MATLAB-Based Graphing Calculator or MATLAB Graphing Calculator: A Comprehensive Visualization Tool

Richmonde Gatchalian[1], Althea Nunag[2], Vian Duran[3]

[1] Computer Engineering
[2] Electronics Engineering
[3] Computer Engineering

richmonde_gatchalian@dlsu.edu.ph, lara_althea_nunag@dlsu.edu.ph, vian_duran@dlsu.edu.ph

Abstract

The Graphing Calculator, developed using MATLAB, is a comprehensive tool designed for students, educators, and professionals across various disciplines. This application provides an interactive platform for dynamically creating and visualizing mathematical functions, going beyond the capabilities of traditional graphing calculators. It serves as a bridge between basic calculators and advanced charting software, thereby enhancing users' productivity and understanding in fields such as mathematics and engineering. By addressing contemporary challenges in data representation and modeling, the Graphing Calculator proves essential for teaching, studying, and professional activities.

This project aims to design a user-friendly interface coupled with high functionality graphing tools that facilitate educational, research, and professional decision-making processes. Initially, we will assess project requirements, design the interface, and outline the functionalities. Implementation involves developing the key features using MATLAB's graphical capabilities. The testing phase ensures all functionalities work as expected, while the documentation phase provides clear instructions for users.

Roles within the project are clearly defined to ensure efficient progress. Tasks are distributed for independent work to minimize dependencies, and progress is tracked through regular updates via Messenger and an action plan with deadlines.

Introduction

The Graphing Calculator, developed using MATLAB, is a tool for students, educators, and professionals in all kinds of fields. It offers an interactive platform to create and visualize various mathematical functions dynamically. Unlike traditional graphing calculators, this application has numerous added functions. It bridges the distance between simple calculators and more advanced charts software, thus improving users' productivity and comprehension in mathematics as well as engineering fields. The latter resolves up-to-date problems of data representation and modeling, which is needed for teaching, studying and engineering activities. Its importance is connected with a user-friendly interface offering high functionality of graphing tools that can facilitate educational, research and professional decision-making processes.

Functionalities

This section describes the list of functionalities that you want to build as part of the application. Preferably, this list should be as granular as possible.

You can create these sets of functionality by formulating a table as shown below.

TABLE I. List of Functionalities

Persona	Description	Benefit
Students	The graphing calculator includes different features such as plotting a function, circle, line, and a dot.	Students can use these features for educational purposes especially in their mathematical subjects by applying what they learn. Students can use these features to visually solve equations.
Students	The graphing calculator can plot numerical data provided by the students. It also has a feature wherein users can pan the graph and to zoom in or out.	This can be helpful to students to visualize their data more precisely by using the panning and zooming feature. It allows them to closely examine specific parts of a graph that may not be apparent at standard view.
Teachers	The graphing calculator includes a save feature wherein it makes it easier to save the graph together with the data provided.	This feature can be helpful for educators as they can quickly access saved graphs to illustrate complex concepts, saving time and ensuring smooth transitions between topics.

System Development

Pseudo-code

This section presents the system code or pseudo-code highlighting the modules covered or used. Include discussion and justification of the use of module/s.

- selectGraph

    If a graph is selected, it restores its original properties
    Saves the properties of the newly selected graph
    Highlights the selected graph

- startupFcn(app)

    Initializes the UI axes with labels, title, and grid
    Sets UserData for UIAxes to store different types of plots

- LineButton_2Pushed(app, event)

    Prompt users for line coordinates
    Plots the line on the graph
    Updates the UserData with the newly plotted line

- CircleButton_2Pushed(app, event)

    Prompts user for the circle’s center and radius
    Plots the circle on the graph
    Updates the UserData with the newly plotted circle

- DotButton_2Pushed(app, event)

    Prompts user for the circle’s center and radius
    Plots the circle on the graph
    Updates the UserData with the newly plotted circle

- GraphFunctionButton_2Pushed(app, event)

Prompts user for a mathematical function and X-range
Plots the mathematical function on the graph
Updates the UserData with the newly plotted mathematical function

- PlotDataButton_2Pushed(app, event)

  Opens a dialog to input data points
  Plots the data points on the graph
  Updates the UserData with the newly plotted data points

- DeleteGraphButton_2Pushed(app, event)

  Checks if the graph is selected
  If selected, deletes the graph

- saveGraphButton_2Pushed(app, event)

  Opens a dialogue to save the graph as an image file

- applyLineProperties(app, ColorDropdown, colorCodes, lineWidthField, dialogHandle)

  Applies the selected color and line settings to the selected graph

- applyDatPlot(app, dataField, dialogHandle)

  Parse and plot the entered data points

- createComponents(app)

  Creates UI components and sets their properties and positions
  Assigns callback functions to buttons
  Makes the UI figure visible

- delete(app)

