MATLAB as a Development Environment for Mathematics Functions & Graphs in GUI

Er.Amrit Kaur[1], Harpreet Kaur[2]
[1]Assistant Professor, Department of Computer Engineering,
[2]Engineer, Department of Computer Engineering,
[1,2]University College of Engineering (UCOE), Punjabi University, Patiala, Punjab, India
[1]amrit.tiet@gmail.com, [2]harpreet8307@gmail.com

***Abstract* - MATLAB is a software that provides a high level programming language, many mathematic libraries and easy implementable Graphic User Interface mechanisms. This paper presents information on wide aspects of the mathematics functions, computer graphics, introduction to MATLAB, GUI and its mathematics Toolbox. This paper focuses on the methods of creating a GUI using built-in GUIDE tool. Mathematics computational model is created with the help of GUIDE tool which performs various mathematic operations, trigonometric functions, logarithmic and exponentional functions, factorial of number , fibonacci series, plotting 2-D and 3-D graphs .**

***Keywords* - MATLAB, GUIDE, M.file, mathematical function, 2-D and 3-D graph.**

I. Introduction

Matrix Laboratory is a numerical computing environment and fourth generation programming language developed by Math Works. MATLAB allows matrix manipulation, implementation of algorithms, plotting of functions and data, creation of user interfaces and interfacing with programs. MATLAB [1] is a high-performance language for technical computing. It integrates computation, visualization, and programming environment. Furthermore, MATLAB is a modern programming language environment: it has sophisticated data structures, contains built-in editing and debugging tools, and supports object-oriented programming. These factors make MATLAB an excellent tool for teaching and research. MATLAB has many advantages compared to conventional computer languages like C, FORTRAN, C++ for solving technical problems MATLAB is an interactive system whose basic data element is an array that does not require dimensioning.[4] The software package has been commercially available considered as a standard tool at most universities and industries worldwide. It has powerful built-in routines that enable a very wide variety of computations. It also has easy to use graphics commands that make the visualization of results immediately available. Specific applications are collected in packages referred to as toolbox. There are toolboxes for signal processing, symbolic computation, control theory, simulation, optimization, and several other fields of applied science and engineering.

II. GUI in MATLAB

GUIDE, the MATLAB graphical user interface development environment, provides a set of tools for creating graphical user interfaces (GUIs). These tools greatly simplify the process of designing and building GUIs. We can use the GUIDE tools to Layout the GUI. Using the GUIDE Layout Editor, we can layout a GUI easily by clicking and dragging GUI components such as panels, buttons, text fields, sliders, menus, and into the layout area. GUIDE stores the GUI layout in a FIG-file Program the GUI GUIDE automatically generates an M-file that controls how the GUI operates. The M-file initializes the GUI and contains a framework for the most commonly used callbacks for each component, the commands that execute when we clicks a GUI component. Using the M-file editor, we can add code to the callbacks to perform the functions as we wants. A GUI (graphical user interface) allows users to perform tasks interactively through controls such as buttons and sliders. Within MATLAB, GUI tools enable you to perform tasks such as creating and customizing plots (plottools), fitting curves and surfaces (cftool), and analyzing and the filtering signals (sptool).[2]

III. Displaying GUIDE window

GUIDE window contains various text buttons and .control button and properties of these buttons are see by right clicking on button and select the property inspector from popup menu where we can see the font size ,sting ,tag ,color, dimension information of the button and also can modify that one.

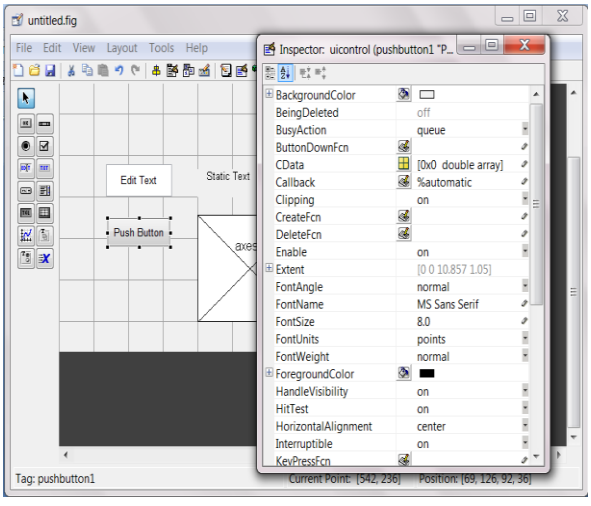

Figure 1: Display the various buttons and property inspector window in GUIDE

IV. Components of GUI

Toggle Button once pressed stays depressed and executes an action, after the second click it returns to the raised state and performs the action again. Check Box generates an action when checked and indicates its state (checked or not checked), many options might be ticked in the same time. Radio Button similar to the check box, but only one option can be selected at any given time, function starts working after the radio button is clicked. Listbox displays a list of items and enables user to select one or more from them. Popup Menu opens a list of choices when the arrow is pressed. Panel groups all components what makes interface easy and understandable, positions of all objects are relative to the panel and do not change while moving the whole panel. Button Group similar to the panel but able to manage specific behavior of radio and toggle buttons that are logically grouped. ActiveX Component allows displaying ActiveX controls that are interactive technology extensions of html. They enable sound, Java applets and animations to be integrated in a Web page.

