Faculty of Science and Technology

MASTER'S THESIS

Study program/ Specialization: Master of Science in Petroleum Technology / Drilling and Well Technology	Spring semester, 2015 Open
Writer: Arne Kristoffer Torsdal	………………………………………… (Writer's signature)
Faculty supervisor: Kjell Kåre Fjelde John Emeka Udegbunam	
Thesis title: Inclusion of temperature in the AUSMV scheme with simulation examples from Underbalanced and Mud Cap Drilling	
Credits (ECTS): 30	
Key words: Matlab Dual gradient drilling simulation Underbalanced drilling Controlled mudcap Numerical diffusion AUSMV scheme Drift flux model	Pages: 71 + enclosure: 39 Stavanger, 15/06 2015 Date/year

Inclusion of temperature in the AUSMV scheme with simulation examples from Underbalanced and Mud Cap Drilling

Arne Kristoffer Torsdal

Juni 2015

MASTER THESIS

Department of Petroleum Engineering

University of Stavanger

Supervisor 1: Kjell Kåre Fjelde

Supervisor 2: John Emeka Udegbunam

Preface

This thesis have been written with the open source writing tool Latex. Latex is a free, flexible and interesting writing tool with a big innovative community. Latex utilizes codes as its primary writing style and this can reflect in how the program handles pictures and pdf's For the simulations in this thesis, Matlab has been used. A pre written Matlab code for the AUSMV scheme is used and modified. The code can be found in Appendix A and B.

Stavanger, 2015-06-14

Arne Kristoffer Torsdal

Acknowledgment

I would like to thank my family for the support they have given me during my five years of study.

I would also like to thank my fellow students of computer room E-353 at the university of Stavanger. Working side by side with great people have made the time on this thesis enjoyable.

I would also like to thank professor Mesfin Agonafir Belayneh for his help. Although I was not his student during this thesis he has always had time to assist and give directions.

And finally I would like to express great thanks to Professor Kjell Kåre Fjelde and John Emeka Udegbunam for their help as supervisor during this thesis.

A special thanks to professor Kjell Kåre Fjelde for always having the time and patience to help me during this semester. His positive attitude and helpfulness towards student are a great source for the University of Stavanger

Arne Kristoffer Torsdal

(A.K.T)

Abstract

Transient flow modelling is actively used to understand dynamic flow scenarios. In this thesis a modification and improvement has been performed for a transient flow model called AUSMV (Advective upstream splitting method). To get a understanding behind some of the simulation scenarios, a brief literary study has been performed on the dual gradient drilling concept and the dynamics behind underbalanced drilling

The dual gradient drilling concept is built on the manipulation of pressure using two different fluid densities. The hydrostatic pressure curve produced using dual gradient makes it easier to handle low margins between pore and fracture pressure. Drilling can be performed faster, cheaper and give better completion solutions. The downside with the dual gradient concept is the need for rig modification, re-training of personnel and new well control procedures.

The AUSMV scheme can be used to solve the transient drift flux model. This transient drift flux model uses a set of conservations and closure laws. The conservations of liquid and gas mass and the conservation of momentum. To close the system, four closure laws are added. Two of these closure laws are for the liquid and gas densities. These two closure laws were changed with new models which incorporate temperature effects.

An existing Matlab code of the AUSMV scheme was used as basis for the modifications and simulations. Two different drilling cases were simulated. The first one was a underbalanced drilling scenario with three different configurations. The main purpose of this simulation was to include and test the temperature dependent density and viscosity models for liquid and gas. The simulation results were compared against results produced from the original code. First the density models were implemented and tested. Then the viscosity was modified to include temperature. Simulation showed that the largest effect of including temperature, was related to the density models and their impact on the hydrostatic pressure which was reduced.

The second simulation was a controlled mudcap drilling scenario. The objective for this simu-

lation was to control the mud level using a suction pump. A suction pump removed mud from the middle of the well during simulation and the hydrostatic column dropped along with the bottom hole pressure. A steady state was reached when the injection rate of mass was equal to the suction rate. By adjusting the mass rates, the mud column could be adjusted and the hydrostatic bottom hole pressure controlled. Several modification had to be done to the model in order to make this simulation possible. First a sink term had to be implemented in the liquid mass conservation equation. In addition, the floating mud level in the well causes challenges for the outlet boundary condition treatment and several approaches for handling this was investigated. Finally the effect of numerical diffusion on the mud level interface was demonstrated and by refining the grid, a more accurate description of the mud interface was obtained. However this type of refinement had a computational time cost.

List of Figures

List of Tables

Acronyms

MPD Managed pressure drilling

RKB Rotary Kelly Bush

DGD Dual gradient drilling

BHP Bottom hole pressure

GOM Gulf of Mexico

FVS Flux-vector splitting

FVD Flux-difference splitting

UBD Underbalanced drilling

RCD Rotational control device

AUSMV Advective Upstream Splitting Method

CMC Controlled Mud Cap

ECD Equivalent circulation density

ROP Rate of penetration

Contents

Chapter 1

Objective

The goal of thesis can be summarized in two main objectives.

Main objective

The main objective is to make improvement to a numerical model called the AUSMV scheme. The AUSMV scheme is a numerical solution to a transient drift flux model for two phase flow and is used to simulate dynamic flow systems. The current AUSMV scheme utilizes functions which does not take into account temperature as a variable. Therefore the first main objective is set to be:

- Modify the AUSMV to be temperature dependant

To achieve this several modifications has to be done. Temperature affects density, viscosity and friction, so new and improved models need to be added to the model. For verification that the modifications are successful, a simulation scenario will be performed.

Secondary objective

The secondary objective is to modify the new temperature dependant AUSMV model to simulate a mud level control scenario for a dual gradient system. The mud level is to be controlled by a suction pump, which removes mud from the system and reduces the hydrostatic pressure.

Chapter 2

Introduction

The era of easy obtainable oil in shallow water is over. Discoveries of natural resources, in deep water areas like the golf of Mexico(GOM) and offshore Brazil drives the industry to develop new drilling techniques. Because of the small drilling window in deep water zones conventional drilling is no longer the optimal choice. One of these new drilling technologies is called dual gradient drilling(DGD).

Dual gradient drilling

DGD is one of the "new" technologies the industry has developed to improve drilling in deep water areas or difficult pressure zones. The concept of the system is to use two different fluid densities to manipulate the down hole pressure window. There are many different DGD systems, but they share the same concept that the riser is filled with a lighter fluid or gas, and the drilling mud and cuttings are diverted into a separate return line. The return line is connected to a subsea pumping system which pumps the fluids and cuttings back to the drilling rig. The main advantage with the DGD system is to the manipulation of the pressure regime. This can lead to a reduced number of casing placements, bigger production tubing and a reduction in number of rigs days.

Transient drift flux model

The transient drift flux model is a partial differential equation system used to simulate two phase flow. The model is build up on three different conservations laws. The conservation of mass (gas

and liquid) and conservation of momentum. When combining these three laws a set of seven unknown variables is found. To close out the amount of unknown variables, four closure laws are added.

AUSMV scheme

The AUSMV scheme is a numerical alternative solution to the transient drift flux model. AUSMV stands for advective upstream splitting method, and the V at the end refer to a modified velocity splitting function. The AUSMV scheme has been extracted into a Matlab model and different improvements and scenarios will be tested.

Simulation scenarios

In this section two different simulations cases will be performed. The first simulation is an underbalanced drilling(UBD) scenario where the well is unloaded by injecting lighter gas. The main objective in this simulation is to make the AUSMV scheme dependant on changes in temperature. To achieve this objective several key functions in the AUSMV scheme need revision. A modification will also be made to the viscosity for gas and liquid.

The second simulation will be a mud cap drilling scenario, where the mud level in the well will be controlled by a "pump" system. The new temperature dependant AUSMV scheme created in the first simulation will be used as a base model. Several modifications and fixes had to be implemented to get a working system.

In the final section of the thesis the results from the two different simulation cases will be presented and discussed.

Chapter 3

Dual gradient drilling

Dual gradient drilling is a drilling technique where two different fluid gradients are used to manipulate the bottom hole pressure(BHP). The DGD system is part of group of drilling techniques dubbed managed pressure drilling (MPD). The idea of a DGD system was discussed as early as in the 1960s, but was concluded to be out of reach with the current technological capabilities. There excists many different configurations for DGD, some of which will be presented in later chapters. This chapter focuses on the general dynamic of the system.

3.1 Concept

The basic principle of the DGD system is the manipulation of the BHP using two different fluid densities. The BHP is the summation of the hydrostatic pressure produced from these two fluid columns.(5) For a single density fluid, as with conventional drilling, the equation for hydrostatic pressure is used (3.1).

$$P_{Conventional} = TVD * g * \rho_{Mud} \tag{3.1}$$

For calculating the pressure created by a dual gradient hydrostatic column, equation 3.2 is used.

$$P_{DGD} = (D_{water} * g * G_{seawater}) + (TVD - D_{water}) * g * MW_{dualgradient} \tag{3.2}$$

An example with a seawater and mud gradient system is used. The first part of the equation represent the pressure from the seawater hydrostatic column. D_{water} represent the water depth, g

the gravitational constant and $G_{seawater}$ the density for seawater(1030 m^3/Kg)(6). The second part represent the hydrostatic pressure created from the mud column. TVD is the "True vertical depth" of the well and $MW_{dualgradient}$ (m^3/Kg) refers to the density of the drilling fluid. It is important to note that in DGD systems the mud gradient reference point is at the top of the mud line and NOT at the rotary Kelly bush(RKB) as with conventional drilling.

Figure 3.1 illustrates the fluid column in a DGD system vs a conventional one. The hydrostatic BHP are the same for both systems, but the height of the fluid columns differs.
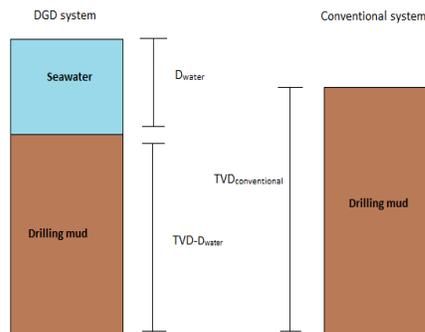
Figure 3.1: A DGD and conventional system with equal BHP

If these two fluid columns are plotted with a given length, a pressure profile can be constructed. Figure 3.2 illustrates the pressure curve for a DGD system and a conventional drilling system.
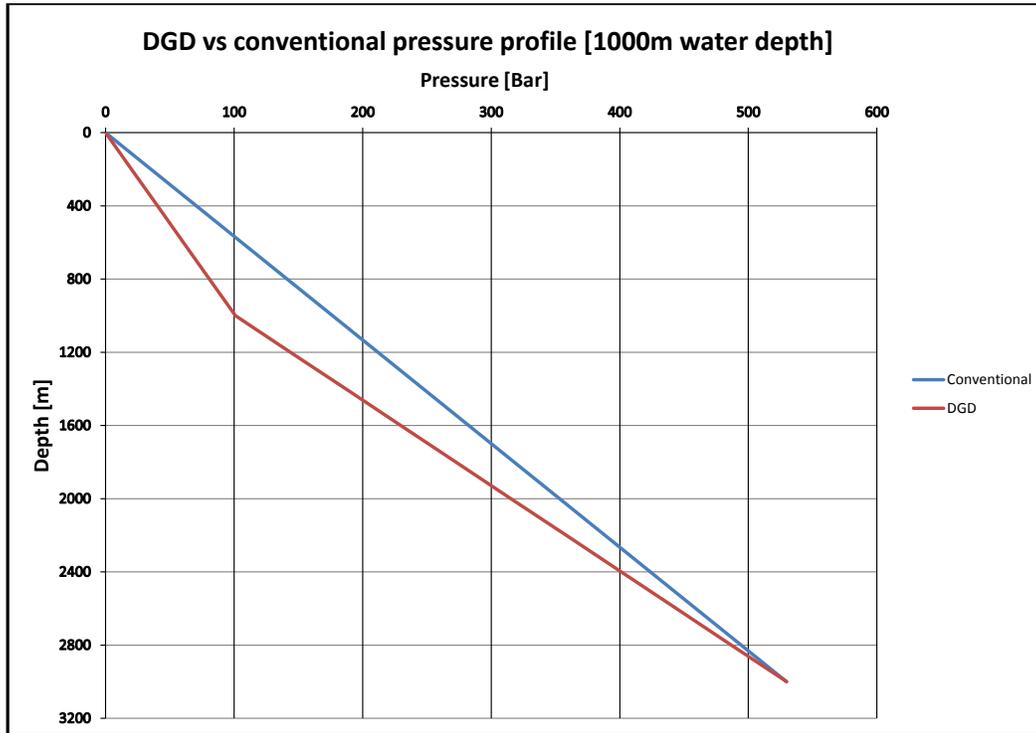
Figure 3.2: DGD vs conventional pressure profisle

The figure have two graphs representing the conventional and DGD pressure curves. The conventional pressure curve is calculated using equation 3.2 with a constant mud weight and a depth of 3000 meter. The graph can be seen to be linear which relates to a homogeneous fluid column. For the DGD system the first 1000 meters represent the seawater gradient that is created with a seawater filled riser. The remaining 2000 meter can then have mud density higher then normal and still produce adequate BHP. This is the essence of a dual gradient system.

Figure 3.3 shows a picture of a conventional drilling system and a DGD system called subsea mudlift drilling.
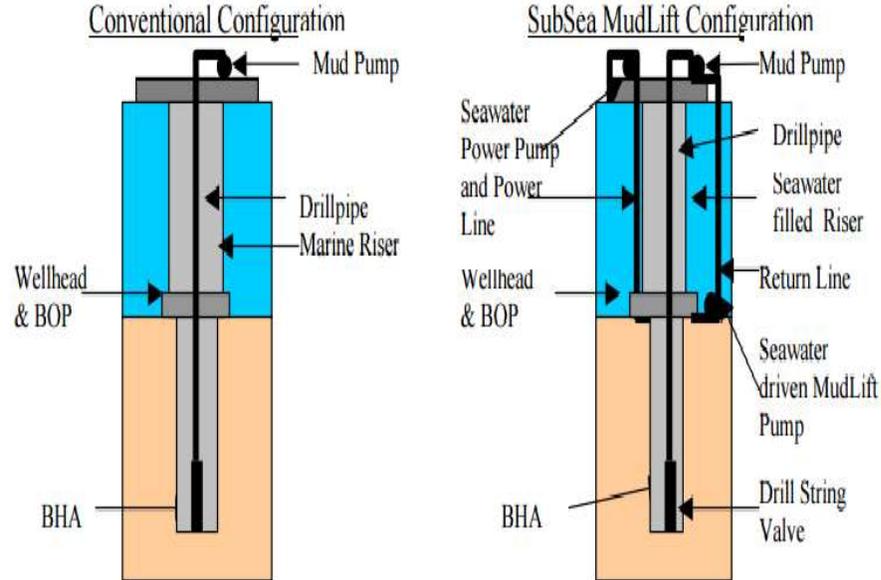
Figure 3.3: Conventional drilling configuration(left) and subsea mudlift system(right)(1)

For a conventional drilling system the drilling fluids and cuttings are returned to the rig trough the annulus and riser. For a DGD systems the drilling fluids moves up the annulus just like with a conventional system, but are diverted at the riser into a return line. The drilling fluid and cuttings are then pumped up this return line via a subsea pumping system and returned to the rig. The riser above the suction point can be filled with seawater or lighter engineered fluids to achieve the dual gradient effect. There exists several different configuration for dual gradient systems. One example is the subsea Mudlift drilling system (SMD) shown on the right side in figure 3.3. (5)

3.2 Join industry project

3.2.1 Initial motivation

In the 1960's the concept of a dual gradient system started to gather interest. The original concept was built around a riser less drilling technique for deep water areas. The concept was at

the time interesting, but deemed unnecessary due to improvements to conventional drilling techniques. In the early 1990's several deep water reservoirs was discovered in the gulf of Mexico(GOM) and revitalized the concept of a DGD system. Due to a high demand on rigs, especially those with deep water capabilities, some operators and drilling contractors wanted to explore new possibilities. This industry desire lead to a weekend workshop, which lead to a five year project now known as the "subsea mudlift drilling joint industry project". (5)

3.2.2 Joint industry project

The need for a new drilling technology was apparent within the oil industry. Conoco and Hydril personnel was the first to investigate the possibility of this new technology dubbed dual gradient drilling. After initial research there was concern that the development of this new technique would be to expensive and risky for just two companies. Therefore in 1996, a workshop was organized and 25 operators, drilling contractors and service companies was invited. The intention was to organize a joint venture to explore and develop this new drilling technology and to confirm the technologies feasibility. This workshop led to the creation of the joint industry project (JIP), now known as "subsea mudlift drilling JIP". Conoco was set up to oversee the project as administrative leader and Hydril as project designer. The JIP was divided into three main phases.(5)

- Phase 1 Conceptual engineering

- Phase 2 Component design and testing

- Phase 3 System design, fabrication and testing

Since this was entirely new ground for the industry, phase one was set up as a research and development phase. A basic design for the system was developed which could incorporate the existing rig fleet and utilize existing technologies. One of the most important factors of this phase was to determine possible problems or "show stoppers" in the later phases.

In phase two the individual component was designed, built and tested. The drilling operation and well control procedures was created and an initial rig integration study was conducted to

prepare the industry for the coming technology.

Phase three was the most expensive and the one with the highest financial risk. In this phase an operator, a test field and a test rig was required. The goal was to assemble and integrate the DGD system on a test rig and successfully drill a test well. One of the most important focuses in this phase was training of personnel. The technology was completely new for the industry and all level of personnel involved received renewed training in preparation for the test well case.

3.3 Problems associated with conventional drilling in deepwater areas

Some problems that are associated with conventional drilling in deep water areas can be eliminated with the use of different DGD systems. These are some the problems encountered when drilling with conventional methods in deepwater areas(7).

- Deck space limitations.

The water depth causes the need for long risers which again affects the deck space. For older and smaller vessels this may cause the need for additional storage on another nearby ship. The deck space is also limited because of the increased size of the mud pits.

- Cost of mud to fill riser

For conventional drilling the mud column goes from the bottom of the well and all the way up to the drill floor. In deep water, a longer riser means extra mud is needed to completed the mud circulation loop.

- Huge deck loads due to riser pipes and mud volumes

Heavier load can be a problem for rigs when drilling deep water. The increase in pipes and mud volumes increase the total load of the rig.

- Large number of casing strings

In deep water drilling, the pressure margins between collapse and fracture tends to be smaller. Because of this pressure regime, multiple casings is needed to reach target depth. This is both expensive and time consuming. When forced to use many casings, the final size of the production tubing might also be reduced.(7)

3.4 Advantages with DGD systems

There are several significant advantages with using a DGD system in deep water or difficult formation pressure zones. The main advantages can be summarized as:

- Reduced well cost

- Reduction of casing size and number

- Improved primary cement capabilities

One of the most important advantages of a DGD system are the reduction in casing numbers. With the flexibility in the pressure management in DGD system, fewer casing strings is often needed(8). With the reduced number of casing strings, the rig and operational costs is also reduced.

On ultra deep water much of the operational time is used for well control and only 1/3 for actual drilling. If successful a dual gradient system can reduce costs with over 50 % (8). Each reduction of casing saves 4 to 6 rig days, the cost of the casings, hole evaluation, and casing and logging. Fewer casing strings also means more options for the completion system of the well.

With bigger spacing between the casing strings, the problem of trapped annular fluids can be reduced. Trapped annular fluid occurs when fluids from one casing string is circulated back into the previous string.

There are also some advantages with regards to the deck space. Since mud is no longer returned trough the riser, but trough the return line, a significant smaller amount of mud is required. This affects the deck space and the cost.

3.5   Challenges for DGD systems

There are some challenges with regards to the use of DGD technology. The main challenges are

- Well control and the u-tube effect

- Rig modification

- Procedure rework and crew training

Most of these challenges comes from inexperience and that the system is of very new design.(9)

U-tube effect

The u-tube effect can best be described as a distribution of the bottom pressure for two pipes until the pressure on both sides are equal. For conventional drilling the pressures in the annulus and drill pipe is for most cases equal. Equation 3.3 shows the relationship between annulus and drill pipe.

$$P_{bot} = \rho_{annulus} * h_{annulus} * g = \rho_{drillpipe} * h_{drillpipe} * g \tag{3.3}$$

This assumption is correct when the well is in static conditions with equal mud density in annulus and drill pipe. During drilling operations the annulus will be filled with cuttings and mud which will give different pressures.

The u-tube effect in a DGD system is different and more complicated. In figure 3.4 the depiction of a dual gradient system can be seen. For most DGD system the riser is filled with water or a lighter than mud gradient. For a riserless dual gradient systems the u-tube effect can be problematic. For riserless DGD systems the dual gradient effect is artificially created using subsea suction pumps. If the pumps stops creating the seawater gradient effect the well pressure will be unbalanced and the mud in the annulus will return into the drill string. (10)
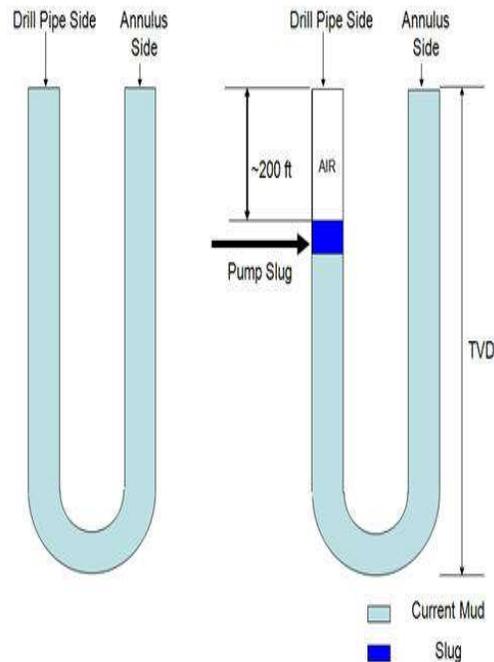
Figure 3.4: U-tube effect for drilling(2)

The main driving force behind the u-tube effect is the relative densities of mud and water, and the water depth. With increased water depth, the mud column in the drill pipe is increased to counteract the pressure. There are four other factors that also influence the u-tube, but these four only affect the time before the u-tube equilibrium is reached. These four are bit nozzle size, inside diameter of the drill string, mud viscosity and total well depth. The u-tube effect can diminished by the installation of flow control valve.(10)

Rig modification

Most of the rig fleet available for drilling are of conventional configuration(10). To utilize a DGD system on a conventional rig several costly and time consuming equipment changes has to be done. Some equipment can vary from different systems, but these are some of the essential components needed for a DGD system:

- Mudlift pumping system

- Mud return line

- BOP suction line attachment point

- Subsea rotating device for barrier control

- Control module

New procedures and training

One of the most important areas for the success of an operation is to have well trained personnel and good procedures in place. With this new drilling system, all the current procedures required revision. Procedures for making a connection and tripping, has been revised to meet the new parameters for a DGD system. Drill crews need retraining and familiarization with the new system and the procedures.