  Initializes the UI components and registers the app
  Runs the startup function

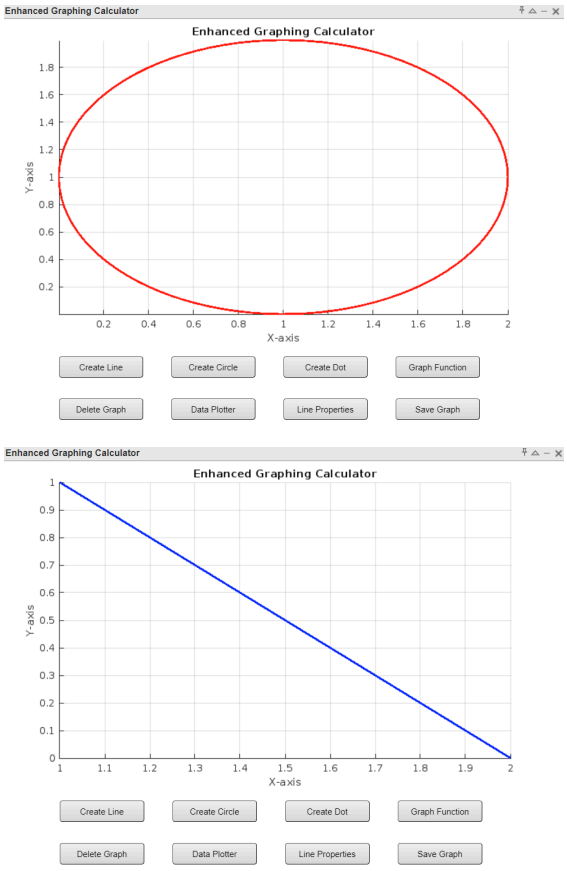
- GraphingCalculator Constructor

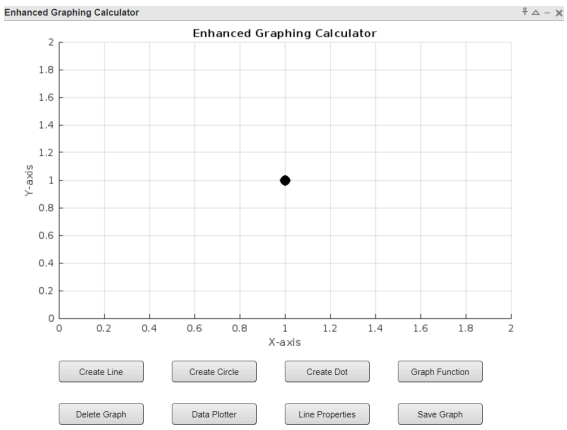
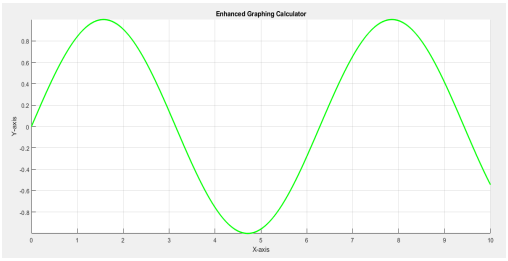
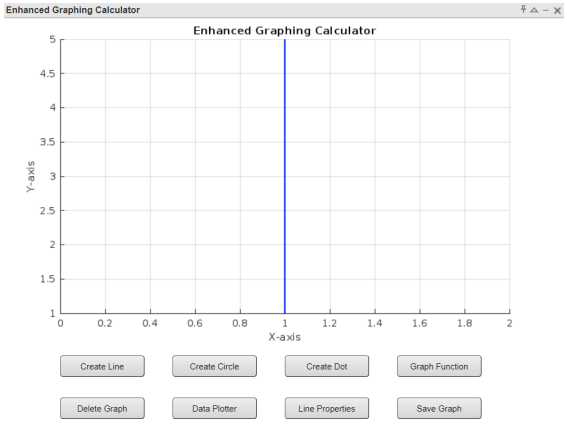
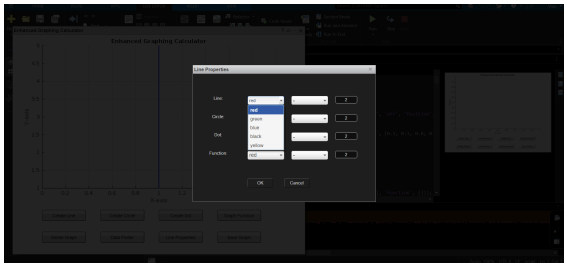
  Deletes the UI figure when the app is deleted

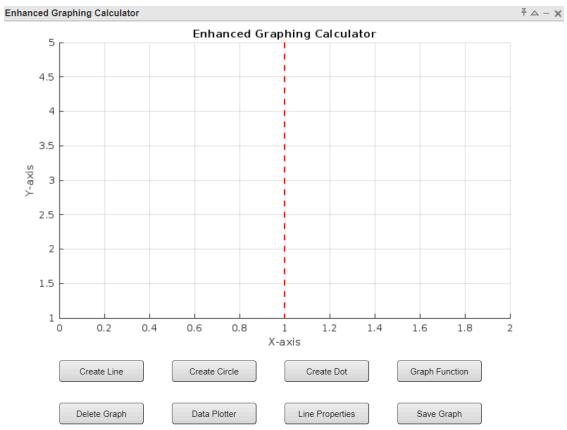
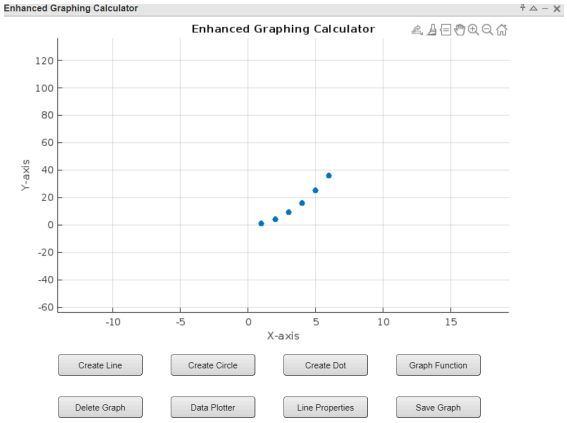
System Output and Results

