FACULTY OF ENGINEERING AND TECHNOLOGY

A REPORT ON DEVELOPMENT OF PROFESSIONAL GRAPHIC USER INTERFACES FOR NUMERICAL METHODS AND YIELD ESTIMATION.

BY GROUP 19

COURSE UNIT: COMPUTOR PROGRAMING

LECTURER: Mr. MASERUKA BENEDICTO

ACKNOWLEDGEMENT

First and foremost, we would like to thank the Almighty God for giving us the strength to carry on with our research in Group 19. We would love to extend our gratitude to all the persons with whose help we managed to make it this far. The willingness of each one of us to invest time and provide constructive feedback has been immensely valuable in this assignment. We wish to extend our gratitude to our lecturer for his consistent guidance and valuable insights throughout this assignment. His teaching and encouragement made it possible for us to understand and practically apply concepts of developing graphical user interfaces in MATLAB.

We also thank our group members for their cooperation and contribution. Each member actively participated in research, coding, and report writing, which ensured the success of this work. Finally, we would like to express our gratitude to all the sources and references that have been cited in this report.

ABSTRACT

We started our first meeting for research on 1st, November, 2025 in the university library. This report details the development of a professional Graphical User Interface (GUI) for the Numerical Methods and yield estimations prepared in prior assignments.

Our GUI functions create an interface designed for predicting crop yields based on several key environmental and agricultural inputs. The user can input specific numerical values for Rainfall (in mm), Fertilizer application rate (in kg/ha), a Soil Quality Index (on a 0-10 scale), average daily Sunlight (in hours/day), and the total Farm Size (in hectares). Our other GUI function also provides a platform for solving mathematical problems using four common numerical techniques: Euler's Method, Runge-Kutta (RK), Newton-Raphson, and the Secant Method. Each method's execution relies on separate classes to compute and display the numerical solutions in an output text area.

DEDICATION

We dedicate this report to all the individuals especially Group 19 members, who have been there with us in the process of formulating and compiling this report. To our lecturer Mr. Maseruka Benedicto whose guidance and expertise have been invaluable, your mentorship and insightful feedback have shaped our understanding.

DECLARATION

We hereby certify and confirm that the information in this report is out of our own efforts, research and it has never been submitted in any institution for any academic award.

STUDENT NAME	**REG.NO**	**COURSE**	**SIGNATURE**
NTALE JOASH KATEREGA	BU/UG/2024/2595	WAR	
WANYAMA JOSEPH EROGO	BU/UP/2024/1077	WAR	
KAKULU ALFRED	BU/UP/2024/0981	MEB	
OKORI DARIOUS	BU/UP/2024/1061	WAR	
ALITEMA VICTOR	BU/UP/2024/5098	WAR	
AMASO SUSAN	BU/UP/2024/5435	PTI	
MAATE PHILEMON	BU/UP/2024/0830	AMI	
SIKUKU BELIZER RUTH	BU/UP/2024/0846	AMI	
TWICHIRIZE FLORENCE	BU/UP/2022/1836	WAR	
OKWI NICHOLAS	BU/UP/2024/4457	WAR	

APPROVAL

We are presenting this report which has been written and produced under our efforts. We carried out research on visualizing our data interfaces that are well labeled ready for easy interpretation by the final user.

DATE OF SUBMISSION: …….

SIGNATURE: …….

Table Of Contents

List of Acronyms/Abbreviations.

MATLAB –Matrix Laboratory.

GUI – Graphics user interfaces

1 CHAPTER 1: INTRODUCTION

1.1 Introduction

It is one of the core competencies of engineering and data science to create and manipulate algorithms to solve functions and equations. MATLAB provides a suite of functions and recursions through which equations can be manipulated to find solutions using numerical method.

For the first part of this assignment, the application enables us solve problems using several fundamental numerical techniques drawn from Assignment 5.1. The GUI features distinct buttons and input fields to execute four methods Euler, Runge-Kutta, Newton-Raphson, and Secant allowing for the efficient calculation of Ordinary Differential Equations (ODEs) and finding roots of equations.

The main objectives of the numerical method were;

- To abstract the complexity of coding in mathematical methods
- To facilitate the comparison and validation of different techniques for a given problem.
- To maximize efficiency in the problem-solving process e.g., quick execution

In part two, the GUI is designed to showcase an application of Machine Learning in agriculture, specifically for Crop Yield Estimation. It provides a user-friendly interface to input five key environmental and farm metrics (Rainfall, Fertilizer, Soil Quality, Sunlight, and Farm Size) and instantly receive a predicted crop yield output from an integrated YieldModel object.

