Matholic — A Math Companion

# Table of Contents

* [Authors](#_Authors)
* [Built With](#_Built_With)
  + [Tech Stack](#_Tech_Stack)
  + [Key Features](#_Key_Features)
* [Problem definition and formulation](#getting-started)
  + [Explanation of the Problem](#_Explanation_of_the)
  + [Descriptions of the tasks](#_Descriptions_of_the)
* [Solution Method](#getting-started)
  + [The calculations required to solve each task of the problem](#_The_calculations_required)
* [Programming in MATLAB](#_Programming_in_MATLAB)
  + [Structure Chart](#_Structure_chart)
  + [Pseudocode](#_Pseudocode)
  + [Flowchart](#_Flow_chart)
  + [Table: list of the matlab codes, figures and I/O variables used](#_Table:_list_of)
  + [Explanation on how the code works](#_Explanation_on_how)
* [Future Features](#_Future_Features)
* [Acknowledgments](#_Acknowledgments)

# Matholic

This MATLAB program features a simple GUI that assists users with a wide range of mathematical calculations, from basic operations to more advanced tasks, based on their selected operations and inputs.

## Authors

**Amanuel Galema**

* Student number: 22316708
* Department: Computer Engineering
* GitHub: [@amexabee](https://github.com/amexabee)

**Abdulahi Ogunlesi**

* Student number: 22114031
* Department: Computer Engineering
* GitHub: [@Niffy024](https://github.com/Niffy024)

## Built With

### Tech Stack

* MATLAB R2017a

### Key Features

* **Easier user interface**
* **Wide range of calculation options**

## Problem Definition and Formulation

### Explanation of the Problem

Creating a single, user-friendly interface that allows users to choose a specific type of mathematical operation, perform the necessary calculations efficiently, and display results. This MATLAB program addresses that challenge by providing a GUI-based solution where users can interactively select operations, input data, and obtain results directly in the MATLAB environment.

### Descriptions of the tasks that need to be performed to solve the problem

The user has to click on one of the six button representing operations. By clicking, the user will be taken to a MATLAB command window where they have to follow the options and the instruction to calculate what they want.

## Solution Method

### The calculations required to solve each task of the problem

* **basic operations**: Find max, min, sum; Calculate length, size and Fibonacci.
* **file operations**: Write/Read to Excel and Publish as pdf.
* **plotting operations**: 2D and 3D plot.
* **polynomial operations**: Find roots, Convolve and Deconvolve polynomials.
* **symbolic computation**: Solve equations, Differentiate, and Integrate.
* **terminate**: Quit program.

## Programming in MATLAB

### Structure chart

We used git (version control system) and github to collaborate and work on the project together [[link](https://github.com/amexabee/matlab-project)]. Both students worked on all of the requirements; the structure chart is as follows:



### Pseudocode

Main Program:

Display GUI with various options:

1. Basic Operations

2. File Operations(excel)

3. Polynomial Operations

4. Plotting Operations

5. Symbolic Computations(calculus)

6. Quit Program

Wait for user to select an option.

Based on the user selection, execute the corresponding module:

- Call basicOperations() for Basic Operations.

- Call fileOperations() for File/excel Operations.

- Call polyOperations() for Polynomial Operations.

- Call plottingOperations() for Plotting Operations.

- Call symbolicComputations() for Symbolic/calculusComputations.

- Close program for Quit.

basicOperations():

Display menu for Basic Operations:

1. Find max, min, and sum of a vector.

2. Calculate length and size of a matrix.

3. Use find to locate elements in a vector.

4. Find Fibonacci sequence up to n.

Prompt user to choose an operation.

Perform the operation based on the user’s choice:

Case 1:

Accept a vector input.

Calculate and display max, min, and sum.

Case 2:

Accept a matrix input.

Calculate and display length and size.

Case 3:

Accept a vector and a value.

Use find to locate indices of the value in the vector.

Display indices.

Case 4:

Accept an integer n.

Compute Fibonacci sequence iteratively up to n.

Display the result.

Otherwise:

Display invalid choice message.

Return to main menu.

fileOperations():

Display menu for File Operations:

1. Write data to an Excel file.

2. Read data from an Excel file.

3. Publish MATLAB script as PDF.

Prompt user to choose an operation.

Perform the operation based on the user’s choice:

Case 1:

Accept data and a filename.

Write data to the specified Excel file.

Display success message.

Case 2:

Accept filename.

Read data from the specified Excel file.

Display the data.

Case 3:

Accept MATLAB script filename.

Publish the script as a PDF.

Display success message.

Otherwise:

Display invalid choice message.

