Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

Rikke Mohn

Numerical Approximation of Temperature Distribution in Deep Water wells

Master's thesis in MTFYMA
Supervisor: Peter Lindqvist, Dag Vavik
June 2019

NTNU
Norwegian University of
Science and Technology

Numerical Approximation of Temperature Distribution in Deep Water Wells

Rikke Mohn [1]

January 2019 - June 2019

MASTER THESIS

Department of Mathematics
Norwegian University of Science and Technology

[1]Supervisors: Peter Lindqvist, Dag Vavik

Preface

This is a Master thesis in mathematics at NTNU as part of the study program Industrial Mathematics. It was carried out during the spring semester of 2019.

The thesis was written in cooperation with Future Well Control, and is intended to be the foundation of one part of a larger system. This system is intended to provide a detailed surveillance of pressure and temperature when drilling deep water wells, in order to further ensure the safety of those working.

The readers of this project are assumed to have a basic background in mathematics at university level. Some background in physics and numerical mathematics would be helpful, as well as basic knowledge of deep water wells. A few figures have been included to aid the understanding of the reader.

Trondheim, June 1, 2019

Rikke Mohn

Abstract

Today's methods of Early Kick Detection tools for drilling deep water wells are based on flow-meters or active volume control. That is, they measure net gain or net loss of circulation at any given point in time. They will therefore not necessarily detect influx of formation fluids due to swap-out, or/and the gas consumption and volume reduction from formed hydrates. This might again cause them to draw faulty conclusions when abnormalities occur. Creation of hydrates is dependent on temperature and pressure conditions, and the dissociation of hydrates is an endothermic reaction, that is, the process releases energy. An accurate method for predicting the temperature when drilling will therefore allow us to compare the predicted temperature with the real time temperature, and so allow us to better predict why certain abnormalities occur, and react accordingly. This could possibly increase security at oil platforms, and is indeed what motivated this thesis.

The temperature distribution has been approximated at several positions based on equations developed using the necessary balance of energy. The system was split into two parts, the slightly simpler system above the seabed, and the one below. The energy balance equations have then been discretized, and approximated using numerical mathematics. Specifically I used the Finite Difference Methods two point Forward Euler and three point Central Differences in space, and two point Backward Euler in time. A system of linear equations was then developed and solved using LU-decomposition. That is, the matrix A in a system $Ax = b$, where X, B are matrices and B is known, is factorized into a lower triangular matrix L, and an upper triangular matrix U. Thus allowing us to solve two separate linear systems of equations. The code was written in C++, and the method was chosen specifically to allow for acceleration using parallel programming by use of a graphic processing unit. Indeed, LU-reduction is often used in the context of parallel computing with the purpose of comparing processing speeds for different computers.

Both the code for the system above the seabed and the system below the seabed were developed. However, because of time constraint due to errors in the data and a few errors in the original equations given to me, only the results of the upper system has been included in this thesis. The upper system gives reasonable results that stabilize after only half an hour, and so suggests that the method and code works. There is however a discrepancy of negative temperatures at low altitudes that I have not been able to explain.

The necessary processing speed for each iteration is ≤ 1 Hz for the program to be useful in comparison with real time data. As of yet the upper part of the program spends only 6 seconds per iteration, while the lower one spends 3 minutes, and so

is by far the most time consuming part. Within the code the most time consuming process is the decomposition of A into LU, and so the decomposition of the lower matrix should be the focus of a future acceleration using parallel programming. If necessary, the number of vertical nodes in the lower system will have to be reduced.

Acknowledgement

I would like to thank my supervisor Peter Lindqvist, for his help during the writing of this masters. I would particularly like to thank him for his support and patience.

I would also like to thank my second supervisor from Future Well Control, Dag Vavik, who has made himself available to me frequently. I hope that this thesis can be of some help to you in the future development of your company.

Lastly I would like to thank Lucas Sevillano for allowing me to base this thesis partly on his article, and for humoring me with the questions and quandaries I had regarding his work.

On a personal note, I would like to thank my family for all their support during this longer than expected process. I would particularly like to thank Arne-Christina Mohn, Thea Mohn and Seba Charaf for proof-reading this thesis, and the latter ones for throwing a wonderful wedding that left me with no tears left to cry in the last week of my masters.

R.M

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Today's surveillance of conditions in deep water wells are based mostly on volume of drilling mud being pumped into the pipes versus the volume of formation and drilling fluid returned. This method will not always give a complete picture of how the drilling is going, and might result in the wrong reaction to certain abnormalities.

An accurate method for predicting the temperature when drilling deep oil wells would allow us to compare the predicted temperature with the real time temperature, and so perhaps allow us to better predict why abnormalities occur, and react accordingly. Thus such a method will be developed using Finite Difference Methods on energy balancing equations developed at certain positions in the well. A temperature distribution model has already been developed by NTNU (Sevillano u. a., 2017), but is too slow and owned by NTNU. The new code should therefore be optimized with respect to coding speed and number of grid points. One possible aid to increase the computing speed is by use of a graphic processing unit (GPU). The goal for this research is to be able to achieve a computational speed for each time step which is faster than the 1Hz, including time for data transfer from real-time sensors and back to a human machine interface (HMI). Once optimized, the model should then be compared with Drillbench. This work is too extensive for a Masters thesis, and so this thesis should be considered as a basis for this work and as a resource for the completion of these goals.

Chapter 2

Background

An oil well is created by drilling a hole of diameter up to 1 m into the earth with a rotating drill string on a drilling rig. When the hole has been drilled, a casing (section of steel pipe), that is slightly smaller in diameter is placed in the hole. This process is repeated with consequently deeper depths and smaller diameters. An example of such a deep water well can be seen in Figure 2.1.

As can be seen from Figure 2.1, the bottom part of the well consists partly of an open hole. To prevent formation fluids and gases from entering into the well bore, a drilling fluid often referred to as drilling mud, provides hydrostatic pressure in the well. The pressure p exerted by the mud is highly dependent on its density ρ, as evident from the equation of hydrostatic pressure:

$$p = \rho g h,$$

where g is the gravitational acceleration constant, and h the depth (White, 2008). This drilling mud is a heavy, viscous fluid mixture that flows down through the center of the well, known as the drill string, through the drill bit at the bottom of the well, into the annulus that surrounds the drill string, and upward through the annulus to the top of the well. The drilling mud thus carries drill cuttings to the surface, while simultaneously lubricating and cooling the drill bit. The mud also suspends the drill cutting while drilling is paused and when the drilling assembly is brought in and out of the hole. The movement of the mud is demonstrated by the arrows in the center of the figure.

In deep water wells the most common types of drilling muds are Water-Based muds (WBM) and Oil-Based muds (OBM), where WBM is the most common of the two. Each of these fluids are well suited for High Pressure High Temperature purposes. WBM has a large potential for cooling down the wellbore during circulation, while its relatively low thermal conductivity leaves the cement acting as an

insulating layer against the formation. Additionally, the seawater surrounding the Riser, as shown in Figure 2.1, will gradually remove heat from the drilling fluid, and thus the temperature of the casing wall will be relatively low.

When drilling deep water wells one will drill through different materials, and so naturally the pressure within the drilled rock may vary. If for instance one suddenly drills into a large gas filled gap, the pressure could fall substantially, causing mud to flow into the hole. This is referred to as lost circulation. If the opposite occurs, that is, the pressure within the drilled rock is higher than the hydrostatic pressure from the mud acting on the borehole or rock face, then the greater formation pressure has a tendency to force formation fluid into the wellbore. This forced fluid flow is called a Kick. To minimize both these occurrences one therefore often refers to a so called Drilling Margin, which denotes the difference between the maximum pore pressure and the minimum fracture pressure.

Current Early Kick detection (EKD) tools are based on flow-meters or active volume control. That is, they measure net gain or net loss of circulation at any given point in time. According to the article by Vavik (Vavik u. a., 2017), these EKD tools do not sufficiently take into account cross flow, other forms of swap-out, loss of drilling fluid, and the gas consumption and volume reduction from formed hydrates. They will therefore not necessarily detect influx of formation fluids, thus increasing the chance of faulty conclusions being drawn. In other words, influx of hydrocarbon gas to the wellbore may occur without any observed gain at the surface if one simultaneously has cross flow, hydrates forming, or partial or total loss of circulation.

2.1 Hydrates

As mentioned above, according to the article by Vavik (Vavik u. a., 2017), one of the dangers of only considering net gain and net circulation loss in deep water wells is the undetected formation of gas hydrates.

Gas hydrates are crystalline components that occur when water forms a cage-like structure around smaller guest molecules. These lattice structures are incredibly strong, and so may allow a large number of guest molecules to be trapped in a small volume high pressure environment inside the cage-like structure.

Gas hydrates may form when natural gas and the water typically in drilling mud are mixed under high pressure p, and the mud has sufficiently low temperature T to cool down the gas to below the temperature required to form hydrates.

When gas hydrates are formed, they may stay stable in large parts of the wellbore casing and the riser annulus, and so may be transported a long distance to the upper

part of the riser. If the hydrates melt deep down in the wellbore, then the high pressure will ensure that the released gas only expands slightly. If the hydrates melt in the upper part of the riser however, the cage-like structure will have prevented the trapped gas from expanding as pressure falls, and so once released, the gas will expand very rapidly. Dissociation of hydrates is an endothermic process, meaning the process requires heat, and so the process is also dependent on how fast the surrounding fluid or solid material is heated up again.

Another study by vavik (Vavik u. a., 2016), suggests that the temperature conditions for forming hydrates are likely present in the well head, the Blow out preventer (BOP) and the drilling riser, as well as deep down inside the wellbore casing, where as mentioned above, the temperature will be relatively low. It also states that because gas hydrates consume gas when formed, the consumed gas will be replaced with more gas from reservoirs or alternatively with drilling fluids. This phenomenon may cause a reduction in casing shut-in pressure (CSIP), and so may mistakenly be interpreted as loss in fluid. This may have safety implications.

2.2 Drilling Advanced Influx Detection

Drilling Advanced Influx Detection (DrillAid^{TM}) is an early kick detection (EKD) tool that detects an influx in a wellbore with pressure transmitters, arranged in a fixed vertical distance h. These pressures can thus be used to calculate the density $\rho_{real\,n}$ of a return flow, and compare it to a calculated expected density $\rho_{calc\,n}$. Additionally, the method predicts the probability of gas hydrates forming in the wellbore, wellhead or riser annulus by measuring or calculating the temperatures in an annulus section.

In order to achieve this there is need for an accurate method for prediciting how temperature distributions about the wellbore change with time. Specifically a computer program that numerically calculates the expected temperature distribution in a deep water well. One such model has already been developed by Sevillano (Sevillano u. a., 2017). This model has been programmed using the programming language Matlab, and is detailed as it has many nodes in the formation. It is however quite slow, and so cannot be used directly. The model is intended to be used in combination with real time data, that will be updated every second. That is, each iteration should be faster than 1Hz, for it to be used as intended.

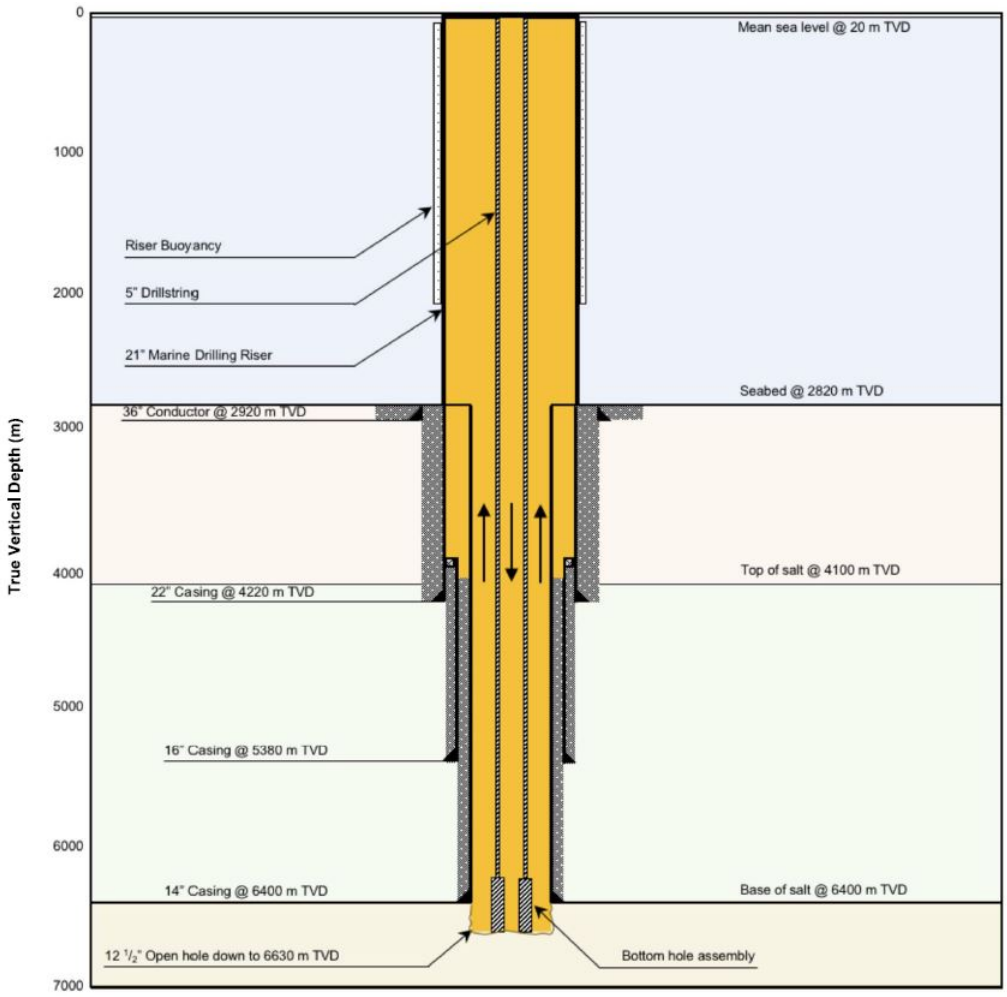


Figure 2.1: Typical deep water well and casing program. Taken from (Vavik u. a., 2017).

Chapter 3

Transient Temperature Distribution

The current trend of drilling deeper and costlier wells requires an increasingly accurate method of measuring the variables involved. This due to the increase in probability for the occurrence in previously not accounted for factors, as stated by Vavik (Vavik u. a., 2017). Among these variables is temperature. An accurate mean of estimating the temperature distribution with time would in fact have many applications. Among these are cementing program design, Injection and production operations, and well design in permafrost regions.