Chapter 4

Review of drilling concepts with relevance for simulation

In this thesis two different simulation cases are performed. The first one is a underbalanced drilling(UBD) scenario. UBD is a drilling concept where the hydrostatic pressure is lowered below the pore pressure. UBD is not of dual gradient nature, but represent a good two phase flow dynamic, which can be simulated with the AUSMV scheme. Therefore the basic concept of UBD will be presented.

The second concept is of a controlled mudcap drilling(CMC) scenario which relates to the second simulation of this thesis. In CMC drilling the mud level in the well is controlled by a pump system.

4.1   Underbalanced drilling

Underbalanced drilling(UBD) is performed with a well pressure that is lower than the pore pressure which allows hydrocarbons production during drilling. The pressure profile for a UDB scenario can be seen in figure 4.1.
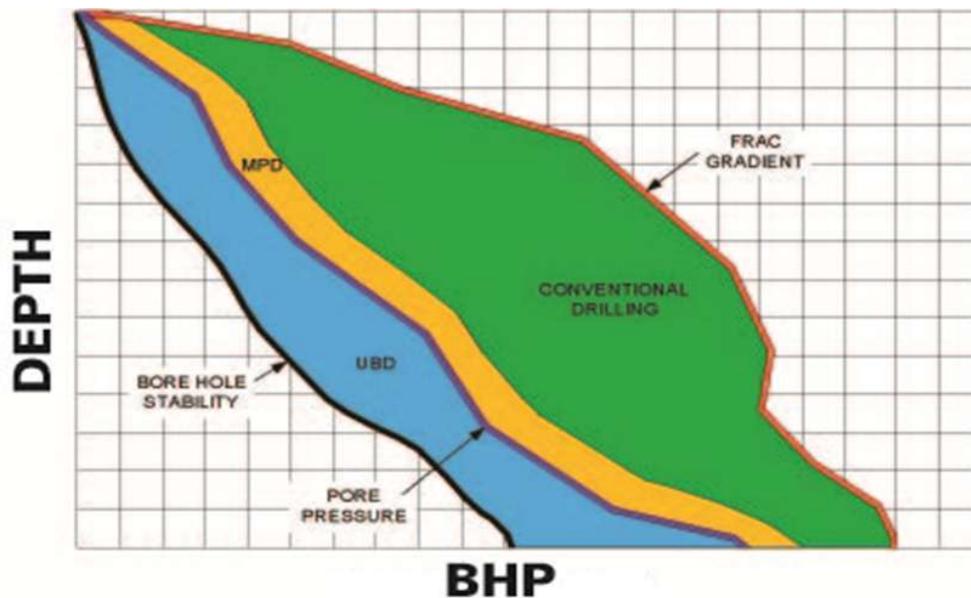
Figure 4.1: Common BHP for different drilling systems (3)

The drilling technique can be used for handling low margin fields, improve ROP and reduce formation damage. It requires additional safety barriers and separation equipment. Well flow is also directed through a choke, which gives an additional pressure control on the drilling system. For low formations pressures underbalanced conditions in the well can be achieved by injection of nitrogen together with the drilling mud. The unloading scenario is when the pressure is reduced to create the underbalanced conditions. When drilling into the reservoir, reservoirs fluids will also contribute in reducing the pressure. During drilling a multiphase flow system will be present which can be highly dynamic in nature. Hence advanced transient flow models must be used to simulate the dynamics of the system. The first simulation scenario performed in this thesis is inspired from a UBD scenario. (11)

4.1.1 Bottom hole pressure

Bottom hole pressure in UBD is comprised of three main components. The pressure are a combination of the hydrostatic, frictional and acceleration pressure. The frictional and acceleration pressure are considered the dynamic part(12).

Due to the complexity of multiphase flow, computer models are needed to simulate hydrostatic and frictional components. In the SPE paper Nessa et al(12), simulations with gas injection were performed. The results showed that a well could be hydrostatic dominated or frictional dominated. The optimal circulation point was found to be when the reduced hydrostatic pressure is balanced against increased annular friction. This optimal point is the minimum achievable BHP for a given liquid injection rate. If an increase in gas occurs at this point the well will move from being hydrostatics dominated, to frictional dominated. During UBD operation it is very important to know which system is dominating. If a reduction in BHP is required injecting more gas to unload the well may have the opposite effect((12).

4.1.2 Fluid selection

The fluid selected as circulation medium are often water or hydrocarbon based Newtonian fluids. Simple fluids which are compatible with the formation liquid are desired. Viscosifiers as Bentonite gel and polymers are not used because their effect on friction. Higher viscosity increases the annular friction and forces a higher rate of gas injection to achieve the optimal BHP. viscosifying materials can also cause emulsion between water and oil, which reduces the surface separation efficiency.(12)

4.1.3 Well control

For UBD system the term well control can be replaced with flow control, as the core concept is to produce while drilling. The system is a closed loop system. In the closed surface control system, a multiple flow path system approach is used to improve safety and redundancy. For conventional drilling a two barrier principle are used, with the hydrostatic mud column and BOP stack as barriers. For UBD it is not possible for the mud column to act as the primary

barrier. Instead the BOP acts as the primary barrier with a secondary system for the separation module.(12)

4.2 Controlled mud-cap drilling

CMC is a MPD drilling technique where the well pressure is controlled by adjusting the mud levels in the well. The pore pressure range for a MPD technique can be seen in figure 3.1. The mudcap is a reference to the top of the mudline inside of the riser. The mudline can be increased and decreased to adjust the BHP. The mud level is controlled by a pump system connected to the riser and will evacuate mud and cuttings into a separate outlet. From this outlet the fluid travels to a subsea pump which is connected to the return line. The return lines function is to transport the mud and cuttings from the seabed and onto the drilling rig. The subsea pump is specially constructed to handle fluids combined with large cutting particles. Depending on the suction pressure at the outlet, and the injection rates through the drill string, the mud column height is regulated. To control and regulate the mud column, a real time, transient multiphase flow simulator is connected to the subsea mud pump system. This system controls the pumps speed, which control the suction pressure. The margins above the mudline inside of the riser will be filled with air or gas contributing to a dual gradient effect.(13)

4.2.1 ECD compensation

During connections or general stop of circulation the equivalent circulation density (ECD) will drop due to loss of friction. Since MPD techniques like the CMC system operates close to the pore pressure, a compensation to the BHP is needed when circulation stops. This compensation is achieved with regulating the mud column in the riser. Since reliable real time BHP data transmissions systems is not yet fully field proven, a engineered approach is needed to adjust for the ECD. With the engineered approach a pressure management system is needed. This system need to take real time well parameters(eq mud weight and column height), calculate the pressure and add for the compensation. Then the speed of the pump is adjusted to either rise or lower the mud level in the riser. With regards to multiphase flow the engineered approach becomes much more complicated. For multiphase flow situations transient modelling are re-

quired to calculate the pressure profiles. (13)

4.2.2 Well control

Well control for a CMC system can be based on two principles, open or closed system. Other MPD systems usually use closed principle, but the CMC system has the advantage that it can operate either as an open system or as a closed system. When a system is refereed to as open it means that the upper part is open to the atmosphere. There are some advantages associated with the use of a open system on a floating rig. In a open system there are no need for a rotational control device(RCD) to keep the BHP pressure constant. This is especially valuable in harsh environments where large heaves caused by rig movements can affect the BHP control. With the open well control option, the CMC system will have a pressure regime that resembles that of a conventional drilling system. The difference lies in the dual gradient effect in the riser, where the mud height is lower but denser, and gas or air is filled above the mud line.

One challenge to CMC system is when a fast shut in scenario occurs and the subsurface BOP pipe rams closes. The resulting u-tube effect can fracture and damage the formation. This can also occur when circulation starts up again with a closed BOP. To control this effect a valve inside the drill string is installed. This valve opens at predetermined pressure and compensate for the static unbalance in the annulus and drill pipe. When the system is in static condition the valve will close if the pressure in the annulus is lower than the drill pipe pressure thus avoiding the u-tube effect.(13)

Chapter 5

Transient drift flux model

When advanced and complicated well operations is being planned transient flow modelling can be a extremely useful tool. A flow model can describe pressure, temperature, density and viscosity changes for a given system. The transient drift flux model is a two phase model of one dimension. The model is comprised of two mass conservation laws, one momentum conservation law and four closure laws. Another model based on the drift flux model will be presented later called the AUSMV scheme.

The theory on the drift model is taken from the SPE paper Udegbunam et al(14) and Evje et al(15).

The drift flux model generic form can be expressed as equation 5.1.

$$\partial_t W + \partial_x F(x, W) = G(x, W) \tag{5.1}$$

Here W represents the conservative variables, F represents the fluxes, G represent the source term, x is the coordinate along the flow direction and T represent the time variable.

5.1   Conservation laws

When it comes to the modelling of fluids in a system, a very important concept is put to use. The notion that some properties are conserved and remain equal throughout a system is a remarkable idea. The realization of these conservation laws can be credited as one of the greatest achievements for modern science and are the backbone of many modern physics fields. The basic idea is that however complicated a systems has become, some basic properties like density and pressure are in effect during the process at all times(16). The three conservations laws used in the development of the drift flux models are the conservation of liquid and gas mass, and the conservation of momentum. Since the model is based on two phase flow, there are need two different mass equations. One equation for the liquid mass and one equation for the gas.

- Conservation of mass (liquid and gas)

The conservation law of mass refer to the preservation of mass during a systems lifetime. Since the system is without any chemical reaction, equation 5.2 describes the most basic assumption.

$$Mass_{inn} = Mass_{out} \tag{5.2}$$

Equation 5.3 and 5.4 are the conservation laws for liquid and gas respectively.

$$\frac{\partial}{\partial t}(A\rho_l\alpha_l) + \frac{\partial}{\partial z}(A\rho_l v_l) = s_1 \tag{5.3}$$

$$\frac{\partial}{\partial t}(A\rho_g\alpha_g) + \frac{\partial}{\partial z}(A\rho_g v_g) = s_2 \tag{5.4}$$

It is assumed that there is zero exchange between the gas and liquid mass. Therefore it is assumed that $s_1 = s_2 = 0$.

- Conservation of momentum

$$\frac{\partial}{\partial t}(A(\rho_l\alpha_l v_l + \rho_g\alpha_g v_g)) + \frac{\partial}{\partial z}(A(\rho_l\alpha_l v_l^2 + \rho_g\alpha_g v_g^2)) + A\frac{\partial}{\partial z}\rho = -A(\rho_{mix}g) + \frac{\Delta p_{fric}}{\Delta z} \tag{5.5}$$

The equation for conservation of momentum is based on newtons laws of motion. It states that in a isolated system the total momentum is preserved. "The rate of change of the momentum of a system is the sum of all external forces acting on a system"(17)

- Conservation of energy

An equation for the conservation of energy is also often used in modelling, but since this model is based on isothermal conditions this part is neglected.

- Conservative vector form

These three conservation equations can be written on a conservative vector form as shown in equation 5.6.

$$
\partial_t \begin{pmatrix} \alpha_l \rho_l \\ \alpha_g \rho_g \\ \alpha_l \rho_l v_l + \alpha_g \rho_g v_g \end{pmatrix} + \partial_x \begin{pmatrix} \alpha_l \rho_l v_l \\ \alpha_g \rho_g v_g \\ \alpha_l \rho_l v_l^2 + \alpha_g \rho_g v_g^2 + P \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -q \end{pmatrix} \tag{5.6}
$$

and again in a more condensed format as shown in equation 5.7.

$$
\partial_t \begin{pmatrix} W_1 \\ W_2 \\ W_3 \end{pmatrix} + \partial_x \begin{pmatrix} W_1 v_l \\ W_2 v_g \\ W_1 v_l^2 + W_2 v_g^2 + p(W_1, W_2) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -q \end{pmatrix} \tag{5.7}
$$

In this set of equations there are a total of 7 unknown variables and only three equations. Therefore a set of four closure laws will be used. The variables needed to be solved are $\alpha_l, \alpha_g, \rho_l, \rho_g, v_l, v_g$ and q. α_l and α_g represents the volume fractions of the liquid and gas. ρ_l and ρ_g represent the fluid densities and v_l and v_g are the phase velocities. The the variable q represent the source term.

5.2 Closure laws

Closure laws are equations that are used to reduce the number of unknown variables in system. Often simplified equations are used that can express the unknown variables with known quantities. In the drift flux model four different closure laws are used. One of the main objective of this thesis is to make the AUSMV scheme temperature dependant. To accomplish this several of the old closure laws will be replaced. Both the old and new laws will be presented and explained. The four closure laws are:

- Gas slippage

- Liquid density

- Gas density

- Friction model (source term)

5.2.1 Gas slippage

In equation 5.7 which represent the flux model in conservative vector form, two equations representing the fluid velocities can be found. To solve this, equation 5.8 is introduced.

$$v_g = K * v_{mix} + S \tag{5.8}$$

v_{mix} is the fluids mixture velocity, v_g represent the gas velocity and K and S are the flow dependant parameters. The variables S is the slip ratio between liquid and gas, and K is the distribution coefficient.

5.2.2 Density models

In equation 5.6 two variables representing the liquid and gas density is included. To solve these two more closure laws is added. Because of the temperature influence on the densities two new closure law is added.

Original liquid density model

The liquid density model used in the origal simulation is based on (Imput here). Equation 5.9 is the basis for the old densisty closure law.

$$\rho_l(p) = \rho_{lo} + \frac{(p - p_o)}{a_l^2} \tag{5.9}$$

Since it is assumed that the liquid drilling fluid is water, the referance density and pressure is set to 1000 kg/m^3 and $p_o = 100000 Pa$ respectably. a_l represent the rate of which speed travel in water and is equal to $1500 m/s$.

New liquid density model

Since temperature affects the density of the fluids, a new function was necessary in the model. Since the temperature and pressure effect on liquid density are diminutive, a linerized equation can be utilized. Equation 5.10 is based on the equation of state for the liquid density. An equation of state is a functional relationship between state variables (18). For this equation the state variables is volum, temperature and pressure.

$$\rho_l = \rho_0 + \frac{\rho_0}{\beta}(p - p_0) - \rho_0 \alpha (T - T_0) \tag{5.10}$$

In equation 5.10 the parameters ρ_0, p_0 and T_0 represent the reference density, reference pressure and reference temperature . β represents the bulk modulus and α is the volumetric thermal expansion coefficient.

Original gas density model

For the gas density equation

$$\rho_g = \frac{p}{a_g^2} \tag{5.11}$$

Here a_g is represented as the speed of sound and p is pressure.

New gas density model

Since temperature also affects the gas density, a new equation with a temperature variable is needed. Equation 5.12 is called the ideal gas law and is used as the basis for a new gas density equation.

$$pV = nRt \tag{5.12}$$

p represent the pressure, V is the volume, n is the mole count, R is the universal gas constant and t the temperature. Equation 5.13 is added to give the ratio of gas mass divided by mole mass.

$$n = \frac{m}{M} \tag{5.13}$$

Equation 5.14 is added to remove the mole mass variable, by dividing the individual gas constant by mole mass. The new constant R_{Spec} represent the specific gas constant.

$$R_{Spec} = \frac{R}{M} \tag{5.14}$$

Equation 5.13 and 5.14 are then imputed in equation 5.12 and the end result is equation 5.15

$$pV = \frac{m}{M} R_{spec} t \tag{5.15}$$

Using the fact that density is mass divide by volume, equation 5.15 is rewritten to equation 5.16

$$\rho_g = \frac{p}{R_{spec} T} \tag{5.16}$$

The next step is to substitute equation 5.10 with the liquid density. Now these new formulas for gas and liquid density, which takes into account temperature can be implemented to the AUSMV scheme.

Phase volume fraction

The phase volume fraction relates the distribution of volume between liquid and gas. The sum of the different fractions will always be equal to one and is expressed with equation 5.17.

$$\alpha_l + \alpha_g = 1 \tag{5.17}$$

5.2.3 Source term

The source term is represented by the variable q. The source term can be divided into two parts, F_g and F_w. F_g is the gravity term for the source expression, but since the well is vertical with zero degree inclination this equation equals zero and is not added to the two phase model. F_w represent the loss of pressure due to friction and can be calculated using equation 5.18.

Friction equation

F_w is calculated by using equation

$$F_w = \frac{2f\rho_{mix}v_{mix}abs(v_{mix})}{d_{out} - d_{in}} \tag{5.18}$$

To find the friction factor f, the type of flow first has to be determined. The type of flow is determined by the Reynold number and can be by equation 5.19. If the Reynold number is at 2000 or bellow the flow is laminar. For Reynold numbers between 2000-3000 the flow is said to be in a transitional phase. For Reynold numbers above 4000 the flow is said to be turbulent. The difference flow patterns determine what kind of friction factor formula is used.

$$N_{Re} = \frac{\rho_m abs(v_m)(d_o - d_i)}{\mu_m} \tag{5.19}$$

5.3 Hyperbolic system

Partial differential equation can come in many forms and higher orders. Partial differential equations can be elliptic, parabolic or hyperbolic. Elliptic equations are often associated with steady state behaviour. Parabolic and hyperbolic systems often describes physical phenomena

that evolves in time. Notable examples are the heat diffusion equation and water acoustic propagation. Hyperbolic system are often used to model transport of a physical medium, e.q fluids or sound waves. The drift flux model is of the hyperbolic type.

Chapter 6

AUSMV scheme

The Advection Upstream Splitting Method(AUSMV) is a numerical alternative to the drift flux model. Due to excessive dissipation at contact discontinuities for flux vector splitting(FVS) schemes another alternative have been created. This new scheme called AUSMV combines the efficiency of the FVS scheme and the accuracy of the flux difference splitting(FDS) scheme. To avoid the dissipation from FVS schemes the velocity splitting function shown in equation 6.1 is used.

$$\tilde{V}^{\pm}(v, c, X) = \begin{cases} \beta, & |v| \leq c \\ \frac{1}{2}(v \pm |v|) & |v| > c \end{cases} \tag{6.1}$$

where $\beta = X^{V^{\pm}}(v, c) + (1 - X)\frac{v \pm |v|}{2}$.

For a more detailed look into the underlying FVS scheme see Udegbunam et al(14) or Evje et al(15).

6.1 Discretization

When building a numerical simulation model the discretization step is an important one. Numerical discretization is often split into two categories, an explicit or implicit structure. For the explicit model the dependent variable is known and set at a fixed amount, while with an implicit setup the dependant variable is calculated from equations, matrixes or iterations. This

makes implicit calculation much more complicated and often require strong processing power, but the end result is more accurate. The whole length of the well is split up into boxes of equal sizing. Each box has a equal length which is denoted with a Δx and represent one part of the well as can be seen in figure 6.1. With an increased amount boxes can give more accurate results, but require more computing power and makes a simulation loop run for a much longer time. ((16)(19))
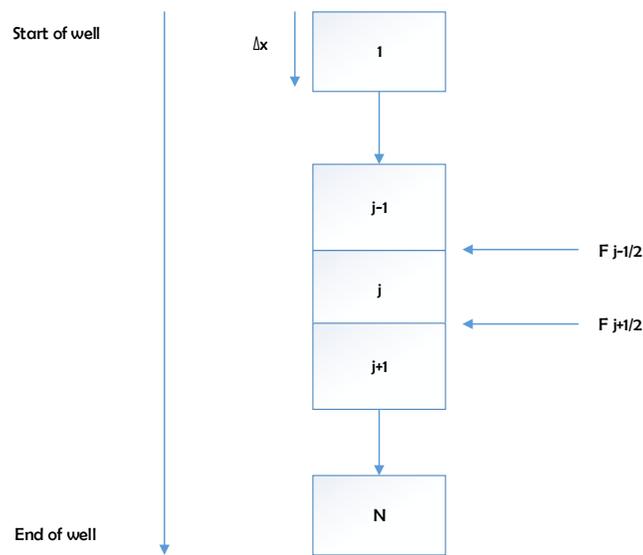
Figure 6.1: Discretization of a numerical system

To calculate the values for W_1, W_2 and W_3 for the next time step equation 6.2 are used. $W_{i,j}$ represents the mass and momentum conservative variables for $i = 1, 2$ & 3, which are found in chapter 5.1 Conservation laws. The term F^{AUSMV} represent the mass fluxes The Δt variable represent a CLF condition which will be discussed later in the chapter.

$$W_{i,j}^{n+1} = W_{i,j}^{n} - \frac{\Delta T}{\Delta Z}(F_{j+\frac{1}{2}}^{AUSMV} - F_{j-\frac{1}{2}}^{AUSMV}) - \Delta t q_i^n \qquad (6.2)$$

6.2 Handling of flow area change

A real well during drilling will have sections where the flow area will change as shown in figure 6.2. Therefore an option to handle any potential well geometric change exist in the AUSMV scheme.
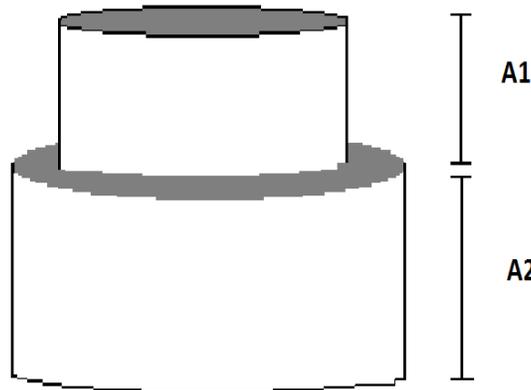
Figure 6.2: Flow area discontinuity

The continuity equations 6.3 is put to use here. This equation relate the fact that the system is locally conserved. The subscript L and R in the equation denotes the left and right of the equation.

$$(A, \alpha_l \rho_l v_l)_L = (A, \alpha_l \rho_l v_l)_R \tag{6.3}$$

$$(A, \alpha_g \rho_g v_g)_L = (A, \alpha_g \rho_g v_g)_R$$

$$w_{i,j}^{n+1} = w_{i,j}^1 - \frac{\Delta T}{\Delta Z}(H^c + H^p) - \Delta t q_i^n \tag{6.4}$$

Equation 6.4 is used when encountering area changes. This equation resembles equation 6.2 for

the dizcretization.

$$H^c = A_R(F^c_{l,g})_{j+\frac{1}{2}} - A_l(F_{l,g^c})_{j-\frac{1}{2}} \tag{6.5}$$

$$H^p = A_{avg}(F^p_{j+\frac{1}{2}} - F^p_{j-\frac{1}{2}}) \tag{6.6}$$

The variable A refer to the cross sectional area and will consist of A_l and A_r which refer to left and right side of the flux cell.

$$A = \frac{\pi}{4}, A_{avg} = \frac{1}{2}(A_l + A_r) \tag{6.7}$$
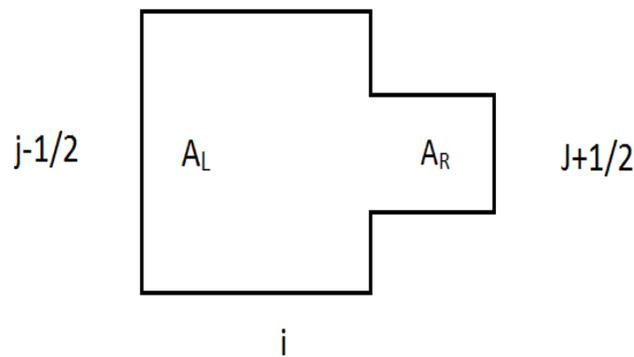
Figure 6.3: Area change within a cell(4)

6.3   Boundary condition and stability

In the numerical scheme the inlet and outlet fluxes in the boundary cells must be treated separately. The AUSMV scheme does not provide formulas for the boundary fluxes. The drift flux model uses a extrapolation method to specify the inlet condition. Equation 6.8 uses the pressures of the first two cells to calculate the inlet pressure flux.

$$P_{Inlet} = p(1) + 0.5(p(1) - p(2)) \tag{6.8}$$

For the outlet boundary conditions two different scenarios can be simulated. Closed or open end well conditions. This describes the well scenario as open or shut in.(15)

For open end boundary conditions the convective fluxes is calculated using extrapolation of the primitive variables. The primitive variables used are the liquid and gas densities, phase velocities and volume fractions. These are used to determine the outlet mass and momentum fluxes of liquid and gas. The outlet pressure flux is set to a fixed value and represent either atmospheric conditions or choke back pressure.

