PE 7063: Advanced Drilling Project

"Numerical simulation of two layer cutting transport model to predict the limiting point (Near Horizontal wells) "

Submitted by:
Yermek Alzhanov (1203819)
Sriram Ravichandran (1457774)
Kriti Singh (1451057)

Submitted to Prof. Stefan Miska

Abstract

Cuttings transport has been modeled with 1 dimensional, two layered model which focuses on the bed formation in the wellbore systems. The model assumes formation of bed layer below a heterogeneous layer of mud and cuttings. The model can predict the variation of bed height in the well bore as a function of various parameters like flow rate, inclination, rate of penetration, mud properties, well bore eccentricity with respect to drill pipe. The model can provide an operational envelope stating the range of conditions to avoid bed formation. Since the model assumes moving bed, the model can very well predict the bed sliding velocity for different inclinations thereby conditions for bed slide back. This model can thus provide a finite range of parameters that could improve the efficiency of solids transport.

Individual Contribution

The successful completion of our project is a result of combined effort from all the team mates. We started with the literature review where all of us read different papers that have made groundbreaking contributions in the field of our study. We discussed various different models available in the literature and designed our scope of the project. The specific contribution of each team mate is shown below:

<u>VBA Code</u>: Sriram and Kriti

<u>Simulations</u>: Kriti and Yermek

<u>Analyis and interpretation of results</u>: Sriram, Kriti and Yermek

<u>Presentation</u>: Sriram and Yermek

<u>Report Writing</u>: Sriram, Kriti and Yermek

Table of Contents

Abstract 2

List of Figures 5

List of Tables 5

Chapter 1: Introduction 6

Chapter 2: Literature Review 8

Chapter 3: Model and Numerical Method proposed 10

Chapter 4: Results and Conclusions 17

Nomenclature 22

REFERENCES 23

Appendix 24

List of Figures

Figure 1. Layers in the 1 dimensional, 2 layered cuttings transport model 10
Figure 2. Force balance for control volume of the heterogeneous layer 12
Figure 3. Three cases of possible bed height relative to the drillpipe 13
Figure 4. Bed height versus flowrate with constant Inclination angle at point 50 17
Figure 5. Velocity of bed layer versus flowrate with constant Inclination angle at point 790 18
Figure 6. Limiting point versus Inclination angle at constant flowrate equal to 250 gpm 19
Figure 7. Velocity of bed layer versus Inclination angle with constant flowrate at point 9 19

List of Tables

Table-1 Operating conditions for 1-D 2 layered model 16

Chapter 1: Introduction

Cutting transport is one of the most important functions performed by the drilling fluid. Improper design will lead to accumulating of cuttings in the well and complete stop of the drilling operation which will lead to the big loss of money. Some of the drilling problems that may happen in result are shown below:

- Stuck pipe – pipe cannot be rotated or moved without damaging the pipe and exceeding maximum allowed hook load.
- Lost circulations – drilling fluid lost to the formation
- High torque and drag – high resistance to move or rotate the drill-string.
- Poor cement jobs – lead to casing damage, integrity problems, casing leakage.
- Reduction in rate of penetration – leads to longer time to perform drilling operation and consequently higher cost of drilling
- Faster bit wear – leads to more trip outs to change the bit which increase the drilling time consequently higher cost of drilling

There are few ways to solve this problem. First is increase the flow rate and effective viscosity of the fluid but the drawback of this method is the reduction in penetration rate. Other ways include using mechanical cleaning devices, foams, sweeps and etc. The cutting transport problems in the vertical wells were very well studied. Nowadays with the introduction of horizontal and deviated wells in drilling practice a good understanding of cutting transport in highly inclined wells is vital.

This report presents the numerical simulation of 1-D two layer cutting transport model to predict cutting bed height. It is governed by mass flux of cuttings that were re-suspended and deposited per unit interface. The model was later tested against a correlation based models.

Chapter 2: Literature Review

Ford and his colleagues[3] from a Heriot-Watt University found that there are two mechanisms that govern the cuttings transport- rolling or sliding and suspension travel. Extensive experimental studies were conducted and it was found that the minimum transport velocity required for cutting transport is sensitive to hole-angle variations. Seven different slurry flow patterns were observed. They also noted that an increase in the viscosity helps in cutting transportations when medium and highly viscous fluids are used. Interestingly they found that when they used water, the minimum transport velocity was even lower than when they used the medium viscosity fluid. They concluded that not only the rheology but also a flow regime greatly affects the effectiveness of a cutting transport. They also noted that the effect of pipe rotation when they used water was very small compared to the effect when they used medium to high viscous fluids.

Bjorndalen and Kuru[2] developed a one dimensional transient mechanistic model of cuttings transport with conventional drilling fluids in horizontal wells. Their model also predicts cutting bed height and solved numerically as a function of mud flow rate, drilling rates, rheological characteristics, drill pipe eccentricity and wellbore geometry. Their sensitivity analysis showed that drill pipe eccentricity, cutting diameter and mud density have very little effect on bed height. They concluded that drilling rate and mud flowrate are the most important factors affecting the formation of bed height. Higher drilling rates increase bed height while higher flowrates decrease it which using common engineering sense were anticipated. When compared thick mud with water or thin mud, thick mud removes cuttings at lower flow rates but water will remove the cuttings more effectively if required critical rate is achieved.

Kamp and Rivero[1] developed a mechanistic model that is able to predict cuttings bed build-up. The model is able to show good quantative preedictions in comparison with a correlation-based model. But model overpredicts cuttings transport at given mud flowrate due to the fact that resuspension flux should not be proportional with the interfacial friction velocity and cuttings concentration profile in the heterogeneous layer is not flat.

Sharma[4] used Gavignet and Sobey(1989) two-layer model and separated the particle layer into two separate layers. He had a stationary bed and a bed sliding up inside the annulus on top of a bed sliding down at the bottom.

Feifei Zhang[5] conducted experimental studies investigating the effects of fluid type, well inclination angle, ROP, mud rate, drill pipe rotation, and temperature on cuttings behavior in directional wells. He developed a mechanistic models to calculate the cutting concentration profile and pressure profile along the entire wellbore. He found that cuttings back sliding have great effect on the cutting transport in inclined wells.

Chapter 3: Model and Numerical Method proposed

Model Description:

When the solids are transported in the hole along with mud, there are different flow pattern observed in the bore hole at different operating conditions and sections of the hole. This can be attributed to the fact that there are many forces acting on the solid particles which can make it to settle down or may be suspended in the fluids. Assuming the basic case where separated flow occurs in the bore hole where the solids settle down and form a bed that moves with a particular velocity in the hole. Not all the solid cutting particles settle down due to drag forces provided by the moving fluid, making a certain portion of the solid particles to remain suspended in the fluids. This creates 2 layers of fluids where there is a heterogeneous layer of mud and cuttings on the top and moving bed in the bottom. The model assumes slippage between both the layers, the bed and the heterogeneous layer move with different velocities, u_b and u_h respectively. The model neglects slippage with the heterogeneous layer and hence the cuttings and the mud are transported with a homogenous no slip velocity u_h. The concentration of cuttings in the heterogeneous layer C_h is a strong function of the velocities of both the layers and also depends on the degree of mass transfer between the layers. The mass transfer fluxes of both the phases between the layers ϕ_s and ϕ_l, is in the form of either deposition into the bed or resuspension in to the heterogeneous layer.