In order to develop a transient temperature distribution model, one can utilize the energy balance equations for the system. To do so we consider the three regions of the flow in the wellbore. In the first region the fluid enters the drill string with a known temperature, and flows downwards through it. The temperature change in this region is thus determined by the rate of thermal convection down the fluid column, and the rate of convective heat transfer radially between the fluid, pipe wall, and annulus. Thermal convection is in this case the transfer of heat cause by the tendency of hotter and thus less dense material to rise, while colder, denser materials sink due to gravity. In the second region, that is at the bottom of the wellbore, the fluid flows from the drill string, through the drill bit, into the annulus. Change in temperature in this region should be determined by vertical and radial heat conduction within the pipe wall. In the last region the fluid flows back up, through the annulus. In this region the temperature change is again dependent on the rate of heat convection up the annulus and the radial convection between the annulus fluid, the drill pipe wall, and the fluid within the drill pipe. Additionally it will be dependent on the rate of radial convection through the formations Kays u. a.

(2005).

Because transient heat transfer is considered, the time of circulation has an important effect on the temperature of the fluid, as has the heat generation within the three flow regions due to frictional forces and the rotational energy of the drill string and the drill bit.

Furthermore, before stating the energy balance equations, certain assumptions has been made. Flow in the wellbore is assumed to be both fully developed and to be turbulent in the drill pipe and drill bit, while laminar in the annulus. It is assumed that both the radial temperature gradient and the heat generation by viscous dissipation within the fluid may be neglected. It is also assumed that the heat transfer within the drilling fluid is by axial convection, and that conduction may be neglected everywhere except where the fluid is still. Lastly it is assumed that properties such as density, thermal conductivity, and specific heat capacity are independent of temperature. The thermal conductivity k is defined as the ability of a material to transfer heat, and is in fact highly dependent on temperature. The same applies to specific heat capacity c_p which is the ability of formations to store heat. For simplicity however, both of these variables are as stated assumed to be constant. Other variables, such as the convection coefficient h and the energy source term Q, are assumed to be constant within certain height intervals.

With these assumptions, then as stated in the article by Marshall and Bentsen (Marshall und Bentsen, 1982), and as developed in the article by (Sevillano u. a., 2017), we have seven equations to consider and discretize.

3.1 Energy Balancing Equations

The energy balances within the systems can be described by seven partial differential equations. In these equations there are several constant and variables to be considered apart from the temperature T. Table 3.1 demonstrates an overview of each of these and their units, and Table 3.2 explains the different subscripts. There are two main areas in which I would like to estimate the temperature distribution; the riser and the well, separated by a mud-line. This is illustrated in Figure 4.1. Additionally, there will be an area right above the seabed, below the riser, where the Blowout Preventer (BOP), Wellhead Housing, and Conductor Housing will be located. This area might demand a separate system of equations as well, but primarily I will consider two sets of equations, leading to two systems of equations. The first three equations are identical for both systems, while the next equations are specific for each system. I begin by presenting the first three equations.

Variable	Notation	Unit
Gravitational acceleration	g	$\mathrm{m\,s^{-2}}$
Specific Heat Capacity	c_p	$\mathrm{J\,Kg^{-1}\,K^{-1}}$
Thermal Conductivity	k	$\mathrm{W\,m^{-1}\,K^{-1}}$
Inclination angle	α	radians
Density	ρ	$\mathrm{Kg\,m^{-3}}$
Volumetric Flow Rate	q	$\mathrm{m^3\,s^{-1}}$
Radius	r	m
Convective Heat Transfer Coefficient	h	$\mathrm{W\,m^{-2}\,K^{-1}}$
Heat generated in a control volume by the different energy sources present	Q	$\mathrm{W\,m^{-1}}$
Time	t	s
Vertical Position	z	m

Table 3.1: List of variables with their respective units

Subscript	Explanation
a	Annulus A
ca	Casing annulus
cw	Casing wall
df	Drilling fluid
ds	Drill string
db	Drill bit
rw	Marine riser wall
rf	Riser Floaters
i	Inner
o	Outer
j	$\mathrm{j^{th}}$ well and riser system component, and radial position of element in mesh grid

Table 3.2: List of subscripts and their meaning

Fluid inside the drill string

The center-most area of the well is the mud within the drill string. The first term in this equation accounts for the fluid's potential energy, where the inclination angle α allows for the use of measured depth instead of true vertical depth. The second term accounts for the fluid's variation of enthalpy as it crosses the control volume

CV, while the third term describes the radial convective heat transfer between the drill string wall and the fluid within. Lastly, the two right-hand terms represent the accumulation of energy within the drill string and the energy source, respectively.

$$q\,\rho_{df_{ds}} g \sin\alpha + \rho_{df_{ds}}\, q\, c_{p_{df}} \frac{\partial T}{\partial z} + 2\pi\, r_{ds_i} h_{ds_i} \left(T_{ds} - T_{df_{ds}}\right) = \rho_{df_{ds}} c_{p_{df}} \pi r_{ds_i}^2 \frac{\partial T}{\partial t} - Q_{ds} \quad (3.1)$$

This equation is based on the equation in both (Marshall und Bentsen, 1982) and (Sevillano u. a., 2017), but there is a discrepancy. Since the latter is itself based on the first, the author suspect an error has been transferred from one to the other. That is, the first two terms should be positive, not negative as they are shown to be in both these papers.

Drill string wall

The natural next area to consider is the drill string itself. The first term of this equation accounts for the vertical heat conduction in the drill string, while the two center terms account for heat exchanged by convection with fluid flowing in its interior and on the surrounding annulus. Here $\pi\left(r_{ds_o}^2 - r_{ds_i}^2\right)$ is clearly the cross sectional area of the drill string wall, where π has been cancelled out. Unlike the previous equation, this equation has been divided by the cross sectional area. Like before, the right-hand term below accounts for the accumulation of energy along the drill string.

$$k_{ds}\frac{\partial^2 T}{(\partial z)^2} + \frac{2 r_{ds_i} h_{ds_i}}{r_{ds_o}^2 - r_{ds_i}^2}\left(T_{df_{ds}} - T_{ds}\right) + \frac{2 r_{ds_o} h_{ds_o}}{r_{ds_o}^2 - r_{ds_i}^2}\left(T_{df_a} - T_{ds}\right) = \rho_{ds} c_{p_{ds}} \frac{\partial T}{\partial t} \quad (3.2)$$

This equation is formulated differently in the two papers cited above. In the latter the third term is negative, that is the temperature of flowing annulus is subtracted from the temperature of the drill string wall. I disagree with this change because it would lead to an addition of energy when the temperature in the drill string wall is in fact higher than that in the flowing annulus. This is the opposite of one would expect, as a lower temperature should subtract from the overall energy, not add to it. This assumed error is repeated in the fourth term in the next equation, and so in analogy with the argument above this term has again been left positive. That is, it is like that of the paper from 1982 (Marshall und Bentsen, 1982).

The upward flow of fluid in the annulus

The next area is the flowing annulus, where the mud and drill cuttings are carried up to the surface. Three of the left-hand terms are very similar to (3.1), while the fourth term accounts for the radial convective heat transfer between the drilling fluid

and either the riser, the casing, or the formation, depending on the vertical position in the well. In the equation below, the subscript cw has been used, and so must be changed for what applies. Energy accumulation and generation are again accounted for by the two right-hand terms.

$$\rho_{df}\, q\, c_{p_{df}} \frac{\partial T}{\partial z} + q\, \rho_{df} g \sin\alpha + 2\pi\, r_{ds_o} h_{ds_o} \left(T_{ds} - T_{df_a}\right) + 2\pi\, r_{cw_i} h_{cw_i} \left(T_{cw} - T_{df_a}\right) = \pi \rho_{df} c_{p_{df}} \left(r_{cw_i}^2 - r_{ds_o}^2\right) \frac{\partial T}{\partial t} - Q_a \quad (3.3)$$

Comparing this equation with equation (3.1) further establishes that there is most likely an error in the first equation. Otherwise the analogy between the two equations would be illogical, as the first two terms of each equation would be of opposite signs.

Riser and Floaters

The next equations are specific for the system above the mud-line. Here we must consider two sets of equations. The first applies to where the riser wall is in direct contact with either the air or the sea. The riser will then exchange heat with the sea or air through both natural and forced convection. The first term in this equation describes the axial heat conduction in the riser wall, while the second and third term account for the radial heat transfer by convection between the riser wall and the fluid it is in contact with. The last term is as before.

$$k_{rw} \frac{\partial^2 T}{(\partial z)^2} + \frac{2 r_{rw_i} h_{rw_i}}{r_{rw_o}^2 - r_{rw_i}^2} \left(T_{df_a} - T_{rw}\right) + \frac{2 r_{rw_o} h_{rw_o}}{r_{rw_o}^2 - r_{rw_i}^2} \left(T_{sea} - T_{rw}\right) = \rho_{rw} c_{p_{rw}} \frac{\partial T}{\partial t} \quad (3.4)$$

The variables in the second and third term have been mislabelled in Sevillano u. a. (2017), and so should be as stated here, in analogy with the terms accounting for radial heat transfer by convection in the previous equations. The temperature of the sea or air is again assumed to be undisturbed by the well due to the constant circulation of both.

When the riser wall is in contact with floaters the equation for the riser wall is similar to the above, but the third term has been changed so to account for the heat transfer by convection between the riser wall and the riser floaters.

$$\frac{1}{r_{rw_o}^2 - r_{rw_i}^2} \cdot \frac{2 k_{rw} k_{rf}}{k_{rw} \ln\left(\frac{r_{rf_o} - r_{rf_i}}{2 r_{rw_o}}\right) - k_{rf} \ln\left(\frac{r_{rw_o} - r_{rw_i}}{2 r_{rw_o}}\right)} \left(T_{rf} - T_{rw}\right) + k_{rw} \frac{\partial^2 T}{(\partial z)^2} + \frac{2 r_{rw_i} h_{rw_i}}{r_{rw_o}^2 - r_{rw_i}^2} \left(T_{df_a} - T_{rw}\right) = \rho_{rw} c_{p_{rw}} \frac{\partial T}{\partial t} \quad (3.5)$$

The equation for the riser floaters is then

$$\frac{1}{r_{rf_o}^2 - r_{rf_i}^2} \cdot \frac{2k_{rw}k_{rf}}{k_{rw} \ln\left(\frac{r_{rf_o} - r_{rf_i}}{2r_{rw_o}}\right) - k_{rf} \ln\left(\frac{r_{rw_o} - r_{rw_i}}{2r_{rw_o}}\right)} (T_{rw} - T_{rf}) + k_{rf} \frac{\partial^2 T}{(\partial z)^2} + \frac{2r_{rf_o}h_{rf_o}}{r_{rf_o}^2 - r_{rf_i}^2} (T_{sea} - T_{rf}) = \rho_{rf} c_{p\_rf} \frac{\partial T}{\partial t} \quad (3.6)$$

Casing string wall and subsequent layers

The next two equations describe the subsequent layers specific to the well, below the mud-line. For the Casing string wall and the subsequent layers there are again several scenarios depending on the vertical position in the well. An annulus filled with mud can be represented by equation (3.3). A Casing string surrounded by two annuli filled with mud can be described by equation (3.2). The next scenario occurs once for each vertical position. That is, a casing string with an annulus filled with mud on its left side, and an annulus with cement on its right side. This can be represented by the equation described in equation (3.5). The subsequent layers can then be represented by the following equation, where the two left-hand terms account for the radial heat transfer by conduction.

$$\frac{1}{r_{j_o}^2 - r_{j_i}^2} \cdot \frac{2\,k_{j-1}k_j}{k_{j-1} \ln\left(\frac{r_{j_o} - r_{j_i}}{2r_{(j-1)_o}}\right) - k_j \ln\left(\frac{r_{(j-1)_o} - r_{(j-1)_i}}{2r_{(j-1)_o})}\right)} (T_{j-1} - T_j) + \frac{1}{r_{j_o}^2 - r_{j_i}^2} \cdot \frac{2\,k_j k_{j+1}}{k_j \ln\left(\frac{r_{(j+1)_o} - r_{(j+1)_i}}{2r_{j_o}}\right) - k_{j+1} \ln\left(\frac{r_{j_o} - r_{j_i}}{r_{j_o}}\right)} (T_{j+1} - T_j) + k_j \frac{\partial^2 T}{(\partial z)^2} = \rho_j c_{p_j} \frac{\partial T}{\partial t} \quad (3.7)$$

The second and fourth term in this equation has also been mislabelled in Sevillano u. a. (2017), and so should again be as stated here, in analogy with previous equations.

Chapter 4

The Numerical Model

4.1 Discretization of The Energy Balancing Equations

I intend to numerically approximate the temperature distribution using the equations above. To do so I discretize the temperatures and approximate the differential terms in the equations using finite difference methods. A way of doing so is by Taylor Series Methods. Its principle is to represent the solution of a differential equation locally by a few terms of its Taylor series (Cheney und Kincaid, 2013). From Taylor's formula we have that the series expansion of a formula around a point $x = a$, where $x - a = \Delta x$ is as follows

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + ...$$
$$f(a+\Delta x) = f(a) + \frac{f'(a)}{1!}\Delta x + \mathcal{O}\left((\Delta x)^2\right),$$

(Suli und Mayers, 2011). Or rewritten as

$$f'(a) = \frac{f(a+\Delta x) - f(a)}{\Delta x} + \mathcal{O}(\Delta x).$$

This is commonly referred to as Forward Euler, and is the most basic explicit method for solving ordinary differential equations. Using the same method, but with $x - a = -\Delta x$, we have the approximation commonly referred to as Backward Euler:

$$f'(a) = \frac{f(a-\Delta x) - f(a)}{-\Delta x} + \mathcal{O}(\Delta x) = \frac{f(a) - f(a-\Delta x)}{\Delta x} + \mathcal{O}(\Delta x),$$

Combing the two series expansions above we see that

$$f(a+\Delta x)+f(a-\Delta x)=2f(a)+f''(a)(\Delta x)^2+\mathcal{O}\left((\Delta x)^3\right)$$

and so

$$f''(a)=\frac{f(a+\Delta x)-2f(a)+f(a-\Delta x)}{(\Delta x)^2}+\mathcal{O}\left(\Delta x\right).$$

In other words we may approximate the first order partial derivatives by two-point forward and backward difference approximations, while the second order derivatives can be represented by three point central differences. In our case the partial derivatives in time have been approximated by two-point backward difference approximations, while the partial derivatives in space have been approximated by two point forward difference approximation and three point central differences (Thomas, 2013). Inserting these approximations as well as the constants defined in table 4.1, leaves the following seven equations.