For closed well condition another extrapolation method is used. Equation 6.9 shows how the outlet boundary is specified. In this case mass and momentum fluxes of liquid and gas, are set to zero while the pressure is extrapolated.

$$P_{Outlet} = p(N) + 0.5(p(N) - p(N-1)) \tag{6.9}$$

Since the system represents two phase flow there are three different eigenvalues. The first and third eigenvalue represent the the wave propagation, upward and downward respectably. The second eigenvalue is the speed of gas volume travelling downstream. The eigenvalues can be written as $\lambda_1 = v_l - \omega$, $\lambda_2 = v_g$ $\lambda_3 = v_l + \omega$. (14)

If the condition in equation 6.10 is met and the liquid is assumed incompressible then an expression for the sound velocity ω can be found.

$$\alpha_g \rho_g \ll \alpha_l \rho_l \tag{6.10}$$

Equation 6.11 is an expression for the sound velocity.

$$\omega = \sqrt{\frac{p}{\alpha_g \rho_l (1 - K\alpha_g)}} \tag{6.11}$$

As mentioned above there are three eigenvalues for a two phase flow system. Figure 6.4 shows how the eigenvalues propagates at the outlet boundary of a given cell. One wave is propagation inwards, while two other are propagating outwards. The direction of the eigenvalues is determined by its sign. If the value of the eigenvalue is negative the wave propagate backwards. If the wave is positive it propagate forward. The number of waves propagating in the system also determines how many constraints has to be put on the inlet and outlet boundary. At the inlet boundary two conditions can be set, in this case fluxes of gas and liquid. At the outlet only one condition can be set, which for open well conditions will be pressure. If the well is closed in, it is natural to specify the gas and liquid fluxes to zero, and extrapolate the pressure.
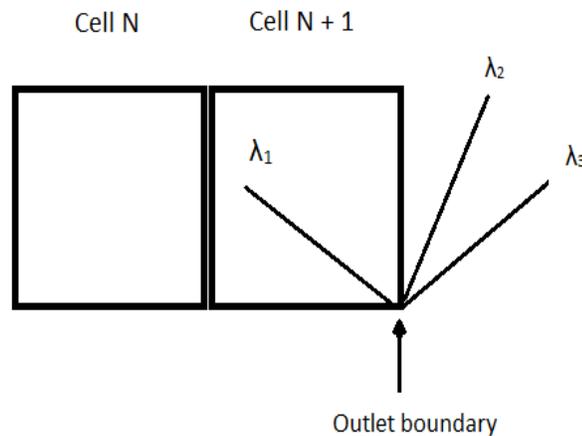
Figure 6.4: Eigenvalue propagation at the outlet boundary for a cell

6.4 CFL condition

When simulation with hyperbolic partial differential equation stability will often be a problem. Therefore a time restriction element named Courant Friedrichs Lewy (CFL) condition is implemented. This condition sets a restraint on how the simulation will seek information. ((16),(19))

$$\Delta t = CFL \frac{\Delta Z}{max(|\lambda_1|, |\lambda_2|, |\lambda_3|)} \tag{6.12}$$

Here $\lambda 1$, $\lambda 2$ and $\lambda 3$ refer to the eigenvalues of the scheme.

In 6.5 an example of a stabile CLF case is shown. The triangle represent the information that propagates in a simulation. The black dots represent different cells. It can be seen that information gathered at a given cell does not affect other time steps.
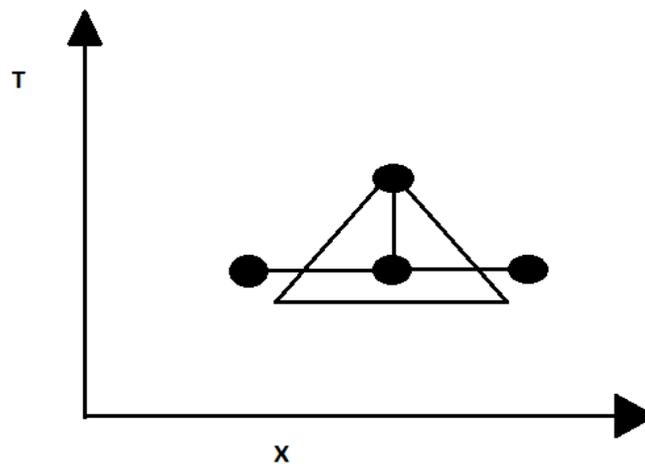
Figure 6.5: Stabile CFL condition

In 6.6 an unstable CFL case is shown. Here again the triangle represent the information propagation, but now it exceeds the next simulation box. This can cause unstable results and affects the reliability of a simulation.
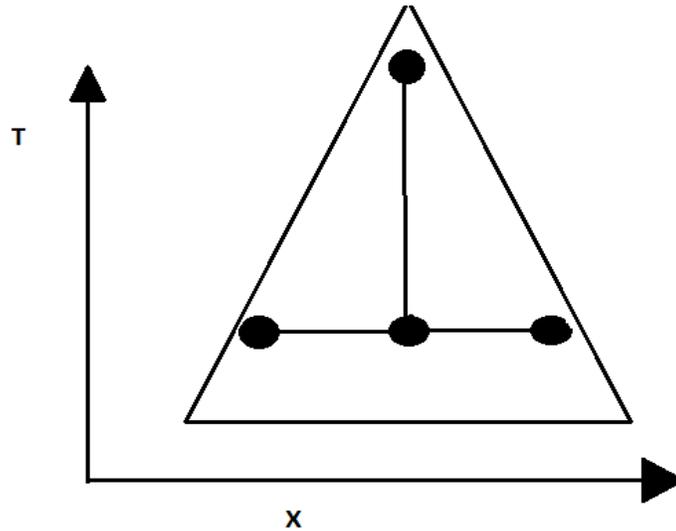
Figure 6.6: Unstabile CFL condition

6.5   Changes to the model

Several changes has been made to the AUSMV model to make it temperature dependant. Some of the changes include the new density models, which was described earlier in chapter 5.2 Closure laws. The other changes are made to how the model solves the primitive variables using the conservative variable. The changes made to the model is based on the in house paper "extension to AUSMV scheme" and be found in Appendix C.

Primitive variables

When temperature was set as a variable in the liquid and gas density, it affected the solution for the primitive variables. The conservative variables for liquid and gas can be expressed as at set of equations. For the complete vect or form including fluxes and source term see equation 5.7 and 5.6 in chapter 5.

Equation 6.13 and 6.14 is a representation of the conservative variable for liquid and gas respec-

tively.

$$w_1 = \rho_l \alpha_l \tag{6.13}$$

$$w_2 = \rho_g \alpha_g \tag{6.14}$$

There are now two different equations that needs to be combined. To solve this one of the closure laws described in chapter 5 is used. The equation for phase volume fractions 5.17 relates the two equations.

$$w_1 = \rho_l \alpha_l = \rho_l(1 - \alpha_g) = \rho_l(1 - \frac{w_2}{\alpha_g}) \tag{6.15}$$

$$w_1 = \rho_l(1 - \frac{w_2 RT}{p}) \tag{6.16}$$

Equation 6.16 is then multiplied with the variable for pressure (p) to obtain the pw_1 expression on the left side of equation 6.17.

$$pw_1 = \rho_l(p - w_2 RT) \tag{6.17}$$

In this new equation there is variable for liquid density that need to be expressed in another form. The new equation for liquid density that was presented earlier(5.10) is used, and combined with equation 6.17. This leads to equation 6.18

$$pw_1 = (\rho_0 - \frac{p_0 \rho_0}{\beta} - \rho_0 \alpha(T - T_0) + \frac{\rho_0 p}{\beta})(p - w_2 RT) \tag{6.18}$$

To make equation 6.18 easier to handle the equation is shortened to equation 6.19, where x_1, x_2 and x_3 replaces some of the variables. $x_1 = \rho_0 - \frac{p_0 \rho_0}{\beta} - \rho_0 \alpha(T - T_0)$, $x_2 = \frac{\rho_0}{\beta}$) and $x_3 = -w_2 RT$.

$$pw_1 = (x_1 + x_2 p)(p + x_3) \tag{6.19}$$

If equation 6.19 is expanded to second order polynomial, the quadric equation can be used.

$$x_2 p^2 + (x_1 + x_2 x_3 + w_1)p + x_1 x_3 = 0 \tag{6.20}$$

A new substitution is done to equation 6.20 and it is reduced to equation 6.21. Here $a = x_2, b = x_1 + x_2 x_3$ and $c = x_1 x_3$

$$ap^2 + bp + c = 0 \tag{6.21}$$

Now equation 6.21 can be solved for p with the quadratic question. This gives the final equation that is needed to solve the pressure term.

$$p = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{6.22}$$

Equation 6.22 is the quadric equation also named the quadric formula(20). These equations can be found implemented into the AUSMV matlab scheme in the appendix.

Viscosity

Viscosity is a measure of the internal resistance to movement for a fluid. Viscosity is a property that is affected by change in temperature. For the original AUSMV scheme the viscosity for liquid and gas were set at a constant value. To improve the AUSMV scheme, a set of equations representing the viscosity were added. These equations are based on the drilling fluids being water and air. These equations were needed to be temperature dependant. Equation 6.23 is the equation used to calculate the liquid viscosity. μ_{w0} represents the reference viscosity for water and T the temperature.

$$\mu_w = \mu_{w0} * 10^{(247,8/(T-140))} \tag{6.23}$$

For the gas viscosity equation 6.24 is used. This equation is often called Sutherland law and describes the relationship between viscosity and temperature for a ideal gas. μ_{w0} represents the reference viscosity, T the temperature, T_{w0} the reference temperature and C the Sutherland's constant for the specific gas.

$$\mu = \mu_0 (\frac{T}{T_0})^{3/2} * \frac{(T_0 + C)}{(T + C)} \tag{6.24}$$

Chapter 7

Simulation and results

In this chapter two different cases will be simulated. The first case is an under balanced drilling (UBD) scenario with three different configuration.

- AUSMV Original model

- AUSMV with temperature.

- AUSMV with temperature and viscosity

The main goal of this simulation is to modify the AUSMV scheme to be dependant on temperature. After temperature is implemented the viscosity will be modified from a fixed value, to a function which take into account the temperature effect. The three scenarios uses the same well configuration, drilling fluids and mass injection rates. The first simulation is used as a base case on which to compare the result from the other two modified simulations. The simulation is of a UBD scenario, where gas is injected into the well in order to reduce the pressure exerted from the hydrostatic column. This is a highly dynamic scenario for two phase flow which represent a good test case for the extended model.

The second simulation case is a mudcap drilling scenario, where the temperature dependant AUSMV scheme, has been modified to simulate a pump system which withdraws fluid from the system. The objective of the simulation is to adjust the height of the mud column using a suction pump system in the middle of a well. All of the simulation will be performed in Matlab,

which excel at calculating complex models and handling input functions. All the codes used to simulate the two scenarios can found in Appendix A and B.

7.1 Underbalanced drilling scenario

This case is comprised of a underbalanced drilling scenario with three different configuration which will be simulated. The simulation setting will be explained and the result analysed and discussed in the result section. Each scenario will be present individually, but the graphs depicting the result will also include the results from the previous case as a comparison. The well configuration and fluid injection rates will be identical for the three scenarios to make the results for the different simulations easier to interpret. The main changes made to the model are the temperature function, the density and viscosity functions for gas and liquid, and how the model calculates the primitive variables of pressure and density.

7.1.1 AUSMV Original model

The first scenario will be the base case for the simulation. The results of the first simulation will be used as a reference point for the other two scenarios and presented alongside. The temperature and viscosity for this model is set at constant values.

Well and fluid configuration

Specifications related to the well geometry, fluid properties and mass rates are stated below.

- **Well depth**

The well is set to be of vertical configuration with a depth of 2000 meter.

- **Wellbore diameter**

The drill pipe outer diameter is set to 0,127m (5 inches) and the well bore inner diameter is 0.2159 (8.5 inches). For simplicity it is assumed that this area configuration is applied for the whole well. This is a simplification made for the simulation and is not realistic with regards to actual configuration. There will always be a difference in diameter due to the tools on a bottom

hole assembly (BHA), drill bit type and size, connections, blast joints, etc. The AUSMV scheme is equipped to handle area discontinuties, but this is not a focus point for the simulation.

- **Fluid selection and properties**

For this simulation water is used as drilling fluid and air used as the displacement medium. Table 7.1 below contains the specification for the fluids in the simulation.

Table 7.1: Fluid properties for water and air

Fluid properties	Density(*)	T(**)	Viscosity(***)	Sound velocity(**)
Water	1000 kg/m^3	20^oC	0,001 (Pa s)	1500 m/s
Air	1 kg/m^3	20^oC	0.0000176 (Pa s)	316 m/s

* The reference density at 20^oC. The density will change according to the density model.
** The temperature for the original scenario. This temperature and the sound velocity is only used in the base case scenario.
*** The viscosity used in the original base case and the second simulation when temperature is implemented. In the third simulation the viscosity will have its own model.

- **Fluid massrates and time interval**

During the simulation different fluid rates will pass trough the system. Table 7.2 shows a overview of the flow rates for gas and liquid. The flow rates are injected at different times during the simulation.

Table 7.2: Mass flow rates

Time (s)	liquid mass rates (Kg/s)	Gas mass rates (Kg/s)
0 - 150	0 Kg/s	0 Kg/s
150 - 160	22*(time-150)/10 Kg/s	2.0*(time-150)/10Kg/s
160-1700	22Kg/s	2Kg/s
1700-1710	22-22*(time-1700)/10Kg/s	2.0-2.0*(time-1700)/10Kg/s
1710 -2000	0 Kg/s	0 Kg/s
2000-2010	22*(time-2000)/10Kg/s	2.0*(time-2000)/10Kg/s
2000 - end	22Kg/s	2.0Kg/s

The same configuration for mass rates are used in all of the three UBD simulations.

Simulation scenario

The simulation is of a UBD scenario, where the well is being displaced with gas to reduce the BHP. When substituting the drilling fluid with a lighter gas component, the pressure created from the hydrostatic column will be reduced. The well will be displaced with gas, and when the pressure has stabilized a connection will be performed. Before the connection, the injection of fluids will be stopped. After the connection is complete the injection of fluids will resume. Below a more detailed layout of the fluid injection rates and time steps is added. For the full table see figure 7.2.

In the first 150 seconds the system is in static condition. Zero fluids are added to the system.

At 150-160 gas and liquid are added to the system, more increasingly with every timestep.

At 160-1700 the system are being filled at a constant rate of 22 Kg/s liquid and 2 Kg/s gas.

At 1700-1710 the fluid injection is slowly stopped over ten timesteps.

At 1700-2000 a connection is being performed. During this operation there is zero injection.

At 2000-2010 the injection of fluids are starting up again, at the same slow rate as before.

At 2000 and until the end of the simulation the mass injection rates are constant.

Results

The results presented here are from the AUSMV original scheme. Figure 7.1 shows the BHP plotted against time. The pressure changed during different operations and injection rates. In the first 150 seconds the model was relatively stable and there was zero fluid injections. The pressure at this point was about 196 bars which represented the hydrostatic column of drilling fluid in static conditions. Between 150 and 160 seconds a spike could be seen in the pressure. This was when gas and liquid was slowly and increasingly injected into the system. When the system started to circulate there was an increase in friction which caused a initial spike in the pressure readings.

From time 160 to 1700 seconds there was a steady inflow of gas and liquid. The pressure can be observed to decrease at a steady pace before it stabilized at 125 bars. This was because of the gas being pumped into the well. With a high amount lighter gas in the hydrostatic column the pressure has decreased.

Next a connection was being performed. In anticipation of this operation the injection of fluids was shut down over a period of 10 seconds. When the pumps was shut down and the connection was being performed, a decline in BHP was observed. This was causedt by the decrease in well friction when circulation stopped. The pressure drop during the connection was at about 23 bars.

After the connection, the pumps were started up again and injection was resumed. The same flow rates was injected as before the connection. The system pressure increased with a peak at 2600 seconds before going towards steady state. The peak can be explained with how the gas and liquid separated during the connection. When the pumps were shut down and the fluids was mostly static, there have been a migration of fluids. The lighter fluid have migrated to the top while the heavier fluid have remained at the bottom. When the pumps were shut on again, the lighter fluids at the top(air) have then been circulated out first before the liquid. This have eventually lead to the peak, where most of the old gas had been circulated out of the well. After the peak the pressure goes towards steady state around 125 bars.
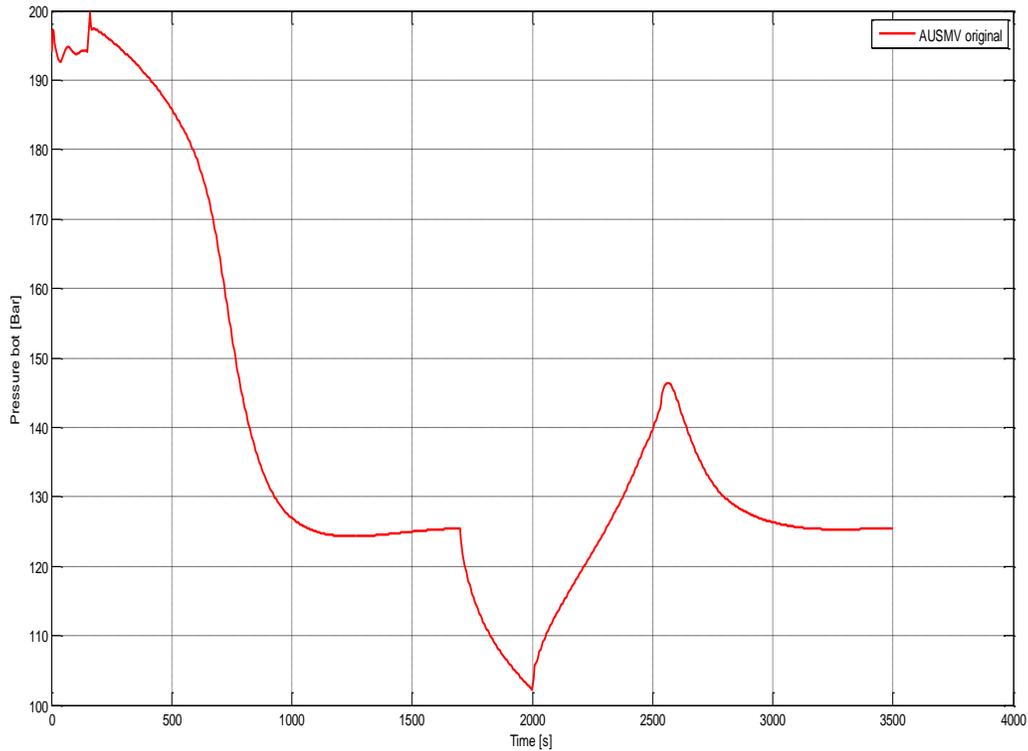
Figure 7.1: Bottom hole pressure measurement vs time for the orginal AUSMV scheme

In figure 7.2 the liquid density was plotted against the well depth. The figure was taken when the system had reached steady state. The liquid density was highest at the bottom and gradually moved toward initial condition as it flowed upward in the well. The reduction in density can be contributed to the change in pressure. At the top the liquid density has reached its initial state.
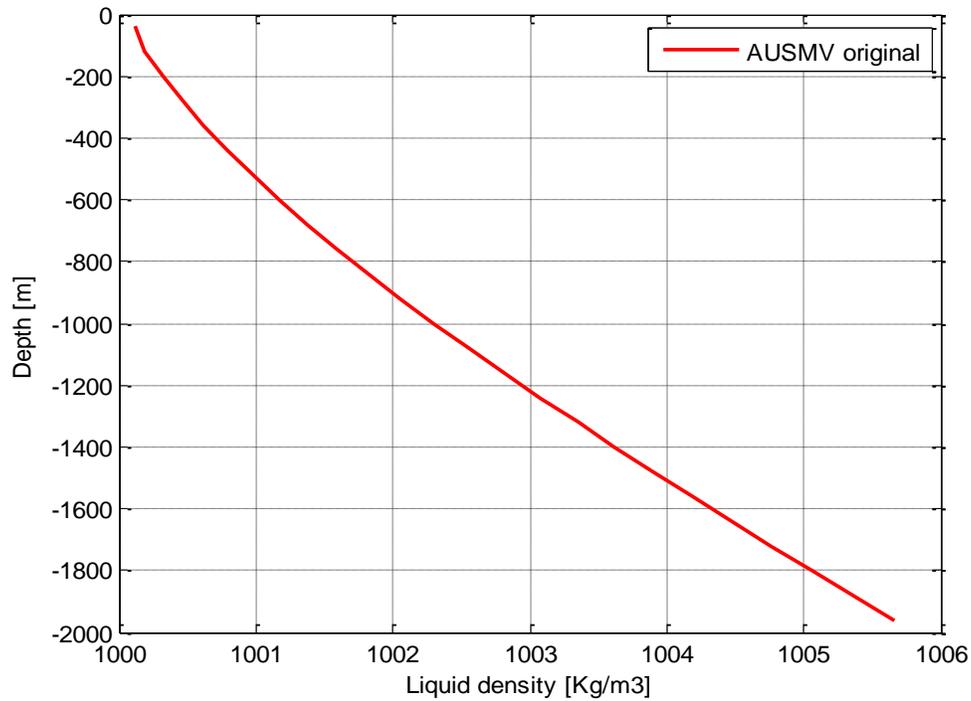
Figure 7.2: Liquid density vs depth for the original AUSMV

In figure 7.3 the density of the gas was plotted against the well depth. This results was taken when the well had reached steady state at the end of the simulation. The density of the gas was observed to be much higher at the bottom of the well and can be contributed to the high pressure. As the gas moved upwards, the density was reduced because of the reduction in pressure.
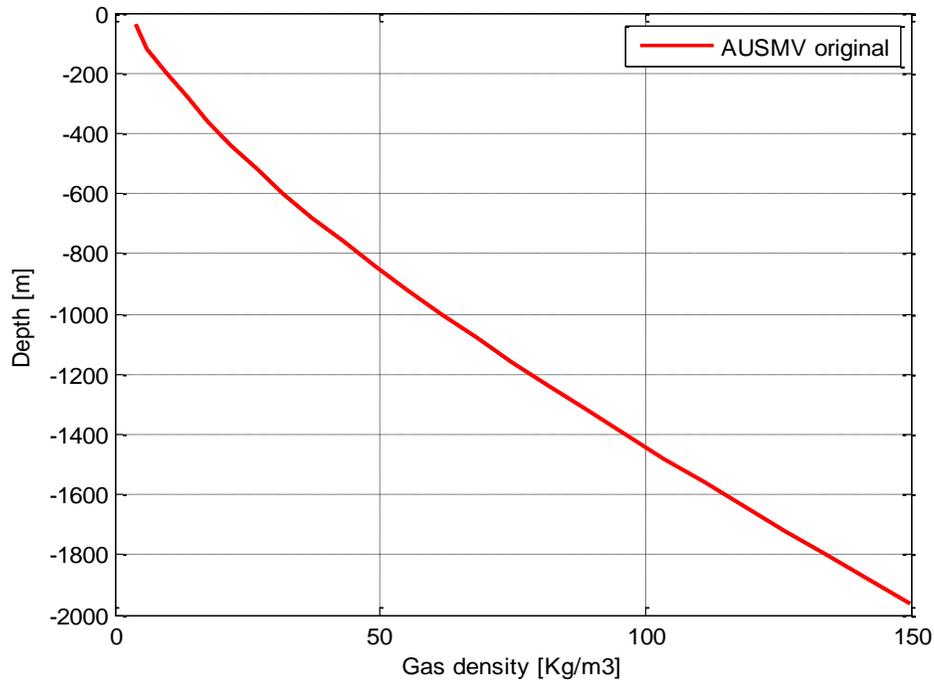
Figure 7.3: Gas density vs depth for the original AUSMV

7.1.2 AUSMV with temperature

In this simulation a temperature dependant version of the AUSMV scheme is tested. The resulting will be presented together with the result from the previous simulation of the orignal AUSMV scheme.