The following assumptions are made for the modelling the rate of cuttings transport in the system

- Steady state transport of cuttings and mud in the hole
- Drill Pipe rotation effects are neglected in the system, thus making it 1-D simulations.
- The concentration of cuttings in the bed are assumed to be constant due to the nature of packing of solids in the bed (cubical packing, C_b=0.52)
- No slippage considered within the layers
- The density of the cuttings and mud remain constant along the wellbore, (incompressible fluids)

The mass balance is done for a small control volume of Δz length in the well bore-hole section which gives rise to the continuity equations as follows

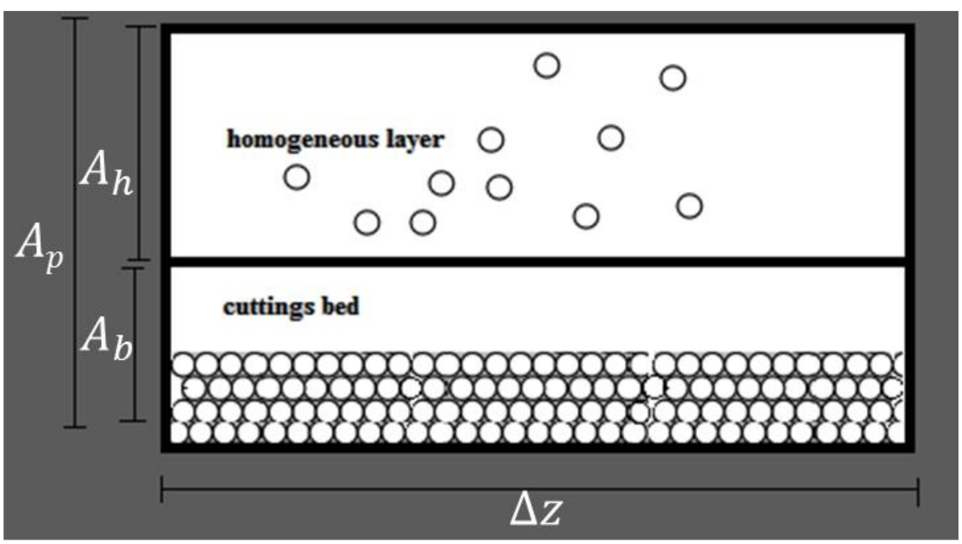

Figure 1. Layers in the 1 dimensional, 2 layered cuttings transport model

In the bed applying the mass balance, considering the mass transfer over the interface of length S_i and no accumulation in the control volume, we have

$$\rho_b A_b u_b - \left(\rho_b A_b u_b + \frac{\partial}{\partial z}(\rho_b A_b u_b)\Delta z\right) + (\varphi_L + \varphi_s) S_i \Delta z = 0 \qquad (1)$$

$$\frac{\partial}{\partial z}\{\rho_b A_b u_b\} = (\varphi_L + \varphi_s) S_i \qquad (2)$$

Similarly applying the individual phase balance in heterogeneous layer, we get following equations as continuity equations

$$\frac{\partial}{\partial z}\{\rho_l (1 - c_h) A_h u_h\} = -\varphi_L S_i \qquad (3)$$

$$\frac{\partial}{\partial z}\{\rho_s c_h A_h u_h\} = -\varphi_s S_i \qquad (4)$$

Here the cuttings bed concentration is assumed to be constant, hence volume ratio of liquid and solids in the bed remain constant, which gives the following relation between the mass transfer fluxes

$$\varphi_L = \frac{(1-c_h)}{c_h}\frac{\rho_l}{\rho_s}\varphi_s \qquad (5)$$

The bed density and heterogeneous layer density is given by the concentration of cuttings and liquid in the respective layers.

$$\rho_b = (1 - c_b)\rho_l + \rho_s c_b \qquad (6)$$

$$\rho_h = (1 - c_h)\rho_l + \rho_s c_h \qquad (7)$$

The force balance in the control volume leads to the momentum equation in the control volume of heterogeneous layer which is as follows

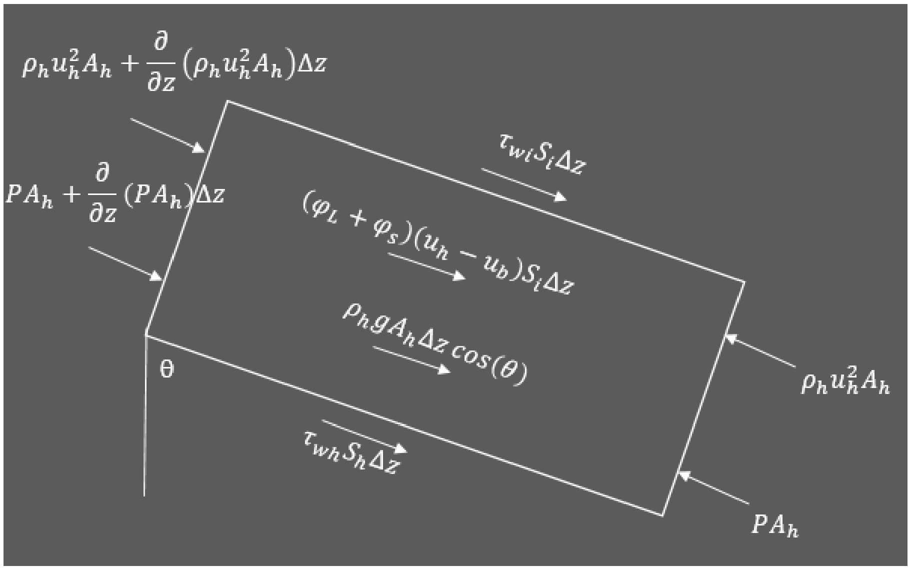

Figure 2. Force balance for control volume of the heterogeneous layer

$$\rho_h u_h^2 A_h - \left(\rho_h u_h^2 A_h + \frac{\partial}{\partial z}(\rho_h u_h^2 A_h)\Delta z\right) + PA_h - \left(PA_h + \frac{\partial}{\partial z}(PA_h)\Delta z\right)$$

$$-(\varphi_L + \varphi_s)(u_h - u_b)S_i\Delta z - \rho_h g A_h \Delta z \cos(\theta) - \tau_{wh} S_h \Delta z - \tau_{wi} S_i \Delta z = 0 \qquad (8)$$

$$\frac{\partial}{\partial z}\{\rho_h u_h^2 A_h\} =$$

$$-A_h \frac{\partial P}{\partial z} - \rho_h g A_h \cos(\theta) - 0.5\rho_h f_h u_h |u_h| S_h - 0.5\rho_h f_i (u_h - u_b)|u_h - u_b| S_i - (\varphi_L + \varphi_s)(u_h - u_b) S_i \qquad (9)$$

Similarly, the momentum equation for the bed is also written as follows

$$\frac{\partial}{\partial z}(\rho_b u_b^2 A_b) =$$

$$-A_b \frac{\partial P}{\partial z} - \rho_b g A_b \cos(\theta) - 0.5\rho_b f_b u_b |u_b| S_b + 0.5\rho_h f_i (u_h - u_b)|u_h - u_b| S_i + (\varphi_s + \varphi_L)(u_h - u_b) S_i \qquad (10)$$

As it is seen in the above equations, the unknown variables for the equations are C_h, u_b,u_h,P and bed height. But the model has five coupled differential equations with five unknown which has to be solved numerically.

