ISSN: 2581-4311
DOI: doi.org/10.51219/IJCRSET/Ekrem-Alaggoz/124

URF PUBLISHERS
connect with research world

International Journal of Current Research in Science, Engineering & Technology

https://urfpublishers.com/journal/ijcrset

Vol: 7 & Iss: 1

Research Article

Computational Tool for Wellbore Stability Analysis and Mud Weight Optimization v1.0

Ekrem Alagoz*, Ahmet E. Mengen, Fethi Bensenouci and Emre Can Dundar

Turkish Petroleum Corporation (TPAO), R&D Department, Sogutozu Nizami Gencevi Street No:10, 06510 Cankaya, Ankara, Turkey

Citation: Alagoz E, Mengen AE, Bensenouci F, Dundar EC. Computational Tool for Wellbore Stability Analysis and Mud Weight Optimization v1.0. *Int J Cur Res Sci Eng Tech* 2023; 7(1), 1-5. DOI: doi.org/10.51219/IJCRSET/Ekrem-Alaggoz/124

Received: 11 January, 2024; **Accepted:** 24 January, 2024; **Published:** 27 January, 2024

***Corresponding author:** Ekrem Alagoz, Turkish Petroleum Corporation (TPAO), R&D Department, Sogutozu Nizami Gencevi Street No:10, 06510 Cankaya, Ankara, Turkey, Email: ealagoz@tpao.gov.tr

ABSTRACT

This paper presents the development of a computational tool for wellbore stability analysis and mud weight optimization, written in Python language. The tool utilizes well log data as input parameters and correlations to calculate stresses, and performs wellbore stability analysis to create a mud weight window. The computational tool's performance is evaluated and compared to traditional methods, and a case study is presented. The results demonstrate that the computational tool provides an efficient and accurate solution for wellbore stability analysis and mud weight optimization. Future research directions are also discussed.

Keywords: Wellbore stability analysis; Mud weight optimization

1. Introduction

Wellbore stability analysis is a critical aspect of drilling operations in the oil and gas industry. It involves assessing the mechanical integrity of the wellbore and ensuring its stability to prevent issues such as wellbore collapse, formation damage, and lost circulation. Mud weight optimization, on the other hand, refers to the process of determining the optimal density of drilling mud to maintain wellbore stability while minimizing the risk of formation damage and ultimately reduce NPTs[1]. Traditionally, wellbore stability analysis and mud weight optimization have relied on manual calculations and empirical models, which can be time-consuming and may not always yield accurate results. With the advancement of computational tools and programming languages, there is a growing opportunity to develop more efficient and reliable solutions. The development of a computational tool for wellbore stability analysis and mud weight optimization addresses the need for a more streamlined and accurate approach in the industry. By automating the calculation process and utilizing well-log data and correlations, the tool can provide a faster and more precise evaluation of wellbore stability conditions[2]. The significance of this computational tool lies in its potential to improve drilling efficiency, reduce operational risks, and optimize drilling mud costs. By enabling engineers and drilling professionals to make well-informed decisions based on comprehensive analysis, the tool can contribute to safer and more cost-effective drilling operations.

The primary objectives of this paper are as follows:

- To present the development of a computational tool for wellbore stability analysis and mud weight optimization.
- To describe the methodology employed in the tool, including the utilization of well log data, correlations, and stress calculations.
- To evaluate the performance of the computational tool and compare it with traditional methods commonly used in the industry.
- To present a case study demonstrating the application of the computational tool and its outcomes.
- To provide the Python code implementation of the tool.
- To discuss the accuracy, efficiency, and potential future research directions for the computational tool.

2. Methodology

The development of the computational tool involved a systematic approach to address wellbore stability analysis and mud weight optimization. The tool was designed and implemented using the Python programming language to ensure flexibility, ease of use, and compatibility with existing industry practices. The computational tool utilizes various input parameters derived from well log data which are tabulated in (**Table 1**). Additionally, the tool incorporates empirical correlations established through previous research and field data to estimate the relevant geomechanical properties of the formation. One of the key components of the computational tool is the calculation of stresses within the wellbore. By analyzing well log data, the tool employs relevant equations and correlations to determine the magnitudes and orientations of the stresses. This information is crucial for assessing wellbore stability conditions and potential failure mechanisms.

Table 1. Well Log input parameters used in the calculations.