To achieve this modification, several functions and sections had to be remade and edited. All of the changes made to this code is based on the in house paper "extension of the AUSMV scheme" and can be found in the appendix C. The matlab code used in the simulation can also be found in appendix A. The changes is highlighted with red color.

The results from this simulation will be plotted together with the result from the original AUSMV scheme

Well and fluid configuration

The same well configurations are used as in the previous scenario. The drilling fluid is water and air is used as displacement medium. Fluid properties and mass rates can be found in section 7.1.1.

- **Temperature**

The main change to this scenario is that the temperature have been changed from a constant to a variable. The temperature now starts at 100^oC at the bottom of the well and is reduced to 20^oC when the well fluids reaches the top. The temperature decline is divided equally over the discretized boxes in the simulation.

- **Density models**

Since the temperature is now added as a variable that directly affected the density, new models was needed. See equation 5.10 and 5.16 for the new liquid and gas density models. Table 7.3 is an overview of the reference values used for the liquid density model.

Table 7.3: Reference values used in liquid density model

$\rho_0 (Kg/m^3)$	$p_0 Pa$	$T_0(^oC)$	β (Pa)	$\alpha(K^-1)$
1000	100000	20	$2.2 * 10^9$	0.000207

For the gas density model the specific gas constant for air is used and is equal to 286.9 J/kg.K

- **Viscosity**

The viscosity for liquid and gas is the same as in the previous section.

Simulation scenario

The same simulation scenario as used in chapther 7.1.1.

Results

Figure 7.4 shows a graph of the BHP plotted against time. The red line represent the pressure from the original AUSMV model, while the blue represent the pressure from the new temperature dependant model. Because the different simulations have the same injection rates, the graph structure will be very similar. When comparing the two plots a clear difference can be observed in the pressure. This reduction in pressure can be contributed to the new density models and the implementation of temperature as a variable. When the new simulation reaches steady state the pressure is recorded at 117 bars. That is equal to a pressure drop of 8 bars from the original simulation.
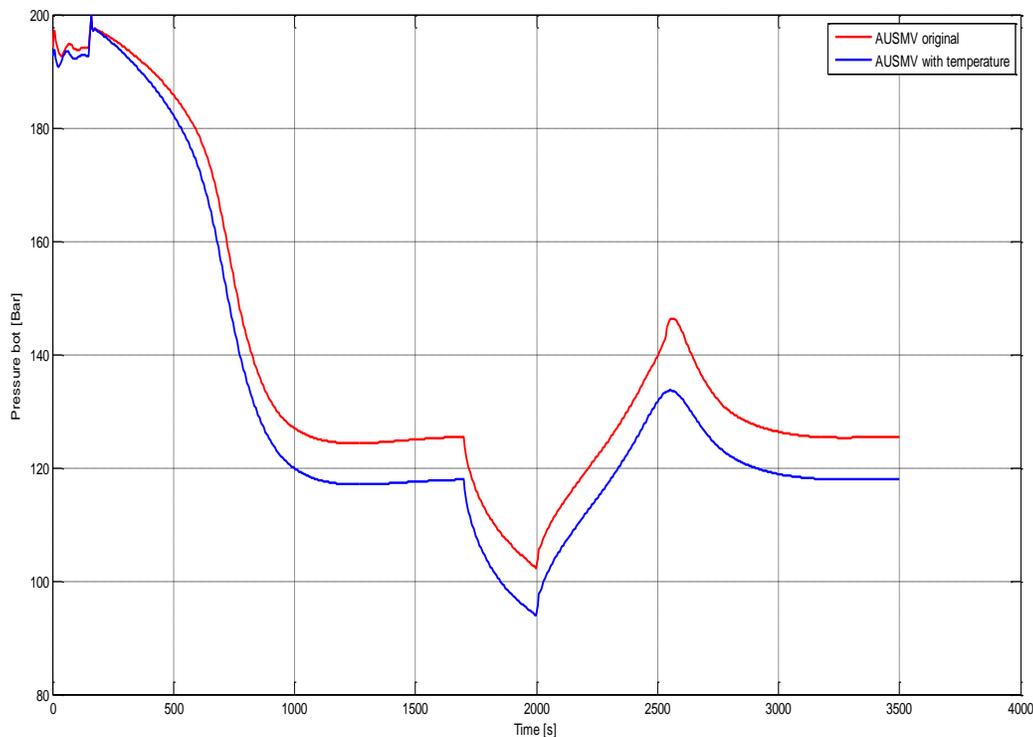
Figure 7.4: Pressure vs time for the original and temperature dependent AUSMV scheme

Figure 7.5 shows a graph of the liquid density plotted against depth. The information is taken when the simulation has reached steady state at 3500 seconds. A significant change in the liquid

density can be observed. The massive drop in density can be explained by the temperature dependency of the new model. For the red curve the temperature is set to a fixed value of 20^oC, while the blue line has a temperature variation of 100^oC at the bottom and 20^oC at the top. The liquid density for the temperature dependent model gradually increases and eventually reaches normal conditions as the pressure and temperature in the well decreases.
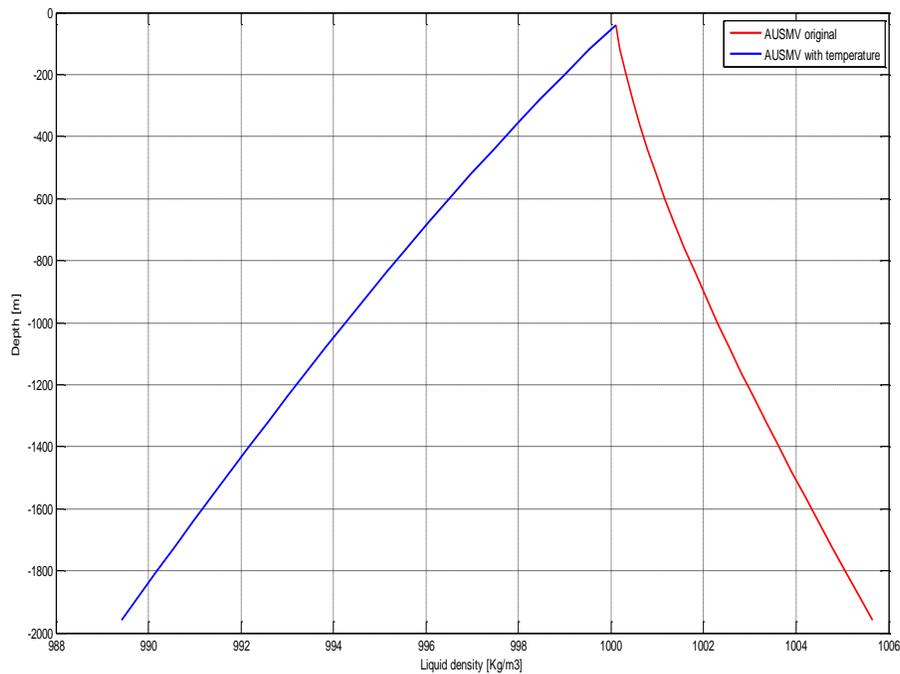
Figure 7.5: Liquid density vs depth for the original and temperature dependent AUSMV scheme

Figure 7.6 shows the gas density for the original and temperature dependant AUSMV scheme at steady state conditions. A clear difference in density at the deeper sections of the well can be observed. The blue line represent the new AUSMV scheme while the red is the original. The addition of temperature to the gas density calculation has caused the density to decrease. The original model have a perm ant temperature at 20^oC the bottom, while the modified code have 100^oC. This is the reason for the big difference in density. With increasing temperature the gas

will become lighter and the density will fall. As the gas ascends in the well, the density falls. This is because of the temperature and pressure reduction at the higher levels of the well.
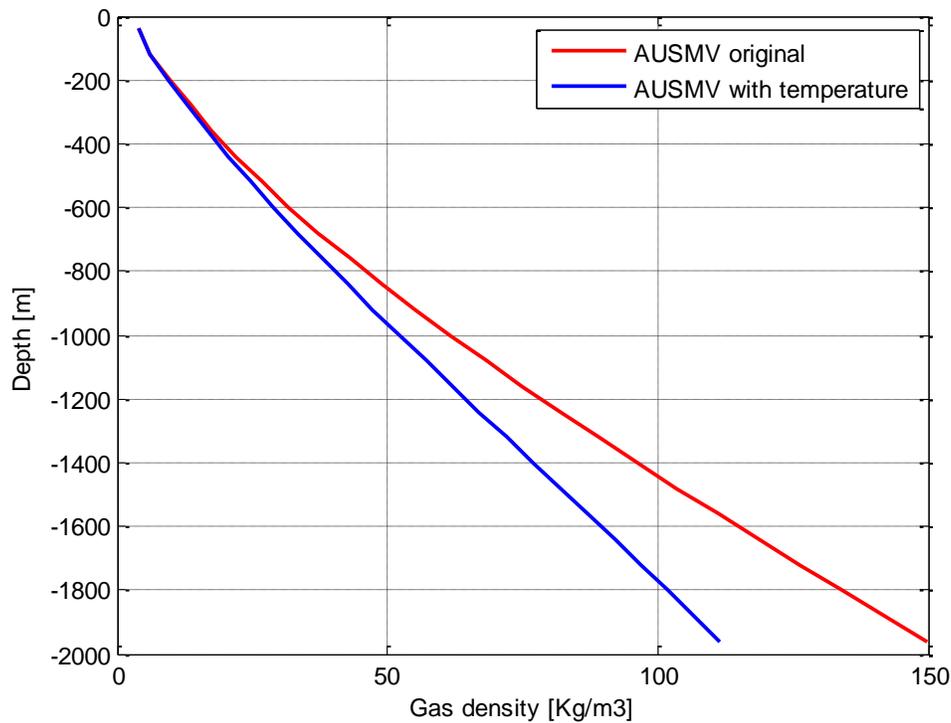
Figure 7.6: Gas density vs depth for the original and temperature dependent AUSMV scheme

7.1.3 AUSMV with temperature and viscosity

For the third and finale scenario of the UBD simulation, the implementation of viscosity as a variable is the goal. In the previous simulations the viscosity has only been defined as a constant value and has not been affected by temperature change. To implement viscosity as a variable two new function had to be added to scheme. One for the liquid and one for the gas viscosity. The viscosity equations used can be found in chapter 6.5. Viscosity is a measure of the fluid internal resistance to flow and is directly affected by temperature (21).

Well and fluid configuration

This simulation has the same well and fluid configuration as the previous scenario. The only change made are to the viscosity functions for liquid and gas. The details can be found in chapter 5.1.2 AUSMV with temperature. Table 7.4 show a list of the properties used to calculate the viscosity

Table 7.4: Properties used for the viscosity model

Fluid	μ_0 (Pa.s)	T_0 (K)	C (K)
Air	$1.827 * 10^{-5}$	291.15	120
Water	$2.414 * 10^{-3}$	-	-

Simulation scenario

The same scenario as in chapter 5.1.1 AUSMV Original.

Simulation

Figure 7.7 shows the results for all three scenarios with the new viscosity plot in yellow. The inclusion of viscosity in the simulation has had a minor effect on the pressure profile of the plots.
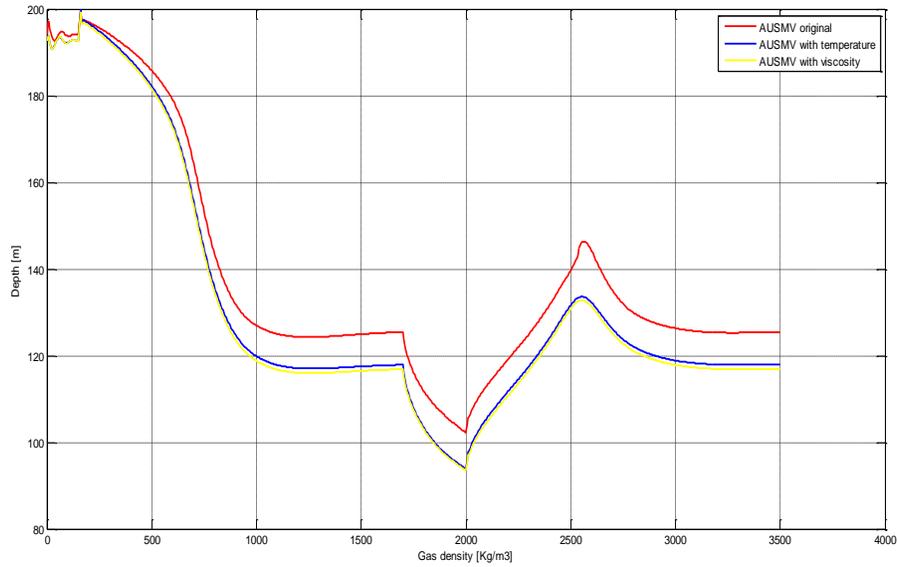
Figure 7.7: Bottom hole pressure measurement vs time for the three AUSMV configurations at steady state

Figures 7.8 and 7.9 shows the density graphs for the liquid and gas. The liquid and gas densities has been very little affected by the change in viscosity.
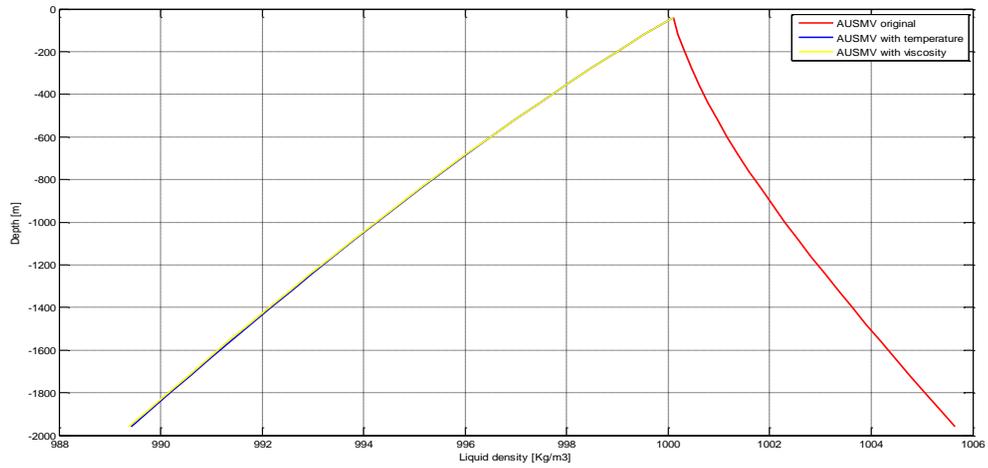
Figure 7.8: Liquid density plotted against depth for the three AUSMV configurations at steady state
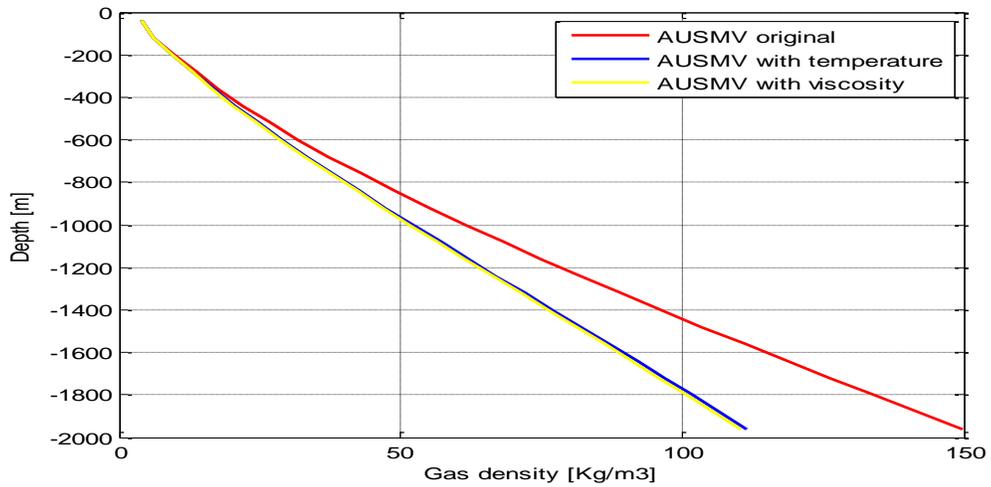
Figure 7.9: Gas density plottet against depth for the three AUSMV configurations at steady state

Figures 7.10 and 7.11 shows the liquid and gas viscosity. The viscosities are plotted against depth. The liquid viscosity is shown to increases when the temperature decreases. This effect is the opposite for the gas. When the temperature in the well decreases, the viscosity for gas also decreases.
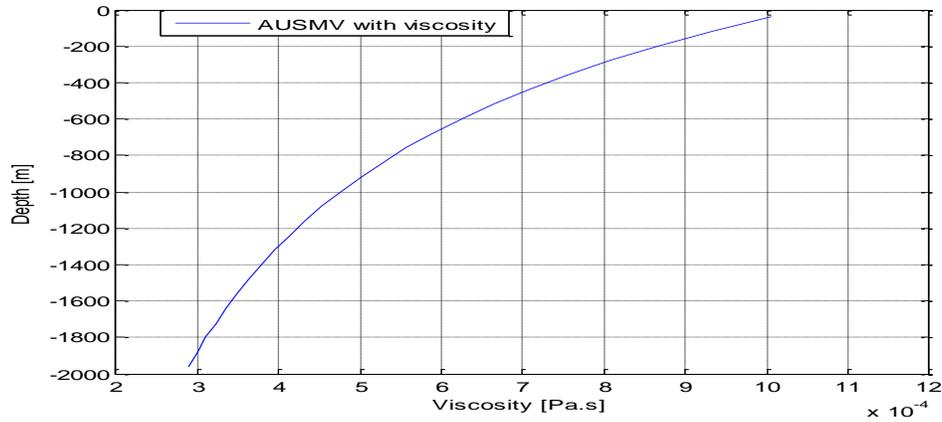
Figure 7.10: Liquid viscosity plotted vs depth at steady state condition
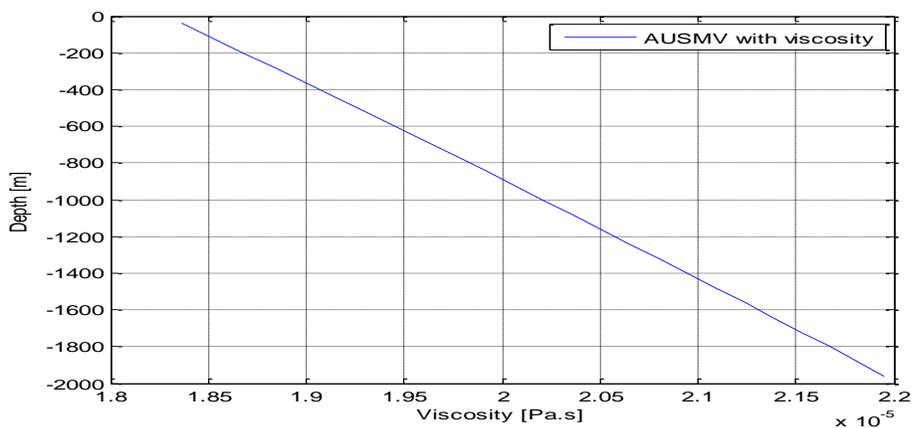
Figure 7.11: Gas viscosity plotted vs depth at steady state condition

7.2 Mudcap drilling case

For this simulation a mudcap drilling scenario is tested. The objective of the simulation is to adjust the mud level in the well, using a "pump" to reduce the height of the hydrostatic column. The mud will be sucked out via a pump at 1000 meter depth, which is in the middle of the well. This achieved with the implemented of a source term in the mass conservation equation in the cell where the pump was installed. The primary goal of this simulation is to analyse the pressure profile of the well during various inlet and outlet pump rates. The phase velocity at different time steps will be analysed and described. Also a look into how the model interprets sharp changes at the liquid/gas interface are performed. The new temperature modified AUSMV scheme from the previous chapter are used as the basis for the simulation.

7.2.1 Well and fluid configuration

Some specifications for this scenario has to be made. The well depth, fluid configuration, mass rates, simulation run time.

- **Well configuration**

The well is assumed to be assumed to be 2000m deep and of vertical configuration. At a depth of 1000m a suction point will be set and will act as a pump. At this suction point mass rates will be sucked out of the well. This is purely a theoretical scenario and does not reflect any real well configuration. This setup is constructed to achieve easy observable changes in the mud level.

- **Wellbore diamater**

The drill pipe inner diameter is set to 0,127m (5 inches) and the well bore outer diameter is 0.2159 (8.5 inches). For simplicity it is assumed that this area configuration is applied for the whole well. This is a simplification made for the simulation and is not realistic with regards to actual configuration. There will always be a difference in diameter due to the tools on a bottom hole assembly (BHA), drill bit type and size, connections, blast joints, etc.

- **Fluid massrates**

The inflow for the simulation is specified in table 7.5. The inflow and outflow reflect the simulation scenario.

Table 7.5: Injected mass flow rates

Time (s)	liquid mass rates (Kg/s)
0-150	$0\ Kg/s$
15-160	22*(time-150)/10 Kg/s
160-300	$22Kg/s$
300-310	22-22*(time-1700)/10Kg/s
310-800	$0\ Kg/s$
800-810	22*(time-2000)/10Kg/s
810-100	$22Kg/s$

Time 0 to 150 is the initialization of the model. There are no liquid injected into the system and conditions are considered static.

From 150-160 the flow is slowly commencing over a time period of 10 seconds.

From 160-300 there are a steady inflow of drilling fluids.

From 300-310 the flow are slowly shut off.

From 310-800 there are zero injected flow, but at 400 seconds a pump start to eject fluid out of the system and mud level decreases.