It is observed that the parameters A_b, A_h, S_b, S_h and S_i are related to the bed height h. Based on the bed height, three different cases are seen where the pipe parameters of wetted area and perimeter would differ significantly.

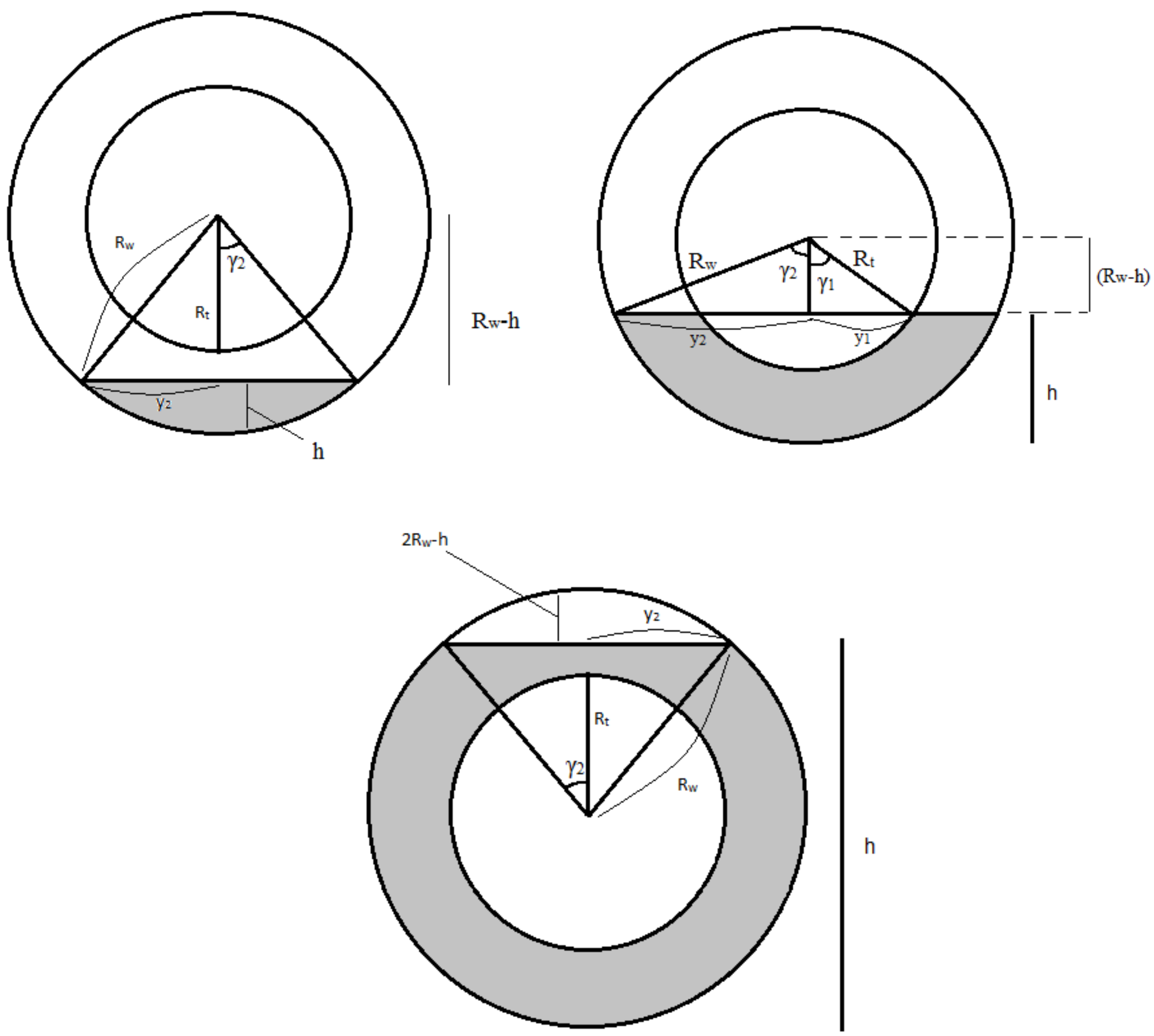

Figure 3. Three cases of possible bed height relative to the drillpipe

The variations of wetted area and perimeter are given in appendix in the code for all the three cases. Closure relationships are available for the friction factors based on whether the flow is in laminar or turbulent conditions.

$$Re_h = \frac{\rho_h D_h u_h}{\mu_h} \qquad Re_b = \frac{\rho_b D_b u_b}{\mu_b} \qquad Re_i = \frac{\rho_h D_h (u_h - u_b)}{\mu_h}$$

$$D_h = \frac{4(A_h)}{(S_h + S_i)} \quad D_b = \frac{4(A_b)}{(S_b + S_i)}$$

The closure relationship for solid mass flux is available as follows

$$\varphi_s = \varphi_{s,dep} - \varphi_{s,sus} \tag{11}$$

$$\varphi_{s,dep} = c_h \rho_s u_s Sin\theta \tag{12}$$

The u_s is the settling velocity under equilibrium conditions which is solved from the equation below

$$\frac{1}{6}\pi d^3 (\rho_s - \rho_l) g = \frac{1}{8}\pi d^2 (C_D \rho_l)\, {u_s}^2 \tag{13}$$

The drag coefficient is evaluated at the settling Reynolds number , thus making equation 13 a non-linear one which is implicit in nature , needed to be solved at particular diameter of particles.

$$\varphi_{s,sus} = c_h \rho_s u_{Ti} H(h) \tag{14}$$

where u_{Ti} is the interface friction velocity and H(h) is a heavy side function.

Thus with equations 12 and 13, the mass flux of solids are computed from which mass flux of liquid is computed from equation 5.

The governing equations for model are (2), (3), (4), (9) and (10). These equations are made dimensionless by treating $h^*=h/(2^*R_w)$; $A_h^*=A_h/A_{pipe}$; $A_b^*=1- A_h^*$; $S_b^*=S_b/2\pi(R_w+ R_t)$; $S_h^*=1-S_b^*$;$S_i^*=S_i/2\pi(R_w+ R_t)$;

The velocities are made by dimensionless by diving with initial homogenous mixture velocity $U_{o,}$ i.e. $u_h^*=u_h/ U_o$ and $u_b^*=u_b/ U_o$.

$$U_o = Q_m/({r_w}^2 - {r_t}^2)\pi$$

(15)

The boundary condition for C_h is given as C_{ho}=ROP $\pi\ r_w{}^2/Q_m$

The purpose of converting all terms into dimensionless would ease the calculation such that all the values would scale down to lower value around 1.