Return to main menu.

plottingOperations():

Display menu for Plotting Operations:

1. 2D Plot

2. 3D Mesh Plot

Prompt user to choose an operation.

Perform the operation based on the user’s choice:

Case 1:

Accept vectors for x and y.

Validate their lengths.

Plot the data using plot() and add labels.

Case 2:

Accept ranges for x and y.

Generate meshgrid for x and y.

Compute Z values for the mesh.

Plot the data using mesh() and add labels.

Otherwise:

Display invalid choice message.

Return to main menu.

polyOperations():

Display menu for Polynomial Operations:

1. Find roots of a polynomial.

2. Convolve two polynomials.

3. Deconvolve two polynomials.

Prompt user to choose an operation.

Perform the operation based on the user’s choice:

Case 1:

Accept polynomial coefficients.

Compute and display roots using roots().

Case 2:

Accept two polynomials.

Perform convolution using conv().

Display the result.

Case 3:

Accept two polynomials.

Perform deconvolution using deconv().

Display quotient and remainder.

Otherwise:

Display invalid choice message.

Return to main menu.

symbolicComputations():

Display menu for Symbolic Computations:

1. Differentiate a function.

2. Integrate a function.

3. Solve an equation.

Prompt user to choose an operation.

Perform the operation based on the user’s choice:

Case 1:

Accept symbolic function.

Compute and display derivative using diff().

Case 2:

Accept symbolic function.

Compute and display integral using int().

Case 3:

Accept equation and variable.

Solve the equation using solve().

Display the solution.

Otherwise:

Display invalid choice message.

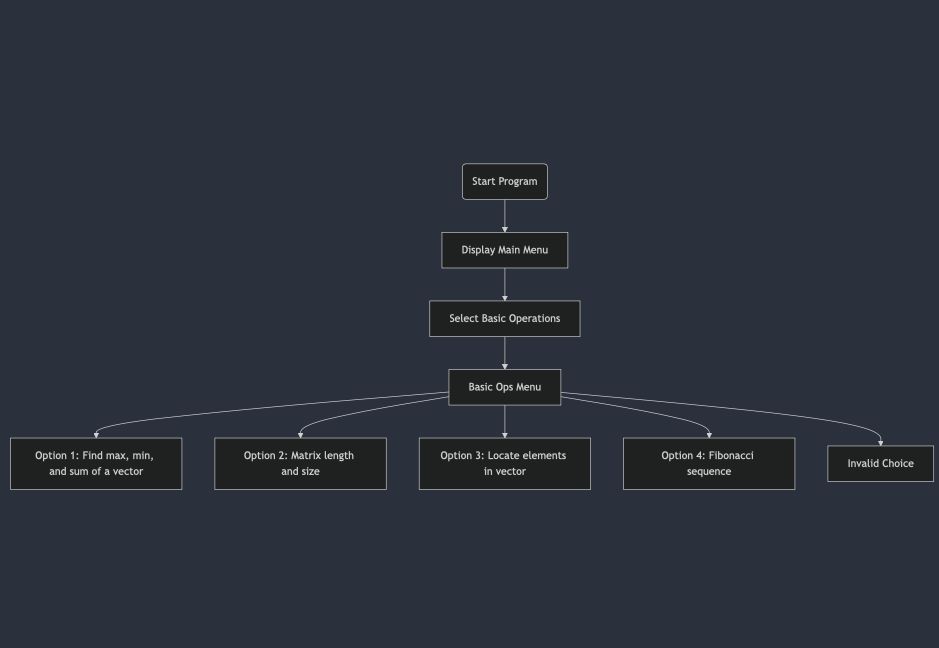
Return to main menu.