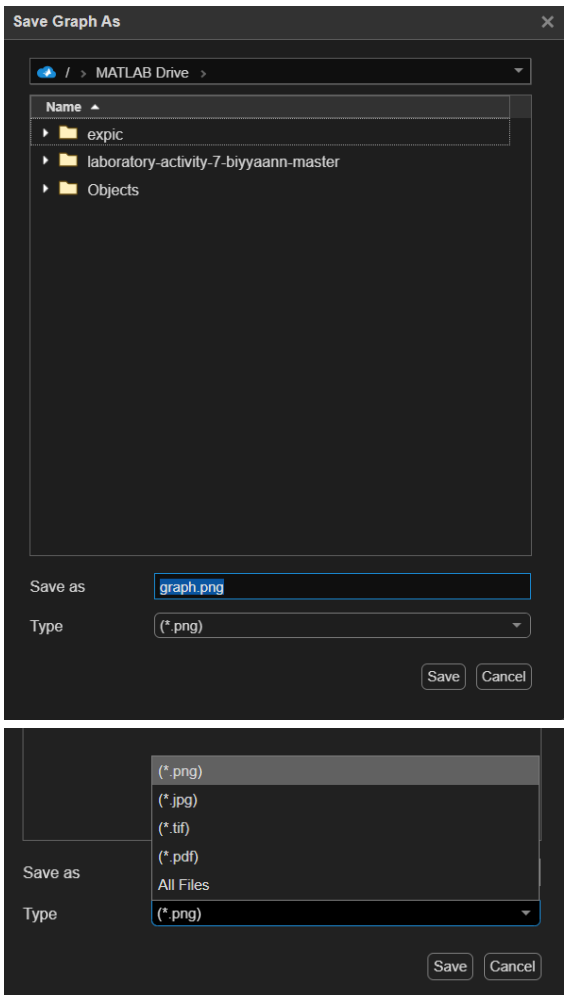
Functionality Displays

TABLE I. Functionality Output Display

Persona	Description	Output Display
Benefactor/user	Describe the expected functionality	(sample screenshot images)
Benefactor/user	The user can create and display various types of graphs such as a circle, line, circle, and dot. It also allows customization of graph elements, including the type of graph and adjusting its properties (line format).	

		
Benefactor/user	The application also supports plotting mathematical functions. The user can input equations and visualize their graphs on the coordinate plane, along with having the ability to pan and zoom in and out for a more detailed analysis.	
Benefactor/user	The application includes features for editing graph properties such as changing the color, line style, and thickness. Users can customize their graph for better visualization and distinction between elements.	Before:  After: 

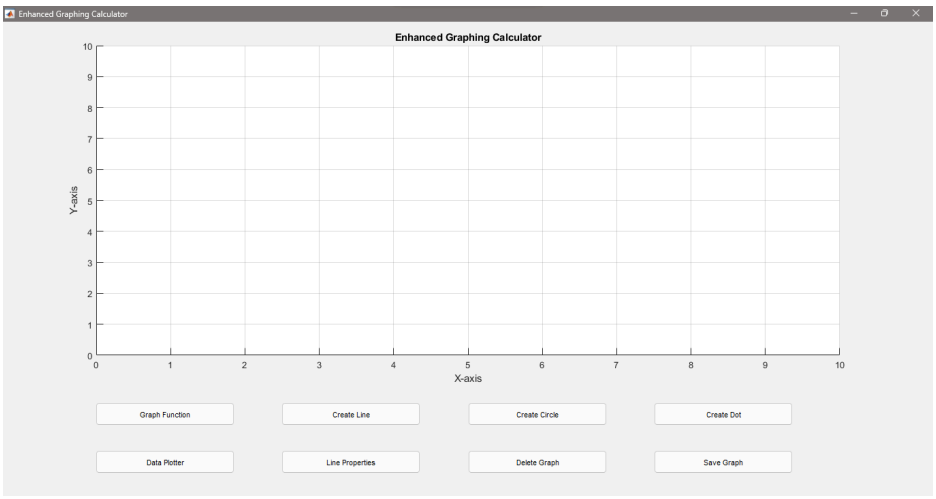
		
Benefactor/user	Users can input datasets to create scatter plots. This enables users to analyze the distribution and correlation between variables.	

Benefactor/user	The application has the ability to save the graphs. Users can export their graph in various formats such as PNG, JPEG, and etc., for further use or presentation purposes. It allows the user to preserve their work and share to others if needed.	

Walkthrough User Interface (Technical User Guide)

Getting Started

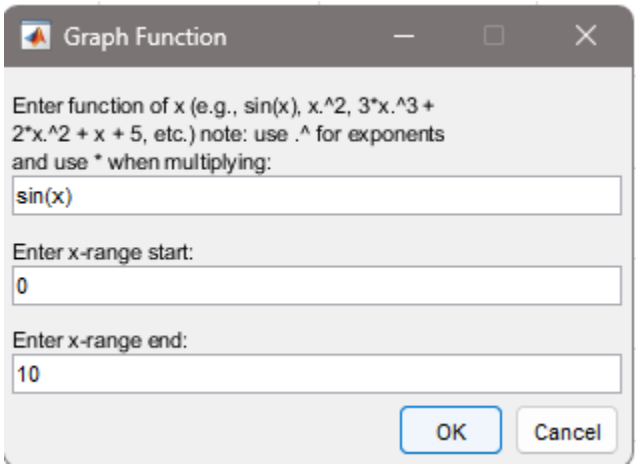
- **Description**: Upon running the application, a dialog will open up and will display the main program which is the Graphing Calculator. This is a tool to help visualize data and graph different functions.
- **Screenshots with Labels**:
    - Opening the application.