Notation	Expression
A_l	$\rho_l\, c_{p_l}$
B_l	$r_{l_o}^2-r_{l_i}^2=r_{l_o}^2-r_{(l-1)_o}^2$
D_l	$\frac{2\,k_{l-1}k_l}{k_{l-1}\ln(r_{l_m})+(k_l-k_{l-1})\ln(r_{(l-1)_o})-k_l\ln(r_{(l-1)_m})}$
E_l	$2r_l h_l$
F_l	$q\,\rho_l\,g\sin\alpha$

Table 4.1: Constants defined in order to simplify the energy balance equations.

Fluid inside the drill string

$$q\,A_{df}\frac{T_{k+1,\,1}^n-T_{k,\,1}^n}{\Delta z_k}+\pi\,E_{ds_i}\left(T_{k,\,2}^n-T_{k,\,1}^n\right)=\pi r_{ds_i}^2A_{df}\frac{T_{k,\,1}^n-T_{k,\,1}^{n-1}}{\Delta t}-Q_{ds}-F$$

Drill string wall

$$k_{ds}\frac{T_{k-1,\,2}^n-2T_{k,\,2}^n+T_{k+1,\,2}^n}{(\Delta z_k)^2}+\frac{E_{ds_i}}{B_{ds}}\left(T_{k,\,1}^n-T_{k,\,2}^n\right)+\frac{E_{ds_o}}{B_{ds}}\left(T_{k,\,3}^n-T_{k,\,2}^n\right)=A_{ds}\frac{T_{k,\,2}^n-T_{k,\,2}^{n-1}}{\Delta t}$$

The upward flow of fluid in the annulus

$$q\,A_{df}\frac{T_{k+1,\,3}^n-T_{k,\,3}^n}{\Delta z_k}+\pi\,E_{ds_o}\left(T_{k,\,2}^n-T_{k,\,3}^n\right)+\pi\,E_{cw_i}\left(T_{k,\,4}^n-T_{k,\,3}^n\right)$$
$$=\pi A_{df}B_a\frac{T_{k,\,3}^n-T_{k,\,3}^{n-1}}{\Delta t}-Q_a-F$$

Riser wall in direct contact with water

$$k_{rw}\frac{T^n_{k-1,\,4}-2T^n_{k,\,4}+T^n_{k+1,\,4}}{(\Delta z_k)^2}+\frac{E_{rw_i}}{B_{rw}}\left(T^n_{k,\,3}-T^n_{k,\,4}\right)+\frac{E_{rw_o}}{B_{rw}}\left(T_{sea}-T^n_{k,\,4}\right)=A_{rw}\frac{T^n_{k,\,4}-T^{n-1}_{k,\,4}}{\Delta t}$$

Riser wall with floaters

$$\frac{D_{rf}}{B_{rw}}\left(T^n_{k,\,5}-T^n_{k,\,4}\right)+k_{rw}\frac{T^n_{k-1,\,4}-2T^n_{k,\,4}+T^n_{k+1,\,4}}{(\Delta z_k)^2}+\frac{E_{rw_i}}{B_{rw}}\left(T^n_{k,\,3}-T^n_{k,\,4}\right)=A_{rw}\frac{T^n_{k,\,4}-T^{n-1}_{k,\,4}}{\Delta t}$$

Floaters

$$\frac{D_{rf}}{B_{rf}}\left(T^n_{k,\,4}-T^n_{k,\,5}\right)+k_{rf}\frac{T^n_{k-1,\,5}-2T^n_{k,\,5}+T^n_{k+1,\,5}}{(\Delta z_k)^2}+\frac{E_{rf_o}}{B_{rf}}\left(T_{sea}-T^n_{k,\,5}\right)=A_{rf}\frac{T^n_{k,\,5}-T^{n-1}_{k,\,5}}{\Delta t}$$

Casing string wall and subsequent layers

$$k_j\frac{T^n_{k-1,\,j}-2T^n_{k,\,j}+T^n_{k+1,\,j}}{(\Delta z_k)^2}+\frac{D_j}{B_j}\left(T^n_{k,\,j-1}-T^n_{k,\,j}\right)+\frac{D_{j+1}}{B_j}\left(T^n_{k,\,j+1}-T^n_{k,\,j}\right)=A_j\frac{T^n_{k,\,j}-T^{n-1}_{k,\,j}}{\Delta t}$$

Using these discretized equations, we may define two systems of linear equations to be solved for each time step. To better visualize how these matrices would look like we rewrite our equations as follows.

Fluid inside the drill string

$$\begin{aligned}\left[\pi\,\Delta t\,\Delta z_k\,E_{ds_i}+q\,\Delta t\,A_{df}+\pi\,\Delta z_k\,r^2_{ds_i}A_{df}\right]T^n_{k,\,1}-\left[\pi\,\Delta t\,\Delta z_k\,E_{ds_i}\right]T^n_{k,\,2}\\ -\left[q\,\Delta t\,A_{df}\right]T^n_{k+1,\,1}=\left[\pi\,\Delta z_k\,r^2_{ds_i}A_{df}\right]T^{n-1}_{k,\,1}+\Delta t\,\Delta z_k\,(Q_{ds}+F)\end{aligned}$$

Drill string wall

$$\begin{aligned}&\left[k_{ds}\Delta t\,B_{ds}\right]T^n_{k-1,\,2}+\left[\Delta t\,(\Delta z_k)^2\,E_{ds_i}\right]T^n_{k,\,1}\\ &\quad-\left[\Delta t\,(\Delta z_k)^2\,E_{ds_i}+(\Delta z_k)^2\,A_{ds}B_{ds}+\Delta t\,(\Delta z_k)^2\,E_{ds_o}+2\Delta t\,k_{ds}B_{ds}\right]T^n_{k,\,2}\\ &\quad+\left[\Delta t\,(\Delta z_k)^2\,E_{ds_o}\right]T^n_{k,\,3}+\left[\Delta t\,k_{ds}B_{ds}\right]T^n_{k+1,\,2}=-\left[(\Delta z_k)^2\,A_{ds}B_{ds}\right]T^{n-1}_{k,\,2}\end{aligned}$$

The upward flow of fluid in the annulus

$$\begin{aligned}&\left[\pi\,\Delta t\,\Delta z_k\,E_{ds_o}\right]T^n_{k,\,2}-\left[\pi\,\Delta t\,\Delta z_k\,E_{ds_o}+q\,\Delta t\,A_{df}+\pi\,\Delta t\,\Delta z_k\,E_{cw_i}+\pi\Delta z_k\,A_{df}B_a\right]T^n_{k,\,3}\\ &\quad+\left[\pi\,\Delta t\,\Delta z_k\,E_{cw_i}\right]T^n_{k,\,4}+\left[q\,\Delta t\,A_{df}\right]T^n_{k+1,\,3}=-\left[\pi\Delta z_k\,A_{df}B_a\right]T^{n-1}_{k,\,3}-\Delta t\,\Delta z_k\,(Q_a+F)\end{aligned}$$

Riser wall in direct contact with water

$$\begin{aligned}&\left[\Delta t k_{rw} B_{rw}\right] T^n_{k-1,\,4} + \left[\Delta t (\Delta z_k)^2 E_{rw_i}\right] T^n_{k,\,3} \\ &\quad - \left[(\Delta z_k)^2 A_{rw} B_{rw} + 2\Delta t k_{rw} B_{rw} + \Delta t (\Delta z_k)^2 E_{rw_i} + \Delta t (\Delta z_k)^2 E_{rw_o}\right] T^n_{k,\,4} \\ &\quad + \left[\Delta t k_{rw} B_{rw}\right] T^n_{k+1,\,4} = -\left[(\Delta z_k)^2 A_{rw} B_{rw}\right] T^{n-1}_{k,\,4} - \Delta t (\Delta z_k)^2 E_{rw_o} T_{sea}\end{aligned}$$

Riser wall in contact with floaters

$$\begin{aligned}&\left[\Delta t k_{rw} B_{rw}\right] T^n_{k-1,\,4} + \left[\Delta t (\Delta z_k)^2 E_{rw_i}\right] T^n_{k,\,3} \\ &\quad - \left[\Delta t (\Delta z_k)^2 D_{rf} + 2\Delta t k_{rw} B_{rw} + \Delta t (\Delta z_k)^2 E_{rw_i} + (\Delta z_k)^2 A_{rw} B_{rw}\right] T^n_{k,\,4} \\ &\quad + \left[\Delta t (\Delta z_k)^2 D_{rf}\right] T^n_{k,\,5} + \left[\Delta t k_{rw} B_{rw}\right] T^n_{k+1,\,4} = -\left[(\Delta z_k)^2 A_{rw} B_{rw}\right] T^{n-1}_{k,\,4}\end{aligned}$$

Floaters

$$\begin{aligned}&\left[\Delta t k_{rf} B_{rf}\right] T^n_{k-1,\,5} + \left[\Delta t (\Delta z_k)^2 D_{rf}\right] T^n_{k,\,4} \\ &\quad - \left[\Delta t (\Delta z_k)^2 D_{rf} + 2\Delta t k_{rf} B_{rf} + \Delta t (\Delta z_k)^2 E_{rf_o} + (\Delta z_k)^2 A_{rf} B_{rf}\right] T^n_{k,\,5} \\ &\quad + \left[\Delta t k_{rf} B_{rf}\right] T^n_{k+1,\,5} = -\left[(\Delta z_k)^2 A_{rf} B_{rf}\right] T^{n-1}_{k,\,5} - \Delta t (\Delta z_k)^2 E_{rf_o} T_{sea}\end{aligned}$$

Casing string wall and subsequent layers

$$\begin{aligned}&\left[k_j \Delta t B_j\right] T^n_{k-1,\,j} + \left[D_j \Delta t (\Delta z_k)^2\right] T^n_{k,\,j-1} \\ &\quad - \left[2k_j \Delta t B_j + D_j \Delta t (\Delta z_k)^2 + D_{j+1} \Delta t (\Delta z_k)^2 + (\Delta z_k)^2 A_j B_j\right] T^n_{k,\,j} \\ &\quad \left[D_{j+1} \Delta t (\Delta z_k)^2\right] T^n_{k,\,j+1} + \left[k_j \Delta t B_j\right] T^n_{k+1,\,j} = -\left[(\Delta z_k)^2 A_j B_j\right] T^{n-1}_{k,\,j}\end{aligned}$$

A representation of how the nodes in these discretized equations are placed in our system is presented later, in Figure 5.1 and 5.2. Generally, these equations allow us to form a matrix system, which again allows us to solve the equations simultaneously. This system would look as follows

$$\mathbf{A}_s \mathbf{T}^n_s = \mathbf{B}'_s \, \mathbf{T}^{n-1}_s + \mathbf{F}_s,$$

where $s = 1, 2$ for the upper and lower system respectively.

It is evident that the equations cannot be applied at the boundaries without further consideration, because it requires us to know the temperatures beyond the borders. Before applying this system of linear equations we must therefore consider our boundary and initial conditions.

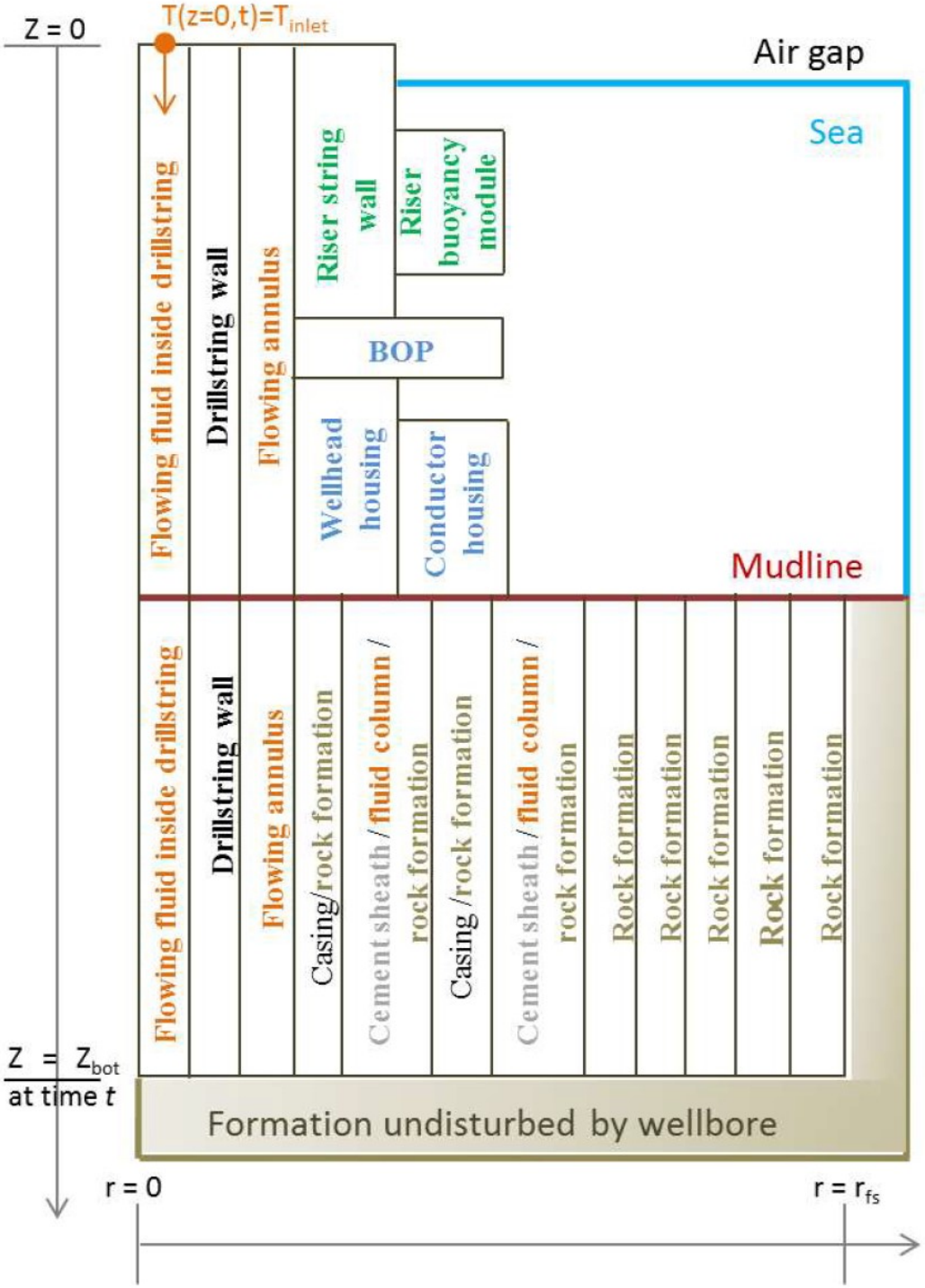


Figure 4.1: Auxiliary sketch of the discretized heat transfer problem and boundary conditions. Taken from (Sevillano u. a., 2017)

4.2 Boundary And Initial Conditions

To implement this system, certain assumptions must be made. For initial conditions it has been assumed that the system has been still for long enough that the well has the same temperature as its surrounding formation or sea, depending on its location. We also assume that there is no heat exchange across the innermost area, so that viewing this system from the center outwards is valid. Additionally, we assume that no heat exchange takes place across the outermost radius, that is $r = r_{max}$. In reality this is simply saying that the node of the sea, and the node of the furthermost formation considered is assumed constant. In mathematical terms these conditions can be expressed as follows

$$\left.\frac{\partial T}{\partial r}\right|_{r=0,\,max} = 0.$$

For further discussion we begin by considering the boundary conditions of our system above the seabed.