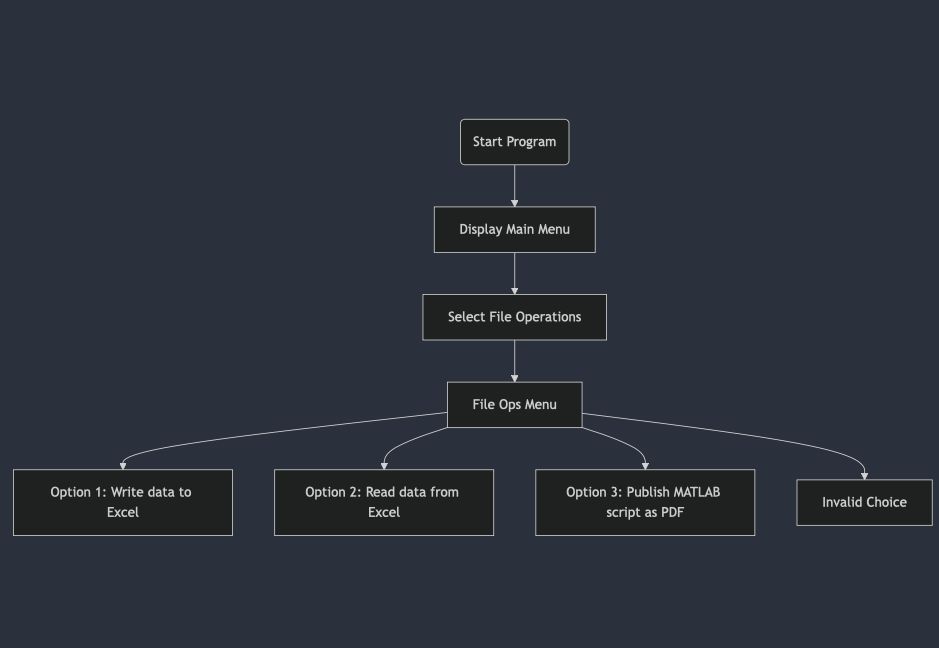
Terminate():

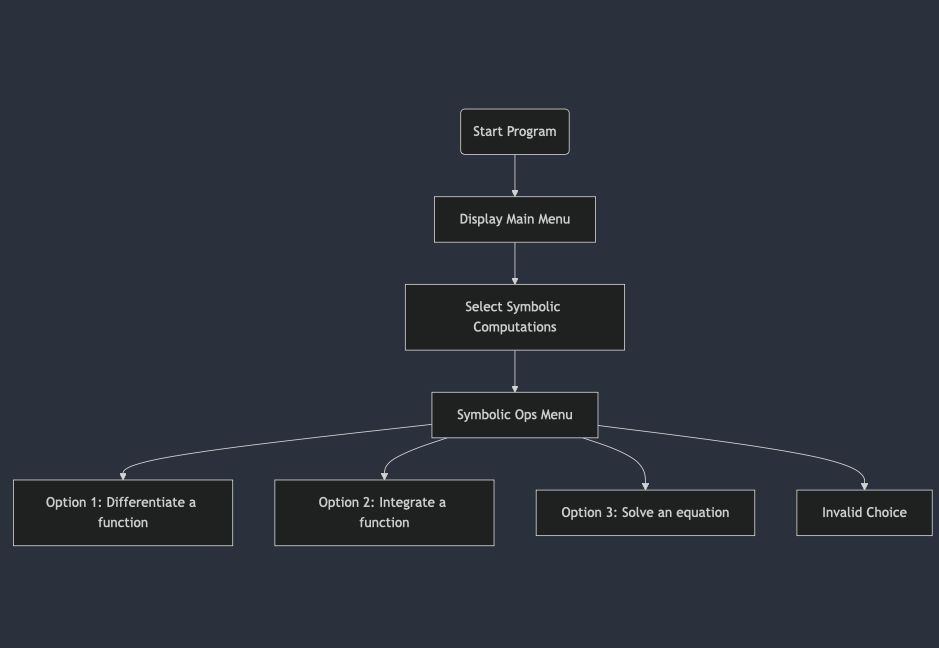
Display exit message.

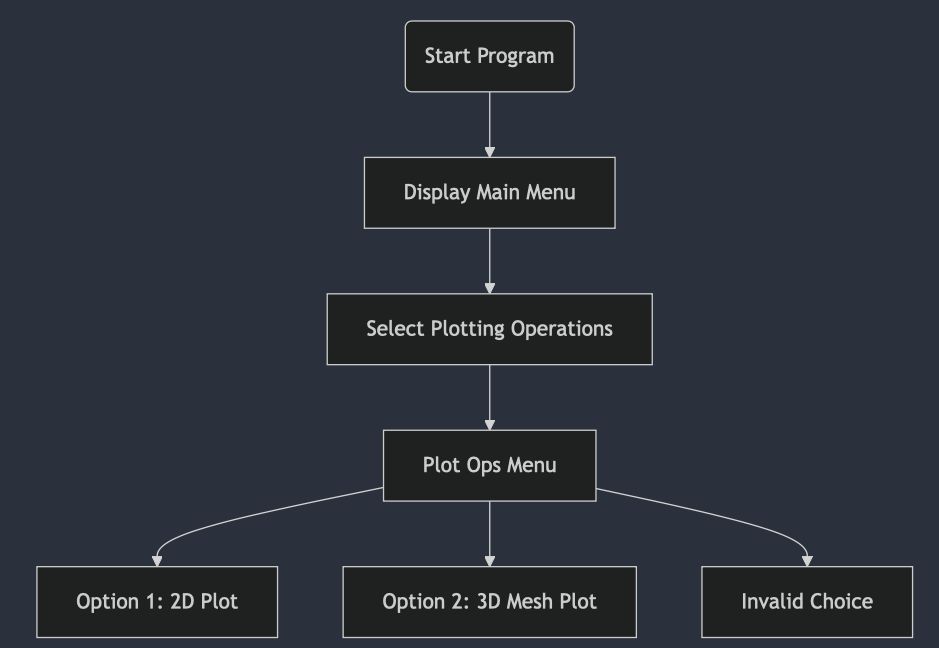
Close the GUI.