The five equations are solved numerically with 4th order Rung- Kutta method as they are self-correcting and stable for coupled non Linear ODEs. The equations have boundary conditions as follows

$$\begin{array}{ll} u_h * & 1 \\ u_b * & 0 \\ h * & 0 \end{array} \qquad \begin{array}{ll} P * & 1 \\ c_h & c_{ho} \end{array}$$

Chapter 4: Results and Conclusions

The model can predict the bed height as function of Q, ROP , eccentricity, fluid viscosity and other operational conditions which has been computed in the Kamp and Rivero)[1]. The operational conditions used for the simulation study has been tabulated below

Parameters	FPS	values		SI	values
Viscosity of Homogenous layer	cp	10	µh	pa s	0.01
Viscosity of Bed	cp	10	µb	pa s	0.01
Density of Fluid used	ppg	9	ρl	kg/m3	1080
Density of solids	ppg	18	ρs	kg/m3	2160
Inclination angle of well	degrees	10	θ	radian	0.523599
Sphericity	-	1	ψ	-	1
diameter of cuttings	inches	0.2	dp	m	0.00508
Concentration of cutting in bed	-	0.52	Cb	-	0.52
Rate of Penetration	ft/hr	50	Rop	m/s	0.004233
	inches	8.75	rw	m	0.22225
	inches	4.5	rt	m	0.1143
Mixture Volumetric flow rate	gpm	100	qw	m3/s	0.0063
Pressure at the Bit	psia	500	P0	pa	3447380
Length	ft	1000	l	m	20
Roughness factor				e_d	0.0005

Table-1 Operating conditions for 1-D 2 layered model

Since more bed height prediction is already computed, the given study aims in the conditions where bed height can be neglected. The point in the well bore from the entrance where the bed height becomes zero is taken as the limiting point. The limiting point can be the farthest point in the wellbore where no more bed formation takes place and all the cuttings are suspended in the fluids. Thus, in order to determine the limiting point, the operational parameters that affect the bed formation must be optimized. The study is also extendable to various inclinations especially for well bores of 50 degrees and above, where the separated 2 layered model hold good. Hence, this model can be used a precursor for the flow pattern predictions and provide an operational envelope in order to avoid bed formations.

It is seen from the simulations that bed formations at a particular location in the wellbore decrease with increase in flow rate. This is pretty much the expected trend as there is more turbulence, the solids gets suspended more and are carried away from the bed.

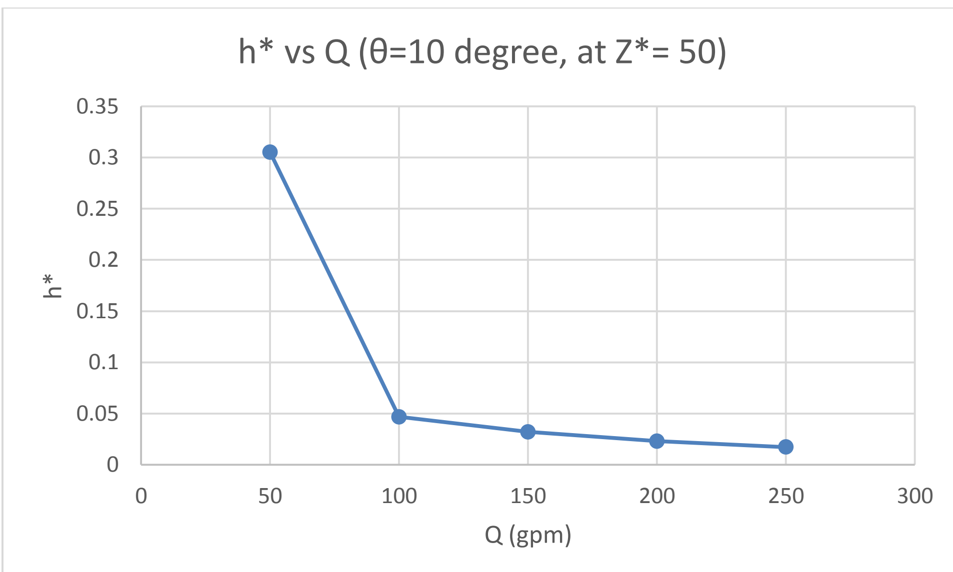

Figure 4. Bed height versus flowrate with constant Inclination angle at point 50

Hence it is anticipated to operate at the maximum flow rate in order to avoid bed formation. But this is not completely true for all cases as, more flow rate would mean more frictional losses and chances of erosion and corrosion in the wellbore are higher. Also, it is seen that for the same inclination, up to a certain point there is bed velocity. The bed becomes stagnant and starts falling back as the shear forces in the bed are overcome by gravity force at that point. Hence, this undesirable. With increase in flow rate, the bed velocity decreases indicating that it becomes zero and also negative. The negative bed velocity indicates that the bed is falling back, hence providing the limit for the flow rate.

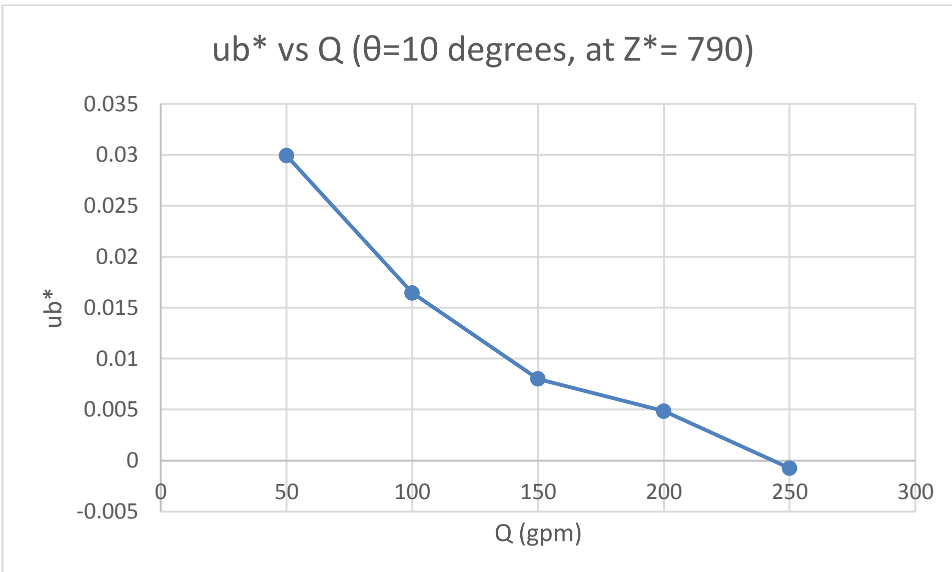

Figure 5. Velocity of bed layer versus flowrate with constant Inclination angle at point 790

The model can also predict the limiting point for different operating conditions and different inclination angles. For a given inclination, it can be seen that the bed height becomes zero at a particular location Z*. Z* (limiting point) strongly depends on the flow rate. As the flow rate is higher, Z* reduces showing that no bed can be formed as early as possible if Q increases. But the bed fall back case must also be taken into account leading to formation of bed again in the previous sections.

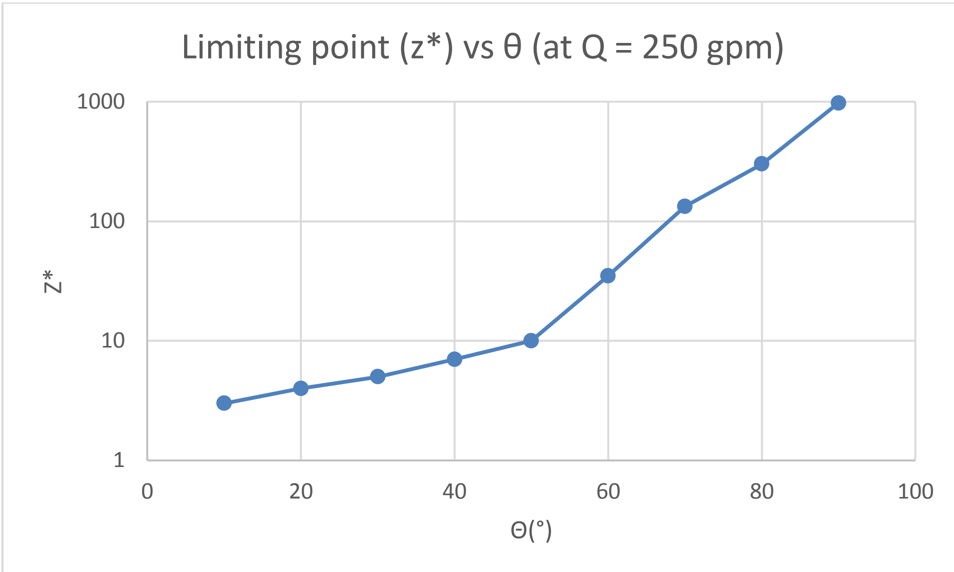

Figure 6. Limiting point versus Inclination angle at constant flowrate equal to 250 gpm

As expected, it is seen that increasing the inclination reduces the bed formation. The limiting point increases as the wellbore inclinations increase i.e. the horizontal wells have the limiting point farthest for a given flow rate of cuttings and mud.