4.2.1 Above The Seabed

For the top of the system we have two assumed conditions. The first is that the inlet temperature is fixed and given, that is

$$T_{0,\,1} = T_{inlet}. \tag{4.1}$$

From this condition it immediately follows that $\partial T_{0,\,1}/\partial t = 0$. The other assumption we make is that no heat exchange takes place across the uppermost surface of the system. In other words,

$$\left.\frac{\partial T}{\partial z}\right|_{z=2820} = 0. \tag{4.2}$$

Evident from the matrix system above, there are a few temperatures at the top and bottom of the system that need to be evaluated before the iteration. For the top of the system these temperatures are $T^n_{0,\,2}$ and $T^n_{0,\,4}$. At the top of the system there is an air gap next to the riser wall, and no floaters. Thus we consider the first four energy balance equations where T_{sea} is replaced by T_{air}. Inserting conditions (4.1) and (4.2) into the discretized version of equation (3.1) for $k = 0$ ($z = 2820$m), we get

$$2\pi\, r_{ds_i} h_{ds_i} \left(T_{inlet} - T_{0,\,2}\right) = Q_{ds} - q\, \rho_{df_{ds}} g \sin\alpha.$$

And so we may already write that

$$T_{0,\,2} = \frac{q\, \rho_{df_{ds}} g \sin\alpha - Q_{ds}}{2\pi\, r_{ds_i} h_{ds_i}} + T_{inlet}. \tag{4.3}$$

In other words, $T_{0,2}$ is constant, and so $\partial T_{0,2}/\partial t = 0$. Applying this and condition (4.1) to equation (3.2) gives

$$\frac{2r_{ds_i}h_{ds_i}}{r_{ds_o}^2 - r_{ds_i}^2}\left(T_{inlet} - T_{0,2}\right) + \frac{2r_{ds_o}h_{ds_o}}{r_{ds_o}^2 - r_{ds_i}^2}\left(T_{0,2} - T_{0,3}\right) = 0.$$

And so using (4.3), we may again define

$$T_{0,3} = \frac{Q_{ds} - q\,\rho_{df_{ds}} g \sin\alpha}{2\pi}\left(\frac{1}{r_{ds_o}h_{ds_o}} - \frac{1}{r_{ds_i}h_{ds_i}}\right) + T_{inlet}. \tag{4.4}$$

In analogy with what our previous steps, we may write (3.3) as follows:

$$2\pi\, r_{ds_o}h_{ds_o}\left(T_{0,2} - T_{0,3}\right) + 2\pi\, r_{rw_i}h_{rw_i}\left(T_{0,3} - T_{0,4}\right) = -Q_a - q\,\rho_{df}\, g \sin\alpha,$$

and so inserting (4.3) and (4.4),

$$\begin{aligned} T_{0,4} = {} & \frac{Q_{ds} - q\,\rho_{df_{ds}} g \sin\alpha}{2\pi}\left(\frac{1}{r_{ds_o}h_{ds_o}} - \frac{1}{r_{ds_i}h_{ds_i}} - \frac{1}{r_{rw_i}h_{rw_i}}\right) \\ & + \frac{Q_a + q\,\rho_{df}\, g \sin\alpha}{2\pi r_{rw_i}h_{rw_i}} + T_{inlet} \end{aligned} \tag{4.5}$$

We have found the two needed constant temperatures for the top of the system.

The next altitude to be considered is the intersection between our upper system of 5 nodes, and our lower of 16. To avoid dependency on the lower system we choose to replace the two-point forward Euler applied on equation (3.1) and (3.3) with two point Backward Euler. In other words we write

$$\frac{\partial T_{k,j}^n}{\partial z} = \frac{T_{k,j}^n - T_{k-1,j}^n}{\Delta z}, \qquad k = \text{first node over seabed.}$$

In equation (3.2) we have approximated $\partial^2 T_{k,j}^n/(\partial z)^2$ using the central differences method, a method found by first applying forward difference, then backward difference. In this instance we choose to apply backward difference twice, and so yield

$$\frac{\partial^2 T_{k,j}^n}{(\partial z)^2} = \frac{T_{k,j}^n - 2T_{k-1,j}^n - T_{k-2,j}^n}{(\Delta z)^2}, \qquad k = \text{first node over seabed.}$$

For the bottom-most altitude of the upper system we now have the following equations:

Fluid inside the drill string

$$[q\,\Delta t\,A_{df}]\,T^n_{k-1,\,1} + \left[\pi\,\Delta t\,\Delta z_k\,E_{ds_i} - q\,\Delta t\,A_{df} + \pi\,\Delta z_k\,r^2_{ds_i}A_{df}\right]T^n_{k,\,1} - [\pi\,\Delta t\,\Delta z_k\,E_{ds_i}]\,T^n_{k,\,2} = \left[\pi\,\Delta z_k\,r^2_{ds_i}A_{df}\right]T^{n-1}_{k,\,1} + \Delta t\,\Delta z_k\,(Q_{ds}+F)$$

Drill string wall

$$[\Delta t\,k_{ds}B_{ds}]\,T^n_{k-2,\,2} - [2k_{ds}\Delta t\,B_{ds}]\,T^n_{k-1,\,2} + \left[\Delta t\,(\Delta z_k)^2\,E_{ds_i}\right]T^n_{k,\,1} - \left[\Delta t\,(\Delta z_k)^2\,E_{ds_i} + (\Delta z_k)^2\,A_{ds}B_{ds} + \Delta t\,(\Delta z_k)^2\,E_{ds_o} - \Delta t\,k_{ds}B_{ds}\right]T^n_{k,\,2} + \left[\Delta t\,(\Delta z_k)^2\,E_{ds_o}\right]T^n_{k,\,3} = -\left[(\Delta z_k)^2\,A_{ds}B_{ds}\right]T^{n-1}_{k,\,2}$$

The upward flow of fluid in the annulus

$$-[q\,\Delta t\,A_{df}]\,T^n_{k-1,\,3} + [\pi\,\Delta t\,\Delta z_k\,E_{ds_o}]\,T^n_{k,\,2} - [\pi\,\Delta t\,\Delta z_k\,E_{ds_o} - q\,\Delta t\,A_{df} + \pi\,\Delta t\,\Delta z_k\,E_{cw_i} + \pi\Delta z_k\,A_{df}B_a]\,T^n_{k,\,3} + [\pi\,\Delta t\,\Delta z_k\,E_{cw_i}]\,T^n_{k,\,4} = -[\pi\Delta z_k\,A_{df}B_a]\,T^{n-1}_{k,\,3} - \Delta t\,\Delta z_k\,(Q_a - F)$$

Riser in direct contact with water

$$[\Delta t k_{rw}B_{rw}]\,T^n_{k-2,\,4} + \left[\Delta t(\Delta z_k)^2E_{rw_i}\right]T^n_{k,\,3} - \left[(\Delta z_k)^2A_{rw}B_{rw} - \Delta t k_{rw}B_{rw} + \Delta t(\Delta z_k)^2E_{rw_i} + \Delta t(\Delta z_k)^2E_{rw_o}\right]T^n_{k,\,4} - [2\Delta t k_{rw}B_{rw}]\,T^n_{k-1,\,4} = -\left[(\Delta z_k)^2A_{rw}B_{rw}\right]T^{n-1}_{k,\,4} - \Delta t(\Delta z_k)^2E_{rw_o}T_{sea}$$

We have now sufficiently considered all boundary and initial conditions for the upper system.

4.2.2 Below The Seabed

For the bottom of the system we again make two assumptions. The first is that the temperature of the flowing fluids and the drill string are equalized at the drill bit ($z = z_{bot}$) during circulation. Thus

$$T_{K,\,1} = T_{K,\,2} = T_{K,\,3}$$

where K is the bottom most node.

The other assumption is that unlike the top of the system, heat exchange does take place across the bottom/lowermost part of the system. However, temperature

variations of formation elements below the drill bit are not computed, and so the temperature of the formations below the drill bit are assumed constant.

At the bottom of the system the first three nodes are as for the rest of the system the drilling fluid, the drill collar, and the flowing annulus. Applying the stated boundary conditions to the first of these three equations yields the following.
Fluid inside the drill string

$$q\,A_{df}\frac{T^n_{k+1,\,1}-T^n_{k,\,1}}{\Delta z_k}=\pi r^2_{ds_i}A_{df}\frac{T^n_{k,\,1}-T^{n-1}_{k,\,1}}{\Delta t}-Q_{ds}-F$$

and so

$$(\Delta t\,q\,A_{df}+\Delta z_k\,\pi r^2_{ds_i}A_{df})\,T^n_{bot,\,1}=\Delta z_k\,\pi r^2_{ds_i}A_{df}T^{n-1}_{bot,\,1}+\Delta t\,(\Delta z_k(Q_{db}+F)+q\,A_{df}T^n_{form.})$$

Since $T_{1,\,bot}=T_{2,\,bot}=T_{3,\,bot}$, this equation can be applied to all three nodes. For the remaining bottom nodes there is simply Limestone at the bottom. Thus we have one node represented by equation (3.5), and the remaining nodes by (3.7). A sketch of the discretized temperatures and boundary conditions can be seen can be seen in Figure 4.1.

When the bottom system is calculated it is assumed that the upper system has already been calculated, and so the lower system may depend on the temperatures of this system. The bottom system is expected to be the most time consuming as it has 16 horizontal nodes vs the 5 horizontal nodes for the upper system. There should therefore not be a substantial delay by calculating one system at a time. If however the reader would prefer to calculate the temperature distribution of the two systems simultaneously there are two ways of doing so The first is to simply use the values of one time-step earlier. The other is to rather than use the central differences method for the double derivatives in space at the top, to apply forward difference twice. And so the temperatures of the first node below the seabed will no longer be dependent on the Temperatures at the node above.

All boundary conditions have now been sufficiently considered, and we may proceed with the system of linear equations.

4.3 System of Linear Equations

Before considering our actual system of equations, we consider a simplified version, so to better visualize the numeric system. That is, we set all variables as listed in Table 4.1 as well as q and π equal to 1. That leaves us with the following equations.

Fluid inside the drill string

$$(\Delta t\,\Delta z_k + \Delta t + \Delta z_k)\,T^n_{k,1} \;-\; \Delta t\,\Delta z_k T^n_{k,2} \;-\; \Delta t\,T^n_{k+1,1} \;=\; \Delta z_k T^{n-1}_{k,1} \;+\; 2\Delta t\,\Delta z_k$$

Drill string wall

$$\Delta t\,T^n_{k-1,2} + \Delta t\,(\Delta z_k)^2 T^n_{k,1} - \left[2\Delta t\,(\Delta z_k)^2 + (\Delta z_k)^2 + 2\Delta t\right] T^n_{k,2} + \Delta t\,(\Delta z_k)^2 T^n_{k,3} + \Delta t\,T^n_{k+1,2} = -(\Delta z_k)^2 T^{n-1}_{k,2}$$

The upward flow of fluid in the annulus

$$\Delta t\,\Delta z_k T^n_{k,2} - [2\Delta t\,\Delta z_k + \Delta t + \Delta z_k]\,T^n_{k,3} + \Delta t\,\Delta z_k T^n_{k,4} + \Delta t\,T^n_{k+1,3} = -\Delta z_k T^{n-1}_{k,3} - 2\Delta t\,\Delta z_k$$

Riser wall in direct contact with water

$$\Delta t T^n_{k-1,4} + \Delta t(\Delta z_k)^2 T^n_{k,3} - \left[(\Delta z_k)^2 + 2\Delta t + 2\Delta t(\Delta z_k)^2\right] T^n_{k,4} + \Delta t\,T^n_{k+1,4} = -(\Delta z_k)^2 T^{n-1}_{k,4} - \Delta t(\Delta z_k)^2 T_{sea}$$

Riser wall in contact with floaters

$$\Delta t\,T^n_{k-1,4} + \Delta t(\Delta z_k)^2 T^n_{k,3} - \left[2\Delta t(\Delta z_k)^2 + 2\Delta t + (\Delta z_k)^2\right] T^n_{k,4} + \Delta t(\Delta z_k)^2 T^n_{k,5} + \Delta t\,T^n_{k+1,4} = -(\Delta z_k)^2 T^{n-1}_{k,4}$$

Floaters

$$\Delta t\,T^n_{k-1,5} + \Delta t(\Delta z_k)^2 T^n_{k,4} - \left[2\Delta t(\Delta z_k)^2 + 2\Delta t + (\Delta z_k)^2\right] T^n_{k,5} + \Delta t\,T^n_{k+1,5} = -(\Delta z_k)^2 T^{n-1}_{k,5} - \Delta t(\Delta z_k)^2 T_{sea}$$

Casing string wall and subsequent layers

$$\Delta t\,T^n_{k-1,j} + \Delta t(\Delta z_k)^2 T^n_{k,j-1} - \left[\Delta t + 2\Delta t(\Delta z_k)^2 + (\Delta z_k)^2\right] T^n_{k,j} + \Delta t(\Delta z_k)^2 T^n_{k,j+1} + \Delta t\,T^n_{k+1,j} = -(\Delta z_k)^2 T^{n-1}_{k,j}$$

Or rather:

Fluid inside the drill string, the drill string wall, and the upward flow of fluid in the annulus

$$\left(1+\tfrac{1}{\Delta t}+\tfrac{1}{\Delta z_k}\right)T^n_{k,1} \;-\; T^n_{k,2} \;-\; \left(\tfrac{1}{\Delta z_k}\right)T^n_{k+1,1} \;=\; \left(\tfrac{1}{\Delta t}\right)T^{n-1}_{k,1} \;+\; 2$$

$$\left(\tfrac{1}{\Delta z_k^2}\right)T^n_{k-1,2}+T^n_{k,1}-\left(2+\tfrac{2}{\Delta z_k^2}+\tfrac{1}{\Delta t}\right)T^n_{k,2}+T^n_{k,3}+\left(\tfrac{1}{\Delta z_k^2}\right)T^n_{k+1,2}=-\left(\tfrac{1}{\Delta t}\right)T^{n-1}_{k,2}$$

$$T^n_{k,2} \;-\; \left(2+\tfrac{1}{\Delta t}+\tfrac{1}{\Delta z_k}\right)T^n_{k,3} \;+\; T^n_{k,4} \;+\; \left(\tfrac{1}{\Delta z_k}\right)T^n_{k+1,3} \;=\; -\left(\tfrac{1}{\Delta t}\right)T^{n-1}_{k,3} \;-\; 2$$

Riser wall in direct contact with water, the riser wall in contact with floaters, and the floaters

$$\left(\tfrac{1}{\Delta z_k^2}\right)T^n_{k-1,4}+T^n_{k,3}-\left(\tfrac{1}{\Delta t}+\tfrac{2}{\Delta z_k^2}+2\right)T^n_{k,4}+\left(\tfrac{1}{\Delta z_k^2}\right)T^n_{k+1,4}=-\left(\tfrac{1}{\Delta t}\right)T^{n-1}_{k,4}-T_{sea}$$

$$\left(\tfrac{1}{\Delta z_k^2}\right)T^n_{k-1,4}+T^n_{k,3}-\left(2+\tfrac{2}{\Delta z_k^2}+\tfrac{1}{\Delta t}\right)T^n_{k,4}+T^n_{k,5}+\left(\tfrac{1}{\Delta z_k^2}\right)T^n_{k+1,4}=-\left(\tfrac{1}{\Delta t}\right)T^{n-1}_{k,4}$$

$$\left(\tfrac{1}{\Delta z_k^2}\right)T^n_{k-1,5}+T^n_{k,4}-\left(2+\tfrac{2}{\Delta z_k^2}+\tfrac{1}{\Delta t}\right)T^n_{k,5}+\left(\tfrac{1}{\Delta z_k^2}\right)T^n_{k+1,5}=-\left(\tfrac{1}{\Delta t}\right)T^{n-1}_{k,5}-T_{sea}$$

Casing string wall and subsequent layers

$$\left(\tfrac{1}{\Delta z_k^2}\right)T^n_{k-1,j}+T^n_{k,j-1}-\left(\tfrac{1}{\Delta z_k^2}+2+\tfrac{1}{\Delta t}\right)T^n_{k,j}+T^n_{k,j+1}+\left(\tfrac{1}{\Delta z_k^2}\right)T^n_{k+1,j}=-\left(\tfrac{1}{\Delta t}\right)T^{n-1}_{k,j}$$

This system would give the following matrices for a system of equations $\mathbf{A}_s\mathbf{T}^n_s = \mathbf{B}'_s\,\mathbf{T}^{n-1}_s+\mathbf{F}_s$, where $s=1,2$ for the upper and lower system respectively. We denote the subscript K_s to mean the bottom most node for each system, respectively. The matrices $\mathbf{A}_s$ are too large to include in the paper, but the other vectors have been included below. They are dependent on the altitude, and so the vectors below are merely examples.

$$\mathbf{T}^n_s=\begin{bmatrix}\tau^n_{s,1}\\ \tau^n_{s,2}\\ \tau^n_{s,3}\\ \vdots\\ \tau^n_{s,K_s-1}\\ \tau^n_{s,K_s}\end{bmatrix},\qquad \tau^n_{1,k}=\begin{bmatrix}T^n_{k,1}\\ T^n_{K_1,2}\\ T^n_{K_1,3}\\ T^n_{K_1,4}\\ T^n_{K_1,5}\end{bmatrix},\quad \tau^n_{2,k}=\begin{bmatrix}T^n_{k,1}\\ T^n_{K_1,2}\\ T^n_{K_1,3}\\ T^n_{K_1,15}\\ T^n_{K_1,16}\end{bmatrix}$$

$$\mathbf{B}_1 = \begin{bmatrix} b_{1,1} \\ b_{1,1} \\ \vdots \\ b_{1,1} \\ b_{1,2} \\ b_{1,2} \\ b_{1,2} \\ \vdots \\ b_{1,2} \\ b_{1,1} \\ \vdots \\ b_{1,1} \end{bmatrix}, \quad b_{1,1} = \begin{bmatrix} 1/\Delta t \\ -1/\Delta t \\ -1/\Delta t \\ -1/\Delta t \\ 1 \end{bmatrix}, \quad b_{1,2} = \begin{bmatrix} 1/\Delta t \\ -1/\Delta t \\ -1/\Delta t \\ -1/\Delta t \\ -1/\Delta t \end{bmatrix}, \quad \mathbf{F}_1 = \begin{bmatrix} 2 \\ -T_{0,2}/(\Delta z_k)^2 \\ -2 \\ -T_{sea} - T_{0,4}/(\Delta z_k)^2 \\ 0 \\ f_{1,1} \\ \vdots \\ f_{1,1} \\ f_{1,2} \\ \vdots \\ f_{1,2} \\ f_{1,1} \\ \vdots \\ f_{1,1} \end{bmatrix}, \quad f_{1,1} = \begin{bmatrix} 2 \\ 0 \\ -2 \\ -T_{sea} \\ 0 \end{bmatrix}, \quad f_{1,2} = \begin{bmatrix} 2 \\ 0 \\ -2 \\ 0 \\ -T_{sea} \end{bmatrix},$$

$$\mathbf{B}_2 = \begin{bmatrix} b_2 \\ b_2 \\ \vdots \\ b_2 \\ b_2 \\ 1/\Delta t \\ 1/\Delta t \\ 1/\Delta t \\ -1/\Delta t \\ -1/\Delta t \\ \vdots \\ -1/\Delta t \end{bmatrix}, \; b_2 = \begin{bmatrix} 1/\Delta t \\ -1/\Delta t \\ -1/\Delta t \\ -1/\Delta t \\ \vdots \\ -1/\Delta t \\ -1/\Delta t \end{bmatrix}, \quad \mathbf{F}_2 = \begin{bmatrix} f_2 \\ f_2 \\ \vdots \\ f_2 \\ f_2 \\ 2 + T_{form.}/\Delta z \\ 2 + T_{form.}/\Delta z \\ 2 + T_{form.}/\Delta z \\ 0 \\ 0 \\ \vdots \\ 0 \\ -T_{form.}/(\Delta z_k)^2 \end{bmatrix}, f_2 = \begin{bmatrix} 2 \\ 0 \\ -2 \\ 0 \\ 0 \\ \vdots \\ 0 \\ -T_{form.}/(\Delta z_k)^2 \end{bmatrix}.$$

where b_2, $f_{2,1}$, and $f_{2,2}$ have lengths equal to 16. The left subscripts $k = 1, 2, ..., K_s$ of $T_{j,k}$ are defined by the row number, and the right subscripts $j = 1, 2, ..., J_s$, where $J_1 = 5$, $J_2 = 16$, are defined by the columns.

We thus have a simplified system of linear equations similar to that of our actual system. The matrices for the actual system will look very similar, but are too large to include here. We thus proceed to consider how to solve such a system, and this system in particular.

Chapter 5

Developing a Program in C++

5.1 LU-Reduction

One of the most efficient and well used methods for solving matrix equations is LU-decomposition, and the algorithm for this decomposition is called LU-Reduction. It is often used in the context of parallel computing with the purpose of comparing processing speeds for different computers. The method of LU decomposition is a matrix version of Gaussian elimination, and is itself one of the most practical and most efficient ways of solving a system of linear equations. The requirement of this method is that the coefficient matrix A is square, and so it may be applied on our system. The method of LU decomposition consists of factorizing the coefficient matrix A into a lower triangular matrix L and an upper triangular matrix U, such that

$$\mathbf{A}x = b \quad \Longrightarrow \quad \mathbf{LU}x = b \quad \Longrightarrow \quad \mathbf{L}\left(\mathbf{U}x\right) = b.$$

Thus we may solve two simpler systems of equations, $\mathbf{L}y = b$ and $\mathbf{U}x = y$, rather than solving the original one in one go. To find the matrices $\mathbf{L}$, $\mathbf{U}$ we have to solve the matrix equation $\mathbf{A} = \mathbf{LU}$,

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{n-1,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \dots & a_{n,n} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{2,1} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ l_{n,1} & \dots & l_{n,n-1} & 1 \end{bmatrix} \begin{bmatrix} u_{1,1} & u_{1,2} & \dots & u_{1,n} \\ 0 & u_{2,2} & \dots & u_{2,n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & u_{n,n} \end{bmatrix}.$$

Immediately from this it is evident that the top row of the matrix $\mathbf{U}$ is identical to the top row of the coefficient matrix $\mathbf{A}$. For the first of $\mathbf{A}$ we first have that

$$a_{2,1} = l_{2,1}u_{1,1} \quad \Longrightarrow \quad l_{2,1} = a_{2,1}/a_{1,1}.$$

In fact, this equation can be extended for all $l_{k,1}$:

$$a_{k,1} = l_{k,1}u_{1,1} \quad \Longrightarrow \quad l_{k,1} = a_{k,1}/a_{1,1}.$$

From simple analysis we also see that for the j^{th} column of row 2 we have

$$a_{2,j} = l_{2,1}u_{1,j} + u_{2,j} \quad \Longrightarrow \quad u_{2,j} = a_{2,j} - l_{2,1}u_{1,j}.$$

Clearly there is a pattern to these equations, and so both the latter equations can be extended to account for all columns j, and all rows k, respectively. In other words we can achieve a general formula for the elements $l_{k,j}$ and $u_{k,j}$ as follows.

$$a_{k,j} = l_{k,j}u_{j,j} + \sum_{i=1}^{j-1} l_{k,i}u_{i,j} \quad \Longrightarrow \quad l_{k,j} = \frac{1}{u_{j,j}}\left(a_{k,j} - \sum_{i=1}^{j-1} l_{k,i}u_{i,j}\right) \qquad \text{for } j < k.$$

$$a_{k,j} = u_{k,j} + \sum_{i=1}^{k-1} l_{k,i}u_{i,j} \quad \Longrightarrow \quad u_{k,j} = a_{k,j} - \sum_{i=1}^{k-1} l_{k,i}u_{i,j} \qquad \text{for } j \geq k,$$

Once the matrices U and L have been found, the outer matrix equation consist of a number of linear equations,

$$\begin{array}{ccccccccccc}
y_1 & & & & & & & & & = & b_1, \\
l_{2,1}y_1 & + & y_2 & & & & & & & = & b_2, \\
l_{3,1}y_1 & + & l_{3,2}y_2 & + & y_3 & & & & & = & b_3, \\
\vdots & & & & & & \ddots & & \vdots & & \\
l_{m,1}y_1 & + & \ldots & + & l_{m,m-1}y_{m-1} & + & y_m & = & b_m,
\end{array}$$

such that

$$y_k = b_k - \sum_{i=1}^{k-1} l_{k,i}y_i.$$

For the inner matrix equation the resulting linear equations are very similar, but opposite, since $\mathbf{U}$ is an upper triangle matrix.

$$\begin{array}{ccccccccccc}
u_{1,1}x_1 & + & u_{1,2}x_2 & + & \ldots & + & u_{1,m}x_m & = & y_1, \\
& + & u_{2,2}x_2 & + & \ldots & + & u_{2,m}x_m & = & y_2, \\
& & & \ddots & & & \vdots & & \vdots \\
& & & & u_{m-1,m-1}x_{m-1} & + & u_{m-1,m}x_m & = & y_{m-1}, \\
& & & & & & u_{m,m}x_m & = & y_m,
\end{array}$$

such that

$$x_k = \frac{1}{u_{k,\,k}} \left(y_k - \sum_{i=k+1}^{m} u_{k,\,i} x_i \right).$$

(Suli und Mayers, 2011).For this inner equation the elements in x must clearly be iterated beginning with the lower-most element, that is $k = m$.

We may now proceed to solve our liner system of equations using this method, but must first elect the number and placement of the nodes.

5.2 Election of Nodes

The system for which the temperature distribution is to be estimated is as presented in Figure 5.1. To simplify the discretizing of the upper system there is simply one horizontal node in the center of each area. Therefore, since the temperature of the sea and air is assumed to be constant, there are only five horizontal nodes to be considered. For the lower system there are at most 11 areas to be considered, as well as the rock. Unlike the upper system however, there is no circulation in the rock. Hence the temperature of the rock close to the drill will likely be affected. To account for this there are 16 nodes in total, that is, a minimum of 5 nodes in the rock area. The placement of the nodes have been chosen to imitate those chosen in Sevillano u. a. (2017), including the placement of where the temperature is assumed constant. That is, at 5.727m from the center of the drill.

The vertical nodes have been chosen based on maximum detail, while limiting time use. For the upper system nodes are first placed mostly with a distance of 20m. Between 19 and 12m a distance of 1m has been used, as this is close to the BOP, as shown in Figure 5.2. From the top of the BOP and downwards a distance of 0.2m is used until the height of 0.2m. This is because, as can be seen from the sketch, there are many variations of widths for both node 3 and node 4 in this area.

For the lower system there is clearly more variation, as there is a shift from mud to cement to rock for each pipe. This can again be visualized in the sketch in Figure 5.1. Again the placement of the nodes have been chosen to imitate those chosen by Sevillano u. a. (2017).