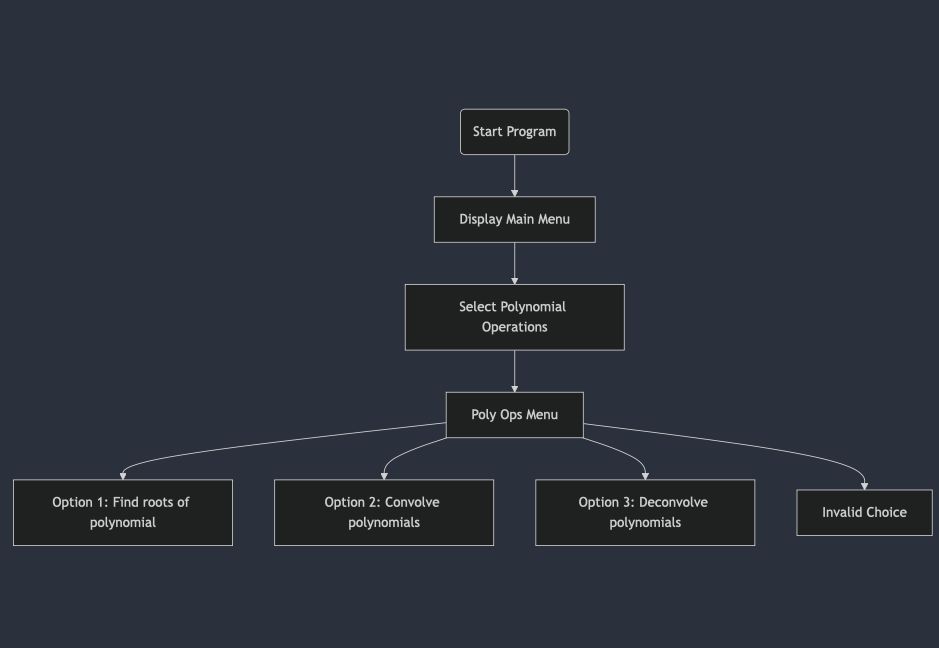
### Flow chart













### Table: list of the matlab codes, figures and I/O variables used

|  |  |  |  |
| --- | --- | --- | --- |
| File Type | File Name | Input variables | Output variables |
| MATLAB code | main.m | - | - |
| MATLAB figure | main.fig | - | - |
| MATLAB code | basicOperations.m | **choice** (user’s choice of operation)  **vec** (vector input from user)  **mat** (matrix input from user)  **value** (value input from user)  **n** (integer input from user) | **fib** (result of nth Fibonnaci sequence) |
| MATLAB code | fileOperations.m | **choice** (user’s choice of operation)  **data** (vector input from user)  **filename** (to write/read/publish the data) | pdf file |
| MATLAB code | plottingOperations.m | **choice** (user’s choice of operation)  **x,y** (vector inputs from user) | - |
| MATLAB code | polyOperations.m | **choice** (user’s choice of operation)  **coeffs** (vector input from user) | **r** (roots of a polynomial)  **result** (multiplication of polynomials)  **quotient** and **reminder** of division of polynomials |
| MATLAB code | symbolicComputations.m | **choice** (user’s choice of operation) | **sol** (solution to an equation), **diff\_expr** (differential expression), **integral** |
| MATLAB code | terminate.m | **Y** (yes) / **N** (no) | - |

### Explanation on how the code works

First open the main.fig file then it opens both the fig file and the MATLAB command window with it. Then from the GUI select an operation you want to perform and click its button. You’ll then be directed to the command window and there will be instructions to follow which are related to your choice. Then follow the instructions and solve your problem. Upon completing you’ll automatically be readirected to the GUI.

## Future Features

* **Integration of other language/framework for a better GUI**
* **Deployment**
* **API Documentation**

## Acknowledgments

* **Dr. Neyre Ersoy**