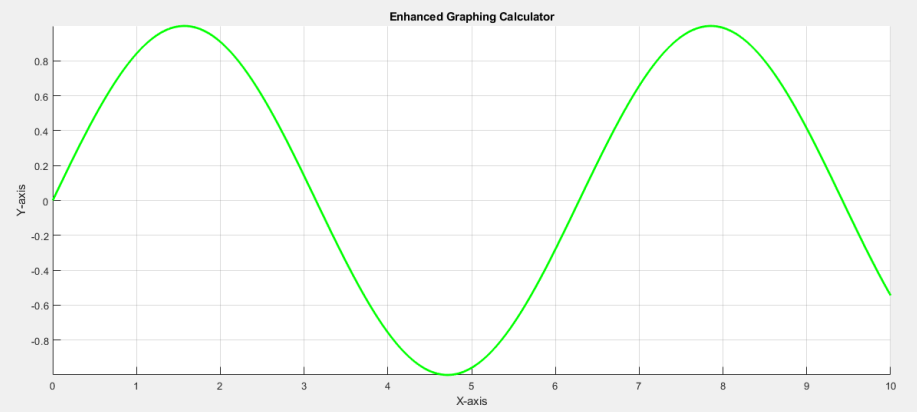
- **Guided Tour**: In the main interface, you will be introduced to different features of the Graphing calculator which includes graphing a function, line, circle, dot, and plotting data.

Main Features and Usage

- **Feature 1**: Graphing a function
  - **Screenshot**:

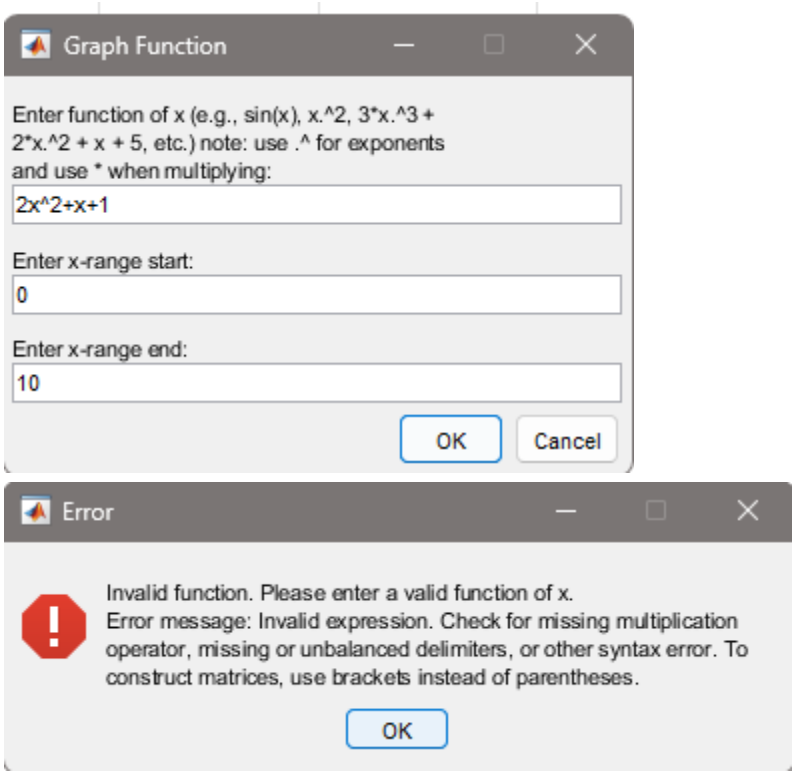


  - **Details**: This feature allows you to input different functions such as trigonometric, polynomials, binomials, and more.
  - **Example Use Case**: You may input any function you want, or you may enter the default value given which is sin(x). The result will then be graphed and displayed in the main page.

