Penerbit Akademia Baru

International Journal of Advanced Research in Future Ready Learning and Education

https://www.akademiabaru.com/submit/index.php/frle/index
ISSN: 2462 - 1951

Akademia Baru Publishing (M) Sdn Bhd
International Journal of
ADVANCED RESEARCH IN FUTURE READY LEARNING AND EDUCATION

Graphical User Interface for Solving Non-Linear Equations for Undergraduate Students

Sharifah Noha Zahirah Syed Abdul Nasir[1], Nurul Ain Ab Wahab[1], Mohd Agos Salim Nasir[1,*]

[1] School of Mathematical Sciences, College of Computing Informatics and Media, Universiti Teknologi MARA, Selangor, Malaysia

ABSTRACT

Solving non-linear equations problems for root-finding is more challenging rather than linear equations problems. Generally, this is because it cannot be solved by using the standard formula. The numerical method is one of the ways that can be applied, consisting of an iterative procedure. A simulator named Root Calculator to Solve Non-Linear Equations was developed, where it is a graphical user interface (GUI) system that can be used to approximate the root based on the equation and interval given. Four methods were chosen which are the Graphical Method, the Bisection Method, Newton's Method, and the Secant Method to produce the approximation root. App Designer in MATLAB R2020b software was used for this development.

Keywords:
Non-linear equations, numerical method, approximation root, graphical user interface (GUI), App Designer

Received: 1 April 2023 **Revised: 20 May 2023** **Accepted: 21 May 2023** **Published: 6 June 2023**

1. Introduction

The root of an equation can be defined as the certain values of the variable which satisfied the given equation. In other words, it is a solution to the equation in the form of a number or an algebraic formula[1], [2]. There are various methods for solving the equation to determine the approximation root depending on the type of the equation itself [3].

The numerical method is one of the ways in solving the problem of the non-linear equation [4], [5]. The Graphical method, Bisection method, Newton's method, and Secant method are examples of the numerical solution [6], [7]. It is an iterative procedure that will approximate the root of the equation by finding the initial estimation of the solution and subsequently followed by successive numerical procedures to get the accurate estimation of the root [8].

A simulator is referred to as a system of real-life program simulation [9]. It is a powerful system that enhances the user's experience by providing them with a much friendlier interface through a visual approach where the user does not need to type the commands line by line as they only must manipulate the symbolic representations provided [10], [11]. It is also has a consistent way of navigating across varieties of applications using visual clues because it already self-explanatory [12]. As in this project, the simulator of approximation root will simulate how the root is obtained

[*] *Corresponding author.*
E-mail address: mohdagos066@uitm.edu.my

based on the function and interval given by applying the four selected numerical methods. The graph as the visualization aid was used that will give a deeper understanding to the end user. The usage of mathematical software which is MATLAB R2021a was used in the development of this simulator as the application of technology that will facilitate the project journey and time efficiency.

So, this project was carried out to develop a simulator to approximate the root using numerical methods. This simulator will approximate the root based on the input keyed in by the end user. The system will process the input based on the selected numerical method by the user to produce the approximated root as the main output. Other than that, the graph of the function and other details are also included to improve the interactivity of the simulator.

1.2 Research Question

Our project aims to develop a system for solving non-linear equations to estimate the root of the function. Hence, our main research question is:

"How we can apply technology in the education system to help the students in their learning process for Mathematics subject?"

1.3 Research Objectives

The objectives of this study are:

1. To develop a system that can find the root of non-linear equations based on the input keyed in by the user.
2. To visualize the approximate root graphically.
3. To help students to check for their answers easily in their learning processes.

2. Methodology

2.1 Project Framework

Table 1 shows the project framework for the development of a Graphical User Interface, GUI for Solving Non-linear Equations for root findings by using the suggested methods which are the Graphical Method, Bisection Method, Newton's Method, and Secant Method.