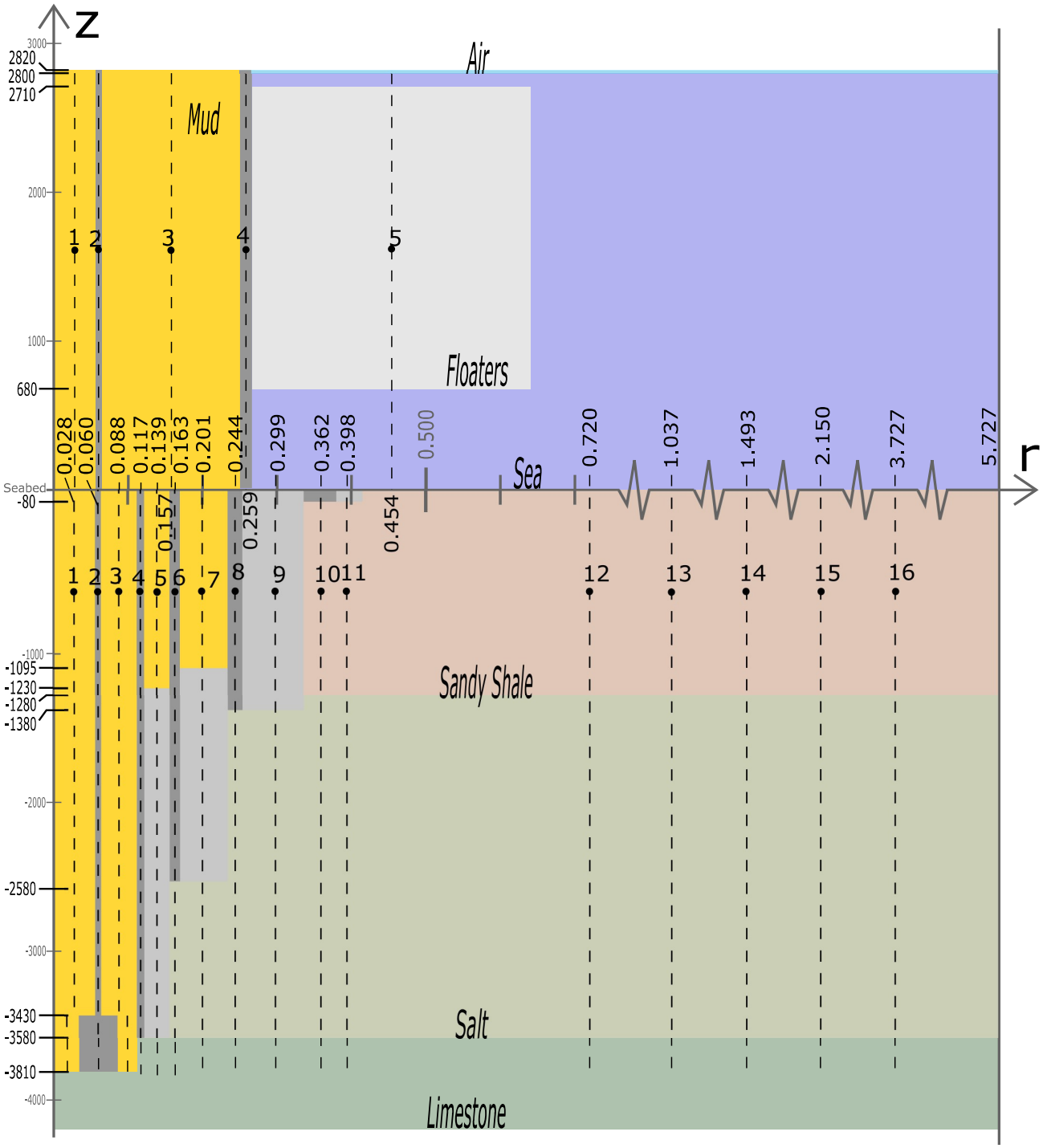


Figure 5.1: Sketch of deep oil well with horizontal nodes.

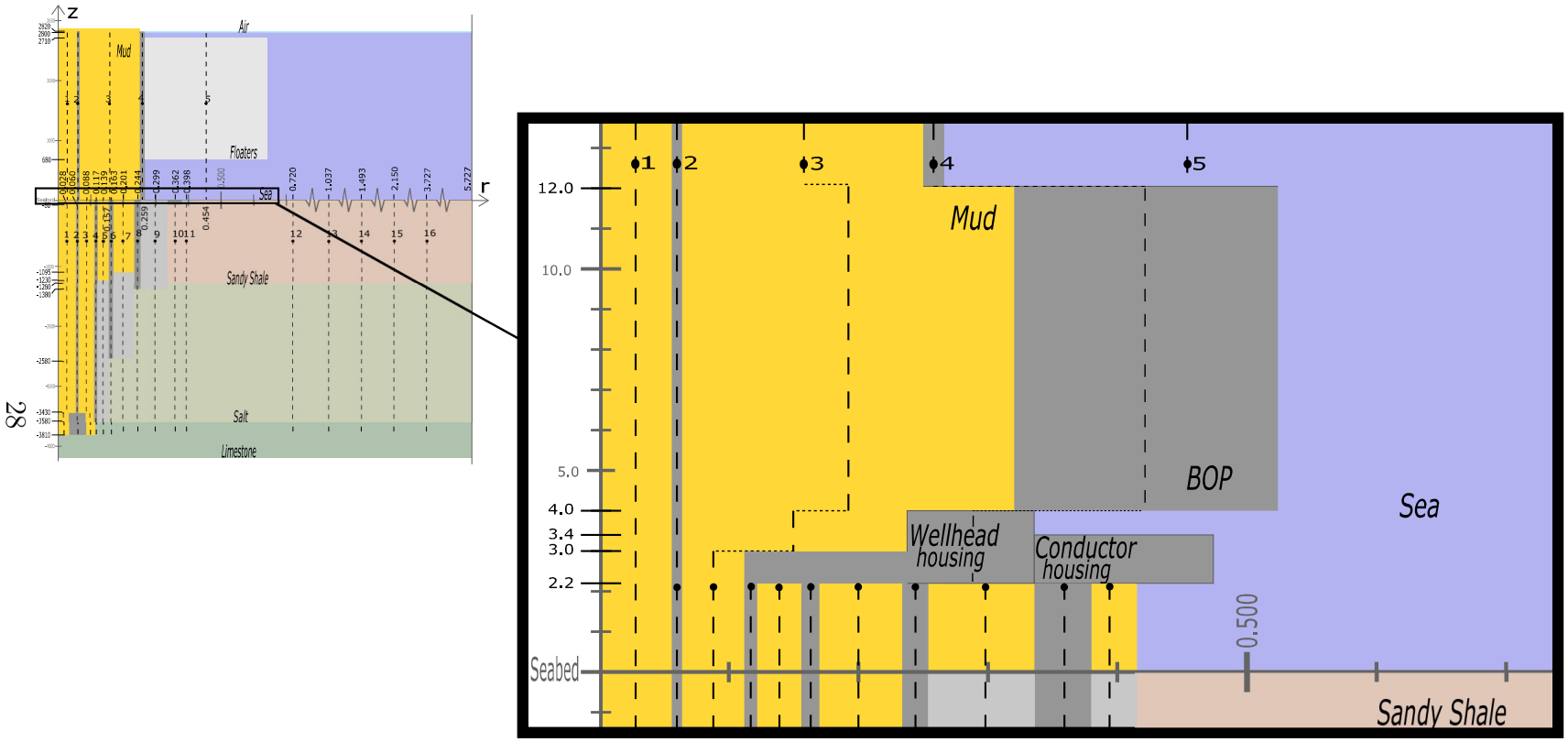


Figure 5.2: Zoomed of Figure 5.1

5.3 Extract From Programming

To better visualize the code built in this thesis, four extracts of the code has been added below. Two headers and one source file.

Z:\Master\Header.h 1

```
1  #ifndef Header
2  #define Header
3  #define _USE_MATH_DEFINES
4
5  #include <iostream>
6  #include <vector>
7  #include <map>
8  #include <math.h>
9  #include <string>
10 #include <cstdlib>
11 #include <iomanip>
12 #include <ctime>
13 #include <sstream>
14
15 using namespace std;
16
17 //***ASSIGNING VALUES TO VARIABLES***
18
19 //Outer radii r, [r] = m
20 //r-Keys 5 nodes: "upper", "BOP", "wellheadHouse", "conductorHouse",
     "casing".
21 //r-Keys 16 nodes: "lower", "bottom".
22 void setOuterRadii(map<string, vector<double>> &rad);
23
24 //Density rho, [rho] = kg / m^3
25 //rho-Keys: "1", "mud", "collar", "drillstring", "steel", "cement",
     "floater", "sea", "shale", "salt", "limestone".
26 void setDensity(map<string, double> &dens);
27
28 //Conductivity k, [k] = W / m K
29 //k-Key: "mud", "steel", "cement", "floater", "sea", "shale", "salt",
     "limestone".
30 void setConductivity(map<string, double> &cond);
31
32 //Specific Heat Capacity Cp, [Cp] = J / Kg K
33 //Cp-Keys: "1", "mud", "collar", "drillstring", "steel", "cement",
     "floater", "sea", "shale", "salt", "limestone".
34 // (1 refers to the first horizontal node. That is, the fluid inside the
     drillstring.)
35 void setHeatCapacity(map<string, double> &cap);
36
37 //Convection Coefficients h, [h] = W K / m ^ 2
38 //h-Key 5 heights: "drillstringInner", "drillstringOuter", "wallInner",
     "wallOuter"
39 void setConvCoeff(map<string, vector<double> > &conv);
40
41 //Energy Source Term Q, [Q] = W/m
42 //Q-Keys: "insideDrillstring"(5), "flowingAnnulus"(5), "drillBit".
43 void setEnergySource(map<string, vector<double>> &energySource);
44
45
46 //***ASSIGNING INITIAL VALUE OF THE FORMATION AND SEA***
47 void setSeaTemp();
```

Figure 5.3: The header for assigning variables and initial temperatures

Z:\Master\Matrix.cpp 1

```
1  #include "stdafx.h"
2  #include "Header.h"
3  #include "Matrix.h"
4
5  //#include <bits/stdc++.h>
6
7
8  Matrix::Matrix(int N) : Matrix(N, N, 1) {}
9  Matrix::Matrix(int colrows, int timeJump) : Matrix(colrows, colrows,
     timeJump) {}
10 Matrix::Matrix(int columns, int rows, int timeJump) : COLUMNS(columns),
     ROWS(rows), SIZE(columns*rows), timeStep(timeJump) {
11     mat = new double[columns*rows]{};
12     //setConstants();
13 }
14 Matrix::~Matrix() {
15     invalidate();
16 }
17
18 bool Matrix::isValid() const {
19     return mat != nullptr;
20 }
21 void Matrix::invalidate() {
22     delete[] mat;
23     mat = nullptr;
24 }
25
26 int Matrix::getRows() const {
27     return this->ROWS;
28 }
29 int Matrix::getColumns() const {
30     return this->COLUMNS;
31 }
32 int Matrix::getSize() const {
33     return this->SIZE;
34 }
35 int Matrix::getTimeStep() const {
36     return this->timeStep;
37 }
38 int Matrix::getPosition(int row, int column) const {
39     return (COLUMNS)*(row)+column;
40 }
41 double Matrix::getValue(int row, int column) const {
42     int pos = getPosition(row, column);
43     return this->mat[pos];
44 }
45 double Matrix::getValue(int row) const {
46     return this->mat[row];
47 }
48
49 double Matrix::getA(string key) {
50     return density[key] * specificHeat[key];
51 }
```

Figure 5.4: A small part of the Matrix Source code

```
153
154
155 //--------Solving the equation--------------
156 void Matrix::LUDecomposition(Matrix & upper, Matrix & lower) {
157     //Matrix lower(ROWS), upper(ROWS);
158     // Decomposing matrix into Upper and Lower
159     // triangular matrix
160     for (int rowk = 0; rowk < ROWS; rowk++) {
161         // Upper Triangular
162         for (int columnj = rowk; columnj < COLUMNS; columnj++) {
163
164             // Summation of L(rowk, i) * U(i, columnj)
165             double sum1 = 0;
166             for (int i = 0; i < rowk; i++)
167                 sum1 += (lower.getValue(rowk,i) * upper.getValue
                      (i,columnj));
168
169             // Evaluating U(rowk, columnj)
170             upper.setValue(rowk, columnj, this->getValue(rowk,columnj) -
                  sum1);
171         }
172
173         // Lower Triangular
174         for (int columnj = rowk; columnj < COLUMNS; columnj++) {
175             if (rowk == columnj)
176                 lower.setValue(rowk,rowk, 1); // Diagonal as 1
177             else {
178
179                 // Summation of L(columnj, i) * U(i, rowk)
180                 double sum2 = 0;
181                 for (int i = 0; i < rowk; i++)
182                     sum2 += (lower.getValue(columnj,i) * upper.getValue
                        (i,rowk));
183
184                 // Evaluating L(k, rowk)
185                 lower.setValue(columnj,rowk, (this->getValue(columnj,rowk)
                        - sum2)/upper.getValue(rowk, rowk) );
186             }
187         }
188         /*
189         if ((rowk % 100) == 0) {
190             std::cout << rowk << endl;
191         }*/
192     }
193 }
194
195
196
197
198
199
200
201
```

Figure 5.5: The LU-reduction within the Matrix class

Z:\Master\UpperMatrix.h 1

```
1  #ifndef UPPMAT
2  #define UPPMAT
3
4  #include "Matrix.h"
5  class UpperMatrix :
6      public Matrix
7  {
8  private:
9      //The variables A, B, D, E as listed in Figure 4.1 in the thesis.
10     vector<double> A;
11     vector<double> B;
12     double D;
13     double E_1; //Dillstring inner
14     vector<double> E_2; //Drillstring outer
15     vector<double> E_3; //Casing wall/Riser wall inner
16
17     //Energy Source Term Q, [Q] = W/m
18     double Q_ds; //Drillstring
19     vector<double> Q_a; //Annulus
20
21     vector<double> r; //Outer radii
22
23     //Conductivity k, [k] = W / m K
24     double ksteel;
25     double kfloat;
26
27 public:
28     UpperMatrix(int timeJump);
29
30     void setConstants(); //Uses the functions from Header.
31     void allocateMatrixConstants(int timeJump); //Allocats the constants in ↵
          the Upper Matrix
32 };
33 #endif
34
35
```

Figure 5.6: The header for the upper matrix

Chapter 6

Results

6.1 Input Data

height z [m]	Convection Coefficient h [$\mathrm{W\,m^{-2}\,K^{-1}}$]			Energy Source Term Q [$\mathrm{W\,m^{-1}}$]	
	Inner wall, Drillstring	Outer wall, Drillstring	Casing String/ Riser Wall	Inside Drillstring	Annulus
[12.79, 2820]	7979.3	1522.4	400.6	2.63	7.70
[4.1,10.98]		1253.8	245.0		3.98
[3.1,4]		1556.4	424.2		8.35
[-3436,3]		2362.3	1333.4		64.97
[-3810,-3444]	14 234.9	1591.0	1212.4	14.50	159.58

Table 6.1: Height dependent variables

As mentioned there are a number of variables that must be defined in order to get temperature outputs. All these variables have been received from Lucas Sevillano, one of the authors of (Sevillano u. a., 2017), thus allowing an easier comparison of data. These variables have been presented in table 6.1 and table 6.2, so to allow the reader the opportunity to reproduce the results. The division of altitudes are also identical to those of Sevillano.