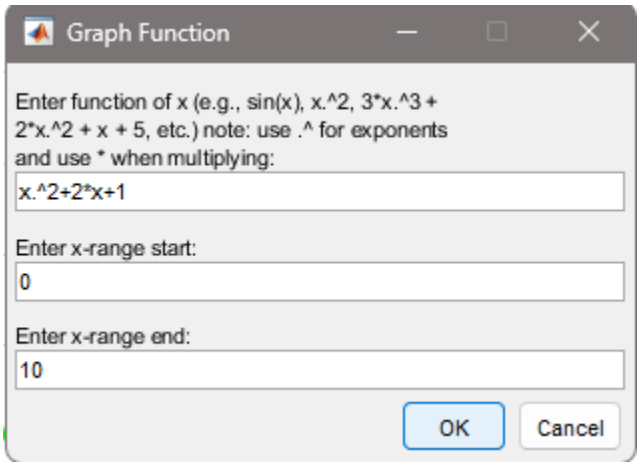


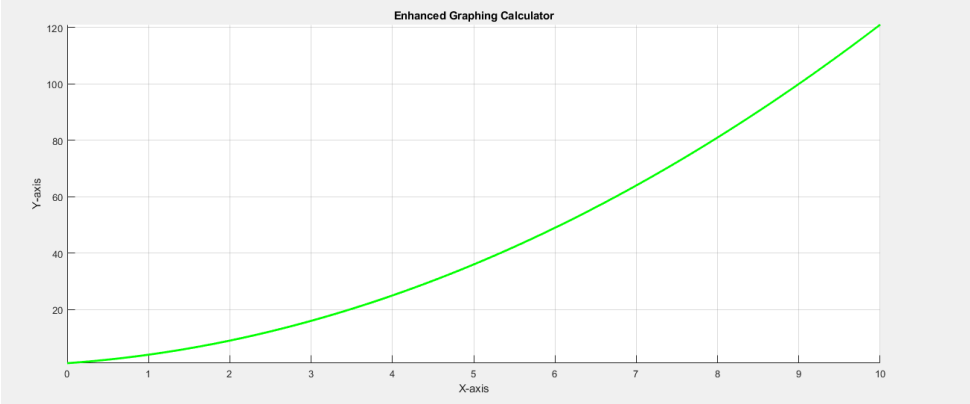
- A note in inputting values, you must use ".^" for exponents and "*" when multiplying. Disregarding this will result in an error to warn the users to input the correct format.

- Wrong format:

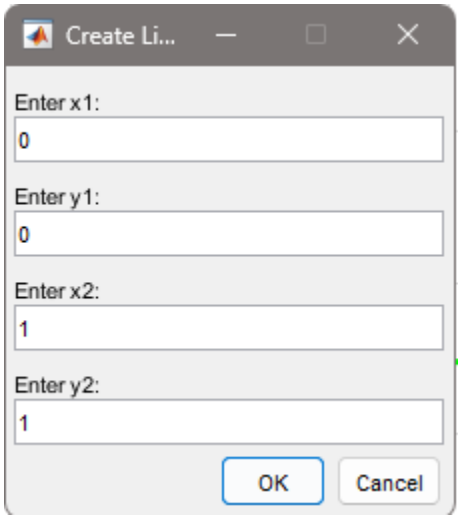


- Correct format:



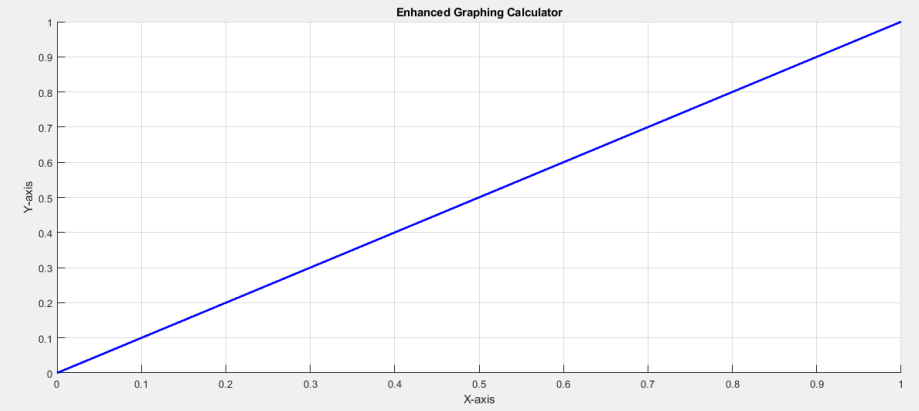


- **Feature 2**: Creating a line
  - **Screenshot**:

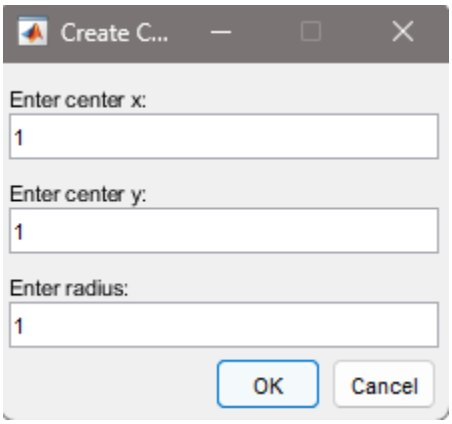


  - **Example Use Case**: In this feature, you are provided with the convenience of simply putting the values of the x and y coordinates to create a line. You can

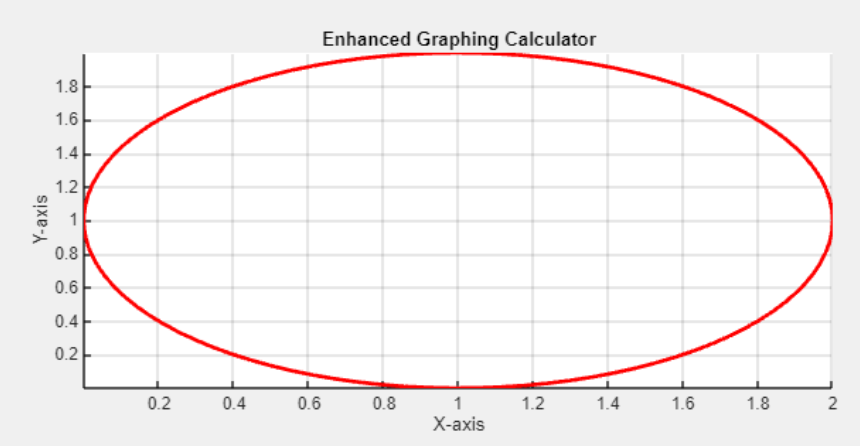
enter the default value given, which will plot a line from (0,0) and (1,1). You have the freedom of entering any values you want!