The main objectives of yield estimation GUI were;

- To model and predict potential crop productivity.
- To serve as a practical decision-making tool for agricultural stake holders,
- To make complex agricultural modeling accessible to a wide audience.

1.2 Design Process

1. We organized meetings during our available time where we went through lecture notes and modules to come up with possible lines of code to put in our script.
2. We went ahead to establish the necessary inputs (like x0, h, rainfall) and planned the output structure for each GUI.
3. We went through different ideologies, flow charts and pseudocodes. That we would then apply into our final scripts.
4. We inquired from other groups about their progress and refined some of ideas from them.
5. The code for both numbers was written down.
6. Debugging was done in the presence of all members that were available to get a better understanding of how it worked.
7. Documentations was carried out in report making and presentation drafting.

2 CHAPTER 2: METHODOLOGY

2.1 Pseudocode

1. We created a main window with dimensions 840x420
2. We created input fields such as
   - text field "Function f (x, y) or f(x)" at position [40,370]
   - text field "Derivative f'(x) or x_1" at position [40,340]
   - numeric field “Rainfall”
   - numeric field "x_0" = 0 at [40,300]
   - numeric field "y_0" = 0 at [160,300] etc.

3. We dedicated an output display using text area.
4. We then created buttons tied to specific function triggers e.g.
   - Button "Euler Method" with a trigger function runEuler()
   - Button "Newton-Raphson" with a trigger function runNR()

2.2 The MATLAB Code

2.2.1 PART A

```
function NumericalMethodsGUI
    % Making Figure window
    fig = uifigure('Name','Numerical Methods GUI','Position',[100 100 840
420]);

    % INPUTS
    uilabel(fig,'Text','Function f(x,y) or f(x):','Position',[40 370 200 22]);
    funcs = uieditfield(fig,'text','Position',[240 370 300 22], ...
        'Value','Enter a Function of x and/or y');

    uilabel(fig,'Text','Derivative f''(x): (Field may be used for x_1 in Secant
method)', ...
        'Position',[40 340 400 22]);
    dFuncs = uieditfield(fig,'text','Position',[440 340 300 22], ...
        'Value','Enter Derivative or x_1 (if required)');

    uilabel(fig,'Text','x_0:','Position',[40 300 30 22]);
    x0Field = uieditfield(fig,'numeric','Position',[70 300 70 22],'Value',0);

    uilabel(fig,'Text','y_0:','Position',[160 300 30 22]);
    y0Field = uieditfield(fig,'numeric','Position',[190 300 70 22],'Value',0);

    uilabel(fig,'Text','h:','Position',[280 300 20 22]);
    hField = uieditfield(fig,'numeric','Position',[300 300 70 22],'Value',0.1);
```

```
    uilabel(fig,'Text','x_end:','Position',[390 300 50 22]);
    x_endField = uieditfield(fig,'numeric','Position',[440 300 70
22],'Value',1);

    uilabel(fig,'Text','Tolerance:','Position',[530 300 70 22]);
    tolField = uieditfield(fig,'numeric','Position',[610 300 100
22],'Value',5e-6);

    % Output Area
    uilabel(fig,'Text','Results:','Position',[40 260 80 22]);
    outputArea = uitextarea(fig,'Position',[40 50 760 200],'Editable','on');

    % Buttons
    uibutton(fig,'Text','Euler Method','Position',[60 10 120 30], ...
        'ButtonPushedFcn',@(btn,event) runEuler());
    uibutton(fig,'Text','Runge-Kutta','Position',[200 10 120 30], ...
        'ButtonPushedFcn',@(btn,event) runRK());
    uibutton(fig,'Text','Newton-Raphson','Position',[340 10 120 30], ...
        'ButtonPushedFcn',@(btn,event) runNR());
    uibutton(fig,'Text','Secant','Position',[480 10 120 30], ...
        'ButtonPushedFcn',@(btn,event) runSecant());

    % Button Functions
    function runEuler()
        f = str2func(funcs.Value);
        x0 = x0Field.Value; y0 = y0Field.Value;
        h = hField.Value; xend = x_endField.Value;
        obj = Integral_Methods(f,x0,y0,h,xend);
        output = evalc('obj.SolutionEulers()');
        outputArea.Value = strsplit(output,newline);
    end

    function runRK()
        f = str2func(funcs.Value);
        x0 = x0Field.Value; y0 = y0Field.Value;
        h = hField.Value; xend = x_endField.Value;
        obj = Integral_Methods(f,x0,y0,h,xend);
        output = evalc('obj.SolutionRungeKutta()');
        outputArea.Value = strsplit(output,newline);
    end

    function runNR()
        f = str2func(funcs.Value);
        fd = str2func(dFuncs.Value);
        x0 = x0Field.Value; tol = tolField.Value;
```

```
        obj = Differential_Methods(f,fd,x0,tol);
        output = evalc('obj.SolutionNewtonRaphson()');
        outputArea.Value = strsplit(output,newline);
    end

    function runSecant()
        f = str2func(funcs.Value);
        x0 = x0Field.Value;
        x1 = str2double(dFuncs.Value);
        tol = tolField.Value;
        obj = Differential_Methods(f,x1,x0,tol);
        output = evalc('obj.SolutionSecant()');
        outputArea.Value = strsplit(output,newline);
    end
end
```