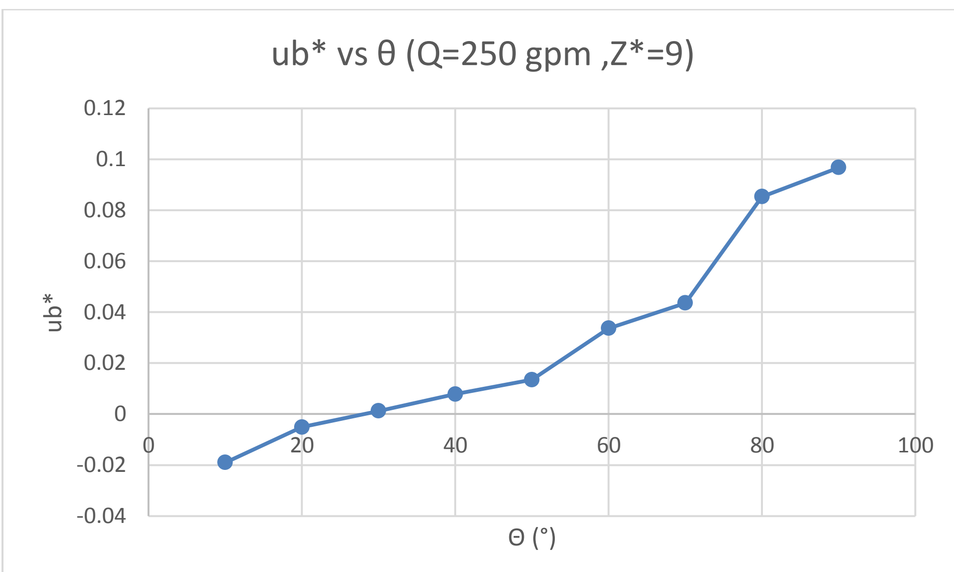

Figure 7. Velocity of bed layer versus Inclination angle with constant flowrate at point 9

The model cannot validate vertical wells as there is no bed formation in the vertical wells and nearly vertical wells. The solution diverges for vertical wells indicating that the model is inappropriate for vertical wells. The bed velocity decreases with decrease in well bore inclination indicating that the bed slides back quickly for nearly vertical wells.

To summarize we can conclude that

- The model can predict the operational envelope for given operational conditions and wellbore inclinations indicating the points of slide back bed velocity and also limiting point.
- Thus, this model can be used a initial predictive tool for developing no bed formation operating conditions
- The model is not consistent with the vertical wells
- For longer wells, the solid particles undergo a projectile and finally land up to the well bore bottom, hence the model becomes inconsistent for very long wells. The model can be coupled with particle tracking to simulate real-time cuttings transport conditions indicating the conditions at which bed does not form at all.

Nomenclature

A =cross-sectional area
C =volumetric cuttings concentration (phase fraction)
CD =drag coefficient
C_0 =cuttings concentration at bed interface
D =hydraulic diameter
_ =particle diffusion coefficient
d = particle diameter
E =absolute eccentricity
e =relative eccentricity
f =friction factor
g =gravity acceleration
p =pressure
Q =flow rate
Rt =drillpipe outer radius
Rw =wellbore radius
Red =particle Reynolds number
ROP =rate of penetration
S =wetted perimeter
vs = cuttings settling velocity
vs,y =cuttings velocity in direction normal to wellbore
Z = limiting point
f =deposition and resuspension flux
m =dynamic viscosity
q =wellbore inclination from vertical
r =density
t =shear stress

Subscripts
h =heterogeneous
b = cuttings bed
l = mud (liquid)
s =solid (cuttings)

REFERENCES

1. KAMP, A.M. and RIVERO, M., Layer Modeling for Cuttings Transport in Highly Inclined Wellbores; SPE 53942, presented at the Latin American and Caribbean Petroleum Engineering Conference, Caracas, Venezuela, April 21-23, 1999.

2. Li, Y., Bjorndalen, N., & Kuru, E. O. (2004, January 1). Numerical Modelling of Cuttings Transport in Horizontal Wells Using Conventional Drilling Fluids. Petroleum Society of Canada. doi:10.2118/2004-227

3. FORD, J.T., PEDEN, J.M., OYENEYIN, M.B., GAO, E. and ZARROUGH, R., Experimental Investigation of Drilled Cuttings Transport in Inclined Boreholes; SPE 20421, presented at the 65th Annual Technical Conference and Exhibition of the SPE held in New Orleans, LA, September 23-26, 1990.

4. Sharma, M.P.: "Cuttings Transport in Inclined Boreholes", paper OSEA-90159 presented at the 8th Offshore South Asia Conference, Singapore (4-7 December 1990).

5. Numerical simulation and experimental study of cuttings transport in intermediate inclined wells by Zhang, Feifei, Ph.D., THE UNIVERSITY OF TULSA, 2015, 261 pages; 3733239