Input Parameters	Units
Depth	m, meter
Bit size	in, inches
caliper	in, inches
DT_compression	µs/ft
DT_shear	µs/ft
Gamma Ray	API
Rhob (density)	g/cc
V_clay (clay volume)	percent

Calculations have been performed in three domains. The first one is the calculation of rock properties. Then, using the calculated rock properties, stresses have been calculated using densities and poroelastic model. Subsequently, wellbore stability analysis based on Mohr-Coulomb criteria has been conducted. Finally, a mud weight window graph has been generated. These steps will be discussed sequentially.

2.1. Rock Properties Calculations

Drilling case is primarily based on the interpretation of rock properties, determination of stresses around the wellbore, in-situ stress characterization, and rock failure criteria[3]. The equations used in the calculation of rock properties are listed below as Equations 1-13. Dynamic elastic properties are converted to static using Equations 5-6[4], whereas Equation 8 is used to calculate UCS calculation[5]. Tensile strength or TSTR is assumed equivalent to 10 % of UCS[4].

$G_{dyn} = (13474.45)*(\rho_b / (\Delta t_{shear})^2)$.................................(1)

$K_{dyn} = (13474.45)*(\rho_b)\,[(\Delta t_{comp})^{-2}] - (4/3)*G_{dyn}$............(2)

$E_{dyn} = (9* G_{dyn*} K_{dyn}) / (G_{dyn} + 3* K_{dyn})$..........................(3)

$PR_{dyn} = [0.5*(\Delta t_{shear} / \Delta t_{comp})^2) - 1] / [(\Delta t_{shear} / \Delta t_{comp})^2 - 1]$........(4)

With G_{dyn} is dynamic shear modulus in Mpsi, K_{dyn} is dynamic bulk modulus in Mpsi, ρ is bulk density in g/cc, E_{dyn} is dynamic Young's modulus in Mpsi and υ_{dyn} is dynamic Poisson's ratio; Δt_{shear} and Δt_{comp} are shear slowness and compressional slowness (both in µs/ft), respectively.

$YME_{stat} = 0.6 * YME_{dyn}$ (for shalaes).............................(5)

$YME_{stat} = 0.8 * YME_{dyn}$ (for sandstones).........................(6)

$PR_{stat} = PR_{dyn}$ (assumed)...(7)

$UCS = 2.280 + 4.1089*E_{stat}$ (Plumb, 1994).......................(8)

TSTR = 0.1 * UCS (assumed)....................................(9)

Vclay > 0.45 (for shales)...(10)

Vclay < 0.45 (for sandstones)....................................(11)

F_{ANG} = 30deg (in shals)..(12)

F_{ANG} = 40deg (in sandstones)..(13)

2.2. Stress Calculations

Understanding of stress conditions is an essential part of drilling and instability analysis. Before drilling operation, stresses are considered in cartesien coordinate system. Once drilling begins, a cyclindrical shape is presented and the analysis is considered in cylindrical coordonate system[6]. In this section, minimum and maximum horizontal stresses have been calculated using the rock properties obtained in the previous section. The equations used in these calculations are listed below as Equation 14-22[4]

$\sigma_h = [v/(1 - v)]*(\sigma_v - \alpha*P_p) + \alpha*P_p + [E / (1 - v^2)]*\varepsilon_h + [(v*E) / (1 - v^2)]*\varepsilon_H$..(14)

$\sigma_H = [v/(1 - v)]*(\sigma_v - \alpha*P_p) + \alpha*P_p + [(v*E) / (1 - v^2)]*\varepsilon_h + [E / (1 - v^2)]*\varepsilon_H$..(15)

$Sv = depth(m)*3.281*\sigma_v$........................(16)

Pp = depth(m)*3.281* (Pore pressure gradient)...............(17)

The overburden and pore pressure gradients are considered here as 1 psi/ft and 0.55 psi/ft respectively. A Closure gradient of 0.79 psi/ft is used to calibrate minimum horizontal stress.

For wellbore stability analysis effective stresses are used:

$\sigma_{h\text{-}min}$ (effective) = $\sigma_{h\text{-}min}$ (calculated) – Pore Pressure ....(18)

$\sigma_{h\text{-}max}$ (effective) = $\sigma_{h\text{-}max}$ (calculated) – Pore Pressure ....(19)

2.3. Failure Criteria

Analysis for the determination of the rock requires to apply failure criteria. One of the most common and applied failure criteria is Mohr-Coulomb. It uses the unconfined compressive strength (UCS) and the Friction Angle (FANG) to construct the yield envelope (**Figure 1**)[7].

