Table of Contents

[**Documentation for plotWellTrajectory Solution (Group 04)** 2](#_Toc182842506)

[1. Introduction 3](#_Toc182842507)

[Key Features: 3](#_Toc182842508)

[2. System Requirements and Setup 3](#_Toc182842509)

[System Requirements: 3](#_Toc182842510)

[Dependencies: 4](#_Toc182842511)

[Installation: 4](#_Toc182842512)

[3. Inputs for plotWellTrajectory Function 4](#_Toc182842513)

[Required Inputs 4](#_Toc182842514)

[1. Surface Location: 4](#_Toc182842515)

[2. Pay Zones: 4](#_Toc182842516)

[3. Pay Zone Shapes: 5](#_Toc182842517)

[4. Kick-Off Point Depth (KOP): 5](#_Toc182842518)

[5. Interval: 5](#_Toc182842519)

[Optional Arguments: 5](#_Toc182842520)

[4. Step-by-Step Usage 6](#_Toc182842521)

[5. Outputs 6](#_Toc182842522)

[6. Explanation of Key Parts in the Code 7](#_Toc182842523)

[1. Path Initialization 7](#_Toc182842524)

[2. Transition to Kick-Off Point 7](#_Toc182842525)

[3. Calculation of Path to Pay Zones 7](#_Toc182842526)

[4. Optimization of Pay Zone Sequence 8](#_Toc182842527)

[4. Plotting and Coating 8](#_Toc182842528)

[5. Excel Data Export 8](#_Toc182842529)

[7. Contact for Support 9](#_Toc182842530)

[8. Appendices 9](#_Toc182842531)

# **Documentation for plotWellTrajectory Solution (Group 04)**

**Project Title:** Hitting the targets

**Course Title:** Drilling methods ɪ

**Course Code:** PGG 317

**Course lecturer:** Dr Darlington Etaje

**Group Number:** 04

**Team Members and their roles:**

Project Manager: Adebisi Adesanwo Israel

Software Developer: Deboh-Ajiga Damisola

Quality Control: Ahmed Afolasewa Mary

Technical Expert: Oladosu Fawaz Olatunbosun.M

# 1. Introduction

* The **plotWellTrajectory** solution is a MATLAB program that enables users to visualize and optimize the **drilling path** of a well that passes through specified **pay zones**. This tool is particularly useful in **drilling engineering** to determine the optimal sequence for reaching multiple pay zones, as well as visualizing both the **3D well trajectory** and the geometry of the pay zones in **2D and 3D**.
* The project was developed with the following objectives:

1. To facilitate the **visual representation** of a well's trajectory through multiple target reservoirs.
2. To implement **optimization algorithm** to minimize the total well path length( Total distance traveled).
3. To provide a **data export feature** that allows users to save trajectory information (MIANET data)in an Excel file for further analysis.

## Key Features:

* **3D Trajectory Visualization** with labeled pay zones.
* **Optimization of Pay Zone Order** to minimize drilling distance.
* **Interactive 2D Visualization** for a top-down view of the well trajectory.
* **Excel Export** for saving detailed drilling information.

# 2. System Requirements and Setup

## System Requirements:

* **MATLAB R2020a** or later.
* **Optimization Toolbox** (optional): Required to enable optimal pay zone sequencing.

## Dependencies:

* Uses MATLAB built-in functions like spline (for path calculation), smoothdata (for smoothing path data), etc.
* The **Optimization Toolbox** is optional but if enabled with produce best functionality.
* Make sure file named as follows “plotWellTrajectory.m”. This is to prevent naming convention error

## Installation:

##### Option 1 Google Drive Link:

* Entirety of solution and other deliverables including video presentation and power point presentation will be submitted on or before 20/11/2024.

##### Option 2 Open GitHub repository:

* Clone git hub repository into desired directory
* Add path to MATLAB path or move into desired directory to use solution

# 3. Inputs for plotWellTrajectory Function

## Required Inputs

* The function plotWellTrajectory requires the following inputs to generate the trajectory and visuals:

### Surface Location:

* + **Type**: Array [Northing, Easting, True Vertical Depth].
  + **Description**: This represents the **starting point** of the wellbore, generally at the surface level.

**Example**:

surfaceLocation = [100, 200, 0]; % Northing = 100,Easting = 200,True Vertical Depth = 0 (surface level).

### Pay Zones:

* + **Type**: Cell array containing [N x 6] elements, where N represents the number of pay zones.
  + **Description**: Each pay zone is defined by a set of boundary coordinates, [Northing\_start, Northing\_end, Easting\_start, Easting\_end, TVD\_start, TVD\_end].