From 800-810 the injection of fluid once again commence over a period of 10 seconds

From timestep 800-1000 there are fluid injection and removal at the same time.

Table 7.6 is an overview of the mass rates that are pumped out of the system. These pump rates are pumped out at a depth of 1000 m during the simulation

Table 7.6: Pump suction rates

Time (s)	liquid mass rates (Kg/s)
400-410	22*(time-400)/10 Kg/s
410-end	22 Kg/s

7.2.2 Boundary condition

Some adjustment to the boundary condition regarding the liquid fluxes was needed. Since the liquid fluxes is extrapolated from the previous interior cell a refill scenario happened. In order to fix this, the mass and momentum outlet fluxes in the last cell was set to zero. The gas mass and momentum fluxes was still extrapolated. In addition the pressure was extrapolated.

After the pump suction was implemented a new problem occurred. When the pump was implemented the boxes in the upper part of the well, started to move towards vacuum conditions and the simulation failed. To avoid this scenario a fixed pressure value was implemented if the simulation started moving towards vacuum. The pressure was set to never go below one Bar.

7.2.3 Numerical diffusion

First order numerical schemes sometimes encounter a problem referred to as numerical diffusion. In this simulation a closer look on the numerical diffusion associated with a rapid change between liquid and gas volume fractions. Numerical diffusion occurs when sharp gradients encounter fast change. One solution to reduce or stop this effect can be to increase the number of boxes in the discretization of the simulation. With an increased of boxes the numerical diffusion can be reduced, but it comes with a cost of computation power and time. For complex simulations with heavy calculations this can be a very costly trade (22).

7.2.4 Results

The results are divided into three sections. The bottom hole pressure, the fluid velocities at different timesteps and the volume fractions with grid adjustments.

Pressure profile

Figure 7.12 shows pressure recording for the bottom of the well over a simulation period of 1000 seconds. The first 150 seconds of graph was the initialization period of the model. The pressure here represents a static hydrostatic column of fluid at 2000m depth. At time 150 an increase in BHP occurred. This was when fluid is being injected and circulation has begun. This rise in

pressure was a result of the frictional increase related to the circulation of fluid. The pressure remains stable when circulation at about 195 bars.
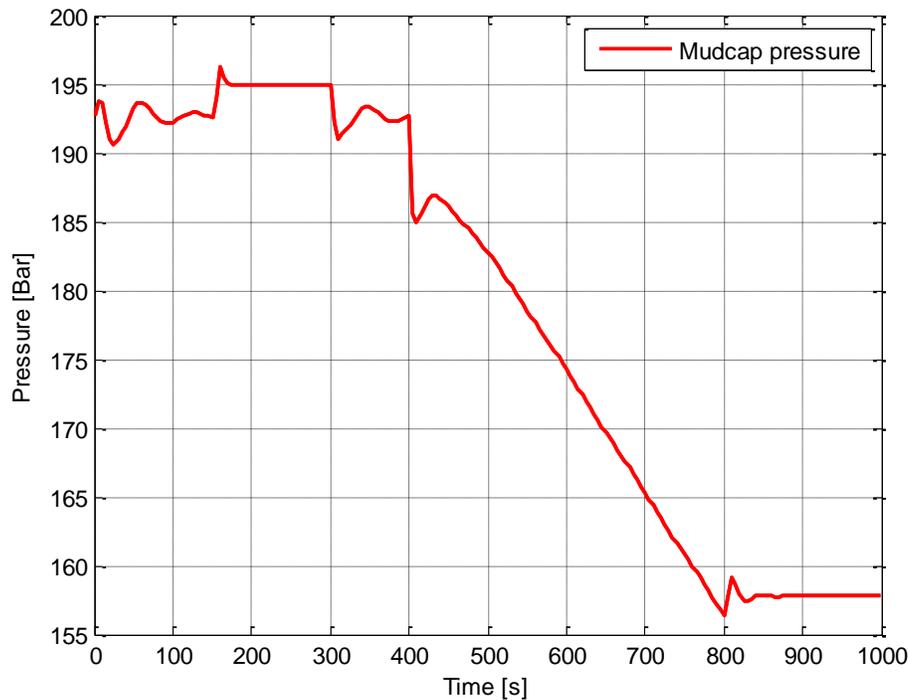
Figure 7.12: Bottom hole pressure measurement plotted against time for the mudcap case.

At 300 seconds there was a pressure drop. At this point the injection of fluid was stopped and circulation ended. The pressure drop is due to the reduction in frictional forces when the well moves towards static conditions.

At 400 seconds a significant pressure drop was recorded. This is when the suction pump start to pump fluids out of the system and the system goes from static to dynamic. The initial pressure drop can be seen to be significantly higher then the previous. This was due to the direction the

of the fluids. Since the pump was located at 1000 meter depth(middle of the simulation) and pumping fluid out of the system, the fluid above the pump was moving downwards. That meant the fluid was moving against the positive direction of the simulation and as a effect caused "negative friction". This negative friction was the cause of the significant initial pressure drop. After the initial drop the simulation stabilizes and the loss of pressure corresponds with the eviction of fluids from the system.

At 800 the graph showed a minor increase in pressure before remaining constant for the rest of the simulation. This was when the injection started up again and the friction started moving in the positive direction. At this point the fluid below the suction pump was starting to circulate at exactly the same rate to that which was being removed by the suction pump. This corresponded well to the mass rates involved in the system. At this time an equal amount of fluids was being removed and injected into the system. The pressure remains stable for the rest of the simulation.

Fluid velocity

The velocity profile at different time intervals describes how the fluids travel in the system. Figure 7.13 shows a graph of the velocity of the drilling fluid at time=250. The system is in circulation and a constant rate of 22Kg/s of fluid is injected. The graph shows that the fluid moved at 1 m/s upwards in the well.
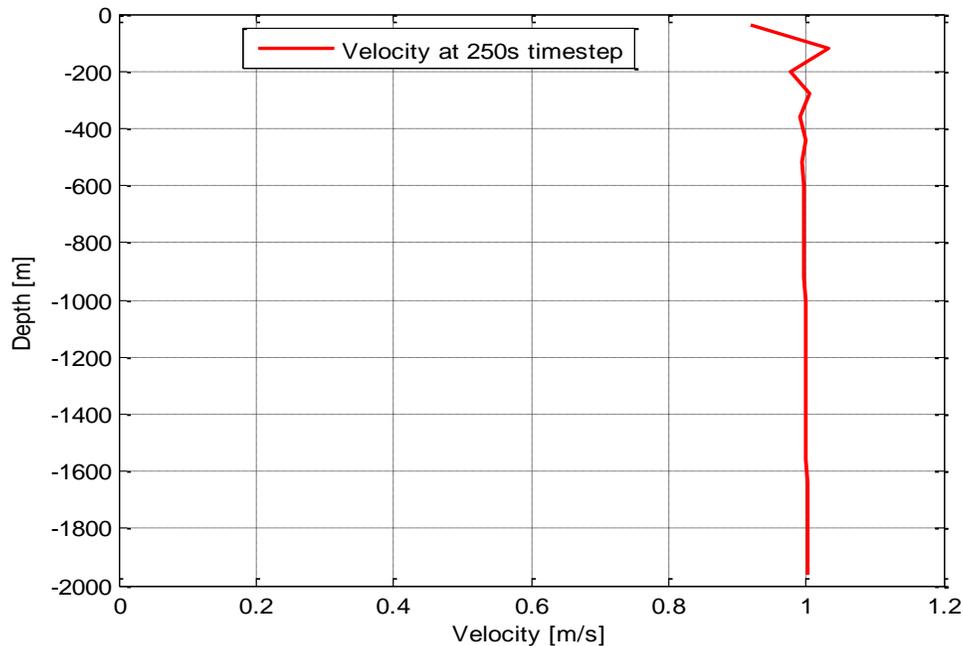
Figure 7.13: Fluid velocity vs depth at timestep 250

Figure 7.14 shows the velocity profile at time = 600. At this time interval drilling fluid was no longer injected into the system, but removed by the suction pump. The fluid at the bottom of the well and up to the suction pump is at almost static conditions with zero to little velocity. Above the suction pump a sharp decrease in velocity is observed. This is because of the direction the fluid was moving. Since the system has movement upwards as its positive direction and the fluid above the suction pump was moving downwards the velocity was recorded as negative. The velocity changed from 0 to -1 m/ at over the suction point. This happened over a period of about 200m and reflects the number of boxes the simulation is discretized with. If the well had been divided into more boxes the graph would appear sharper. At about 300 meter depth the velocity start to increase even more. This the boundary between liquid and gas. Because drilling fluid has been removed from the system the upper section consist of gas. The fluid level

has dropped with about 300 meter since the suction pump started.
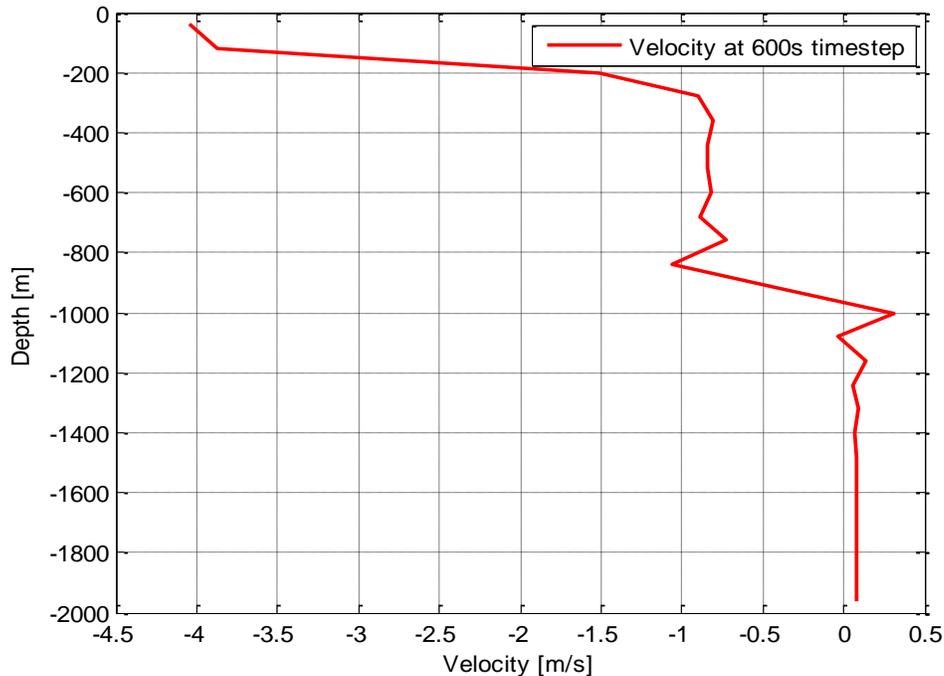
Figure 7.14: Fluid velocity vs depth at timestep 600

Figure 7.15 shows the velocity profile at time step 1000 and is the end of the simulation. At this point injection had started up again and now fluid was injected and removed at the same time. From the bottom of the well and up to the suction point the liquid had a constant speed of 1 m/s. This means that the fluid are flowing from the bottom of the well and up towards the suction point. At the suction point the velocity gradually decreased, before reaching zero. The liquid above the suction point remained static with zero velocity. Since the the same amount of fluid was injected and sucked out, the mud level above the suction pump remained constant. If an increase in injected fluid were to happen the mud level would increase and so would the BHP. The same effect would happen in reverse if the fluid injection rates were to decrease and the suction rates remained constant.
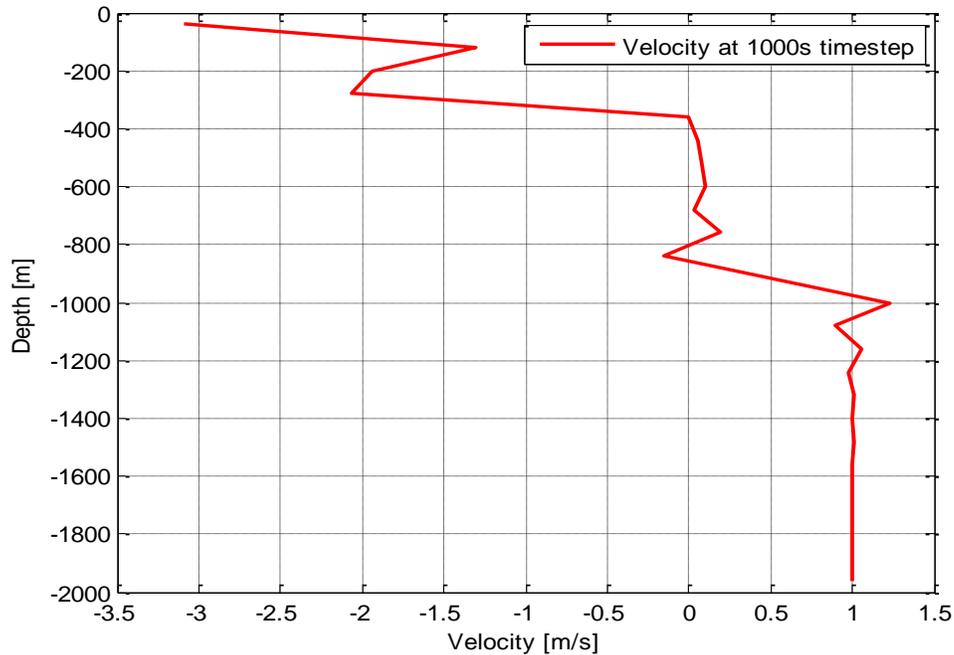
Figure 7.15: Fluid velocity vs depth at time step 1000

Volume fractions with grid adjustment

Figure 7.16 depicts the gas volume fraction throughout the well with a 25 box discretization. The figure shows that from the bottom and up to about 420 meter the gas fractions was equal to zero. This means that there was only liquid in this section and represent the mud column in the well. The 420 meters of gas or air at the top is an effect of the decrease in mud level. The graph also show that the change between liquid and gas occurs over a depth period of almost 150 meters. This is an effect of the discrization configuration for the well. The well is discretized into 25 boxes of equal length and a sudden change in volume fraction is split over multiple boxes. In a optimal configured model this graph should be almost horizontal for a vertical well.
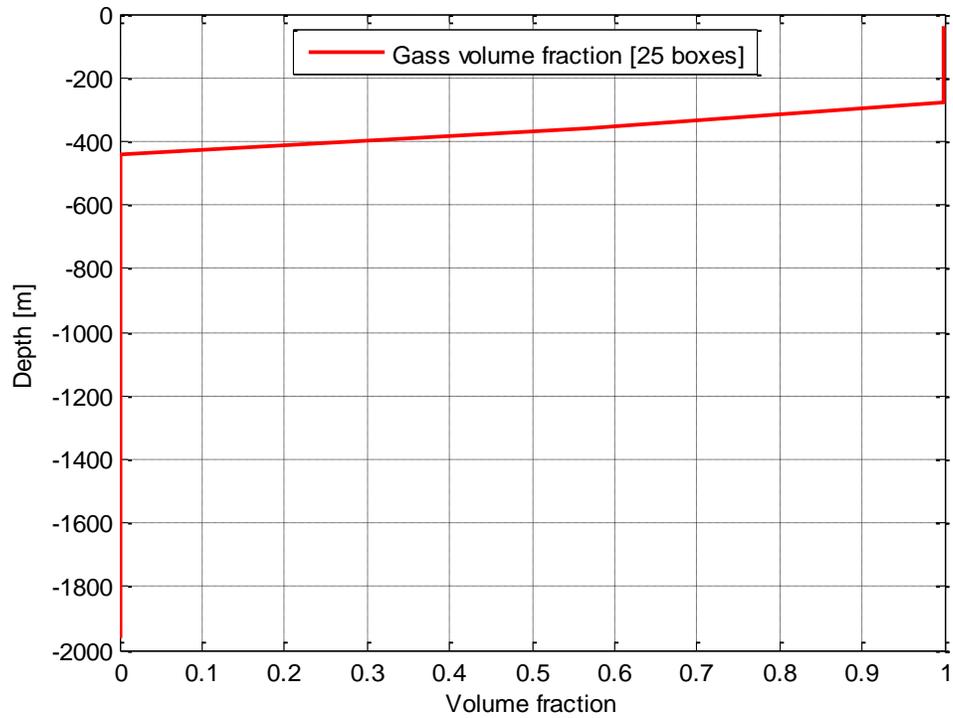
Figure 7.16: Gas and fluid interface with a 25 box discretization

Figure 7.17 shows the volume fraction of gas at time step 1000, but with a discretization of 50 boxes. The change from liquid to gas can be observed to be sharper and occurs over a smaller period of depth.
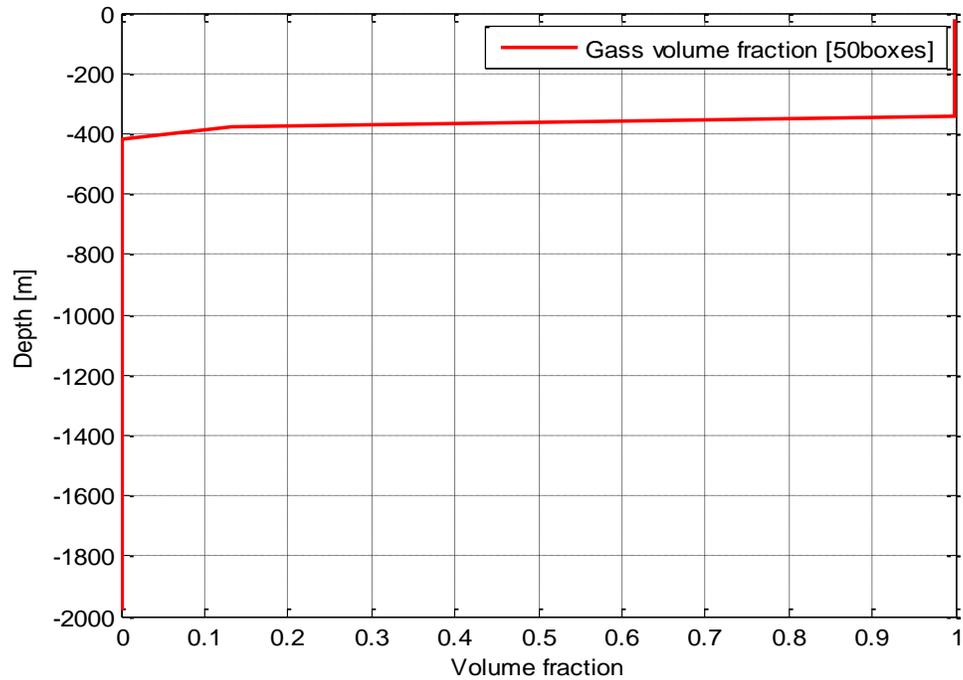
Figure 7.17: Gas and fluid interface with a 50 box discretization

In figure 7.18 a discretization configuration of 100 boxes was attempted. This resulted in an even sharper change, but required significant more time to perform the simulation.
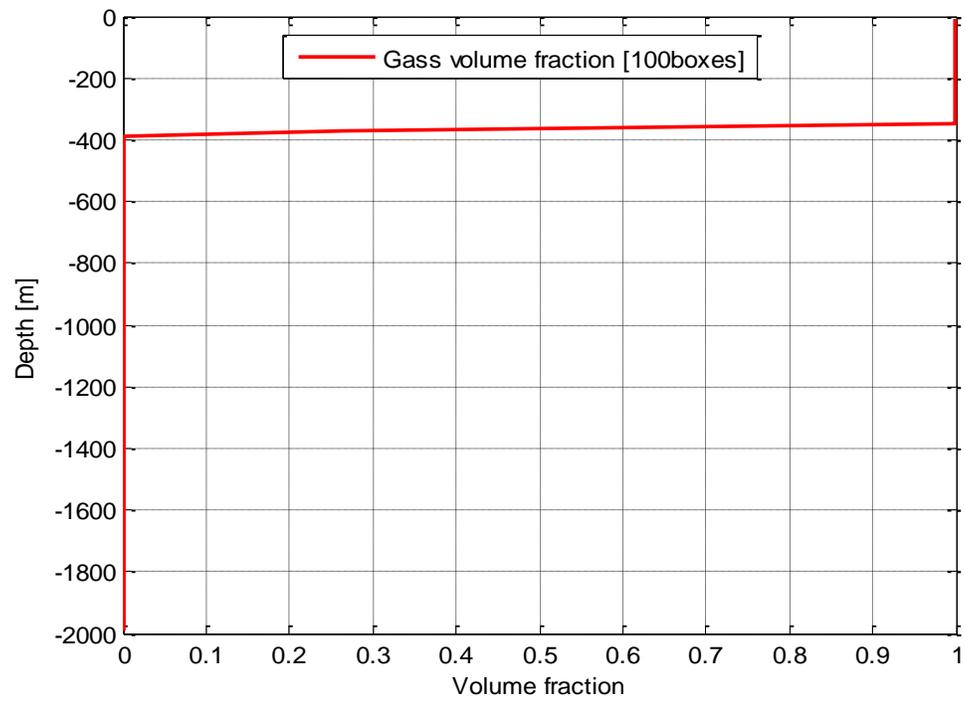
Figure 7.18: Gas and fluid interface with a 100 box discretization

Chapter 8

Discussion

The discussion of the results are split into two sections. One for each simulation case. The main objective of this thesis was to make the AUSMV scheme dependant on temperature. The second objective was to create a modeling structure with the AUSMV scheme that could simulate a moving mud column.

8.1   Discussion of UBD case

When analysing the different graphs and results produced during the simulation there are some sections that should be discussed.

- **Temperature effect on the AUSMV scheme**

When comparing the pressure profile for the three cases a clear drop in pressure can be seen when temperature is implemented. The original model and the two new simulation shares the same profile due to similar injection rates of gas and liquid, but a lower pressure is recorded. The pressure drop can be credited to more realistic density model which incorporate temperature as a variable. As the original model used a fixed value of 20^oC for the whole model, the new simulation has a variation of 100^oC and down to 20^oC. For the the pressure graph 7.7 which shows the BHP readings between the cases the difference in temperature is 80^oC. This temperature difference combined with the new density models are the main contributor to the reduction in bottom hole pressure.

- **Liquid density change in the AUSMV scheme**

In 7.8 two obviously different density profiles was shown. This figure illustrated that the density of liquid has been reduced with the introduction of the new model and a variable temperature. The density can be observed to have to dropped due to the increase in temperature. As the liquid travelled upwards in the well, the temperature was reduced and the density moved toward normal conditions. In the simulation water was used as drilling liquid, so a density graph for water can be used as tool to verify this effect. In appendix D, a figure showing the change of water density due temperature and pressure effects are added. The figure shows that the water density is reduced with higher temperature which correlates with the results in the simulation.

- **Changes in gas density in the AUSMV scheme**

Much as the same as in the previous section, the gas density shown in figure 7.9 has also been reduced with the implementation of temperature. The gas density at the bottom of the well has been reduced from 150 kg/m^3 to about 110 kg/m^3. This is because of the new gas density model which is directly affected by temperature.