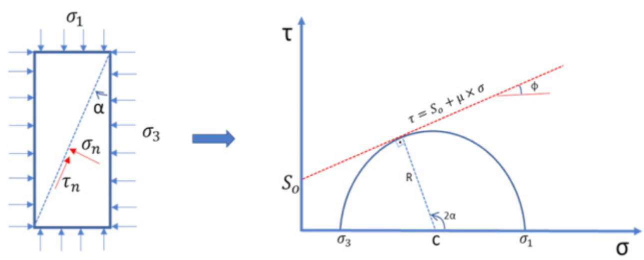

Figure 1: Mohr-Coulomb representation under triaxial test[7]

2.4. Wellbore stability analysis and mud weight optimization process

The computational tool performs wellbore stability analysis by evaluating the mechanical integrity of the wellbore under various operational conditions. It considers factors such as formation strength, pore pressure, mud weight, and wellbore geometry. Based on the calculated stresses and other relevant parameters, the tool determines the stability of the wellbore and identifies potential issues that may arise during drilling operations such as drill-pipe sticking, lost circulation, wellbore collapse, and non-productive time[8]. Furthermore, the tool facilitates mud weight optimization required to limit drilling problems by generating a mud weight window i.e MWW[9]. MWW is defined by four boundaries (**Figure 2**):

From the left, the first is the pore pressure limit, which should be exceeded to maintain an overbalance in the well and avoid kicks.

The second lowest boundary, is the shear failure limit. If the mud weight drops below this limit shear failure at the wellbore wall occurs. This can lead to diametrically opposing pairs of breakouts.

The third limit from the left is fracture gradient assumed overall as mud loss limit. It is defined by the least principal stress in equivalent mud weight. When the mud weight exceeds this limit mud losses can start to occur as a result of pre-existing fractures being opened.

The fourth limit represents the tensile failure of the rock (i.e Breakdown), is caused when the mud weight is too high. Severe mud losses can occur when this pressure is exceeded in intact formations.

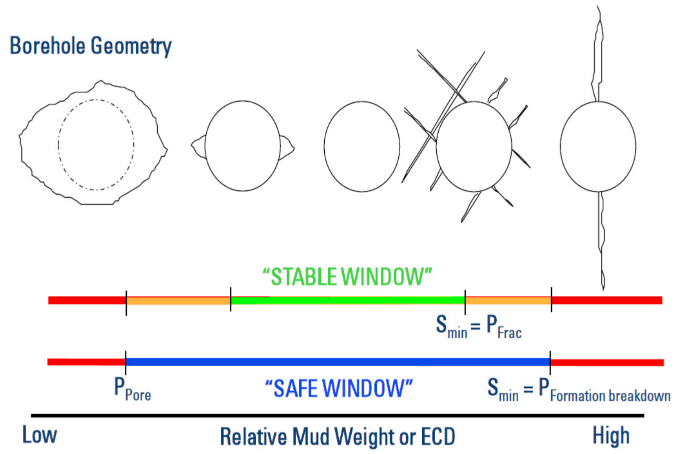

Figure 2: Mud Weight Window Explanation.

The principal stress tensor around the borehole is computed as a function of the far field principal stresses, mud weight, the borehole orientation and azimuth with respect to the principal stress axes[4]. The Mohr-Coulomb criterion can thus be used to predict the minimum mud weight needed to prevent shear failure of the wellbore. The equations used in these calculations are listed below as Equations 23-30.

$N = [1 + \sin(F_{ANG})] / [1 - \sin(F_{ANG})]$............................(23)

P_w = (Mud Weight/2.31)*Normal Gradient*Depth(m)*3.281 ......(24)

$\sigma_R = Pw - Pp$ ....(25)

$\sigma_{\Theta min} = 3*[\sigma_{h\text{-}min}(\text{effective})] - [\sigma_{h\text{-}max}(\text{effective})] - \sigma_R$........(26)

$\sigma_{\Theta max} = 3*[\sigma_{h\text{-}max}(\text{effective})] - [\sigma_{h\text{-}min}(\text{effective})] - \sigma_R$........(27)

Kick Limit (Pore pressure) = 8.34*2.31*Pp / (Depth*3.281) ....(28)

Break out Point (Shear Failure) = ($3*[\sigma_{h\text{-}max}(\text{effective})] - [\sigma_{h\text{-}min}(\text{effective})]$ + Pp – UCS*1000 + N*Pp) / ([1 + N]*[3.281*depth(m)*0.052])....(29)

Fracture Gradient (Sh-min) = (8.34*2.31* $\sigma_{h\text{-}min}$) / (Depth*3.281).........(30)