2.2.2 PART B

```
function YieldEstimationGUI
    % Making our Figure Window
    fig = uifigure('Name','Yield Estimation GUI','Position',[100 100 700 400]);

    % Inputs
    uilabel(fig,'Text','Rainfall (mm):','Position',[40 340 150 22]);
    rainField = uieditfield(fig,'numeric','Position',[180 340 120
22],'Value',1000);

    uilabel(fig,'Text','Fertilizer (kg/ha):','Position',[40 300 150 22]);
    fertField = uieditfield(fig,'numeric','Position',[180 300 120
22],'Value',500);

    uilabel(fig,'Text','Soil Quality Index (0–10):','Position',[40 260 150
22]);
    soilField = uieditfield(fig,'numeric','Position',[180 260 120
22],'Value',3);

    uilabel(fig,'Text','Sunlight (hours/day):','Position',[40 220 150 22]);
    sunField = uieditfield(fig,'numeric','Position',[180 220 120
22],'Value',12);

    uilabel(fig,'Text','Farm Size (hectares):','Position',[40 180 150 22]);
    sizeField = uieditfield(fig,'numeric','Position',[180 180 120
22],'Value',100);
```

```
    % Buttons
    uibutton(fig,'Text','Estimate Yield','Position',[320 180 150 30], ...
        'ButtonPushedFcn',@(btn,event) estimateYieldButton());

    % Output Area
    uilabel(fig,'Text','Results:','Position',[40 140 80 22]);
    outputArea = uitextarea(fig,'Position',[40 30 620 110],'Editable','off');

    function estimateYieldButton()

        rain = rainField.Value;
        fert = fertField.Value;
        soil = soilField.Value;
        sunlight = sunField.Value;
        farmSize = sizeField.Value;

        obj = YieldModel(rain, fert, soil, sunlight, farmSize);
        outputText = evalc('obj.estimateYield()');
        outputArea.Value = strsplit(outputText, newline);
    end
end
```