The temperature of the mud, before entered into the pipe is set to $20°C = 293.15K$, and so inserting the variables into equations (4.3) and (4.5), we have the

		Density ρ [kg m^{-}3]	Specific Heat Capacity c_p [m^{2}K^{-1}s^{-}2]	Conductivity k [kg m K^{-1} s^{-3}]
Mud In Drill string		1 200	1 600	1.750
Drill wall		7 800	400	43.750
Riser Floater		2 700	900	13.750
Mud		1 350	1 600	1.750
Cement		1 917	2 000	0.700
Steel		7 700	400	43.750
Sea		1 070	3 988	0.575
	Shale	2 057	2 151	1.9
Rock	Salt	2 160	920	4.5
	Limestone	2 700	851	2.2

Table 6.2: Height independent variables

Energy Source term at drill bit Q_{db} [W]	150151
Gravity g [m s^{-2}]	9.81
Mass flow q [m^{3}s^{-1}]	0.0473

following temperatures at the top of the system, $z = 0$.

$$T_{\text{inlet}} = 293.15K \tag{6.1}$$

$$T_{2,0} = \frac{9.81 \cdot 1200 \cdot 0.0473 - 2.63}{2\pi \cdot 0.112 \cdot 7979.3} K + 293.15K = 293.25K \tag{6.2}$$

$$\begin{aligned} T_{4,0} = \frac{2.63 - 0.0473 \cdot 1200 \cdot 9.81}{2\pi} & \left(\frac{1}{0.127 \cdot 1522.4} - \frac{1}{0.112 \cdot 7979.3} - \frac{1}{0.501 \cdot 400.6} \right) K \\ & + \frac{7.70 + 0.0473 \cdot 1350 \cdot 9.81}{2\pi \cdot 0.501 \cdot 400.6} K + 293.15K \\ = 293.73K & \end{aligned}$$

The initial temperatures for the entire system is assumed to be equal to that of the formation and the sea. That is, it is as presented in Figure 6.1.

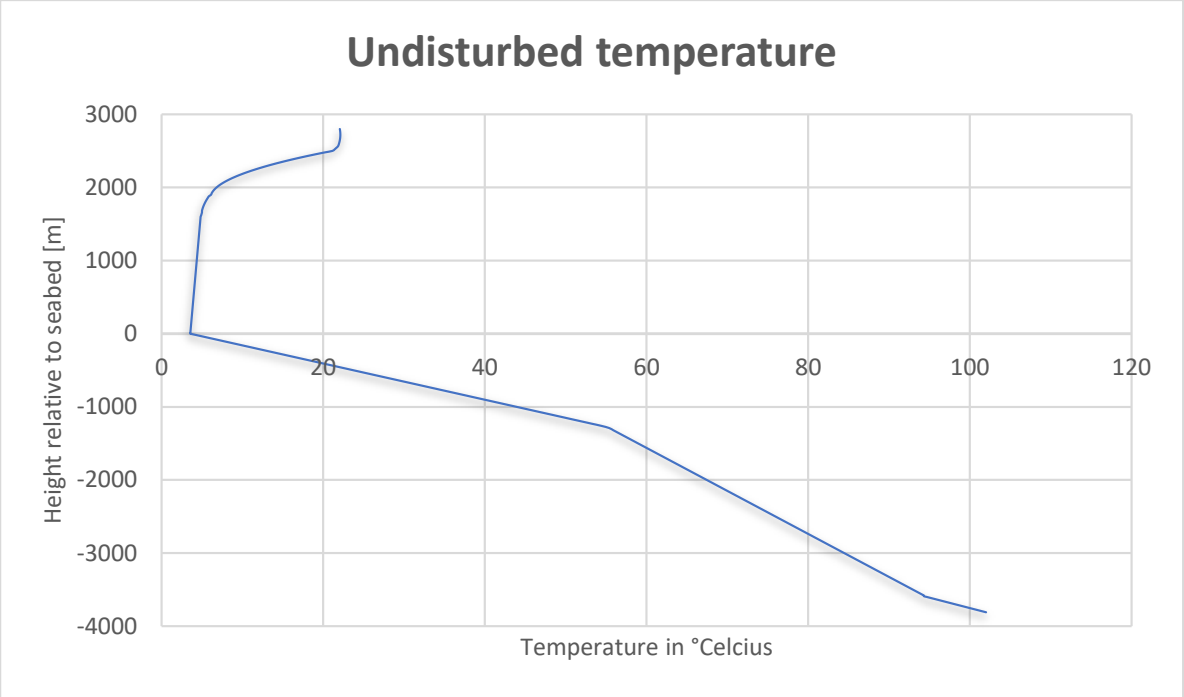


Figure 6.1: Undisturbed temperatures. Sea temperatures have been taken from (Bergman, 1011), while the formation temperature is from(Vavik u. a., 2017)

Natural convection occurs where the nodes are in direct contact with water, due to oceanic circulation (Kays u. a., 2005). These values were calculated and given to me along with the rest of the initial data, but as will be explained in the next section, after much trial and error I opted not to use the calculated natural convection. Hence this concludes our initial data.

6.2 Above Seabed

The code for solving our system of linear system of equations for specific input data was built based on the equations and nodes as explained above. However, when first attempting to run this part of the code it returned impossible temperatures after only a few iterations. That is, it resulted in positive and negative temperatures to the power of six. In addition, each node had temperatures of opposite sign to those next to it. To simplify and better analyze a possible mistake in the coding, the BOP, wellhead housing and conductor housing presented in Figure 5.2 were removed. This

did not significantly improve the results.

After some analysis it was clear that the temperatures reached this altitude at the outer nodes after only one iteration, which suggested the mistake was in the outer nodes. The forced convection for the nodes in direct contact with water, and affected by the current of the water, was calculated and supplied to me. I thus opted to not use these data. That is, both the floaters and the riser were assumed to be the temperature of the water at that altitude. This immediately gave much more sensible results, that is, temperatures in the hundreds.

There was another discrepancy. The temperatures of the innermost nodes were significantly negative. After going back to the original equations I realized there were differences between the equations as stated by (Sevillano u. a., 2017) and that of (Marshall und Bentsen, 1982), and that I in fact disagreed with the first equation. Then, after further trial and error, a few adjustments were made to the original equations, that is, those explained in chapter 3. This, at last, resulted in the data as presented in Figure 6.2 - 6.7.

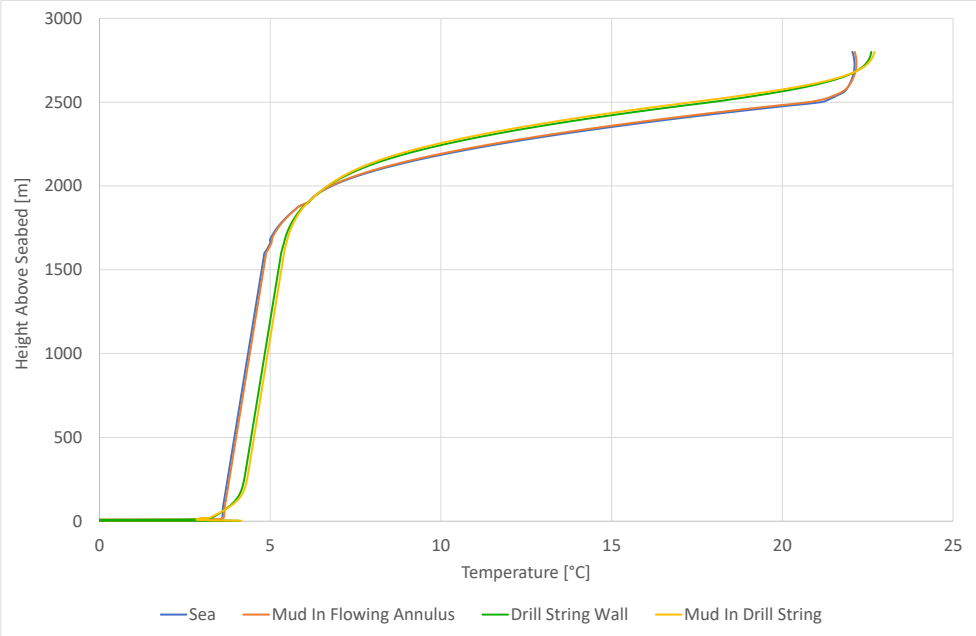


Figure 6.2: Temperature distribution above seabed after half a minute.

From Figure 6.6 and Figure 6.7 it is evident that above the seabed the temperatures stabilize after only 30 minutes of drilling. The distribution of these temperatures are as expected, and can be compared to those of (Vavik u. a., 2017). However,

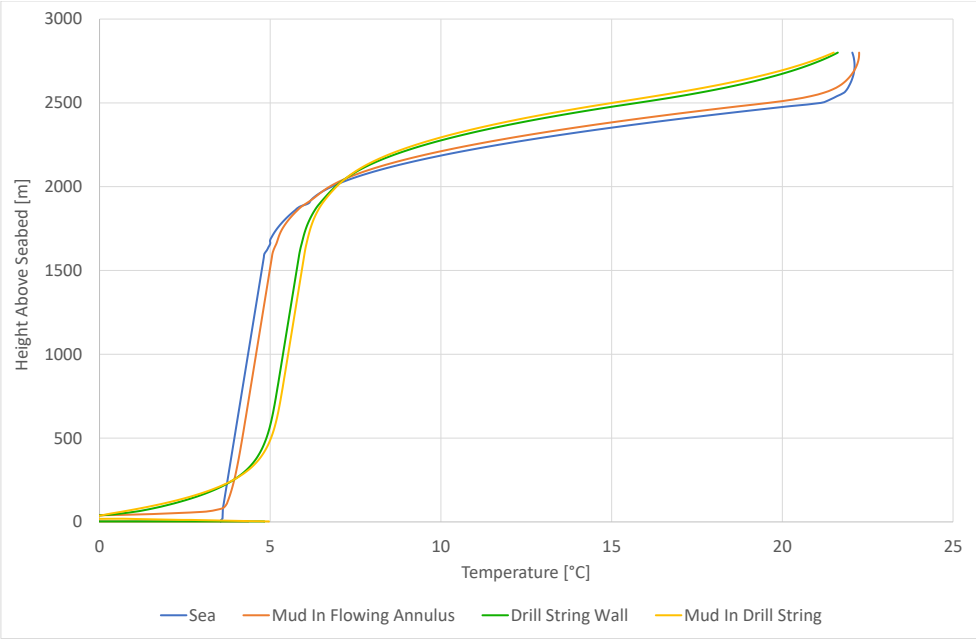


Figure 6.3: Temperature distribution above seabed after two minutes.

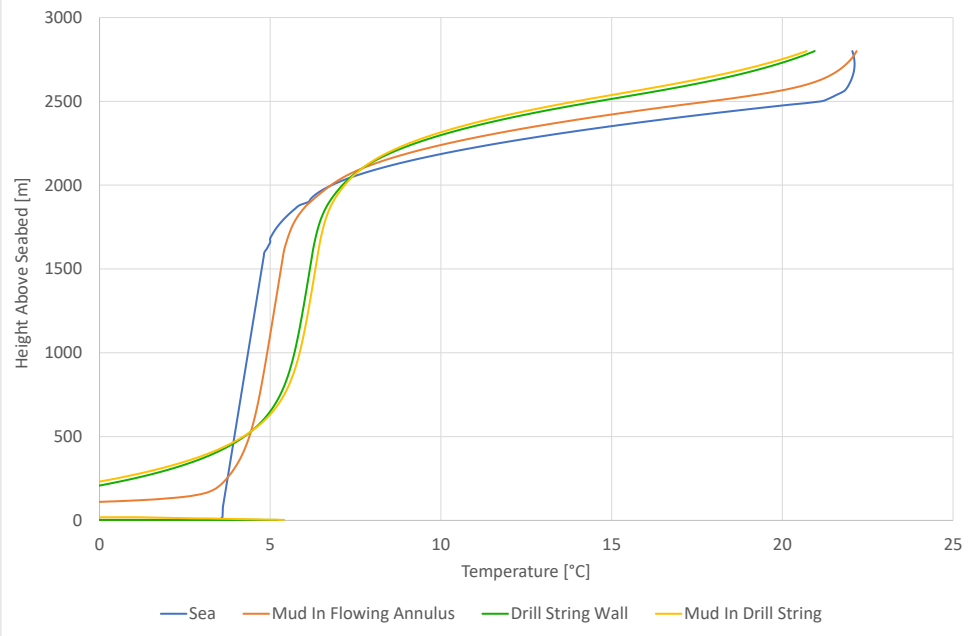


Figure 6.4: Temperature distribution above seabed after five minutes.

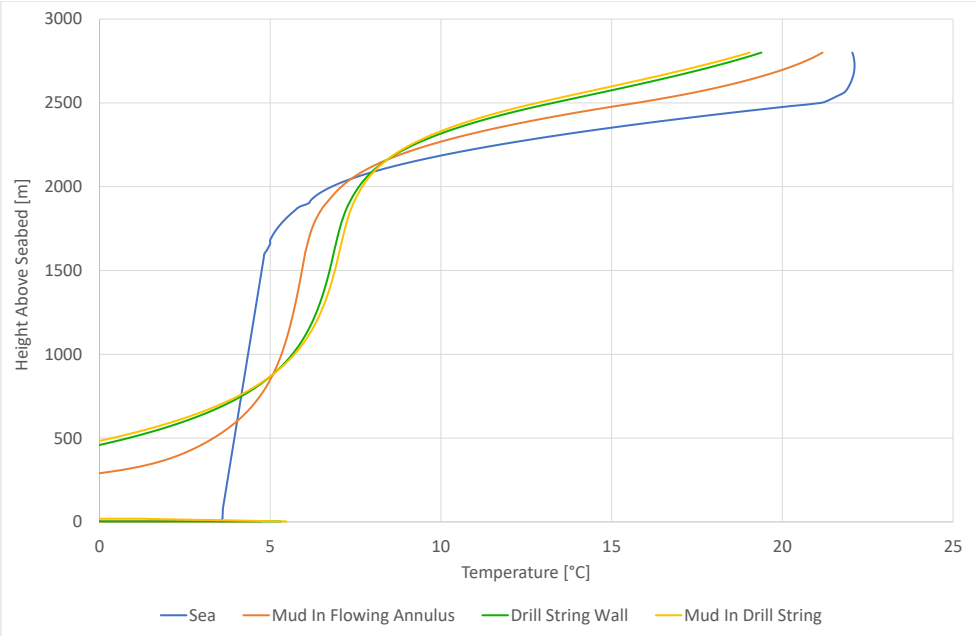


Figure 6.5: Temperature distribution above seabed after 15 minutes.