- **Changes in liquid and gas viscosity in the AUSMV scheme**

In the original model the viscosity was set at a constant value. To get a clear view of the effect viscosity had on the model a separate scenario was done in the simulation, were the viscosity was changed. The viscosity was then changed from the constant value of 0,001(Pa.s) for the liquid and 0,0000176(Pa.s) for the gas. The new viscosity models implemented was directly affected by temperature. In chapter 7.1.3 several of the previous discussed graphs can be seen with viscosity as a variable. The effect of viscosity on the pressure and density can be described to be a minor one. With regards to the BHP plot in figure 7.7, the pressure can be seen to have a minor reduction in simulation with temperature dependant viscosity. In figure 7.10 and 7.11, the viscosity for liquid and gas respectively is plotted against depth. These graphs shows the fact that the viscosity for liquid increase with decreasing temperature, while the opposite effect occured for the gas.

8.2 Discussion of floating mudcap scenario

The results from the floating mudcap scenario is divided into three different parameters. The pressure, fluid velocity and the phase volume fractions. The results for this simulation case is based on the new modified AUSMV scheme, so the effect of temperature and viscosity will is not be discussed here.

- **Pressure profile**

The pressure profile presented in figure 7.12 illustrate the pressure reading for the floating mudcap case. The object of the simulation was to adjust the mud level by implementation a suction system. This graph shows the pressure as a function of the mass rates circulating in the system. At timestep 400 in the simulation, mass rates are removed from the system rather then injected and the pressure decreases as the mud column diminishes. So the mud level has been proved to be controllable with the injection and removal rates. At timestep 800 the model reached a equilibrium as the injection rate was equal to the removal rate of fluid.

- **Fluid velocity**

The fluid velocity graphs presented in figure 7.13, 7.14 and 7.15 illustrates which direction the fluid moved during the simulation. This is a very important observation into how the fluid behave during different times of the simulation. The velocity was plotted against depth, which illustrated the flow from the bottom of the well and to the top. Figure 7.13 showed only positive velocity which means the fluid flowed only upwards, as this is the positive direction. Figure 7.14 illustrated a different movement pattern. Here the liquid at the bottom moved upwards, while the liquid at the top moved downwards. This corresponded well with the setting of the scenario, that a suction pump was removing the liquid at the middle of the well. This is what cause the negative velocity for the upper sections of the well and also causes a bigger pressure drop due to the negative friction. Figure 7.15 illustrates the well in equilibrium, with equal injection and suction rates. The velocity showed that below the pump fluid was flowing upwards and the fluid over the pump was stationary.

- **Grid adjustment for gas volume fraction**

The gas volume fraction for the simulation illustrates the volume fraction change from liquid to gas. When the volume fraction of gas is plotted against depth, it gives a good description of where the top of the mud column lies. Figure 7.16 shows the phase interface between gas and liquid. This does not represent a realistic scenario for a vertical well, because of the gradually change between gas and liquid. In a realistic scenario the curve would be almost horizontal.This effect occurs most likely because of the dizcretization configuration. The simulation was based on a discretization of 25 boxes. This meant that when the model encountered abrupt changes it had a limited amount of boxes to adjust to these changes. Therefore a grid adjustment was tested to observe the effect the volume fraction reading in the model. The grid was adjusted to 50 and 100 boxes and compared against the original 25. Figure 7.17 and 7.18 illustrates the simulation results, with a different number of boxes. A much sharper transition between the liquid and gas interface is observed. In a real situation the interface between the mud and gas will be at a single position, but in the simulation this transition is smeared out. This is due to numerical diffusion discussed in chapter 7.2.3 and this can be remedied by increasing the number of boxes. The was a computational cost with using 100 boxes instead of the previous 25. The increase in boxes lead to a much longer simulation run time. The simulation run time went from 12 minutes to over 2 hours.

- **Challenges in handling the floating mud cap interface**

There were several problem encountered with the implementation of the floating mudcap scenario. The first problem that occurred was a refill scenario. After the suction point was implemented and fluid was removed from the system, the model behaved as if it was being refilled from the top of the well. The problem here was that extrapolation of the mass rates at the outlet from the interior neighbour cell, lead to negative liquid mass rates which effectively filled the well with liquid on top. The refill problem was handled with the adjustment of the boundary conditions at the outlet. The gas fluxes was extrapolated from the interior cell while liquid fluxes was set to zero. This allowed gas to be filled in at the top of the well, but no liquid. This made it possible to reduce the mud level. Pressure at the outlet was determined by extrapolation from the interior cells.

The second problem was that the model crashed when fluids was sucked out of the system and the pressure decreased in cells at the top of the well. The gas over the mud/gas interface

went towards vacuum conditions and crashed the simulation. This was fixed by enforcing a condition that the pressure of the gas could never go below 1 bar.

Chapter 9

Conclusion

The original AUSMV scheme has been modified to incorporate temperature as a variable instead of a fixed value. This new AUSVM scheme have been used to successfully simulate two different well scenarios. The AUSMV scheme has proven to be a very reliable and robust tool for two phase flow simulation. The model has been proven to be adaptable for different multiphase flow scenarios with adjustable flow rates for liquid and gas. Several conclusion can be drawn from the results of the two simulations.

- Making the AUSMV scheme dependent on temperature proved to have a significant effect on the pressure. To make the AUSMV scheme temperature dependent new gas and liquid density models as well as new viscosity functions were added.

- The change in liquid and gas models made a clear effect on the density profile of the model. The density for gas and liquid were both reduced with temperature. A clear difference could be seen at the bottom of the well, where the temperature difference between the old and new model was highest.

- The viscosity was changed from a constant value, to a model which was temperature dependent. This had a minor effect on the pressure and densities for liquid and gas in this case. The pressure decreased with about 1 bar, from 117 to 116 and a very small change in the liquid and gas density was observed.

- When comparing the results for the three different simulations for the UBD case, a clear

change in pressure was observed. The pressure dropped from 125 bar in the original model, to 117 Bar in the model where temperature was included in the density models. When simulating with temperature dependent viscosity another pressure drop was found. The pressure went down to 116 bars which shows that viscosity had a impact on the frictional pressure loss. The largest effect was caused by including temperature in the density models thereby effecting the hydrostatic pressure in the well.

- The AUSMV scheme was successfully modified to simulate a floating mudcap scenario where the mud level was controlled by adjusting the mass injection and suction rates. The mud level changed with different inlet and outlet settings for mass rates.

- By analysing the velocity profile at different times during the simulation it was seen that the numerical scheme was able to handle a reverse of the flow direction above the suction point and thereby reducing the mud level in the well.

- To handle the problem of numerical diffusion with regards to the mud/gas interface, a grid adjustment was performed. In reality the mud/gas interface will be sharp, however in the simulations the interface tends to be smeared out. This is due to numerical diffusion. The adjustment from 25 to 50 and 100 boxes, proved to give a sharper transition zone for the liquid and gas interface. This grid adjustment came with a trade off which increased the simulation run time. The simulation runtime went from 12 minutes to over 2 hours. Which is impractical if multiple simulations are to be run. Other ways of reducing the numerical diffusion should be investigated in further works.

- To get a working simulation for the floating mudcap several fixes was used. To avoid a simulation crash a forced minimum pressure was set for the numerical cells located above the mud/gas interface. The second fix was for the extrapolation of the liquid mass and momentum fluxes at the outlet boundary. The liquid mass and momentum fluxes were forced to zero, while the gas mass and momentum fluxes were still extrapolated from the previous cell. For further studies a look into these temporary fixes should be made and alternative solution proposed.

Bibliography

[1] JJ Schubert. Subsea mudlift drilling: from jip to the classroom. In *American Society for Engineering Education annual meeting held in Albuquerque, NM*, 2001.

[2] Illustration of the u-tube effect. http://www.drillingformulas.com/barrels-of-slug-required-for-desired-length-of-dry-pipe. Accessed: 2015.02.15.

[3] Pore pressure graph for ubd, mpd and conventional drilling. http://www.drillingcontractor.org/wp-content/uploads/2010/02/KenSara-graph.jpg. Accessed: 2015.04.17.

[4] Ali Mohammed Haj. Dual gradient drilling and use of the ausmv scheme for investigating the dynamics of the system. Master's thesis, 2012.

[5] Kenneth L Smith, AD Gault, DE Witt, CE Weddle, et al. Subsea mudlift drilling joint industry project: delivering dual gradient drilling technology to industry. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2001.

[6] Properties of seawater. http://sam.ucsd.edu/sio210/lect2/lecture2.html. Accessed: 2015.03.10.

[7] Jerome J Schubert, Hans C Juvkam-Wold, Jonggeun Choe, et al. Well control procedures for dual gradient drilling as compared to conventional riser drilling. *SPE Drilling & Completion*, 21(04):287–295, 2006.

[8] Robert P Herrmann and John M Shaughnessy. Two methods for achieving a dual gradient in deepwater spe/iadc 67745. 2001.

[9] JP Schumacher, JD Dowell, LR Ribbeck, JC Eggemeyer, et al. Subsea mudlift drilling: Planning and preparation for the first subsea field test of a full-scale dual gradient drilling system at green canyon 136 gulf of mexico. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers, 2001.

[10] Bill Rehm, Jerome Schubert, Arash Haghshenas, Amir Saman Paknejad, and Jim Hughes. *Managed pressure drilling*. Elsevier, 2013.

[11] Jeff Saponja et al. Challenges with jointed pipe underbalanced operations. *SPE drilling & completion*, 13(02):121–128, 1998.

[12] DO Nessa, MJ Tangedahl, and J Saponja. Development of system for underbalanced drilling offshore, approach for planning and execution. In *Paper IADC/SPE 36399, presented at the IADC/SPE Asia Pacific Drilling Technology Conference held in Kuala Lumpur, Malaysia, September*, pages 9–11, 1996.

[13] Borre Fossli, Sigbjorn Sangesland, et al. Controlled mud-cap drilling for subsea applications: well-control challenges in deep waters. *SPE Drilling & Completion*, 21(02):133–140, 2006.

[14] John Emeka Udegbunam, Kjell Kayre Fjelde, Steinar Evje, Gerhard Nygaard, et al. A simple transient flow model for mpd and ubd applications. In *SPE/IADC Managed Pressure Drilling & Underbalanced Operations Conference & Exhibition*. Society of Petroleum Engineers, 2014.

[15] Steinar Evje and Kjell K Fjelde. Hybrid flux-splitting schemes for a two-phase flow model. *Journal of Computational Physics*, 175(2):674–701, 2002.

[16] H Hirsch. *Numerical computation of internal and external flows*, volume 1. John Wiley & Sons, 1990.

[17] Theory on conservation of momentum. http://www.cs.cdu.edu.au/homepages/jmitroy/eng247/sect06.pdf. Accessed: 2015.04.27.

[18] Theory on equation of state. https://www.e-education.psu.edu/png520/m7_p3.html. Accessed: 2015.03.24.

[19] H Hirsch. *Numerical computation of internal and external flows*, volume 2. John Wiley & Sons, 1990.

[20] Quadric formula for solving second degree equations. http://mathworld.wolfram.com/QuadraticEquation.html. Accessed: 2015.02.25.

[21] Internal properties of viscosity theory. http://physics.info/viscosity. Accessed: 2015.02.17.

[22] Kjell K Fjelde, Rolv Rommetveit, Antonino Merlo, Antonio CVM Lage, et al. Improvements in dynamic modeling of underbalanced drilling. In *IADC/SPE Underbalanced Technology Conference and Exhibition*. Society of Petroleum Engineers, 2003.

Appendix A UBD Matlab code with functions

```matlab
% Transient two-phase code based on AUSMV scheme: Gas and Water
% The code can handle area changes. The area changes are defined inside
% the cells such that the where the fluxes are calculated, the geometry is
% uniform.

clear;

% Geometry data/ Must be specified
welldepth = 2000;
nobox = 25; %Number of boxes in the well
nofluxes = nobox+1;
dx = welldepth/nobox; % Boxlength
%dt = 0.005;


% Welldepth array
x(1)= -1.0*welldepth+0.5*dx;
for i=1:nobox-1
 x(i+1)=x(i)+ dx;
end

dt= 0.01;  % Timestep
dtdx = dt/dx;
time = 0.0;
endtime = 3500; % Rime for end of simulation
nosteps = endtime/dt;  %Number of total timesteps
timebetweensavingtimedata = 5;  % How often in s we save data vs time for
plotting.
nostepsbeforesavingtimedata = timebetweensavingtimedata/dt;

% Slip parameters used in the gas slip relation. vg =Kvmix+S
k = 1.1;
s = 0.5;

%Temperatur distribution
tempbot = 100+273
temptop = 20+273
tempdist = (tempbot-temptop)/nobox
for i=1:nobox
   temperatur(i)= tempbot - tempdist*i
  % temperatur(i)=293
end
%Variables
rho0 = 1000;
P0 = 100000;
Bheta = 2.2*10^9;
Alpha = 0.000207;
T0=20+273;
Rair=286,9;
```

```matlab
% Viscosities (Pa*s)/Used in the frictional pressure loss model.
for i=1:nobox
viscl(i) = viscosl(temperatur(i)); % Liquid phase
viscg(i) = viscosg(temperatur(i)); % Gas phase
end
% Density parameters. These parameters are used when finding the
% primitive variables pressure, densities in an analytical manner.
% Changing parameters here, you must also change parameters inside the
% density routines roliq and rogas.

% liquid density at stc and speed of sound in liquid
  dstc = 1000.0;    %Base density of liquid, See also roliq.
  pstc = 100000.0; % Pressure at standard conditions, 100000 Pascal
  al = 1500; % Speed of sound/compressibility of liquid phase.
  t1 = dstc-pstc/(al*al); % Help variable for calc primitive variables from
  % conservative variables
% Ideal gas law constant
  rt = 100000;

% Gravity constant

  grav = 9.81;

% Well opening. opening = 1, fully open well, opening = 0 (<0.01), the well
% is fully closed. This variable will control what boundary conditions that
% will apply at the outlet (both physical and numerical): We must change
% this further below in the code if we want to change status on this.

  wellopening = 1.0



% Specify if the primitive variables shall be found either by
% a numerical or analytical approach. If analytical = 1, analytical
% solution is used. If analytical = 0. The numerical approach is used.
% using the itsolver subroutine where the bisection numerical method
% is used.

  analytical = 1;


% Define and intilalize flow variables



%%IMPORTANT. HERE We specify the area changes. The indexes need to
% be changed if we change the grid size. Here we have assumed a
% 8.5 inch x 5 inch annulus space where diameteres have been specified in
% meters.Box i = 1 starts at bottom. By dividing it into two loops one can
% possibly introduce flow area changes (then one must keep track on where
% we are

   for i = 1:12
```

```
    do(i) = 0.2159;
    di(i) = 0.127;
    areal(i) = 3.14/4*(do(i)*do(i)- di(i)*di(i));
    arear(i) = 3.14/4*(do(i)*do(i)- di(i)*di(i));
%    area(i) = 3.14/4*(do(i)*do(i)- di(i)*di(i));
%    ang(i)=3.14/2;
   end

   for i = 14:nobox
    do(i) = 0.2159;
    di(i) = 0.127;
    areal(i) = 3.14/4*(do(i)*do(i)- di(i)*di(i));
    arear(i) = 3.14/4*(do(i)*do(i)- di(i)*di(i));
%    area(i) = 3.14/4*(do(i)*do(i)- di(i)*di(i));
%    ang(i)=3.14/2;
   end

   do(13)=(0.2159+0.2159)*0.5;
   di(13)=0.127;
   areal(13)=3.14/4*(0.2159^2-0.127^2);
   arear(13)=3.14/4*(0.2159^2-0.127^2);




% Now comes the intialization of the physical variables in the well.
% First primitive variables, then the conservative ones.
    for i = 1:nobox
% Here the well is intialized. This code does not need change.
% The extension letter o refers to the table represententing the
% values at the previous timestep (old values).

        % Density of liquid and gas:
        dl(i) = 1000.0;
        dg(i)= 1.0;
        %"Old" density is set equal to new density to calculate new values
        %based on the old ones:
        dlo(i)= dl(i);
        dgo(i)=dg(i);
        % Velocity of liquid and gas at new and previous timesteps:
        vl(i) = 0.0;
        vlo(i)= 0.0;
        vg(i)= 0.0;
        vgo(i)= 0.0;
        %The pressure in the horisontal pipe is the same
        %all over:
        p(i) = 100000.0;
        po(i) = p(i);
        %Phase volume fractions of gas and liquid:
        eg(i)= 0.0;    %Gas
        ego(i)=eg(i);
        ev(i)=1-eg(i); % Liquid
        evo(i)=ev(i);

        vg(i)=0.0;
```

```
        vgo(i)=0.0;
        vl(i)=0.0;
        vlo(i)=0.0;

        % Variables related to the velocity of the flux boundaries at old
        %and new times, and on the left and right side of the boxes
        % reflecting that area changes can take part inside cells (i.e :
        % (A x v)left = (A x v)right, continuity equation.
        vgr(i)=0.0;
        vgor(i)= 0.0;
        vgl(i)= 0.0;
        vgol(i)= 0.0;

        vlr(i)=0.0;
        vlor(i)=0.0;
        vll(i)=0.0;
        vlol(i)=0.0;

  % Conservative variables:

        qv(i,1)=dl(i)*ev(i)*(areal(i)+arear(i))*0.5;
        qvo(i,1)=qv(i,1);

        qv(i,2)=dg(i)*eg(i)*(areal(i)+arear(i))*0.5;
        qvo(i,2)=qv(i,2);

        qv(i,3)=(qv(i,1)*vl(i)+qv(i,2)*vg(i))*(areal(i)+arear(i))*0.5;
        qvo(i,3)=qv(i,3);

    end

% Intialize fluxes between the cells/boxes

for i = 1:nofluxes
  for j =1:3
   flc(i,j)=0.0; % Flux of liquid over box boundary
   fgc(i,j)=0.0; % Flux of gas over box boundary
   fp(i,j)= 0.0; % Pressure flux over box boundary
  end
end


% CODE BELOW HAVE BEEN ADDED TO INITIALIZE FLOWVARIABLES IN A
% VERTICAL WELL:

p(nobox)= 100000.0+0.5*dx*9.81*dstc;
dl(nobox)=rholiq(p(nobox),temperatur(nobox));
dg(nobox)=rogas(p(nobox),temperatur(nobox));

for i=nobox-1:-1:1
p(i)=p(i+1)+dx*9.81*dl(i+1);
dl(i)=rholiq(p(i),temperatur(i));
dg(i)=rogas(p(i),temperatur(i));
end
```

```matlab
for i=nobox-1:-1:1
p(i)=p(i+1)+dx*9.81*(dl(i+1)+dl(i))*0.5;
dl(i)=rholiq(p(i),temperatur(i));
dg(i)=rogas(p(i),temperatur(i));

end


for i=1:nobox
  dlo(i)=dl(i);
  dgo(i)=dg(i);
  po(i)=p(i);
  qv(i,1)=dl(i)*ev(i)*(areal(i)+arear(i))*0.5;
  qvo(i,1)=qv(i,1);

  qv(i,2)=dg(i)*eg(i)*(areal(i)+arear(i))*0.5;
  qvo(i,2)=qv(i,2);

  qv(i,3)=(qv(i,1)*vl(i)+qv(i,2)*vg(i))*(areal(i)+arear(i))*0.5;
  qvo(i,3)=qv(i,3);
end


%  Main program. Here we will progress in time. First som intializations
% and definitions to take out results. The for loop below runs until the
% simulation is finished.

countsteps = 0;
counter=0;
printcounter = 1;
pbot(printcounter) = p(1);
pchoke(printcounter)= p(nobox);
liquidmassrateout(printcounter) = 0;
gasmassrateout(printcounter)=0;
timeplot(printcounter)=time;

for i = 1:nosteps
   countsteps=countsteps+1;
   counter=counter+1;
   time = time+dt;

   g = grav;

% Then a section where specify the boundary conditions.
% Here we specify the inlet rates of the different phases at the
% bottom of the pipe in kg/s. We interpolate to make things smooth.
% It is also possible to change the outlet boundary status of the well
% here. First we specify rates at the bottom and the pressure at the outlet
% in case we have an open well. This is a place where we can change the
% code.
```

```
if (time < 150)

  inletligmassrate=0.0;
   inletgasmassrate=0.0;

elseif ((time>=150) & (time < 160))
  inletligmassrate = 22*(time-150)/10;
  inletgasmassrate = 2.0*(time-150)/10;

elseif ((time >=160) & (time<1700))
  inletligmassrate = 22;
  inletgasmassrate = 2.0;

elseif ((time>=1700)& (time<1710))
  inletligmassrate = 22-22*(time-1700)/10;
  inletgasmassrate = 2.0-2.0*(time-1700)/10;
elseif ((time>=1710)&(time<2000))
  inletligmassrate =0;
  inletgasmassrate =0;
elseif ((time>=2000)& (time<2010))
  inletligmassrate= 22*(time-2000)/10;
  inletgasmassrate= 2.0*(time-2000)/10;
elseif (time>2010)
  inletligmassrate= 22;
  inletgasmassrate= 2.0;
end
```

```
% specify the outlet pressure /Physical. Here we have given the pressure as
% constant. It would be possible to adjust it during openwell conditions
% either by giving the wanted pressure directly (in the command lines
% above) or by finding it indirectly through a chokemodel where the
wellopening
% would be an input parameter. The wellopening variable would equally had
% to be adjusted inside the command line structure given right above.

 pressureoutlet = 100000.0;

% Based on these boundary values combined with use of extrapolations
techniques
% for the remaining unknowns at the boundaries, we will define the mass and
% momentum fluxes at the boundaries (inlet and outlet of pipe).

% inlet fluxes first.

    flc(1,1)= inletligmassrate/areal(1);
    flc(1,2)= 0.0;
    flc(1,3)= flc(1,1)*vlo(1);
```

```matlab
    fgc(1,1)= 0.0;
    fgc(1,2)= inletgasmassrate/areal(1);
    fgc(1,3)= fgc(1,2)*vgo(1);

    fp(1,1)= 0.0;
    fp(1,2)= 0.0;
    fp(1,3)= po(1)+0.5*(po(1)-po(2)); %Interpolation used to find the
% pressure at the inlet/bottom of the well.




% Outlet fluxes (open & closed conditions)

    if (wellopening>0.01)

% Here open end condtions are given
        flc(nofluxes,1)= dlo(nobox)*evo(nobox)*vlo(nobox);

        flc(nofluxes,2)= 0.0;
        flc(nofluxes,3)= flc(nofluxes,1)*vlo(nobox);

        fgc(nofluxes,1)= 0.0;
        fgc(nofluxes,2)= dgo(nobox)*ego(nobox)*vgo(nobox);
        fgc(nofluxes,3)= fgc(nofluxes,2)*vgo(nobox);

        fp(nofluxes,1)= 0.0;
        fp(nofluxes,2)= 0.0;
        fp(nofluxes,3)= pressureoutlet;
    else

% Here closed end conditions are given

        flc(nofluxes,1)= 0.0;
        flc(nofluxes,2)= 0.0;
        flc(nofluxes,3)= 0.0;

        fgc(nofluxes,1)= 0.0;
        fgc(nofluxes,2)= 0.0;
        fgc(nofluxes,3)= 0.0;

        fp(nofluxes,1)=0.0;
        fp(nofluxes,2)=0.0;
        fp(nofluxes,3)= po(nobox)-0.5*(po(nobox-1)-po(nobox));

    end

% Now we will find the fluxes between the different cells.
```

```matlab
% NB - IMPORTANE -  Note that if we change the compressibilities/sound velocities of
% the fluids involved, we need to do changes inside the csound function.

    for j = 2:nofluxes-1
     cl = csound(ego(j-1),po(j-1),dlo(j-1),k);
     cr = csound(ego(j),po(j),dlo(j),k);
     c = max(cl,cr);
     pll = psip(vlor(j-1),c,evo(j));
     plr = psim(vlol(j),c,evo(j-1));
     pgl = psip(vgor(j-1),c,ego(j));
     pgr = psim(vgol(j),c,ego(j-1));
     vmixr = vlol(j)*evo(j)+vgol(j)*ego(j);
     vmixl = vlor(j-1)*evo(j-1)+vgor(j-1)*ego(j-1);

     pl = pp(vmixl,c);
     pr = pm(vmixr,c);
     mll= evo(j-1)*dlo(j-1);
     mlr= evo(j)*dlo(j);
     mgl= ego(j-1)*dgo(j-1);
     mgr= ego(j)*dgo(j);

     flc(j,1)= mll*pll+mlr*plr;
     flc(j,2)= 0.0;
     flc(j,3)= mll*pll*vlor(j-1)+mlr*plr*vlol(j);

     fgc(j,1)=0.0;
     fgc(j,2)= mgl*pgl+mgr*pgr;
     fgc(j,3)= mgl*pgl*vgor(j-1)+mgr*pgr*vgol(j);

     fp(j,1)= 0.0;
     fp(j,2)= 0.0;
     fp(j,3)= pl*po(j-1)+pr*po(j);
    end

% Fluxes have now been calculated. We will now update the conservative
% variables in each of the numerical cells.

   sumfriclossgrad = 0;
   sumhyd=0;

    for j=1:nobox


     densmix = dlo(j)*evo(j)+dgo(j)*ego(j);
     sumhyd = sumhyd+dx*g*densmix;
     a2 = arear(j);
     a1 = areal(j);
     avg = (a2+a1)*0.5;

     pressure=p(j);

%     We calculate the frictional gradient by calling upon the dpfric
function.
```

```
    friclossgrad(j) =
dpfric(vlo(j),vgo(j),evo(j),ego(j),dlo(j),dgo(j),pressure,temperatur(j),do(
j),di(j),viscl(j),viscg(j));

    sumfriclossgrad = sumfriclossgrad+dx*friclossgrad(j);


    qv(j,1)=qvo(j,1)-dtdx*((a2*flc(j+1,1)-a1*flc(j,1))...
                        +(a2*fgc(j+1,1)-a1*fgc(j,1))...
                        +(avg*fp(j+1,1)-avg*fp(j,1)));

    qv(j,2)=qvo(j,2)-dtdx*((a2*flc(j+1,2)-a1*flc(j,2))...
                        +(a2*fgc(j+1,2)-a1*fgc(j,2))...
                        +(avg*fp(j+1,2)-avg*fp(j,2)));

    qv(j,3)=qvo(j,3)-dtdx*((a2*flc(j+1,3)-a1*flc(j,3))...
                        +(a2*fgc(j+1,3)-a1*fgc(j,3))...
                        +(avg*fp(j+1,3)-avg*fp(j,3)))...
                -dt*avg*((friclossgrad(j).')+g*densmix);

    end




% Section where we find the physical variables (pressures, densities etc)
% from the conservative variables. Some trickes to ensure stability. These
% are induced to avoid negative masses.


    for j=1:nobox


% Remove the area from the conservative variables to find the
% the primitive variables from the conservative ones.

    qv(j,1)= qv(j,1)/(areal(j)+arear(j))*2.0;
    qv(j,2)= qv(j,2)/(areal(j)+arear(j))*2.0;

    if (qv(j,1)<0.00000001)
      qv(j,1)=0.0000001;
    end

    if (qv(j,2)< 0.00000001)
      qv(j,2)=0.0000001;
    end


% Below, we find the primitive variables pressure and densities based on
% the conservative variables q1,q2. One can choose between getting them by
% analytical or numerical solution approach specified in the beginning of
```

```matlab
% the program.

% %Variables
% rho0 = 1000;
% P0 = 100000;
% Bheta = 2.2*10^9;
% Alpha = 0.000207;
% T0=20+273;
% R=286,9;

    if (analytical == 1)
      % Coefficients: (OLD CODE)
     % a = 1/(al*al);
     % b = t1-qv(j,1)-rt*qv(j,2)/(al*al);
     % c = -1.0*t1*rt*qv(j,2);

        x1 = rho0-(P0*rho0/Bheta)-(rho0*Alpha*(temperatur(j)-T0));
        x2 = rho0/Bheta;
        x3 = -1.0*qv(j,2)*Rair*temperatur(j);

% % %        x1=dl0-(dl0/B)*p0-dl0*A*(tk(j)-t0);
% % %        x2=dl0/B;
% % %        x3=-1.0*qv(j,2)*R*tk(j);

        a=x2;
        b=x1+x2*x3-qv(j,1);
        c=x1*x3;
% abc formel
  % solution =(-b +sqrt(b.^2-4.*a.*c))/2.*a

      % Analytical solution:
       p(j)=(-b+sqrt(b*b-4*a*c))/(2*a);   % Pressure
       dl(j)= rho0+(rho0/Bheta)*(p(j)-P0)-(rho0*Alpha*(temperatur(j)-T0)); %
Density of liquid
       dg(j) = p(j)/(Rair*temperatur(j));                    % Density of gas
    else
      %Numerical Solution:
      [p(j),error]=itsolver(po(j),qv(j,1),qv(j,2)); % Pressure
      dl(j)=rholiq(p(j),temperatur(j)); % Density of liquid
      dg(j)=rogas(p(j),temperatur(j)); % Density of gas

      % Incase a numerical solution is not found, the program will write out
"error":
      if error > 0
         error
      end
    end


 % Find the phase volume fractions based on new conservative variables and
 % updated densities.

      eg(j)= qv(j,2)/(dg(j));
      ev(j)=1-eg(j);
```

```matlab
%     Reset average conservative varibles in cells with area changes inside.

      qv(j,1)=qv(j,1)*(areal(j)+arear(j))/2.0;
      qv(j,2)=qv(j,2)*(areal(j)+arear(j))/2.0;


%     The section below is used to find the primitive variables vg,vl
%    (phase velocities) based on the updated conservative variable q3 and
%     the slip relation.


% Part where we interpolate in the slip parameters to avoid a
% singularities when approaching one phase gas flow.
% In the transition to one-phase gas flow, we need to
% have a smooth transition to no-slip conditions.

      xint = (eg(j)-0.75)/0.25;
      k0 = k;
      s0 = s;
      if ((eg(j)>=0.75) & (eg(j)<=1.0))
        k0 =1.0*xint+k*(1-xint);
        s0 = 0.0*xint+s*(1-xint);
      end

      if (eg(j)>=0.999999)
        k1 = 1.0;
        s1 = 0.0;
      else
        k1 = (1-k0*eg(j))/(1-eg(j));
        s1 = -1.0*s0*eg(j)/(1-eg(j));
      end



%     Below we operate with gas vg and liquid vl velcoities specified
%     both in the right part and left part inside a box. (since we have
%     area changes inside a box these can be different. vgl is gas velocity
%     to the left of the disconinuity. vgr is gas velocity to the right of
%     the discontinuity.
%

      help1 = dl(j)*ev(j)*k1+dg(j)*eg(j)*k0;
      help2 = dl(j)*ev(j)*s1+dg(j)*eg(j)*s0;

      vmixhelpl = (qv(j,3)/areal(j)-help2)/help1;
      vgl(j)=k0*vmixhelpl+s0;
      vll(j)=k1*vmixhelpl+s1;

      vmixhelpr = (qv(j,3)/arear(j)-help2)/help1;
      vgr(j)=k0*vmixhelpr+s0;
      vlr(j)=k1*vmixhelpr+s1;
```

```matlab
 %  Averaging velocities.

     vl(j)= 0.5*(vll(j)+vlr(j));
     vg(j)= 0.5*(vgl(j)+vgr(j));

   end



% Old values are now set equal to new values in order to prepare
% computation of next time level.
   for j = 1:nobox
    po(j)=p(j);
    dlo(j)=dl(j);  %Liquid density
    dgo(j)=dg(j);  %Gas density
    vlo(j)=vl(j);  %Liquid velocity
    vgo(j)=vg(j);  %Gas velocity
    ego(j)=eg(j);  %Gas fration
    evo(j)=ev(j);  %Liquid fraction.

    vlor(j)=vlr(j);
    vlol(j)=vll(j);
    vgor(j)=vgr(j);
    vgol(j)=vgl(j);

     for m =1:3
      qvo(j,m)=qv(j,m);
     end
   end


% Section where we save some timedependent variables in arrays.
% e.g. the bottomhole pressure. They will be saved for certain
% timeintervalls defined in the start of the program in order to ensure
% that the arrays do not get to long!

  if (counter>=nostepsbeforesavingtimedata)
    printcounter=printcounter+1;
    time
    pbot(printcounter)= p(1);
    pchoke(printcounter)=p(nobox);
    pcasingshoe(printcounter)=p(25); %NB THIS MUST BE DEFINED IN CORRECT BOX

liquidmassrateout(printcounter)=dl(nobox)*ev(nobox)*vl(nobox)*arear(nobox);
    gasmassrateout(printcounter)=dg(nobox)*eg(nobox)*vg(nobox)*arear(nobox);
    timeplot(printcounter)=time;
    sumfriclossgrad(printcounter)= friclossgrad(1);
    sumhyd(printcounter)= densmix(1);
    counter = 0;



  end
end
```

```
% end of stepping forward in time.




% Printing of resultssection


countsteps % Marks number of simulation steps.


% Plot commands for variables vs time. The commands can also
% be copied to command screen where program is run for plotting other
% variables.

plot(timeplot,pbot/100000)
%plot(timeplot,pchoke/100000)
%plot(timeplot,liquidmassrateout)F
%plot(timeplot,gasmassrateout)
%plot(vg)

%Plot commands for variables vs depth/Only the last simulated
%values/endtime is visualised

%plot(vl,x);
%plot(vg,x);
%plot(eg,x);
%plot(p,x);
%plot(dl,x);
%plot(dg,x);
```

