherical particles in a Lagrangian frame of reference.

One limitation of using DPM is that the particle sizes must be much smaller than the grid sizing. Therefore, DPM should only be used in the case of fine-sized drill cuttings or sand production. Another limitation of DPM is that the DPM option in Fluent does not allow for particle-particle interaction. The discrete element method (DEM) must be enabled for this interaction. However, in ANSYS Fluent, DEM cannot be enabled when dynamic meshing motion is also enabled. To simulate larger particles, macroscopic particle modelling (MPM) is available to model larger particles which span the size of several cells, though MPM cannot be enabled with dynamic meshing. Because of the limitations discussed above, simulation of solid particles during the transient closure process may be too challenging of a problem for an inexperienced CFD user. Either Fluent can be used to investigate some steady state simulation cases, or another software package can be used to allow for dynamic meshing simulations with particle simulations.

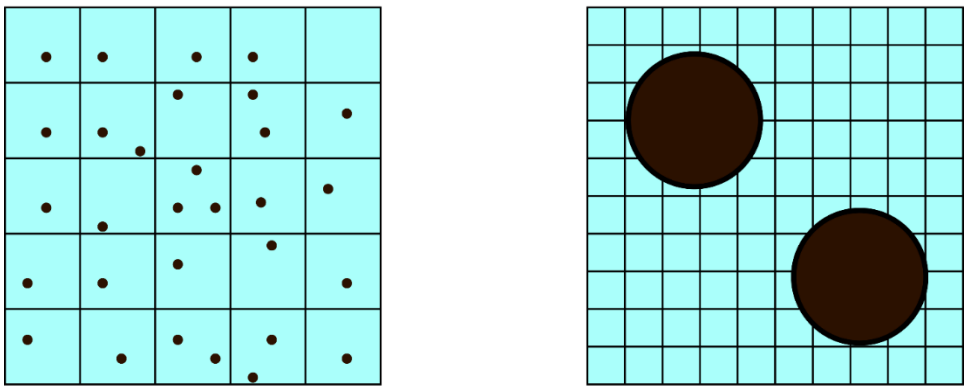


Figure C.5. Solid Particle Modelling Options in ANSYS Fluent

References

Anderson, J.D. and Wendt, J., 1995. *Computational fluid dynamics* (Vol. 206). New York: McGraw-Hill.

Bauer, W., 2010. *Hydropneumatic suspension systems*. Springer Science & Business Media.

Bergant, A., Ross Simpson, A. and Vìtkovsk, J., 2001. Developments in unsteady pipe flow friction modelling. *Journal of Hydraulic Research*, *39*(3), pp.249-257.

Beune, A., Kuerten, J.G. and Van Heumen, M.P.C., 2012. CFD analysis with fluid–structure interaction of opening high-pressure safety valves. *Computers & Fluids*, *64*, pp.108-116.

Curtiss III, J.P. and Buckley, M., 2003, January. Subsea Accumulators-Are they a False Reliance?. In *SPE/IADC Drilling Conference*. Society of Petroleum Engineers.

Davis, J.A. and Stewart, M., 2002. Predicting globe control valve performance—Part I: CFD modeling. *Journal of Fluids Engineering*, *124*(3), pp.772-777.

Delong, Z., Jun, J., Yuwen, H., Wei, W. and Wenjian, Z., 2014. Study on Well Control Technology of High Temperature Geothermal Drilling. *Procedia Engineering*, *73*, pp.337-344.

Dobroserdova, T., Olshanskii, M. and Simakov, S., 2016. Multiscale coupling of compliant and rigid walls blood flow models. *International journal for numerical methods in fluids*, *82*(12), pp.799-817.

Ghidaoui, M.S., Zhao, M., McInnis, D.A. and Axworthy, D.H., 2005. A review of water hammer theory and practice. *Applied Mechanics Reviews*, *58*(1), pp.49-76Srikanth, C. and Bhasker, C., 2009. Flow analysis in valve with moving grids through CFD techniques. *Advances in Engineering Software*, *40*(3), pp.193-201.

Good, C.A. and McAdams, J.P., 2001, January. Mathematical prediction and experimental verification of deep water accumulator capacity. In *Offshore Technology Conference*. Offshore Technology Conference.

Han, G., Ling, K., Khor, S.H., Zhang, H. and Thakur, R.K., 2013. Simulation of Multiphase Fluid-Hammer Effects During Well Startup and Shut-in. Oil and Gas Facilities, 2(06), pp.68-77.

Hati, A., Verma, N. and Chhabra, R., 2001. Transient analysis of gas flow in a straight pipe. *The Canadian Journal of Chemical Engineering*, 79(1), pp.18-27.

H. G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, COMPUTERS IN PHYSICS, VOL. 12, NO. 6, NOV/DEC 1998. (OPENFOAM ORIGINAL PAPER FOR CITATION)

Feng, T., Zhang, D., Song, P., Tian, W., Li, W., Su, G.H. and Qiu, S., 2017. Numerical research on F phenomenon of parallel pump-valve system by coupling FLUENT with RELAP5. *Annals of Nuclear Energy*, *109*, pp.318-326.

Fluent, A.N.S.Y.S., 2013. 14.0 ANSYS Fluent UDF Manual. *ANSYS, Inc. Canonsburg, PA, USA*.

Fluent, A.N.S.Y.S. "Ansys fluent theory guide." *ANSYS Inc., USA* 15317 (2011): 724-746.