2.3 Graphics User Interfaces

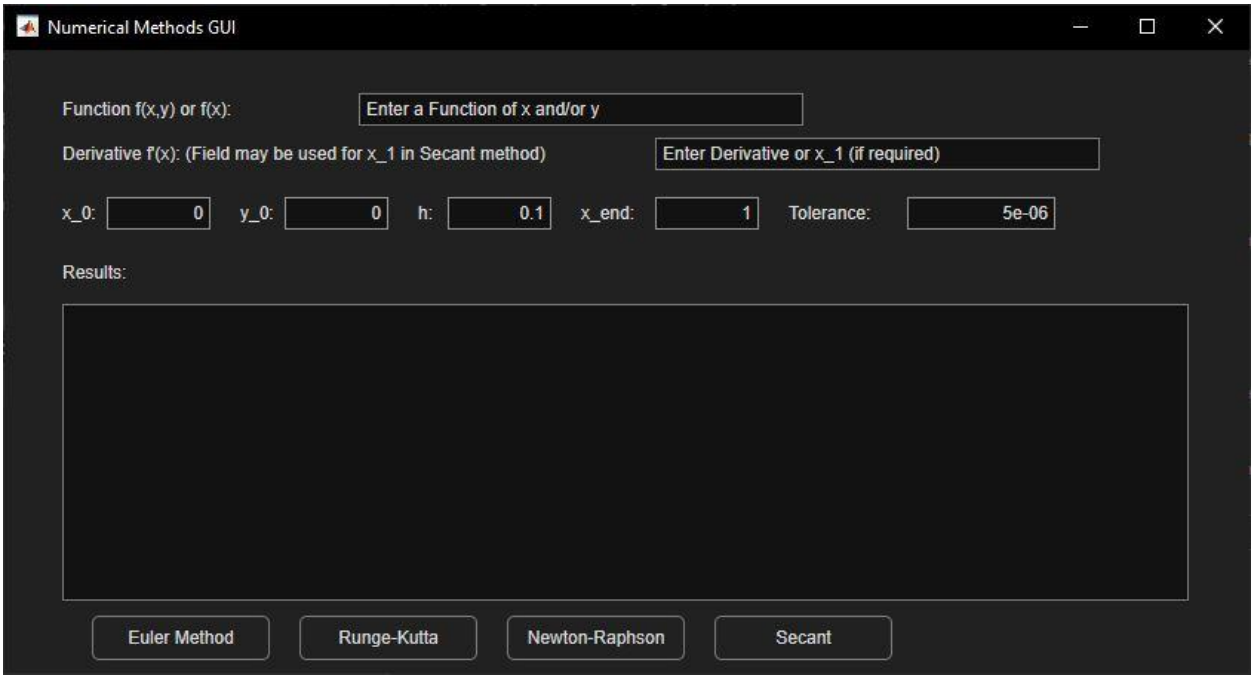

Figure 1 Numerical Methods GUI Window on default settings

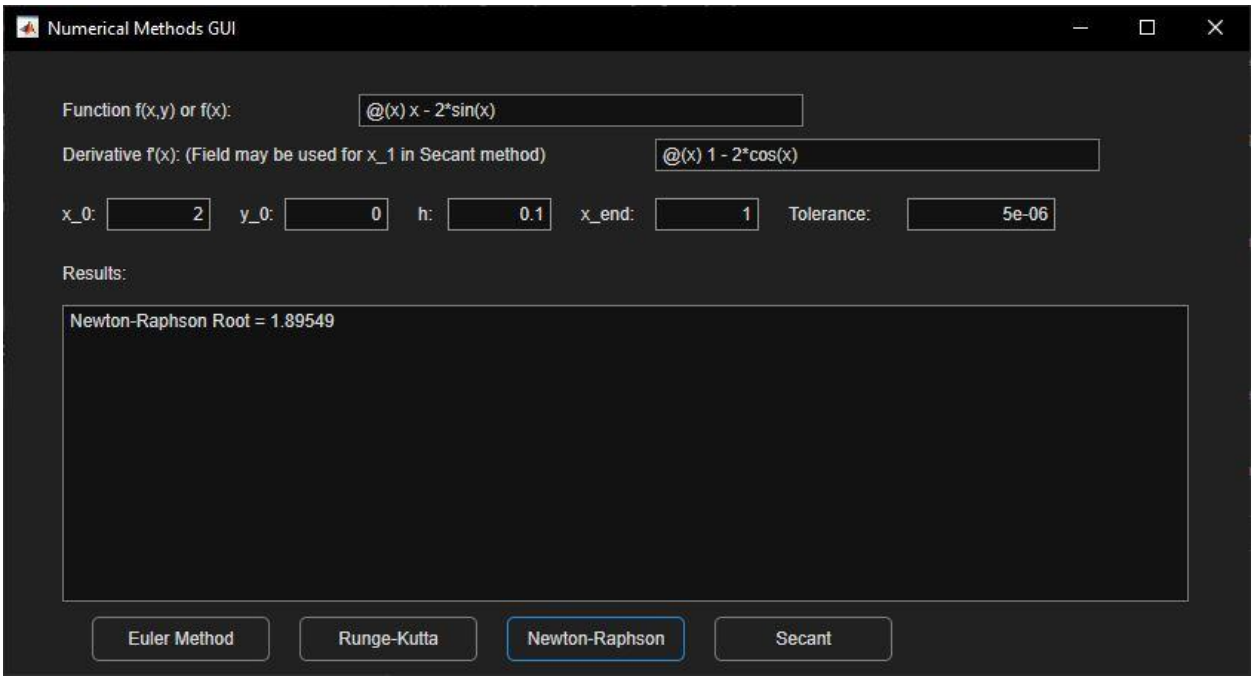

Figure 2 Numerical Methods GUI after solving a Newton Raphson Methods

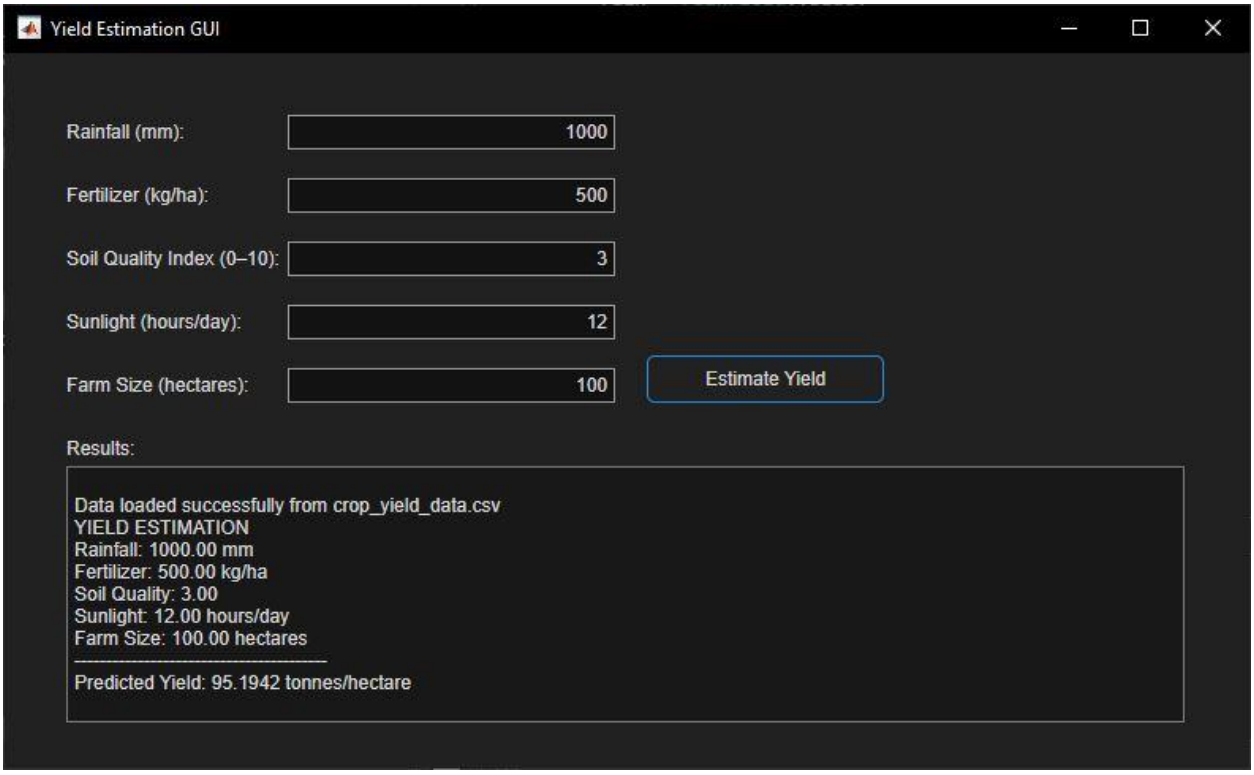

Figure 3 GUI for Yield Estimation showing a solved example

3 CONCLUSION

The development of these two MATLAB (GUIs) successfully demonstrates the practical application of numerical methods and predictive modeling to solve real-world engineering problems. The Numerical Methods GUI provides a relatively user-friendly platform for implementing core computational algorithms, including the Euler and Runge-Kutta methods for solving ordinary differential equations, and the Newton-Raphson and Secant methods for finding roots of equations.

Simultaneously, the Yield Estimation GUI showcases the specialized application of Machine Learning Toolbox in agricultural engineering. By integrating key factors rainfall, fertilizer, soil quality, sunlight, and farm size into a predictive model, the GUI offers a practical support system for estimating crop yield.

We faced a challenge on trying to find specific ways to loop back the functions to buttons especially for each Numerical methods operation. Luckily this is where we applied knowledge from documentation function from MATLAB better understand how each is done. We also faced a problem of unstable power supply reducing our effective coding time.

4 REFERENCES

• MATLAB Documentation Retrieved from: https://www.mathworks.com/help/matlab/

• Kirani Singh, T., & Chanduri, B. B. MATLAB Programming. PHI Learning Pvt. Ltd.

• Course Notes of Modules 5: Computer Programming with MATLAB, Mr. Maseruka Benedicto