Appendix

```
Const pi = 3.14159265358979
Const So = 2.114606015
Const Ao = 0.11413586
Const L = 1
Const uo = 0.027587584
Const Po = 3447380
Option Explicit
Function gamma1(h As Double) As Double
' functions for case 2
Dim Rw As Double
Dim rt As Double
Rw = Cells(14, 6)
rt = Cells(15, 6)
gamma1 = WorksheetFunction.Acos(Rw / rt * (1 - 2 * h))

End Function
Function gamma2(h As Double) As Double
' functions for case 1 & 2
Dim Rw As Double
Dim rt As Double
Rw = Cells(14, 6)
rt = Cells(15, 6)
gamma2 = WorksheetFunction.Acos(1 - 2 * h)
End Function
Function y1(h As Double) As Double
' function for case 1
Dim Rw As Double
Dim rt As Double
Rw = Cells(14, 6)
rt = Cells(15, 6)
y1 = rt * Sin(gamma1(h))
End Function
Function y2(h As Double) As Double
' functions for case  2
Dim Rw As Double
Dim rt As Double
Rw = Cells(14, 6)
rt = Cells(15, 6)
y2 = Rw * Sin(gamma2(h))
End Function
Function alpha(h As Double) As Double
' functions for case 3
Dim Rw As Double
Dim rt As Double
Rw = Cells(14, 6)
rt = Cells(15, 6)
alpha = WorksheetFunction.Acos(-1 + 2 * h)

End Function
Function y3(h As Double) As Double
'' functions for case 3
Dim Rw As Double
```

```
Dim rt As Double
Rw = Cells(14, 6)
rt = Cells(15, 6)
y3 = Rw * Sin(alpha(h))
End Function

Function Ab(h As Double) As Double
' wetted area of bed for all three cases
Dim Rw As Double
Dim rt As Double
Dim c As Double
Rw = Cells(14, 6)
rt = Cells(15, 6)
c = (Rw - rt) / (2 * Rw) ' clearance
If h = 0 Then Ab = 0.0001
If h <= c And h > 0 Then Ab = (gamma2(h) * (Rw) ^ 2 - y2(h) * Rw * (1 - 2 * h)) / Ao
If h > c And h <= c + rt / Rw Then
                        Ab = (gamma2(h) * (Rw) ^ 2 - gamma1(h) * (rt) ^ 2 - y2(h) * (Rw - h) + y1(h) * (Rw - h)) /
Ao
                        End If
If h > c + rt / Rw Then Ab = pi * ((Rw) ^ 2 - (rt) ^ 2) - alpha(h) * (Rw) ^ 2 + y3(h) * (h - Rw)
End Function
Function ah(h As Double) As Double 'fucntion for wetted fraction of heterogenous layer
Dim Rw As Double
Dim rt As Double
Rw = Cells(14, 6)
rt = Cells(15, 6)
ah = 1 - Ab(h)
End Function
Function Si(h As Double) ' interface length for all 3 cases
Dim Rw As Double
Dim rt As Double
Rw = Cells(14, 6)
rt = Cells(15, 6)
Dim c As Double
c = (Rw - rt)
If h <= c / (2 * Rw) Then Si = 2 * y2(h) / So
If h > c / (2 * Rw) And h <= ((c + 2 * rt) / (2 * Rw)) Then
                                 Si = 2 * (y2(h) - y1(h)) / So
                                 End If
If h > ((c + 2 * rt) / (2 * Rw)) Then Si = 2 * y3(h) / So
End Function
Function Sb(h As Double) As Double ' wetted perimeter for bed
Dim Rw As Double
Dim rt As Double
Dim c As Double
Dim d As Double
Rw = Cells(14, 6)
rt = Cells(15, 6)
c = (Rw - rt)
d = gamma2(h)
If h < (Rw - rt) / (2 * Rw) Then
   Sb = 2 * d * Rw / So
ElseIf h > c / (2 * Rw) And h <= ((c + 2 * rt) / (2 * Rw)) Then
                              Sb = (2 * gamma2(h) * Rw + 2 * gamma1(h) * rt) / So
ElseIf h > ((c + 2 * rt) / (2 * Rw)) Then Sb = 1 - 2 * alpha(h) * Rw / So
```

```
End If
End Function
Function sh(h As Double) As Double ' wetted perimeter for heterogenous layer
Dim Rw As Double
Dim rt As Double
Rw = Cells(14, 6)
rt = Cells(15, 6)
Dim c As Double
sh = 1 - Sb(h)

End Function
Function Dh(h As Double) As Double
Dim a As Double
Dim s As Double
' hydraulic diameter of heterogenous layer
Dh = 4 * ah(h) / (1 - Sb(h) + Si(h)) * Ao / So
End Function
Function Db(h As Double) ' hydraulic diameter of bed
If h = 0 Then Db = 0
If h > 0 Then Db = 4 * Ab(h) / (Sb(h) + Si(h)) * Ao / So
End Function
Function Reh(h As Double, uh As Double, ch As Double) ' Reynolds number for heterogenous layer
Dim muh As Double
muh = Cells(5, 6)
Reh = uh * Dh(h) * rhoh(ch) / muh * uo
End Function
Function Reb(h As Double, ub As Double) ' reynolds number for bed
Dim mub As Double
mub = Cells(6, 6)
Reb = ub * Db(h) * rhob() / mub * uo
If Reb = 0 Then Reb = 0.01

End Function
Function Rei(h As Double, uh As Double, ub As Double, ch As Double) ' reynolds number at interface
Dim muh As Double
muh = Cells(5, 6)
Rei = Abs(uh - ub) * Dh(h) * rhoh(ch) / muh * uo
End Function
Function fh(h As Double, uh As Double, ch As Double) ' friction factor for 1st layer
Dim re As Double
Dim e_d As Double
Dim fh1 As Double
Dim fh2 As Double
e_d = Cells(19, 6)
re = Reh(h, uh, ch)
If re < 2300 Then fh = 16 / re
If re > 2300 And e_d = 0 Then fh = 0.079 * (re) ^ -0.25
If re > 2300 And e_d > 0 Then
            fh1 = 0.079 * (re) ^ -0.25
             Do
            fh2 = (1 / (-4 * Log((e_d / 3.7) + (1.256 / (re * (fh1) ^ 0.5))))) ^ 2
            fh1 = fh2
            Loop Until Abs((fh1 - fh2) / fh1) < 0.0000000001
            fh = fh2
            End If
End Function
```

```
Function fb(h As Double, ub As Double) ' friction factor for bed
Dim re As Double
Dim e_d As Double
Dim fh1 As Double
Dim fh2 As Double
e_d = Cells(19, 6)
re = Reb(h, ub)
If re < 2300 Then fb = 16 / re
If re > 2300 And e_d = 0 Then fb = 0.079 * (re) ^ -0.25
If re > 2300 And e_d > 0 Then
   fh1 = 0.079 * (re) ^ -0.25
   Do
   fh2 = (1 / (-4 * Log((e_d / 3.7) + (1.256 / (re * (fh1) ^ 0.5))))) ^ 2
   fh1 = fh2
   Loop Until Abs((fh1 - fh2) / fh1) < 0.0000000001
 fb = fh2
End If
End Function
Function fi(h As Double, uh As Double, ub As Double, ch As Double) ' interface friction factor
Dim re As Double