Khandoker, S., Hydril LLC, 2008. Seal for blowout preventer with selective debonding. U.S. Patent Application 11/829,697.

Kolmogorov, A.N., 1942. Equations of turbulent motion of an incompressible liquid. *Izv. Akad. Nauk SSSR, Ser. Fiz*, *6*, pp.1-2.

Launder, B.E. and Spalding, D.B., 1973. The numerical computation of turbulent flows. In *Numerical prediction of flow, heat transfer, turbulence and combustion* (pp. 96-116). Pergamon.

Leutwyler, Z. and Dalton, C., 2008. A CFD study of the flow field, resultant force, and aerodynamic torque on a symmetric disk butterfly valve in a compressible fluid. *Journal of Pressure Vessel Technology*, *130*(2), p.021302.

McCain Jr, W.D., 2017. *Properties of petroleum fluids*. PennWell Corporation.

McCleney, A.B., Green, S.T. and Mueschke, N.J., 2018, April. Modeling of Subsea BOP Shear and Sealing Ability Under Flowing Conditions. In Offshore Technology Conference. Offshore Technology Conference.

McNutt, M.K., Camilli, R., Crone, T.J., Guthrie, G.D., Hsieh, P.A., Ryerson, T.B., Savas, O. and Shaffer, F., 2012. Review of flow rate estimates of the Deepwater Horizon oil spill. *Proceedings of the National Academy of Sciences*, *109*(50), pp.20260-20267.

Omana, R., Houssiere Jr, C., Brown, K.E., Brill, J.P. and Thompson, R.E., 1969, January. Multiphase flow through chokes. In *Fall Meeting of the Society of Petroleum Engineers of AIME*. Society of Petroleum Engineers.

Patankar, S.V. and Spalding, D.B., 1967. *Heat and mass transfer in boundary layers*. Morgan-Grampian.

Patankar, S., 1980. *Numerical heat transfer and fluid flow*. CRC press.

Sattler, J.P., 2002, January. BOP Subsea Hydraulic Accumulator Energy Availability, How To Ensure You Have What You Need. In *IADC/SPE Drilling Conference*. Society of Petroleum Engineers.

Shih, T., Liou, W.W., Shabbir, A., Yang, Z. and Zhu, J., 1994. A New Eddy Viscosity Model for High Reynolds Number Turbulent Flows: Development and Validation. *NASA TM*, *106721*.

Sirovich, L., 1987. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Quarterly of applied mathematics*, *45*(3), pp.561-571.

Slater, J. "Tutorial on CFD Verification and Validation." *Retrieved August* 6 (2008): 2014.

Song, X., Cui, L., Cao, M., Cao, W., Park, Y. and Dempster, W.M., 2014. A CFD analysis of the dynamics of a direct-operated safety relief valve mounted on a pressure vessel. *Energy conversion and management*, *81*, pp.407-419.

Springett, F.B., Ensley, E.T., Yenzer, D. and Weaver, S., 2011, January. Low force shear rams: The future is more. In *SPE/IADC Drilling Conference and Exhibition*. Society of Petroleum Engineers.

Springett, F.B., Pickett, G. and Franchek, M.A., National Oilwell Varco LP, 2016. *Method of closing a blowout preventer seal based on seal erosion*. U.S. Patent Application 14/960,263.

Tekin, A., 2010. *Blind shear ram blowout preventers: estimation of shear force and optimization of ram geometry*(Doctoral dissertation, The Ohio State University).

TRANSOCEAN Macondo Well Incident Investigation Report Volume I, June 2011.

Tulimilli, B.R., Naik, P., Chakraborty, A., Sawant, S. and Whooley, A., 2014, June. Design Study of BOP Shear Rams Based on Validated Simulation Model and Sensitivity Studies. In ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering (pp. V01BT01A026-V01BT01A026). American Society of Mechanical Engineers.

Versteeg, H.K. and Malalasekera, W., 2007. *An introduction to computational fluid dynamics: the finite volume method*. Pearson education.

Xu, Y., Guan, Z., Liu, Y., Xuan, L., Zhang, H. and Xu, C., 2014. Structural optimization of downhole float valve via computational fluid dynamics. *Engineering Failure Analysis*, *44*, pp.85-94.

Wilcox, D.C., 1988. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA journal*, *26*(11), pp.1299-1310.

Wilcox, D.C., 1998. Turbulence modeling for CFD (Vol. 2, pp. 172-180). La Canada, CA: DCW industries.

Wu, D., Yang, S., Wu, P. and Wang, L., 2015. MOC-CFD coupled approach for the analysis of the fluid dynamic interaction between water hammer and pump. *Journal of Hydraulic Engineering*, *141*(6), p.06015003.

Xing, L., Yeung, H. and Lo, S., 2011, June. Investigation of slug flow induced forces on pipe bends applying STAR-OLGA coupling. In *15th International Conference on Multiphase Production Technology*. BHR Group.

Zhang, X., Cheng, Y., Xia, L. and Yang, J., 2016, November. CFD simulation of reverse water-hammer induced by collapse of draft-tube cavity in a model pump-turbine during runaway process. In *IOP Conference Series: Earth and Environmental Science* (Vol. 49, No. 5, p. 052017). IOP Publishing.

Vita

Daniel Barreca was born in 1995 in Louisiana. He earned his Bachelor of Science in Petroleum Engineering from the Craft and Hawkins Petroleum Department at LSU in May 2017. He started his Master of Science in the same department in August 2017. He plans to graduate in August 2019.