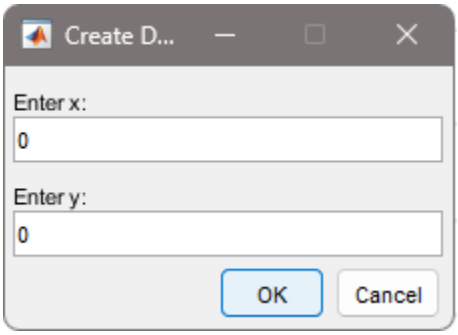
- **Feature 3**: Creating a circle
  - **Screenshot**:



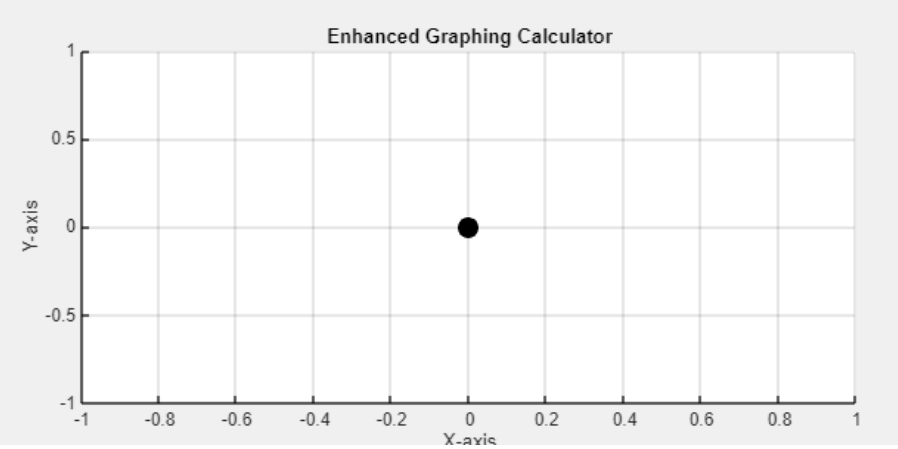
  - **Example Use Case**: In this feature, you will be asked for the coordinates of the center of the circle and what would be its radius. So you can input any values you want to locate your circle and determine its radius! By default, we can create a circle in the coordinate of (1,1) with a radius of 1. Note that the circle may look like an ellipse because of the default axis settings of MATLAB.



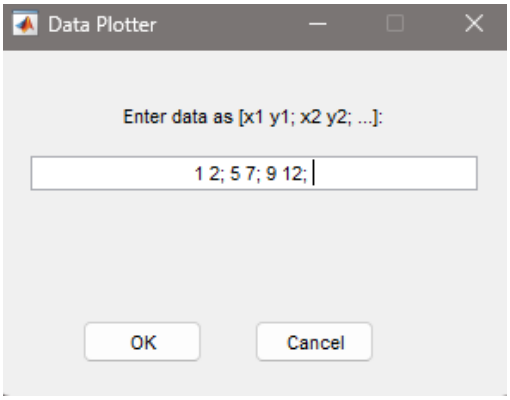
- **Feature 4**: Creating a dot
  - **Screenshot**:



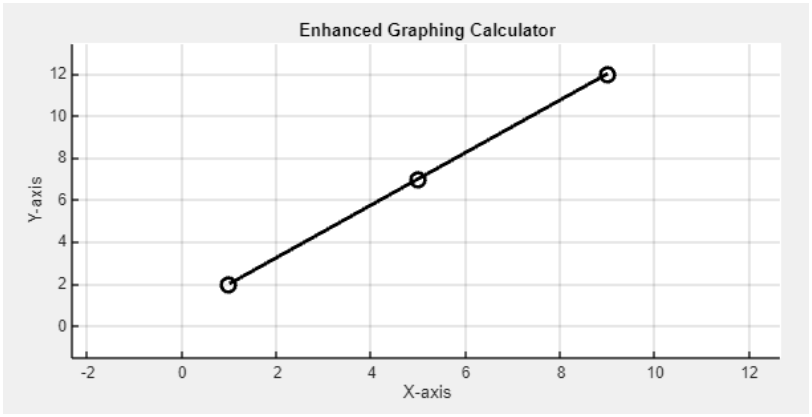
  - **Example Use Case**: In this feature, you are asked for the xy coordinates of the dot to determine its location. The default value is in origin (0,0).



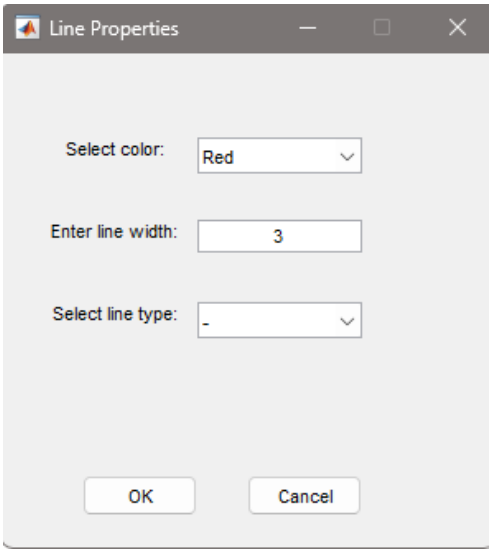
- **Feature 5**: Plotting values of x and y
  - **Screenshot**:



- **Example Use Case**: This feature allows you to enter multiple values of x and y which can be an existing data that you have. This will then be plotted in the graph and allows you to visualize your data!