re = Rei(h, uh, ub, ch)
If re < 2300 Then fi = 16 / re
If re > 2300 Then fi = 0.079 * (re) ^ -0.25
End Function
Function Tauh(h As Double, uh As Double, ch As Double) ' shear stress at wall for heterogeneous layer
Tauh = 0.5 * rhoh(ch) * fh(h, uh, ch) * (uh) ^ 2 * uo ^ 2
End Function
Function Taub(h As Double, ub As Double) ' shear stress at wall for bed layer
Taub = 0.5 * rhob * fb(h, ub) * (ub) ^ 2 * uo ^ 2
End Function
Function Taui(h As Double, uh As Double, ub As Double, ch As Double) ' shear stress at wall for interface
Taui = 0.5 * rhoh(ch) * fi(h, uh, ub, ch) * (uh - ub) ^ 2 * uo ^ 2
End Function
Function rhob() ' density of bed
Dim rhol As Double
Dim rhos As Double
rhol = Cells(7, 6)
rhos = Cells(8, 6)
Dim cb As Double
cb = Cells(12, 6)
rhob = rhol * (1 - cb) + rhos * cb
End Function
Function rhoh(ch As Double) ' density of heterogeneous layer
Dim rhol As Double
Dim rhos As Double

rhol = Cells(7, 6)
rhos = Cells(8, 6)
rhoh = ch * rhos + rhol * (1 - ch)
End Function
Function phisdep(ch As Double) ' mass flux deposition term
Dim rhos As Double
rhos = Cells(8, 6)
phisdep = ch * rhos * vsy(ch)
End Function
Function vsy(ch As Double)
```

```
Dim radian As Double
Dim teta As Double
teta = Cells(9, 6)
radian = Cells(9, 6)
vsy = vs(ch) * Cos(radian)
End Function
Function vs(ch As Double) ' evaluation of settling velocity at equlibrium conditions
Dim vs2 As Double
Dim vs1 As Double
Dim re As Double
Dim cd As Double
Dim spheri As Double
Dim tol As Double
Dim rhoh1 As Double
Dim rhol As Double
Dim rhos As Double
Dim dp As Double
Dim mul As Double
mul = Cells(5, 6)
rhol = Cells(7, 6)
rhos = Cells(8, 6)
spheri = Cells(10, 6)
dp = Cells(11, 6)
rhoh1 = rhoh(ch)
vs1 = 0.00001 ' inital solving of vs
' solve for vs frm the below equation
Do
re = rhoh1 * vs1 * dp / mul
cd = (30 / re) + 67.289 / (Exp(5.03 * spheri))
vs2 = (2 / 3 * (rhos / rhol - 1) * 9.81 * dp / cd) ^ 0.5
tol = (vs2 - vs1) / vs1
vs1 = vs2
Loop Until Abs(tol) < 0.0000000001
vs = vs2
End Function
Function uti(h As Double, uh As Double, ub As Double, ch As Double)
uti = (Abs(Taui(h, uh, ub, ch)) / rhoh(ch)) ^ 0.5 ' interface friction velocity
End Function
Function phisus(h As Double, uh As Double, ub As Double, ch As Double) ' mass flux suspension term
Dim uti1 As Double
Dim vsy1 As Double
Dim vs1 As Double
Dim radian As Double
Dim zc As Double
Dim zm As Double
Dim z As Double
Dim rep As Double
Dim rhol As Double
Dim rhos As Double
Dim dp As Double
Dim teta As Double
Dim mul As Double

teta = Cells(9, 6)
dp = Cells(11, 6)
rhol = Cells(7, 6)
```

```
rhos = Cells(8, 6)
mul = Cells(5, 6)
vs1 = vs(ch)
radian = Cells(9, 6)
vsy1 = vs1 * Cos(radian)
rep = ((rhos / rhol - 1) * 9.81 * dp) ^ 0.5 * dp * rhol / mul
zc = 5
zm = 13.2
z = rep * uti(h, uh, ub, ch) / vsy1
' heavy side function expansion for 3 cases
If z >= zm Then phisus = rhos * vs1 * 0.3
If z < zm And z > zc Then phisus = rhos * vs1 * 3 * z * z * z * z * z * z * z * z * z * z * (1 - zc / z) /
1000000000000#
If z <= zc Then phisus = 0
End Function
Function phis(h As Double, uh As Double, ub As Double, ch As Double)
phis = phisdep(ch) - phisus(h, uh, ub, ch)
End Function
Function phil(h As Double, uh As Double, ub As Double, ch As Double)
Dim rhol As Double
Dim rhos As Double
Dim cb As Double
cb = Cells(12, 6)
rhol = Cells(7, 6)
rhos = Cells(8, 6)
phil = (1 - cb) / cb * rhol / rhos * phis(h, uh, ub, ch)
End Function
Function dch_dz(h As Double, uh As Double, ub As Double, ch As Double) ' derivative function of Ch
Dim rhos As Double
Dim rhol As Double
rhos = Cells(8, 6)
rhol = Cells(7, 6)
dch_dz = (-1 * phis(h, uh, ub, ch) * Si(h) * So * L / (rhos * uo * Ao * uh * ah(h))) + (ch * phis(h, uh, ub, ch) *
Si(h) * So * L / (uo * Ao * rhos * uh * ah(h))) + _
(ch * phil(h, uh, ub, ch) * Si(h) * So * L / (uo * Ao * rhol * uh * ah(h)))
End Function
Function duh_dz(h As Double, uh As Double, ub As Double, ch As Double) ' derivative function of uh
Dim rhol As Double
Dim rhos As Double
rhol = Cells(7, 6)
rhos = Cells(8, 6)
duh_dz = L / ah(h) / uo / Ao * (((phis(h, uh, ub, ch) / rhos) + (phil(h, uh, ub, ch) / rhol)) * (-Si(h) * So) - uh *
uo * Ao * dah_dz(h, uh, ub, ch) / L)
End Function
Function dub_dz(h As Double, uh As Double, ub As Double, ch As Double) ' derivative function of ub
Dim e As Double
If h < 0.01 Then e = 0
If h > 0.01 Then e = Si(h) / Ab(h)
dub_dz = ub * dah_dz(h, uh, ub, ch) + (phil(h, uh, ub, ch) + phis(h, uh, ub, ch)) / rhob * e * So * 1 / Ao / uo
'1 / Ab(h) * (ub * dah_dz(h, uh, ub, ch) + (phil(h, uh, ub, ch) + phis(h, uh, ub, ch)) / rhob * Si(h))
End Function
Function dh_dz(h As Double, uh As Double, ub As Double, ch As Double) ' derivative function of h
Dim teta As Double
Dim radian As Double
Dim rhol As Double
Dim rhos As Double
```

```
Dim Rw As Double
Dim rt As Double
Dim c As Double
Dim d As Double
Dim e As Double

Rw = Cells(14, 6)
rt = Cells(15, 6)
c = 0.5 * (Rw - rt) / Rw
e = (1 - (1 - 2 * h)) ^ 0.5
d = 2 - 2 * Cos(gamma2(h)) + 4 * h * Cos(gamma2(h)) + 2 * Sin(gamma2(h)) * e
rhol = Cells(7, 6)
rhos = Cells(8, 6)
teta = Cells(9, 6)
radian = 22 / 7 * teta / 180


If h <= c Then
            e = 0.001
            d = 0.001
           dh_dz = -dah_dz(h, uh, ub, ch) * Ao / Rw * e / d
           End If