Function for gas density

```
function rhog = rogas(pressure,temperatur)

%Simple gas density model. Temperature is neglected.
% rhogas = pressure / (velocity of sound in the gas phase)^2 = pressure /
% rT --> gas sound velcoity = SQRT(rT)
r = 296.8; % (j/kg*K)

  rhog = pressure/(r*temperatur);
```

```
Function for liquid density
```

```matlab
function [rhol] = rholiq(pressure,temperatur)
%Simple model for liquid density
p0 = 100000; % Assumed
t0 = 20+273.5;



beta = 2.2*10^9; % (Pa)  bulk modulus of liquid
alpha = 0.000207; % (K^-1)  Volumetric thermal expansion of
liquid
rho0 = 1000;


rhol = rho0 +((rho0/beta)*(pressure-p0))-(rho0*alpha*(temperatur-
t0));
end
```

```
Function for liquid viscosity
```

```matlab
function [ viscosityliq ] = viscosl(temperatur)


ref_water_vis=2.414*10^-5 %(Pa.s)


viscosityliq=ref_water_vis*10^(247.8/(temperatur-140))
end
```

```
Function for gas viscosity
```

```matlab
function [ viscositygas ] = viscosg(temperatur)

%ref_vis_gas = 1.781*10^-5 %Nitrogen
%ref_temp =300.55 %nitrogen
%C=111 % Nitrogen
ref_vis_gas = 1,827*10^-5 air
ref_temp =291.15 %air
C=120 % Air



viscositygas=ref_vis_gas*(temperatur/ref_temp)^(3/2)*((ref_temp+C
)/(temperatur+C))

end
```

Function for frictional pressure loss

```
function friclossgrad =
dpfric(vlo,vgo,evo,ego,dlo,dgo,pressure,temperatur,do,di,viscl,viscg)

%friclossgrad =
%dpfric(vlo,vgo,evo,ego,dlo,dgo,pressure,do,di,viscl,viscg)
% Works for two phase flow. The one phase flow model is used but mixture
 % values are introduced.

  rhol = rholiq(pressure,temperatur);
  rhog = rogas(pressure,temperatur);
 vmixfric = vlo*evo+vgo*ego;
 viscmix =  viscl*evo+viscg*ego;
 densmix = dlo*evo+dgo*ego;

 % Calculate mix reynolds number
 Re = ((densmix*abs(vmixfric)*(do-di))/viscmix);

 % Calculate friction factor. For Re > 3000, the flow is turbulent.
 % For Re < 2000, the flow is laminar. Interpolate in between.

 if (Re<0.001)
   f=0.0;
 else
  if (Re >= 3000)
   f = 0.052*Re^(-0.19);
  elseif ( (Re<3000) & (Re > 2000))
   f1 = 24/Re;
   f2 = 0.052*Re^(-0.19);
   xint = (Re-2000)/1000.0;
   f = (1.0-xint)*f1+xint*f2;
  else
   f = 24/Re;
  end
 end

  friclossgrad = ((2*f*densmix*vmixfric*abs(vmixfric))/(do-di));


end
```