**Example**:

payZonePositions = { [500, 600, 400, 500, 600, 800];% Define boundaries for PayZone1

[550, 750, 600, 700, 800, 1000];% Define boundaries for PayZone2 };

### Pay Zone Shapes:

* + **Type**: Cell array of strings.
  + **Options**: 'Cylindrical', 'Cuboid', 'Spherical'.
  + **Description**: Defines the **shape of each pay zone**, which will be used for plotting and visualization.

**Example**:

payZoneShapes = {'Cylindrical', 'Cuboid'};

### Kick-Off Point Depth (KOP):

* + **Type**: Scalar (double).
  + **Description**: Defines the **depth at which the wellbore starts deviating** from a vertical path. Typically, this is set at a point where the drill needs to begin its curve towards a specific target.

**Example**:

kopDepth = 300; % Kick-off depth at 300 meters.

### Interval:

* + **Type**: Scalar (double).
  + **Description**: Defines the **step size** for calculating points along the well path. Smaller values yield more **detailed and smoother paths**, while larger intervals result in faster calculations.

**Example**:

interval = 10; % Calculate points every 10 meters.

### Optional Arguments:

* 'singleColor': Plot the well path using a **single color** rather than different colors for different segments.
* 'plot2D': Generate a **2D projection** of the well trajectory for an alternative top-down view.
* 'saveExcel': Save the computed data, including **measured depth, inclination, azimuth**, etc., to an **Excel file**.

# 4. Step-by-Step Usage

**Step 1: Defining Inputs**

* You start by defining the **surface location**, **pay zones**, **shapes**, **kick-off point**, and **interval** as mentioned above.

**Step 2: Calling the Function**

* Below is an example call to plotWellTrajectory with all required and optional parameters:

surfaceLocation = [100, 200, 0];

payZonePositions = { [500, 600, 400, 500, 600, 800];

[550, 750, 600, 700, 800, 1000]; };

payZoneShapes = {'Cylindrical', 'Cuboid'};

kopDepth = 300; interval = 10;

% Call the function to generate the well trajectory

[M, I, A, N, E, T, MIANETTable] = plotWellTrajectory(surfaceLocation, payZonePositions, payZoneShapes, kopDepth, interval, 'singleColor', 'plot2D', 'saveExcel');

# 5. Outputs

**3D Plot**

* The **3D plot** shows the well trajectory from the surface to the **pay zones**, illustrating the sequence in which the zones are reached.
* Segments of the trajectory can be colored differently or uniformly, based on user preference.

A screenshot of a graph

Description automatically generated

**2D Plot (Optional)**

* If the 'plot2D' option is used, a **top-down view** of the trajectory is generated. This 2D representation can help understand the lateral deviation of the path.

A screen shot of a graph

Description automatically generated

**Excel File (Optional)**

* If the 'saveExcel' argument is used, an Excel file (MIANETTable.xlsx) is generated. This file contains:
  + **Measured Depth**: The cumulative length drilled.
  + **Inclination** and **Azimuth**: Indicate the orientation of the well path.
  + **Northing, Easting, True Vertical Depth**: Coordinates of the path.

# 6. Explanation of Key Parts in the Code

## 1. Path Initialization

* **Function**: initializePath(surfaceLocation)
* **Description**: Sets up initial parameters like measured depth (MD), inclination (I), azimuth (A), and coordinates (N, E, T) from the surface location.

## 2. Transition to Kick-Off Point

* **Function**: moveToKOPAndCalculatePath()
* **Explanation**: Moves vertically to the **kick-off point** depth (KOP), where deviation begins.

## 3. Calculation of Path to Pay Zones

* **Function**: calculateCurvedPath(...)
* **Details**: After reaching the KOP, this function generates the **curved paths** to reach each of the pay zones.
* **Optimization Step**: If the **Optimization Toolbox** is available, an optimal pay zone sequence is calculated.

## 4. Optimization of Pay Zone Sequence

* **Function**: optimizePayZoneOrder(payZonePositions)
* **Purpose**: Determines the optimal sequence to visit pay zones, minimizing the total path length.
* **How It Works**:
  + **Distance Calculation**: Computes the distances between the surface location and each pay zone, and then between pay zones.
  + **Sequence Optimization**: Uses a nearest-neighbor heuristic to determine the closest pay zone to visit at each step.
* **Optimization Step**:
  + If the **MATLAB Optimization Toolbox** is available, the optimal sequence is calculated to minimize the total travel distance.
  + **Fallback Scenario**: If the toolbox is not available, the function proceeds in the default pay zone order and displays a warning message.

## 4. Plotting and Coating

* **Function**: plotSegmentWithColor(...)
* **Explanation**: Plots each segment in either a uniform or varied color scheme. Adds a **cylindrical coating** around the trajectory for better visualization.