Breakdown limit (Tensile) = ($3*[\sigma_{h\text{-}min}(\text{effective})] - [\sigma_{h\text{-}max}(\text{effective})]$ + Pp + 145.037*TSTR) / ([3.281*depth (m)*0.052])...............(31)

3. Case Study and Implementation of Codes

To demonstrate the practical application of the developed computational tool for wellbore stability analysis and mud weight optimization, real data is used. The wellbore stability analysis and mud weight optimization results obtained from the computational tool are smmarized in (**Figure 3**). The analysis includes the assessment of wellbore stability conditions, identification of potential failure mechanisms, and the determination of an optimal mud weight range for maintaining wellbore stability while minimizing formation damage. The results are presented in a clear and concise manner. For instance, it is easy to notice from (**Figure 3**) that a mud weight of 16 ppg would be able to stabilise the boreholes walls, avoid formations kicks, mud losses and formation breakdown.

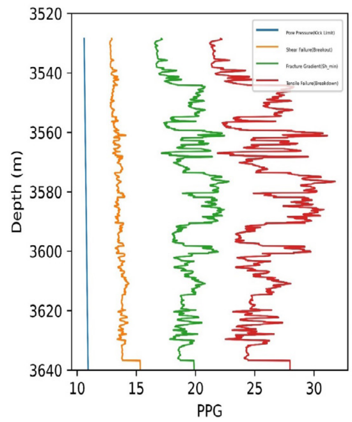

Figure 3: Example of wellbore stability analysis using our Program.

4. Conclusion

The equations used in Equation 5-13 can be modified based on the laboratory studies of the fields for which the data will be used. Additional correlation will be added to enrich the program library.

- Biot constant can be modified and instead of assuming 1.
- The code successfully calculated rock properties, in-situ stresses using real data which allows to perform wellbore stability analysis and define the optimum mud weight for one well section
- This code will be converted into a program with interface to be much more user friendly.
- More applications like Pore Pressure Prediction and impact of wellbore trajectory are planned to be developped.