V. Designing a GUI

After we design a GUI, we need to program each of its controls to operate correctly and consistently. Finally, we should test the completed or prototype GUI to make sure that it behaves as intended under realistic conditions. (Or better yet, have someone else test it.).If testing reveals design or programming flaws, iterate the design until the GUI works to your satisfaction. The following diagram illustrates the main aspects of this process. [3]

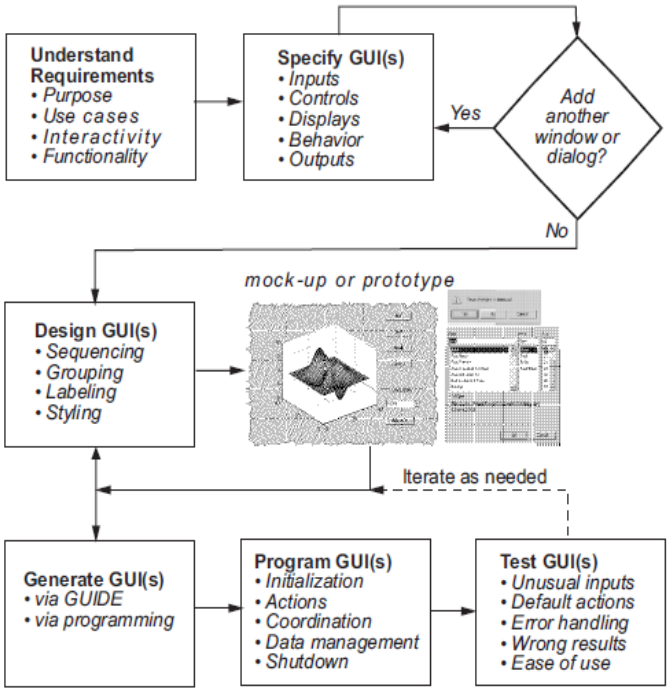

Figure 2: Design flow of GUI

VI. M-File creation and processing

When we save our GUI layout, GUIDE automatically generates an M-file that we can use to control the way the GUI works. This M-file provides code to initialize the GUI and organizes the GUI callbacks. Callbacks are functions that execute in response to user generated events, such as a mouse click. Using the M-file editor, we can add code to the callbacks to perform the functions as we want.

VII. Application of GUI

MATLAB may behave as a calculator or as a programming language and MATLAB combine nicely calculation and graphic plotting. Mathematics computational model is created with the help of mathematics functions and graph plotting functions.

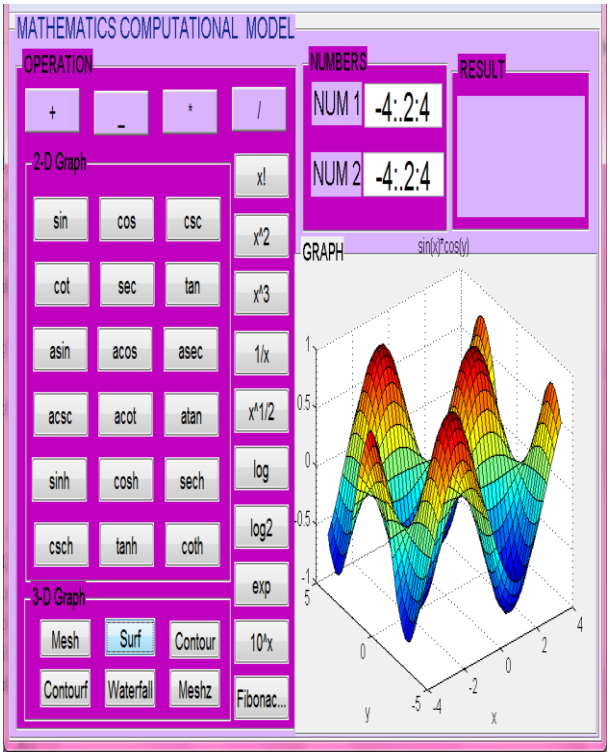

Figure 3: This window contains various mathematical functions and graph axes with surf graph plot.

MATLAB offers many predefined mathematical functions for technical computing which contains a large set of mathematical functions. There is a long list of mathematical functions that are built into MATLAB. These functions are called built in function. Many standard mathematical functions, such as sin(x), cos(x), tan(x), e^x, log(x), are evaluated by the functions sin, cos, tan, exp, and log respectively in MATLAB. [3]

This model contain mathematic functions and computation is easily done by just clicking on the selected button and input of the data is given in the NUMBERS

panel by NUM1, NUM2 edit text and output is displayed in RESULT panel. If output is graphics related then it is displayed in GRAPH panel.

VIII. Conclusion

MATLAB is a software that provides a high level programming language, with the help of this we can create a mathematical computational model which consists of various mathematics functions, factorial of the given number, fibonacci series, 2-D and 3-D graphs. This model saves times for doing all this computation as it contains all these functions in the form of button, only need to click the button and the result is displayed.

References

[1] The MathWorks Inc. MATLAB 7.0 (R14SP2). The MathWorks Inc., 2005.

[2] MATLAB® 7 Creating Graphical User Interfaces by MathWorks.

[3] David Houcque Northwestern University ,"Introduction to MATLAB for engineering students", (version 1.2, August 2005).

[4] Jaun A. Fraile, Javier Bajo, Juan M. Corchando and Ajith Abraham. "Applying wearable solutions in dependent environments". IEEE Transaction on Information Technology in Biomedicine, Vol. 14, No. 6, pp 1459-1467, November 2010.