## 5. Excel Data Export

* **Function**: saveMIANETTableToExcel()
* **Details**: Saves the computed values to an **Excel file** for easy reference and further analysis.

# 7. Contact for Support

* For questions or support regarding the use of this tool, please contact:
* LinkedIn Profile:

1. **Project Manager**: https://www.linkedin.com/in/israel-adebisi-adesanwo-116463217
2. **Software Developer:** <http://www.linkedin.com/in/damisola-da33355236>
3. **Quality Control Officer**: https://www.linkedin.com/in/afolasewa-ahmed-58a079255
4. **Technical Expert**:

# 8. Appendices

**Appendix A: Glossary**

* **Measured Depth (MD)**: The actual distance along the well path.
* **True Vertical Depth (TVD)**: The vertical distance from the surface.
* **Kick-Off Point (KOP)**: The point where the well deviates from the vertical.

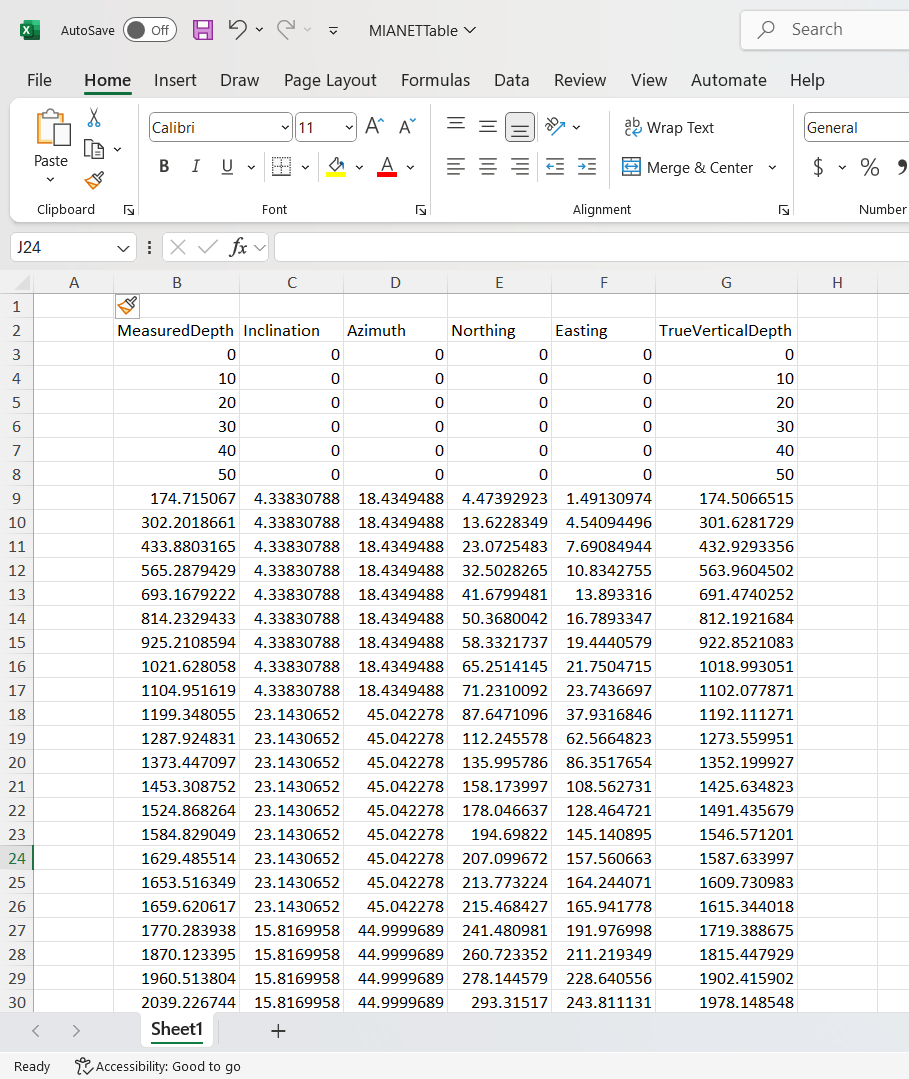
**Appendix B: Example Outputs**

* **MIANET Table**

A screenshot of a table

Description automatically generated

* Excel Spreadsheet



**Final Notes**:

* The development of this project has been a highly enriching experience for the entire team. We gained significant insights into well path optimization, data visualization, and problem-solving.
* We sincerely hope that this tool proves useful for educational purposes and helps in understanding well trajectory visualization.
* We are open to any feedback, which would help us improve the project further and refine our skills.
* **Feel free to reach out for further clarifications or improvements.** Your input is invaluable to us, and we are eager to learn from your perspective.