Appendix

The appendix section provides a description of the Python codes that have been included as supplementary material to the paper. Each code is briefly explained, highlighting its purpose and functionalities within the context of wellbore stability analysis and mud weight optimization.

```
“””
Created on Wed Apr 12 12:26:09 2023
@author: ealagoz
“””
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# Load log data from Excel file
df = pd.read_excel(‘04-Well_A_data.xlsx’)
```

```
# Calculate rock properties
Rhob = df[‘Rhob’]
DT_shear = df[‘DT_shear’]
DT_comp = df[‘DT_comp’]
V_clay = df[‘V_clay’]
Bit_size = df[‘bit size’]
depth = df[‘depth’]
caliper = df[‘caliper’]
G = (13474.45 * Rhob) / (DT_shear**2)
K = (13474.45 * Rhob / (DT_comp**2)) - (4/3) * G
YME_dyn = (9*G*K) / (3*K+G)
PR_dyn = (0.5*(DT_shear/DT_comp)**2-1) / ((DT_shear/DT_comp)**2-1)
df[‘Rock Type’] = np.where(V_clay > 0.45, ‘Shale’, ‘Sandstone’)
df[‘FANG’] = np.where(df[‘Rock Type’] == ‘Shale’, 30, 40)
df[‘YME_stat’] = np.where(df[‘Rock Type’] == ‘Shale’, 0.6 * YME_dyn, 0.8 * YME_dyn)
YME_stat_GPa = 6.895 * YME_dyn
UCS_MPa = (4.1089 * YME_stat_GPa + 2.28) / 2
UCS_Mpsi = UCS_MPa / 6.895
TSTR_MPa = 0.1 * UCS_MPa
# User inputs
normal_gradient = float(input(“Enter normal gradient in psi/ft: “))
pore_pressure_gradient = float(input(“Enter pore pressure gradient in psi/ft: “))
closure_gradient = float(input(“Enter closure gradient in psi/ft: “))
alpha = float(input(“Enter biot constant, alpha: “))
MW_SG = float(input(“Enter mud weight in SG: “))
# Calculate stresses
df[‘Sv’] = df[‘depth’] * 3.281 * normal_gradient
df[‘Pp’] = df[‘depth’] * 3.281 * pore_pressure_gradient
df[‘Sh_min’] = df[‘depth’] * 3.281 * closure_gradient
# Calculate PR_stat
PR_stat = PR_dyn
# Calculate teta_hmin and teta_hmax
epsilon_h = 0.0005
epsilon_H = 0.00108
teta_hmin = (PR_stat/(1-PR_stat)) * (df[‘Sv’]-alpha*df[‘Pp’]) + alpha*df[‘Pp’] + (df[‘YME_stat’]/(1-PR_stat**2))*epsilon_h + ((PR_stat*df[‘YME_stat’])/(1-PR_stat**2))*epsilon_H
teta_hmax = (PR_stat/(1-PR_stat)) * (df[‘Sv’]-alpha*df[‘Pp’]) + alpha*df[‘Pp’] + ((PR_stat*df[‘YME_stat’])/(1-PR_stat**2))*epsilon_h + (df[‘YME_stat’]/(1-PR_stat**2))*epsilon_H
teta_hmin_eff = teta_hmin - df[‘Pp’]
teta_hmax_eff = teta_hmax - df[‘Pp’]
FANG_rad = np.radians(df[‘FANG’])
# Calculate N
N = (1 + np.sin(FANG_rad)) / (1 - np.sin(FANG_rad))
# Calculate pore pressure, tensile, and shear failure
Pw = (MW_SG/2.31)*normal_gradient*df[‘depth’]*3.281
sigma_teta_R = Pw - df[‘Pp’]
sigma_teta_min = (3*teta_hmin_eff) - teta_hmax_eff - sigma_teta_R
sigma_teta_max = (3*teta_hmax_eff) - teta_hmin_eff - sigma_teta_R
Pore_Pressure = (8.34 * 2.31 * df[‘Pp’]) / (df[‘depth’] * 3.281)
Shear_Failure = ((3*teta_hmax_eff) - teta_hmin_eff + df[‘Pp’] - UCS_Mpsi*1000 + N * df[‘Pp’]) / ((1+N)*(3.281*df[‘depth’]*0.052))
Tensile_Failure = ((3*teta_hmin_eff) - teta_hmax_eff + df[‘Pp’] - TSTR_MPa*145.037) / ((3.281*df[‘depth’]*0.052))
Fracture_Gradient = (8.34 * 2.31 * teta_hmin) / (df[‘depth’] * 3.281)
# plot Depth vs Pore Pressure, Shear Failure, Tensile Failure and Fracture Gradient
plt.plot(Pore_Pressure, df[‘depth’], label=’Pore Pressure(Kick Limit)’)
plt.plot(Shear_Failure, df[‘depth’], label=’Shear Failure(Breakout)’)
plt.plot(Tensile_Failure, df[‘depth’], label=’Tensile Failure(Breakdown)’)
plt.plot(Fracture_Gradient, df[‘depth’], label=’Fracture Gradient(Sh_min)’)
plt.gca().invert_yaxis()
#plt.xlim(4, 24) # Set x-axis limits
#plt.ylim(3520, 3660) # Set y-axis limits
plt.xlabel(‘PPG’)
plt.ylabel(‘Depth (m)’)
plt.legend()
plt.savefig(‘figure2.jpg’, dpi=1000)
plt.show()
```

4. References

1. Cole P, Young K, Doke C, Duncan N, Eustes B. Geothermal drilling: a baseline study of nonproductive time related to lost circulation. In Proceedings of the 42nd Workshop on Geothermal Reservoir Engineering 2017;13-15.
2. Zhang J, Keaney G, Standifird W. Wellbore stability with consideration of pore pressure and drilling fluid interactions. OnePetro 2006.
3. Alpkiray M, Nguyen T, Saasen A. Thermal Effects on Wellbore Stability and Fluid Loss in High-Temperature Geothermal Drilling. AADE Conference, Houston 2022.
4. Fjaer E, Holt RM, Horsrud P, Raaen AM. Petroleum Related Rock Mechanics. Elsevier, Amsterdam 2008; 53

5. Plumb RA. Influence of composition and texture on the failure properties of clastic rocks. Paper presented at the Rock Mechanics 1994.

6. Jaeger JC, Cook NG, Zimmerman R. Fundamentals of rock mechanics. John Wiley & Sons 2007.

7. Alpkiray M. Implementation of Geomechanical Approach in Geothermal Drilling During Lost Circulation Phenomena. New Mex Inst Min Technol 2021.

8. Aadnoy BS, Fazaelizadeh M, Hareland G. A 3d analytical model for wellbore friction. J Can Pet Technol 2010; 49(10):25-36

9. Zoback Mark, Daniel Moos, Larry Mastin, Roger NA. Well bore breakouts and in situ stress. J Geophys Res Solid Earth 1985;5523-5530.

10. Zoback M. Reservoir Geomechanics. 2010