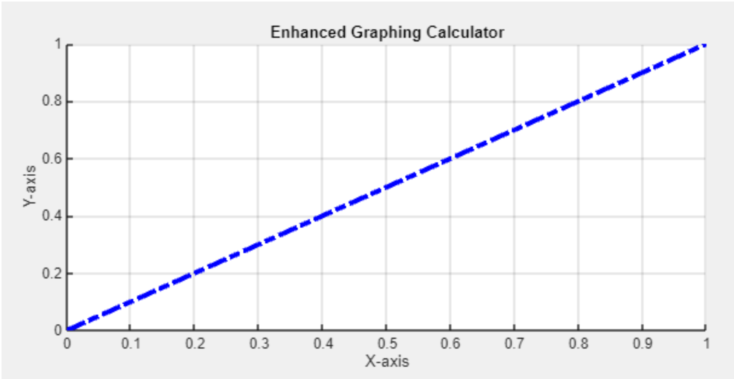


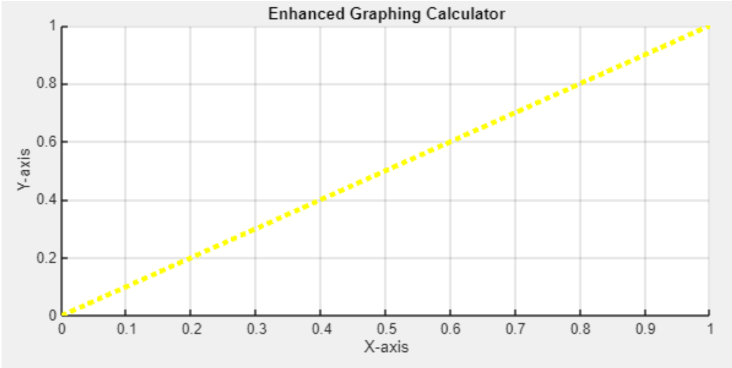
- **Feature 6**: Line properties
  - **Screenshot**:

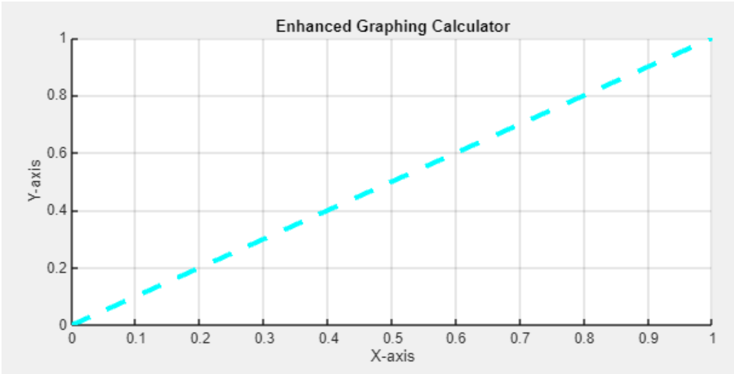


  - **Example Use Case**: This feature allows you to edit your line properties if you want different color, line type, or line width. Simply select an object that you

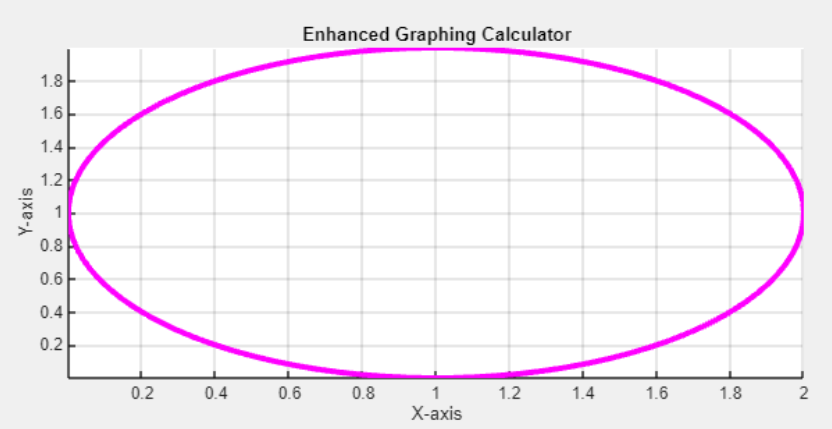
want to edit and press the “Line Properties” button. You can choose within the given options in the dropdown box.

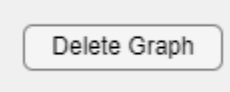






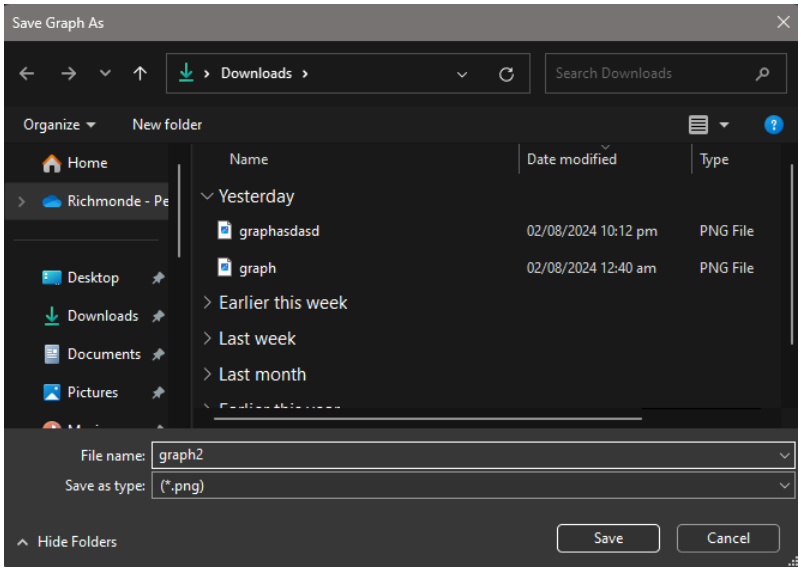
- **Feature 7**: Delete graph
    - **Screenshot**:

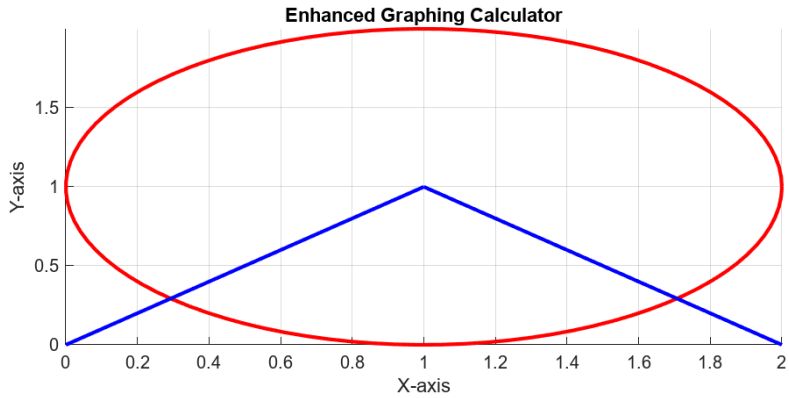




- **Example Use Case**: This feature allows you to delete a specific object in the graph. Simply select an object and press the delete button.

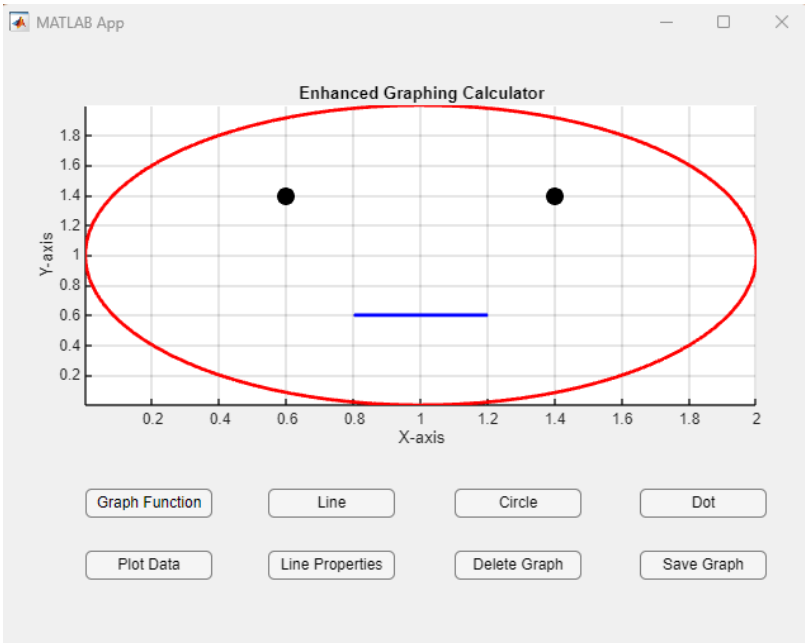
- **Feature 8**: Saving the graph
  - **Screenshot**:





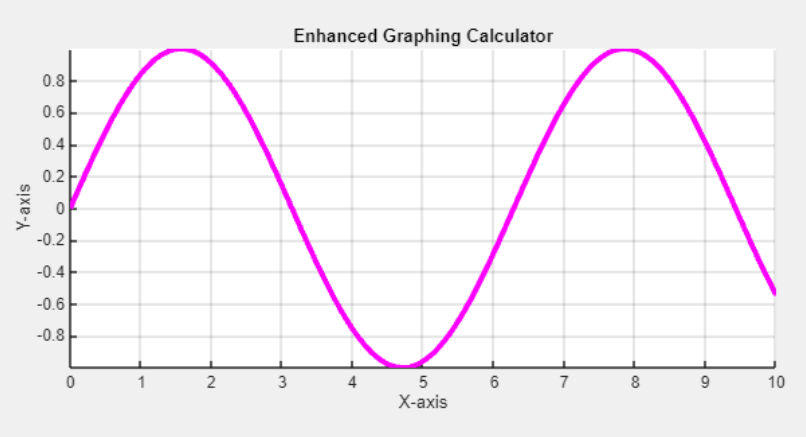
- **Example Use Case**: This feature allows you to save the graph that you have created so that you can have your own copy. Simply press the “Save Graph” button and enter your preferred file name.

- Enjoy creating figures by combining all the features in the app or use this tool as a visualization for your data!



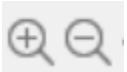
Advanced Features

- **Feature 1**: Selecting a graph
  - **Screenshot**: Interface for Feature 2.



- **Details**: This feature selects a graph that will allow the user to either edit its line properties or delete the selected graph.

- **Feature 2**: Small icons on the upper right side of the graph
  - **Screenshot**:

  - **Details**: These small icons can do the following features:
    a.  - Allows you to save the graph in the file form you prefer
    b.  - Allows you to Pan the view and move wherever you want.
    c.  - Allows you to zoom in or zoom out the graph by clicking or scrolling your mouse.
    d.  - Allows you to go back to the main graph when you are lost or zoomed too much.