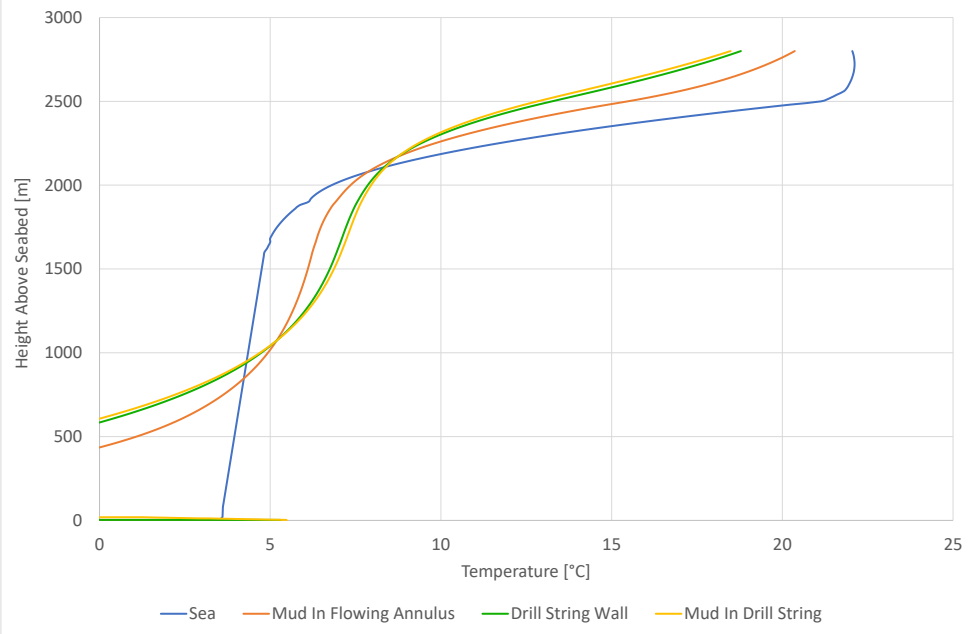


Figure 6.6: Temperature distribution above seabed after 30 minutes.

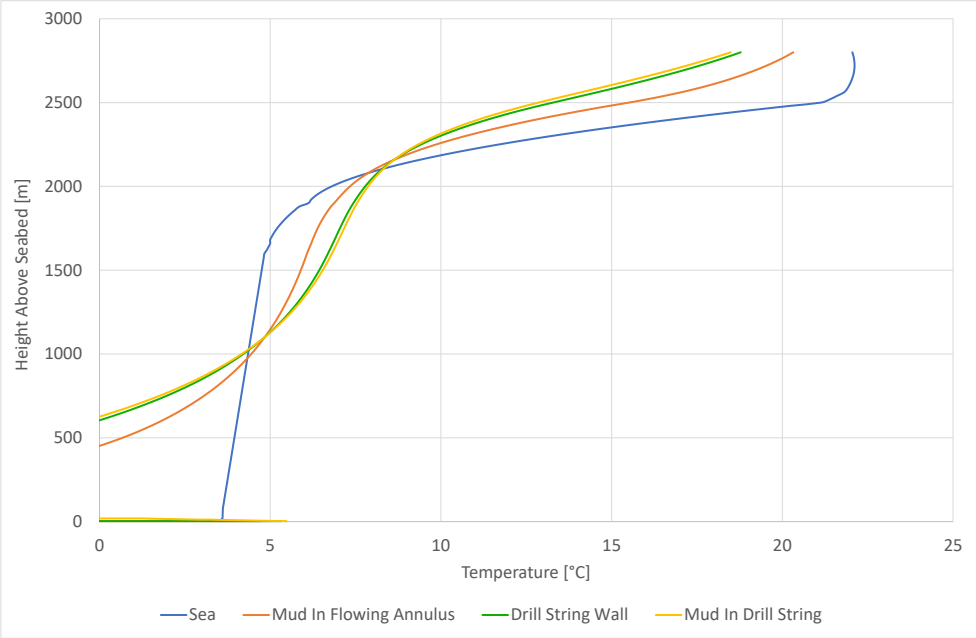


Figure 6.7: Temperature distribution above seabed after four hours.

it does have a significant discrepancy at low altitudes. After only half a minute this discrepancy is evident. I have attempted to find out why this occurs, but have not been able to fix it. Since the riser is assumed to have the temperature of water, the same sets of equations are used at lower altitudes as those higher up. Thus it seems unlikely that there is a mistake in the equations. If however, there is a mistake in the equations, it could lie in the sign of the first term. This term will give a negative addition to the temperature of the fluid inside the drill string. It is also divided by Δz_k, and so it would naturally have a higher value when Δz_k is smaller. Which is indeed what happens closer to the seabed. However, when attempting to change this in the code, the resulting temperatures are again impossible.

Another possible explanation is that the method simply does not work for low values of Δz_k compared with the value of $\Delta t = 0$. If we assume the method and the supplied data are all correct, then the mistake must lie in the code. The LU-decomposition was tested on several known systems of equations, and so is known to be correct. Additionally it is not likely to give such a nice curve if the calculations are all together wrong. Another option is therefore that there is an error when assigning values to the matrix A. I have, however, not been able to find such an error.

For the linear system of equations above the seabed only 5 nodes have been used,

and so has a much lower computation time than that of the lower system. Despite setting both the riser and the floaters equal to the sea temperature, I kept the nodes in the programming. This system spends only 6 seconds per iteration, with the time step set equal to 1s. Thus it is quite fast, and should easily be fast enough by applying parallel programming. In fact, applying parallel programming will likely make it so fast, that it its processing time is practically insignificant compared to that of the lower system.

6.3 Below Seabed

The numerical system for the calculations of the temperatures below the seabed is much more time demanding than that of the upper system, as there are 16 nodes to consider. As of yet the program for the system below the seabed uses 3 minutes and 24 seconds. And so it is clearly the most time consuming part of the code, and should be prioritized if accelerating it.

Chapter 7

Conclusion

Both the code for the system above the seabed and the system below the seabed were developed, but due to time constraint only the results of the upper one has been included in this thesis. The results of the upper system gives reasonable results that stabilize after only half an hour, and so suggests that the method and code works. There is however a discrepancy of negative temperatures at low altitudes that I have not been able to explain.

The necessary processing speed for each iteration is ≤ 1 Hz for the program to be useful in comparison with real time data. As of yet the upper part of the program spends only 6 seconds per iteration, while the lower one spends 3 minutes and 24 seconds, and so is by far the most time consuming part. Within the code the most time consuming process is the decomposition of A into LU, and so the decomposition of the lower matrix should be the focus of a future acceleration using parallel computing.

This thesis should be considered a basis for the development of this program and as a resource for the completion of the above-mentioned goals. The thesis shows that the method developed could be of use, and a processing speed of 3 minutes and 30 seconds suggests that 1 Hz per iteration is indeed an achievable goal once accelerated using parallel computing.

Chapter 8

Further research

8.1 Graphic Processing Unit (GPU)

It was originally my intention to attempt to accelerate the processing speed of the program by applying parallel computing using the programming language CUDA. Normally program are made in a simple serial fashion, as is the program I have made. Parallel computing however, consists of splitting your code into blocks of work so that you may send each block to a different processor, and so the computations can happen simultaneously (Cook, 2013). There are several ways of doing this, and LU-reduction is in fact highly suited for parallel programming.

In the calculations the decomposition of the matrix A into two triangular matrices L and U was by far the most time consuming part. Accelerating this part using parallel programming would therefore be quite advantageous.

8.2 Development of Horizontal Nodes

I have no particular background in the physics of oil wells, or indeed of petroleum in general. Spending my time attempting to recalculate the natural convection variables used in the coding would therefore be an inefficient use of my time. Therefore, as explained, I opted to simplify the system and so not be dependent on this data. It is however my recommendation that this is done in further research.

Bibliography

[Bergman 1011] Bergman, Jennifer: *Temperature of Ocean Water*. 1011. – URL https://www.windows2universe.org/?page=/earth/water/temp.html. – Zugriffsdatum: 05.10.2018

[Cheney und Kincaid 2013] Cheney, W. ; Kincaid, D.: *Numerical Mathematics and Computing, 7th edition*. Brooks/Cole, Cengage Learning, 2013. – URL https://books.google.no/books?id=ztpELgEACAAJ&dq. – ISBN 1133491812

[Cook 2013] Cook, S.: *CUDA programming : a developer's guide to parallel computing with GPUs, 1st edition*. Elsevier/MK, Amsterdam, 2013. – URL https://books.google.no/books?id=EX2LNkSqViUC. – ISBN 1283716453

[Kays u. a. 2005] Kays, W. ; Crawford, M. ; Weigand, B.: *Convective Heat and Mass Transfer, 4th edition*. McGraw-Hill, 2005. – URL https://books.google.no/books?id=Qh5RAAAAMAAJ. – ISBN 9780072990737

[Marshall und Bentsen 1982] Marshall, D. W. ; Bentsen, R. G.: A Computer Model to Determine the Temperature Distributions In a Wellbore. In: *Journal of Canadian Petroleum Technology* 21 (1982), Nr. 1. – URL http://doi.org/10.2118/82-01-05

[Sevillano u. a. 2017] Sevillano, L. C. ; De Andrade, J. ; Sangesland, S.: Estimation of Undisturbed Geothermal Gradient in Wells From Measured Drilling Data: A Numerical Approach. In: *ASME 2017 36th International Conference on Ocean, Offshore and Arctic Engineering* 8 (2017). – URL http://doi.org/10.1115/OMAE2017-62205

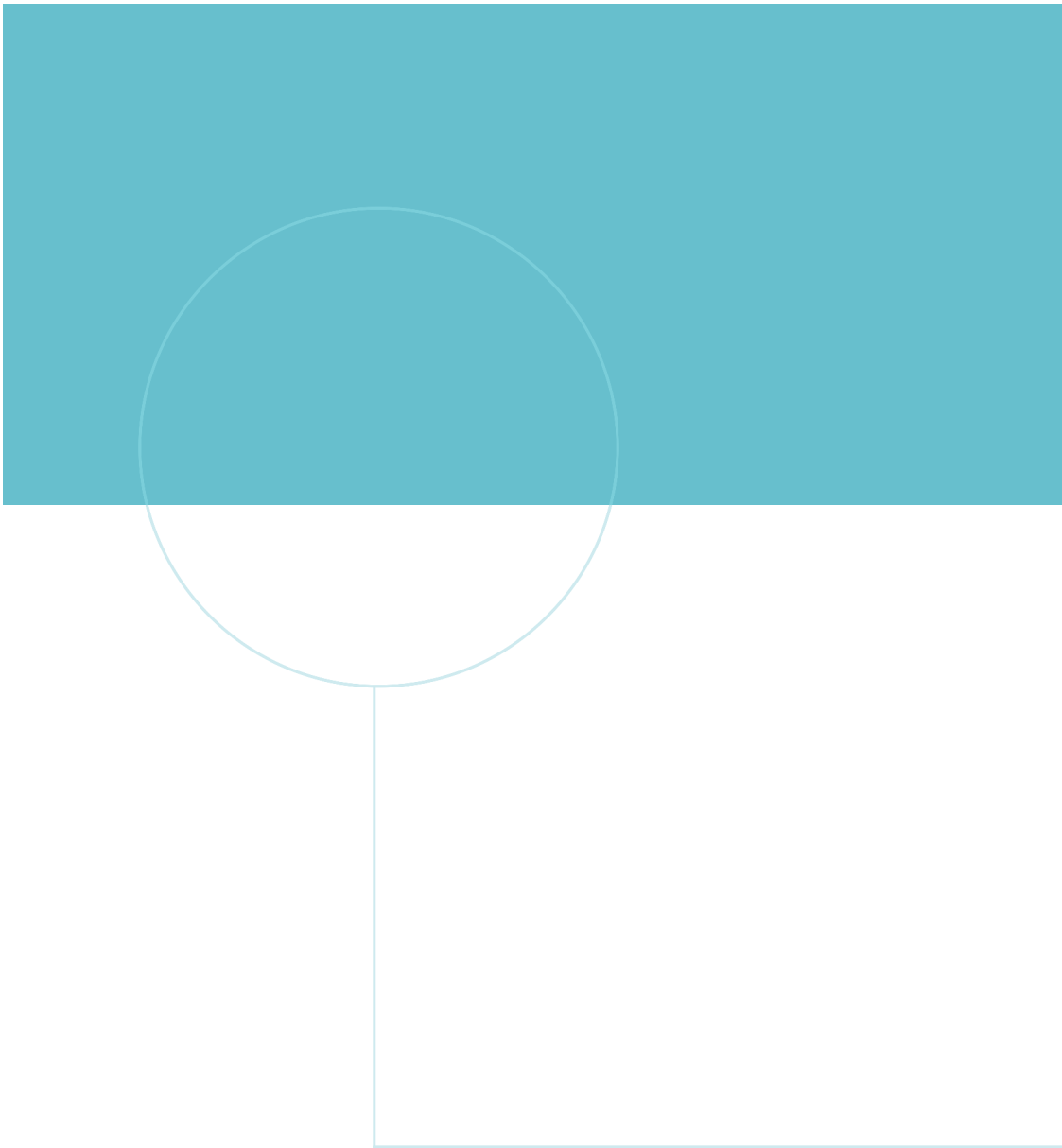
[Suli und Mayers 2011] Suli, E. ; Mayers, D.: *An Introduction to Numerical Analysis, 5th edition*. Camebridge University Press, 2006, 2011. – URL https://books.google.no/books?id=hj9weaqJTbQC. – ISBN 9780521007948

[Thomas 2013] Thomas, J. W.: *Numerical Partial Differential Equations: Finite Difference Methods. Part of the Texts in Applied Mathematics book series (TAM, volume 22)*. Springer Science & Business Media, 2013. – URL https://books.google.no/books?id=83v1BwAAQBAJ. – ISBN 1489972781

[Vavik u. a. 2016] Vavik, D. ; Sangesland, S. ; Shayegh, M.: Loss of Circulation an Indication of Hydrocarbon Influx? (2016). – URL http://doi.org/10.2118/179712-MS

[Vavik u. a. 2017] Vavik, D. ; Sevillano, L. C. ; Sangesland, S. ; Red, B. K.: Gas in Riser - The Elephant in the Room. (2017). – URL http://doi.org/10.2118/185291-MS

[White 2008] White, F. M.: *Fluid Mechanics, 6th edition.* McGraw-Hill, 2008. – URL https://books.google.no/books?id=Z1OlPwAACAAJ. – ISBN 9780071286459



NTNU
Norwegian University of
Science and Technology