If h > c And h < (c + 2 * rt / (2 * Rw)) Then dh_dz = -dah_dz(h, uh, ub, ch) * Ao / ((2 * (Rw) ^ 2 / e) - (2 * (rt) ^ 2 * (Rw / rt) / (1 - (Rw / rt * (1 - 2 * h))) ^ 0.5) - (Rw * (1 - 2 * h) * 2 * Rw * Cos(gamma2(h)) / e) + (Rw * (1 - 2 * h) * 2 * Rw * Cos(gamma1(h)) / (1 - (Rw / rt * (1 - 2 * h))) ^ 0.5) + ((y2(h) + y1(h)) * 2 * Rw))
If h > (c + 2 * rt / (2 * Rw)) Then dh_dz = dah_dz(h, uh, ub, ch) * Ao * e / (4 * (Rw) ^ 2 * h * Cos(alpha(h)) - 2 * (Rw) ^ 2 * Cos(alpha(h)) - 2 * (Rw) ^ 2 - 2 * y3(h) * Rw * e)
dh_dz = Abs(dh_dz)
End Function
Function dah_dz(h As Double, uh As Double, ub As Double, ch As Double) ' derivative function of Ah
Dim teta As Double
Dim radian As Double
Dim rhol As Double
Dim rhos As Double
Dim Rw As Double
Dim rt As Double
Dim c As Double
Dim e As Double
Rw = Cells(14, 6)
rt = Cells(15, 6)
c = Rw - rt
rhol = Cells(7, 6)
rhos = Cells(8, 6)
teta = 0
radian = Cells(9, 6)
If h <= 0.01 Then e = 0
If h > 0.1 Then e = (Si(h) / Ab(h))
dah_dz = (L / uo ^ 2) * ((uh) ^ 2 * (uo) ^ 2 / L * dch_dz(h, uh, ub, ch) * (rhol - rhos) + (rhob - rhoh(ch)) * 9.8 * Sin(radian) * L + (Taub(h, ub) * Sb(h) * So / Ab(h) / Ao) - (Tauh(h, uh, ch) * sh(h) * So / Ao / ah(h)) _
- ((phis(h, uh, ub, ch) + phil(h, uh, ub, ch)) * (uh - ub) + Taui(h, uh, ub, ch)) * So / Ao * ((Si(h) / ah(h)) + e) + _
-2 * rhoh(ch) * uo * uh / Ao / ah(h) * (phis(h, uh, ub, ch) / rhos + phil(h, uh, ub, ch) / rhol) * Si(h) * So + 2 * ub * uo / Ao / Ab(h) * (phis(h, uh, ub, ch) + phil(h, uh, ub, ch)) * Si(h) * So) _
/ (-1 * (rhoh(ch) * (uh) / ah(h) + rhob * (ub) / Ab(h)))
dah_dz = (dah_dz)
End Function
```

```
Function dp_dz(h As Double, uh As Double, ub As Double, ch As Double) ' derivative of P
Dim teta As Double
Dim radian As Double

radian = Cells(9, 6)
dp_dz = (1 / Po) * (rhob * (uo) ^ 2 * ub * dah_dz(h, uh, ub, ch) / Ab(h) / L - 2 * rhob * uo ^ 2 * ub * dub_dz(h,
uh, ub, ch) / Ab(h) / L _
+ rhob * 9.8 * Sin(radian) * L - Taub(h, ub) * Sb(h) * So * L / Ab(h) / Ao + Taui(h, uh, ub, ch) * Si(h) * So * L
/ Ab(h) / Ao + (phis(h, uh, ub, ch) _
+ phil(h, uh, ub, ch)) * Si(h) * So * L * (uh - ub) / Ab(h) / Ao)
End Function
Sub adv_drilling()
Dim z As Integer
Dim dz As Double
Dim length As Double
Dim nlsteps As Integer
Dim i As Integer
Dim ach() As Double
Dim auh() As Double
Dim a1h() As Double
Dim aub() As Double
Dim ap() As Double
Dim ach1 As Double
Dim auh1 As Double
Dim a1h1 As Double
Dim aub1 As Double
Dim ap1 As Double
Dim Rop As Double
Dim qW As Double
Dim Rw As Double
Dim rt As Double
Dim rhos As Double
Dim rhol As Double
rhol = Cells(7, 6)
rhos = Cells(8, 6)
Rop = Cells(13, 6)
Rw = Cells(14, 6)
rt = Cells(15, 6)
qW = Cells(16, 6)
length = Cells(18, 6)
nlsteps = 50
dz = 0.0002

ReDim ach(nlsteps + 1) As Double
ReDim auh(nlsteps + 1) As Double
ReDim a1h(nlsteps + 1) As Double
ReDim aub(nlsteps + 1) As Double
ReDim ap(nlsteps + 1) As Double


Dim k1 As Double
Dim k2 As Double
Dim k3 As Double
Dim k4 As Double
Dim l1 As Double
Dim l2 As Double
```

```
Dim l3 As Double
Dim l4 As Double
Dim m1 As Double
Dim m2 As Double
Dim m3 As Double
Dim m4 As Double
Dim n1 As Double
Dim n2 As Double
Dim n3 As Double
Dim n4 As Double
Dim o1 As Double
Dim o2 As Double
Dim o3 As Double
Dim o4 As Double

' all are dimensionless
auh(0) = 1
ach(0) = Rop / (7 * qW / (22 * ((Rw) ^ 2 - (rt) ^ 2)) * (1 - (rt / Rw) ^ 2))
ap(0) = 1
a1h(0) = 0
aub(0) = 0

For z = 0 To nlsteps - 1
' passing the parameters into function
ach1 = ach(z)
auh1 = auh(z)
a1h1 = a1h(z)
aub1 = aub(z)
ap1 = ap(z)
' evaluating runga kutta constants
k1 = dch_dz(a1h1, auh1, aub1, ach1) * dz
l1 = dh_dz(a1h1, auh1, aub1, ach1) * dz
m1 = duh_dz(a1h1, auh1, aub1, ach1) * dz
n1 = dub_dz(a1h1, auh1, aub1, ach1) * dz
o1 = dp_dz(a1h1, auh1, aub1, ach1) * dz

k2 = dch_dz(a1h1 + l1 / 2, auh1 + m1 / 2, aub1 + n1 / 2, ach1 + k1 / 2) * dz
l2 = dh_dz(a1h1 + l1 / 2, auh1 + m1 / 2, aub1 + n1 / 2, ach1 + k1 / 2) * dz
m2 = duh_dz(a1h1 + l1 / 2, auh1 + m1 / 2, aub1 + n1 / 2, ach1 + k1 / 2) * dz
n2 = dub_dz(a1h1 + l1 / 2, auh1 + m1 / 2, aub1 + n1 / 2, ach1 + k1 / 2) * dz
o2 = dp_dz(a1h1 + l1 / 2, auh1 + m1 / 2, aub1 + n1 / 2, ach1 + k1 / 2) * dz
k3 = dch_dz(a1h1 + l2 / 2, auh1 + m2 / 2, aub1 + n2 / 2, ach1 + k2 / 2) * dz
l3 = dh_dz(a1h1 + l2 / 2, auh1 + m2 / 2, aub1 + n2 / 2, ach1 + k2 / 2) * dz
m3 = duh_dz(a1h1 + l2 / 2, auh1 + m2 / 2, aub1 + n2 / 2, ach1 + k2 / 2) * dz
n3 = dub_dz(a1h1 + l2 / 2, auh1 + m2 / 2, aub1 + n2 / 2, ach1 + k2 / 2) * dz
o3 = dp_dz(a1h1 + l2 / 2, auh1 + m2 / 2, aub1 + n2 / 2, ach1 + k2 / 2) * dz
k4 = dch_dz(a1h1 + l3, auh1 + m3, aub1 + n3, ach1 + k3) * dz
l4 = dh_dz(a1h1 + l3, auh1 + m3, aub1 + n3, ach1 + k3) * dz
m4 = duh_dz(a1h1 + l3, auh1 + m3, aub1 + n3, ach1 + k3) * dz
n4 = dub_dz(a1h1 + l3, auh1 + m3, aub1 + n3, ach1 + k3) * dz
o4 = dp_dz(a1h1 + l3, auh1 + m3, aub1 + n3, ach1 + k3) * dz
' runga kutta approximation
ach(z + 1) = ach(z) + (k1 + 2 * (k2 + k3) + k4) / 4
a1h(z + 1) = a1h(z) + (l1 + 2 * (l2 + l3) + l4) / 4
auh(z + 1) = auh(z) + (m1 + 2 * (m2 + m3) + m4) / 4
aub(z + 1) = aub(z) + (n1 + 2 * (n2 + n3) + n4) / 4
```

```
ap(z + 1) = ap(z) + (o1 + 2 * (o2 + o3) + o4) / 4
Cells(z + 20, 7) = ach(z + 1)
Cells(z + 20, 8) = a1h(z + 1)
Cells(z + 20, 9) = auh(z + 1)
Cells(z + 20, 10) = aub(z + 1)
Cells(z + 20, 11) = ap(z + 1)
Next z

For z = 0 To nlsteps

Next z


End Sub
```