Appendix B Mudcap Matlab code

```
% Transient two-phase code based on AUSMV scheme: Gas and Water
% Re-customized to a mudcap drilling scenario


% The code can handle area changes. The area changes are defined inside
% the cells such that the where the fluxes are calculated, the geometry is
% uniform.

clear;

% Geometry data/ Must be specified
welldepth = 2000;
nobox = 25; %Number of boxes in the well
nofluxes = nobox+1;
dx = welldepth/nobox; % Boxlength
%dt = 0.005;


% Welldepth array
x(1)= -1.0*welldepth+0.5*dx;
for i=1:nobox-1
 x(i+1)=x(i)+ dx;
end

dt= 0.01;  % Timestep
dtdx = dt/dx;
time = 0.0;
endtime = 1000; % Rime for end of simulation
nosteps = endtime/dt;  %Number of total timesteps
timebetweensavingtimedata = 5;  % How often in s we save data vs time for
plotting.
nostepsbeforesavingtimedata = timebetweensavingtimedata/dt;

% Slip parameters used in the gas slip relation. vg =Kvmix+S
k = 1.1;
s = 0.5;

%Temperatur distribution
tempbot = 100+273
temptop = 20+273
tempdist = (tempbot-temptop)/nobox
for i=1:nobox
   temperatur(i)= tempbot - tempdist*i
  % temperatur(i)=293
end
%Variables
rho0 = 1000;
P0 = 100000;
Bheta = 2.2*10^9;
Alpha = 0.000207;
T0=20+273;
Rair=286,9;
```

```matlab
% Viscosities (Pa*s)/Used in the frictional pressure loss model.
for i=1:nobox
viscl(i) = viscosl(temperatur(i)); % Liquid phase
viscg(i) = viscosg(temperatur(i)); % Gas phase
end
% Density parameters. These parameters are used when finding the
% primitive variables pressure, densities in an analytical manner.
% Changing parameters here, you must also change parameters inside the
% density routines roliq and rogas.

% liquid density at stc and speed of sound in liquid
  dstc = 1000.0;    %Base density of liquid, See also roliq.
  pstc = 100000.0; % Pressure at standard conditions, 100000 Pascal
  al = 1500; % Speed of sound/compressibility of liquid phase.
  t1 = dstc-pstc/(al*al); % Help variable for calc primitive variables from
  % conservative variables
% Ideal gas law constant
  rt = 100000;

% Gravity constant

  grav = 9.81;

% Well opening. opening = 1, fully open well, opening = 0 (<0.01), the well
% is fully closed. This variable will control what boundary conditions that
% will apply at the outlet (both physical and numerical): We must change
% this further below in the code if we want to change status on this.

  wellopening = 1.0

% Specify if the primitive variables shall be found either by
% a numerical or analytical approach. If analytical = 1, analytical
% solution is used. If analytical = 0. The numerical approach is used.
% using the itsolver subroutine where the bisection numerical method
% is used.

  analytical = 1;

% Define and intilalize flow variables

%%IMPORTANT. HERE We specify the area changes. The indexes need to
% be changed if we change the grid size. Here we have assumed a
% 8.5 inch x 5 inch annulus space where diameteres have been specified in
% meters.Box i = 1 starts at bottom. By dividing it into two loops one can
% possibly introduce flow area changes (then one must keep track on where
% we are

  for i = 1:12
    do(i) = 0.2159;
    di(i) = 0.127;
```

```matlab
      areal(i) = 3.14/4*(do(i)*do(i)- di(i)*di(i));
      arear(i) = 3.14/4*(do(i)*do(i)- di(i)*di(i));
%     area(i) = 3.14/4*(do(i)*do(i)- di(i)*di(i));
%     ang(i)=3.14/2;
    end

    for i = 14:nobox
     do(i) = 0.2159;
     di(i) = 0.127;
     areal(i) = 3.14/4*(do(i)*do(i)- di(i)*di(i));
     arear(i) = 3.14/4*(do(i)*do(i)- di(i)*di(i));
 %   area(i) = 3.14/4*(do(i)*do(i)- di(i)*di(i));
 %   ang(i)=3.14/2;
    end

    do(13)=(0.2159+0.2159)*0.5;
    di(13)=0.127;
    areal(13)=3.14/4*(0.2159^2-0.127^2);
    arear(13)=3.14/4*(0.2159^2-0.127^2);




% Now comes the intialization of the physical variables in the well.
% First primitive variables, then the conservative ones.
    for i = 1:nobox
% Here the well is intialized. This code does not need change.
% The extension letter o refers to the table representing the
% values at the previous timestep (old values).

        % Density of liquid and gas:
        dl(i) = 1000.0;
        dg(i)= 1.0;
        %"Old" density is set equal to new density to calculate new values
        %based on the old ones:
        dlo(i)= dl(i);
        dgo(i)=dg(i);
        % Velocity of liquid and gas at new and previous timesteps:
        vl(i) = 0.0;
        vlo(i)= 0.0;
        vg(i)= 0.0;
        vgo(i)= 0.0;
        %The pressure in the horisontal pipe is the same
        %all over:
        p(i) = 100000.0;
        po(i) = p(i);
        %Phase volume fractions of gas and liquid:
        eg(i)= 0.0;     %Gas
        ego(i)=eg(i);
        ev(i)=1-eg(i); % Liquid
        evo(i)=ev(i);

        vg(i)=0.0;
        vgo(i)=0.0;
        vl(i)=0.0;
```

```matlab
        vlo(i)=0.0;

        % Variables related to the velocity of the flux boundaries at old
        %and new times, and on the left and right side of the boxes
        % reflecting that area changes can take part inside cells (i.e :
        % (A x v)left = (A x v)right, continuity equation.
        vgr(i)=0.0;
        vgor(i)= 0.0;
        vgl(i)= 0.0;
        vgol(i)= 0.0;

        vlr(i)=0.0;
        vlor(i)=0.0;
        vll(i)=0.0;
        vlol(i)=0.0;

  % Conservative variables:

        qv(i,1)=dl(i)*ev(i)*(areal(i)+arear(i))*0.5;
        qvo(i,1)=qv(i,1);

        qv(i,2)=dg(i)*eg(i)*(areal(i)+arear(i))*0.5;
        qvo(i,2)=qv(i,2);

        qv(i,3)=(qv(i,1)*vl(i)+qv(i,2)*vg(i))*(areal(i)+arear(i))*0.5;
        qvo(i,3)=qv(i,3);

    end

% Intialize fluxes between the cells/boxes

for i = 1:nofluxes
  for j =1:3
   flc(i,j)=0.0; % Flux of liquid over box boundary
   fgc(i,j)=0.0; % Flux of gas over box boundary
   fp(i,j)= 0.0; % Pressure flux over box boundary
  end
end


% CODE BELOW HAVE BEEN ADDED TO INITIALIZE FLOWVARIABLES IN A
% VERTICAL WELL:

p(nobox)= 100000.0+0.5*dx*9.81*dstc;
dl(nobox)=rholiq(p(nobox),temperatur(nobox));
dg(nobox)=rogas(p(nobox),temperatur(nobox));

for i=nobox-1:-1:1
p(i)=p(i+1)+dx*9.81*dl(i+1);
dl(i)=rholiq(p(i),temperatur(i));
dg(i)=rogas(p(i),temperatur(i));
end

for i=nobox-1:-1:1
```

```
p(i)=p(i+1)+dx*9.81*(dl(i+1)+dl(i))*0.5;
dl(i)=rholiq(p(i),temperatur(i));
dg(i)=rogas(p(i),temperatur(i));

end


for i=1:nobox
  dlo(i)=dl(i);
  dgo(i)=dg(i);
  po(i)=p(i);
  qv(i,1)=dl(i)*ev(i)*(areal(i)+arear(i))*0.5;
  qvo(i,1)=qv(i,1);

  qv(i,2)=dg(i)*eg(i)*(areal(i)+arear(i))*0.5;
  qvo(i,2)=qv(i,2);

  qv(i,3)=(qv(i,1)*vl(i)+qv(i,2)*vg(i))*(areal(i)+arear(i))*0.5;
  qvo(i,3)=qv(i,3);
end


%  Main program. Here we will progress in time. First som intializations
% and definitions to take out results. The for loop below runs until the
% simulation is finished.

countsteps = 0;
counter=0;
printcounter = 1;
pbot(printcounter) = p(1);
pchoke(printcounter)= p(nobox);
liquidmassrateout(printcounter) = 0;
gasmassrateout(printcounter)=0;
timeplot(printcounter)=time;

for i = 1:nosteps
   countsteps=countsteps+1;
   counter=counter+1;
   time = time+dt;

   g = grav;

% Then a section where specify the boundary conditions.
% Here we specify the inlet rates of the different phases at the
% bottom of the pipe in kg/s. We interpolate to make things smooth.
% It is also possible to change the outlet boundary status of the well
% here. First we specify rates at the bottom and the pressure at the outlet
% in case we have an open well. This is a place where we can change the
% code.



if (time < 150)
```

```matlab
   inletligmassrate=0.0;
    inletgasmassrate=0.0;


elseif ((time>=150) & (time < 160))
  inletligmassrate = 22*(time-150)/10;
  inletgasmassrate = 0;


elseif ((time >=160) & (time<300))
  inletligmassrate = 22;
  inletgasmassrate = 0;


elseif ((time>=300)& (time<310))
  inletligmassrate = 22-22*(time-300)/10;
  inletgasmassrate = 0;
elseif ((time>=310)&(time<800))
  inletligmassrate =0;
  inletgasmassrate =0;
elseif ((time>=800)& (time<810))
  inletligmassrate= 22*(time-800)/10;
  inletgasmassrate= 0;
elseif (time>810)
  inletligmassrate= 22;
  inletgasmassrate= 0;
end
```

```matlab
% specify the outlet pressure /Physical. Here we have given the pressure as
% constant. It would be possible to adjust it during openwell conditions
% either by giving the wanted pressure directly (in the command lines
% above) or by finding it indirectly through a chokemodel where the
wellopening
% would be an input parameter. The wellopening variable would equally had
% to be adjusted inside the command line structure given right above.

 pressureoutlet = 100000.0;

% Based on these boundary values combined with use of extrapolations
techniques
% for the remaining unknowns at the boundaries, we will define the mass and
% momentum fluxes at the boundaries (inlet and outlet of pipe).
```

```matlab
% inlet fluxes first.

    flc(1,1)= inletligmassrate/areal(1);
    flc(1,2)= 0.0;
    flc(1,3)= flc(1,1)*vlo(1);

    fgc(1,1)= 0.0;
    fgc(1,2)= inletgasmassrate/areal(1);
    fgc(1,3)= fgc(1,2)*vgo(1);

    fp(1,1)= 0.0;
    fp(1,2)= 0.0;
    fp(1,3)= po(1)+0.5*(po(1)-po(2)); %Interpolation used to find the
% pressure at the inlet/bottom of the well.




% Outlet fluxes (open & closed conditions)

    if (wellopening>0.01)

% Here open end condtions are given
        flc(nofluxes,1)= dlo(nobox)*evo(nobox)*vlo(nobox);

        flc(nofluxes,2)= 0.0;
        flc(nofluxes,3)= flc(nofluxes,1)*vlo(nobox);

        fgc(nofluxes,1)= 0.0;
        fgc(nofluxes,2)= dgo(nobox)*ego(nobox)*vgo(nobox);
        fgc(nofluxes,3)= fgc(nofluxes,2)*vgo(nobox);

        fp(nofluxes,1)= 0.0;
        fp(nofluxes,2)= 0.0;
        fp(nofluxes,3)= pressureoutlet;
    else

% Here closed end conditions are given

        flc(nofluxes,1)= 0.0;
        flc(nofluxes,2)= 0.0;
        flc(nofluxes,3)= 0.0;

        fgc(nofluxes,1)= 0.0;
        fgc(nofluxes,2)= 0.0;
        fgc(nofluxes,3)= 0.0;

        fp(nofluxes,1)=0.0;
        fp(nofluxes,2)=0.0;
        fp(nofluxes,3)= po(nobox)-0.5*(po(nobox-1)-po(nobox));
```

```
        end

        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



        %Alternativ 1!
.
% % %       suckrate = 0;
% % %       if ((time> 400)&(time<410))
% % %          suckrate = 100*(time-400)/10;
% % %       elseif(time>410)
% % %           suckrate = 100;
% % %       end


        This is the suction point code and the fix for the boundary conditio

        suckrate = 0;
        if ((time> 400)&(time<410))
          suckrate = 22*(time-400)/10;
             flc(nofluxes,1) = 0;
             flc(nofluxes,3) = 0;
             fp(nofluxes,3)= po(nobox)-0.5*(po(nobox-1)-po(nobox));
            % fp(nofluxes,3)=100000;


        elseif(time>410)
          suckrate = 22;
             flc(nofluxes,1) = 0;
             flc(nofluxes,3) = 0;
           fp(nofluxes,3)= po(nobox)-0.5*(po(nobox-1)-po(nobox));
          %  fp(nofluxes,3)=100000;
        end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

% Now we will find the fluxes between the different cells.
%

        for j = 2:nofluxes-1
         cl = csound(ego(j-1),po(j-1),dlo(j-1),k);
         cr = csound(ego(j),po(j),dlo(j),k);
         c = max(cl,cr);
         pll = psip(vlor(j-1),c,evo(j));
         plr = psim(vlol(j),c,evo(j-1));
         pgl = psip(vgor(j-1),c,ego(j));
         pgr = psim(vgol(j),c,ego(j-1));
         vmixr = vlol(j)*evo(j)+vgol(j)*ego(j);
```

```matlab
        vmixl = vlor(j-1)*evo(j-1)+vgor(j-1)*ego(j-1);


        pl = pp(vmixl,c);
        pr = pm(vmixr,c);
        mll= evo(j-1)*dlo(j-1);
        mlr= evo(j)*dlo(j);
        mgl= ego(j-1)*dgo(j-1);
        mgr= ego(j)*dgo(j);

        flc(j,1)= mll*pll+mlr*plr;
        flc(j,2)= 0.0;
        flc(j,3)= mll*pll*vlor(j-1)+mlr*plr*vlol(j);

        fgc(j,1)=0.0;
        fgc(j,2)= mgl*pgl+mgr*pgr;
        fgc(j,3)= mgl*pgl*vgor(j-1)+mgr*pgr*vgol(j);

        fp(j,1)= 0.0;
        fp(j,2)= 0.0;
        fp(j,3)= pl*po(j-1)+pr*po(j);
      end

% Fluxes have now been calculated. We will now update the conservative
% variables in each of the numerical cells.

    sumfriclossgrad = 0;
    sumhyd=0;

      for j=1:nobox


        densmix = dlo(j)*evo(j)+dgo(j)*ego(j);
        sumhyd = sumhyd+dx*g*densmix;
        a2 = arear(j);
        a1 = areal(j);
        avg = (a2+a1)*0.5;

        pressure=p(j);

%      We calculate the frictional gradient by calling upon the dpfric
function.
        friclossgrad(j) =
dpfric(vlo(j),vgo(j),evo(j),ego(j),dlo(j),dgo(j),pressure,temperatur(j),do(j),
di(j),viscl(j),viscg(j));

        sumfriclossgrad = sumfriclossgrad+dx*friclossgrad(j);


        qv(j,1)=qvo(j,1)-dtdx*((a2*flc(j+1,1)-a1*flc(j,1))...
                            +(a2*fgc(j+1,1)-a1*fgc(j,1))...
                            +(avg*fp(j+1,1)-avg*fp(j,1)));

        qv(j,2)=qvo(j,2)-dtdx*((a2*flc(j+1,2)-a1*flc(j,2))...
                            +(a2*fgc(j+1,2)-a1*fgc(j,2))...
```

```
                              +(avg*fp(j+1,2)-avg*fp(j,2)));

      qv(j,3)=qvo(j,3)-dtdx*((a2*flc(j+1,3)-a1*flc(j,3))...
                            +(a2*fgc(j+1,3)-a1*fgc(j,3))...
                            +(avg*fp(j+1,3)-avg*fp(j,3)))...
                    -dt*avg*((friclossgrad(j).')+g*densmix);


      if (j==14)
      qv(14,1)=qv(14,1)-dt*suckrate/dx;

      end


      end




% Section where we find the physical variables (pressures, densities etc)
% from the conservative variables. Some trickes to ensure stability. These
% are induced to avoid negative masses.


      for j=1:nobox


% Remove the area from the conservative variables to find the
% the primitive variables from the conservative ones.

      qv(j,1)= qv(j,1)/(areal(j)+arear(j))*2.0;
      qv(j,2)= qv(j,2)/(areal(j)+arear(j))*2.0;

      if (qv(j,1)<0.00000001)
        qv(j,1)=0.0000001;
      end

      if (qv(j,2)< 0.00000001)
        qv(j,2)=0.0000001;
      end



% Below, we find the primitive variables pressure and densities based on
% the conservative variables q1,q2. One can choose between getting them by
% analytical or numerical solution approach specified in the beginning of
% the program.

% %Variables
% rho0 = 1000;
% P0 = 100000;
% Bheta = 2.2*10^9;
% Alpha = 0.000207;
```

```
% T0=20+273;
% R=286,9;

    if (analytical == 1)
      % Coefficients: (OLD CODE)
     % a = 1/(al*al);
     % b = t1-qv(j,1)-rt*qv(j,2)/(al*al);
     % c = -1.0*t1*rt*qv(j,2);

       x1 = rho0-(P0*rho0/Bheta)-(rho0*Alpha*(temperatur(j)-T0));
       x2 = rho0/Bheta;
       x3 = -1.0*qv(j,2)*Rair*temperatur(j);

% % %        x1=dl0-(dl0/B)*p0-dl0*A*(tk(j)-t0);
% % %        x2=dl0/B;
% % %        x3=-1.0*qv(j,2)*R*tk(j);


       a=x2;
       b=x1+x2*x3-qv(j,1);
       c=x1*x3;
% abc formel
  % solution =(-b +sqrt(b.^2-4.*a.*c))/2.*a

     % Analytical solution:
      p(j)=(-b+sqrt(b*b-4*a*c))/(2*a);  % Pressure
      dl(j)= rho0+(rho0/Bheta)*(p(j)-P0)-(rho0*Alpha*(temperatur(j)-T0)); %
Density of liquid
      dg(j) = p(j)/(Rair*temperatur(j));              % Density of gas
    else
     %Numerical Solution:
     [p(j),error]=itsolver(po(j),qv(j,1),qv(j,2)); % Pressure
     dl(j)=rholiq(p(j),temperatur(j)); % Density of liquid
     dg(j)=rogas(p(j),temperatur(j)); % Density of gas

     % Incase a numerical solution is not found, the program will write out
"error":
     if error > 0
        error
     end
    end


 % Find the phase volume fractions based on new conservative variables and
 % updated densities.

     eg(j)= qv(j,2)/(dg(j));
     ev(j)=1-eg(j);

      % THE CODE BELOW WAS ADDED TO HELP ON STABILITY WHEN
 % LOWERING MUDLEVEL IN RISER. FORCE THE PRESSURE TO BE 1 BAR
 % WHEN ONLY GAS IS PRESENT IN RISER. WE MAY HAVE TO ADJUST THE TEST
 % CONDITIONS SINCE IT IS UNCERTAIN HOW IT AFFECTS THE MUD LEVEL
 % CHANGES IN RISER.
```

```
% %   if (time < 1000)
% %       %   if (eg(j)>0.999999)
% %           if (eg(j)>0.99)
% %       %     if (p(j)< 100000)
% %             p(j)=100000;
% %             dg(j)= rogas(p(j));
% %             dl(j)= rholiq(p(j));
% %             eg(j)= 1.0;
% %             el(j)= 0.0;
% %             qv(j,1)=dl(j)*el(j);
% %             qv(j,2)=dg(j)*eg(j);
% %       %    end
% %         end
% %
% %   end

 if (p(j)<100000)
   p(j)=100000;
   dg(j)= rogas(p(j),temperatur(j));
   dl(j)= rholiq(p(j),temperatur(j));
   eg(j)=1-qv(j,1)/dl(j);
   el(j)=1-eg(j);
   qv(j,1)=dl(j)*el(j);
   qv(j,2)=dg(j)*eg(j);
 end

%     Reset average conservative varibles in cells with area changes inside.

      qv(j,1)=qv(j,1)*(areal(j)+arear(j))/2.0;
      qv(j,2)=qv(j,2)*(areal(j)+arear(j))/2.0;


%     The section below is used to find the primitive variables vg,vl
%     (phase velocities) based on the updated conservative variable q3 and
%     the slip relation.


% Part where we interpolate in the slip parameters to avoid a
% singularities when approaching one phase gas flow.
% In the transition to one-phase gas flow, we need to
% have a smooth transition to no-slip conditions.

      xint = (eg(j)-0.75)/0.25;
      k0 = k;
      s0 = s;
      if ((eg(j)>=0.75) & (eg(j)<=1.0))
        k0 =1.0*xint+k*(1-xint);
        s0 = 0.0*xint+s*(1-xint);
      end

      if (eg(j)>=0.999999)
        k1 = 1.0;
        s1 = 0.0;
```

```matlab
      else
        k1 = (1-k0*eg(j))/(1-eg(j));
        s1 = -1.0*s0*eg(j)/(1-eg(j));
      end




%     Below we operate with gas vg and liquid vl velcoities specified
%     both in the right part and left part inside a box. (since we have
%     area changes inside a box these can be different. vgl is gas velocity
%     to the left of the disconinuity. vgr is gas velocity to the right of
%     the discontinuity.
%

      help1 = dl(j)*ev(j)*k1+dg(j)*eg(j)*k0;
      help2 = dl(j)*ev(j)*s1+dg(j)*eg(j)*s0;

      vmixhelpl = (qv(j,3)/areal(j)-help2)/help1;
      vgl(j)=k0*vmixhelpl+s0;
      vll(j)=k1*vmixhelpl+s1;

      vmixhelpr = (qv(j,3)/arear(j)-help2)/help1;
      vgr(j)=k0*vmixhelpr+s0;
      vlr(j)=k1*vmixhelpr+s1;

%  Averaging velocities.

      vl(j)= 0.5*(vll(j)+vlr(j));
      vg(j)= 0.5*(vgl(j)+vgr(j));

    end




% Old values are now set equal to new values in order to prepare
% computation of next time level.
    for j = 1:nobox
     po(j)=p(j);
     dlo(j)=dl(j);   %Liquid density
     dgo(j)=dg(j);   %Gas density
     vlo(j)=vl(j);   %Liquid velocity
     vgo(j)=vg(j);   %Gas velocity
     ego(j)=eg(j);   %Gas fration
     evo(j)=ev(j);   %Liquid fraction.

     vlor(j)=vlr(j);
     vlol(j)=vll(j);
     vgor(j)=vgr(j);
     vgol(j)=vgl(j);

      for m =1:3
       qvo(j,m)=qv(j,m);
      end
    end
```

```
% Section where we save some timedependent variables in arrays.
% e.g. the bottomhole pressure. They will be saved for certain
% timeintervalls defined in the start of the program in order to ensure
% that the arrays do not get to long!

  if (counter>=nostepsbeforesavingtimedata)
    printcounter=printcounter+1;
    time
    pbot(printcounter)= p(1);
    pchoke(printcounter)=p(nobox);
    pcasingshoe(printcounter)=p(25); %NB THIS MUST BE DEFINED IN CORRECT BOX

liquidmassrateout(printcounter)=dl(nobox)*ev(nobox)*vl(nobox)*arear(nobox);
    %if (liquidmassrateout(printcounter))<0
       %liquidmassrateout(printcounter)=0;
    %end
    %liquidmassrateout1(printcounter)=suckrate;

%liquidmassrateout2(printcounter)=liquidmassrateout(printcounter)+suckrate;
    gasmassrateout(printcounter)=dg(nobox)*eg(nobox)*vg(nobox)*arear(nobox);
    timeplot(printcounter)=time;
    sumfriclossgrad(printcounter)= friclossgrad(1);
    sumhyd(printcounter)= densmix(1);
    counter = 0;



  end
end

% end of stepping forward in time.




% Printing of resultssection


countsteps % Marks number of simulation steps.



% Plot commands for variables vs time. The commands can also
% be copied to command screen where program is run for plotting other
% variables.

plot(timeplot,pbot/100000)
%plot(timeplot,pchoke/100000)
%plot(timeplot,liquidmassrateout)
%plot(timeplot,gasmassrateout)
%plot(vg)

%Plot commands for variables vs depth/Only the last simulated
%values/endtime is visualised
```

```
%plot(vl,x);
%plot(vg,x);
%plot(eg,x);
%plot(p,x);
%plot(dl,x);
%plot(dg,x);

 The same functions are used as in appendix A
```

Appendix C ExtensionAUSMV

Extension of the AUSMV Scheme for Simulation of Automated Drilling Hydraulics Laboratory

Introduction

The multiphase transient model is formulated under isothermal conditions. Therefore, the effect of wellbore temperature is neglected. However, the assumption is not valid for real drilling conditions, where fluid properties such as density and viscosity vary with temperature and pressure. Before the model should be used simulate a real drilling operation, there is a need to include temperature effects.

Here, some of the relevant changes to the scheme will be highlighted. The two-phase circulating system consists of water and nitrogen.

Wellbore temperature

A linear model is used for temperature modeling. It is assumed that the wellbore temperature increases linearly with depth.

Eq. 1(Kårstad and Aadnøy 1998) will be used for temperature calculation at any depth.

$$T = G_T z + T_{sf} \tag{1}$$

where T = wellbore temperature (K), G_T = geothermal gradient (K/m), z = vertical depth (m), T_{sf} = surface temperature.

Fluid densities

Simplified density models are used for both liquid and gas in the AUSMV formulation. These models are only pressure dependent.

At any depth, however, effective mud density depends on the variations in pressure and temperature.

Liquid density. Because changes in the density due to pressure and temperature are small, a linearized model—based on the Equation of State—can be used to estimate the liquid density.

The Eq. 2 (Stamnes 2011) gives the linearized Equation of State for the liquid density.

$$\rho_l = \rho_0 + \frac{\rho_0}{\beta}(p - p_0) - \rho_0 \alpha (T - T_0) \tag{2}$$

where ρ_0, p_0, and T_0 are the reference point density, pressure and temperature respectively, β is the bulk modulus of the liquid (which is the reciprocal of the compressibility of the liquid), and α is the volumetric thermal expansion coefficient.

For most drilling fluids, Eq. 2 is quite accurate for pressures in the range of 0–500 bar and temperatures, 0 – 200 °C (Stamnes 2011).

Table 1 gives the bulk modulus and volumetric thermal expansion coefficient for water at the reference point.

TABLE 1—BULK MODULUS AND VOLUMETRIC THERMAL EXPANSION COEFFICIENT FOR WATER				
ρ_0 (kg/m^3)	p_0 (Pa)	T_0 (°C)	β (Pa)	α (K^{-1})
1000	100000	20	2.2×10^9	0.000207

Gas density. For the gas density, a simple equation can be used, based on the Ideal Gas Law given by Eq. 3.

$$pV = mRT \tag{3}$$

where p = pressure, V = volume, m = mass, T = temperature, R = individual gas constant.

Eq. 3 can be written as:

$$\rho_g = \frac{p}{RT} \tag{4}$$

The above equation gives the gas density as function of pressure and temperature. For Nitrogen, R = 296.8 J/kg.K. The individual gas constant for air, R_{air} = 286.9 J/Kg.K.

Fluid viscosities

Viscosity is an important to internal property of drilling fluids. It is a measure of a fluid's resistance to flow. The viscosities of liquid and gas are affected by downhole temperature variation. It is also found to depend on pressure. However, this quantity has been specified a constant fluid property in the AUSMV scheme. Here, the fluid viscosities will be given in terms of pressure only.

Liquid Viscosity. Water has a viscosity of 1 cp at 20 °C and 0.890 cp at 20 °C. Thus, its viscosity decreases with temperature.
Eq. 7 gives the dynamic viscosity of water as a function temperature. The equation is based on the exponential model for temperature-dependence of shear viscosity (Eq. 8), first proposed by Reynolds (Reynolds 1886).

The expression can predict the viscosity of water at temperatures in the rage of 0–370 °C, with 2.5% error (Al-Shemmeri 2012). The equation is based on the exponential model for temperature-dependence of shear viscosity, first proposed by Reynolds (Reynolds 1886).

$$\mu_w = \mu_{w0} 10^{(247.8/(T-140))} \tag{5}$$

$$\mu_0 = \mu_0^{-C(T-T_0)} \tag{6}$$

where $\mu_{w0} = 2.414 \times 10^{-5}$ Pa.s (reference water viscosity), T = temperature (K), μ_0 = reference viscosity at reference temperature T_0.

Gas viscosity. The viscosity of gas can be calculated with Eq. 6 (Sutherland 1893). The equation, often called Sutherland's law, expresses the relationship between the viscosity, μ, and temperature, T, of an ideal gas. The formula is valid over a wide range of temperatures (0–555 K), with minimum errors due to pressures.

$$\mu = \mu_0 \left(\frac{T}{T_0}\right)^{3/2} \frac{T_0 + C}{T + C} \Rightarrow \mu = \eta \frac{T^{3/2}}{T + C}, where$$

$$\eta = \frac{\mu_0(T_0 + C)}{T_0^{3/2}}$$

(7)

where μ_0 = reference viscosity at reference temperature T_0, and C = Sutherland's constant for the specified gas.

Table 2 presents the Sutherland's law coefficients for air and Nitrogen.

TABLE 2—SUTHERLAND'S LAW COEFFICIENTS FOR AIR AND NITROGEN			
Gas	μ_0 (Pa.s)	T_0 (K)	C (K)
Air	1.827×10^{-5}	291.15	120
Nitrogen	1.781×10^{-5}	300.55	111

Well Inclination

The previous well configuration that was simulated with the AUSMV scheme was vertical. However, many well drilled in the Norwegian Continental Shelf are either horizontal or inclined. Therefore, the transient model should be transformed to include well inclination, θ. In this case, two types of depth—measured and true vertical depths—are considered. The inclination will also affect the gravity source term, such that the vertical component of acceleration due to gravity, g, become:

$$g_z = g \sin \theta$$

(8)

References

Al-Shemmeri, T. 2012. Engineering Fluid Mechanics. UK: Al-Shemmeri & Ventus Publishing ApS. pp. 17–18. ISBN: 978-87-403-0114-4.

Kårstad, E. and Aadnøy, B.S. 1998. Paper SPE 47806, presented at the 1998 IADC/SPE Asia Pacific Drilling Technology Conference, Jakarta, Indonesia, 7–9 September. http://dx.doi.org/10.2118/47806-MS

Reynolds, O. 1886. On the Theory of Lubrication and Its Application to Mr. Beauchamp Tower's Experiments, Including an Experimental Determination of the Viscosity of Olive Oil. *Phil. Trans. R. Soc. Lond.* vol. **177**: 175–234. http://dx.doi.org/10.1098/rstl.1886.0005.

Stamnes, Ø.N. 2011. Nonlinear Estimation with Applications to Drilling. PhD thesis, Norwegian University of Science and Technology (NTNU), Norway.

Appendix A: Well Pressure Calculation

Mass conservative variables (liquid and gas):

$$w_1 = \rho_l \alpha_l ; w_2 = \rho_g \alpha_g$$

$$w_1 = \rho_l \alpha_l = \rho_l (1 - \alpha_g) = \rho_l (1 - \frac{w_2}{\rho_g}) \tag{A-1}$$

Substituting Eq. 4 for gas density,

$$w_1 = \rho_l (1 - \frac{RTw_2}{p}) \Rightarrow pw_1 = \rho_l (p - w_2 RT) \tag{A-2}$$

Substituting Eq. 2 for liquid density,

$$pw_1 = (\rho_0 + \frac{\rho_0}{\beta}(p - p_0) - \rho_0 \alpha(T - T_0))(p - w_2 RT) \tag{A-3}$$

$$pw_1 = (\rho_0 - \frac{p_0 \rho_0}{\beta} + \frac{p \rho_0}{\beta} - \rho_0 \alpha(T - T_0))(p - w_2 RT)$$

$$pw_1 = (\rho_0 - \frac{p_0 \rho_0}{\beta} - \rho_0 \alpha(T - T_0) + \frac{p \rho_0}{\beta})(p - w_2 RT)$$

$$pw_1 = (x_1 + x_2 p)(p + x_3) \tag{A-4}$$

where:

$$x_1 = \rho_0 - \frac{p_0 \rho_0}{\beta} - \rho_0 \alpha(T - T_0) , \ x_2 = \frac{\rho_0}{\beta}, \text{ and } x_3 = -w_2 RT$$

$$pw_1 = (x_1 p + x_2 x_3 + x_2 p^2 + x_2 x_3 p)$$

$$x_2 p^2 + (x1 + x_2 x_3 - w1)p + x1 - x3 = 0$$

$$ap^2 + bp + c = 0 \tag{A-5}$$

$$p = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \tag{A-6}$$

where:

Where:

$a = x_1$, $b = x_1 + x_2 x_3 - w_1$, and $c = x_1 - x_3$

Appendix D Temperature and pressure effect on water density

engineeringtoolbox.com