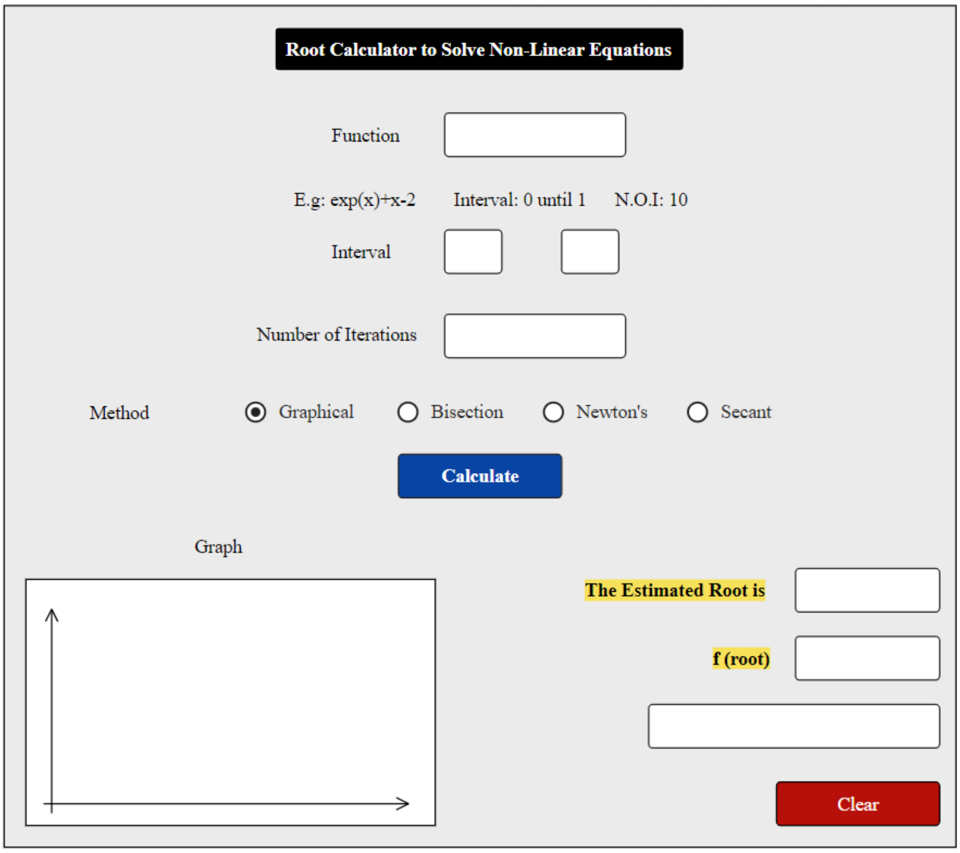
Table 1
Project Framework

Phase	Activities	Techniques / Tools	Deliverables / Outcomes
Analysis	1. Preliminary study 2. Information gathering	1. Articles 2. Online journals	3. Background of the study 4. Problem statement 5. Objective

			6. Scope and limitation of the project 7. Literature review
Design	1. Determine software and hardware requirements 2. Design a graphical user interface 3. Design a flowchart	1. Microsoft Office Word 2019 2. Moqups 3. Draw.io	4. Software and hardware requirements 5. Graphical user interface 6. Flowchart
Implementation	1. Implement the suggested methods 2. Set up the development environment 3. Start coding activity	1. MATLAB R2020b 2. AppDesigner	3. Simulator of root findings
Testing	1. Performance testing 2. Accuracy testing 3. Fixing the error	1. Performance test 2. Accuracy test	3. Result analysis
Documentation	1. Finalize project 2. Reporting	1. Microsoft Office Word 2019 2. Grammarly	3. Final report

2.2 *Graphical User Interface*

The suggested early design for the graphical user interface of this simulator was designed using the Moqups web application before the real design will be developed. Moqups is a visual collaboration tool that combines whiteboard, diagram, and design features in a single-online app. This web application is easy to access and convenient in order to design the early graphical user interface. It has most of the shapes needed in application design and it can be dragged and dropped easily. Figure 1 below shows the early design for the graphical user interface of the simulator that will be developed.



E.g: exp(x)+x-2 Interval: 0 until 1 N.O.I: 10

Method ● Graphical ○ Bisection ○ Newton's ○ Secant

Fig. 1. Graphical User Interface Design

2.3 Flowchart

A flowchart diagram gives a clear picture of how this simulator will function. Figure 2 shows the flowchart of the simulator for Root Calculator to Solve Non-Linear Equations. This root-finding simulator for solving non-linear equations problems is designed for the purpose to approximate the root based on the function and interval given as the input from the user. After the user inserts the function and interval needed, they need to choose which method to use either Graphical Method, Bisection Method, Newton's Method, or Secant Method in order to solve the function. Next, the user needs to click on the Calculate button for the system to start the calculation and produce the output which is the estimated root and graph of the function. If the user wants to insert a new function, they just need to click on the Clear button so that the system can be reset and the user can repeat the same procedure again.

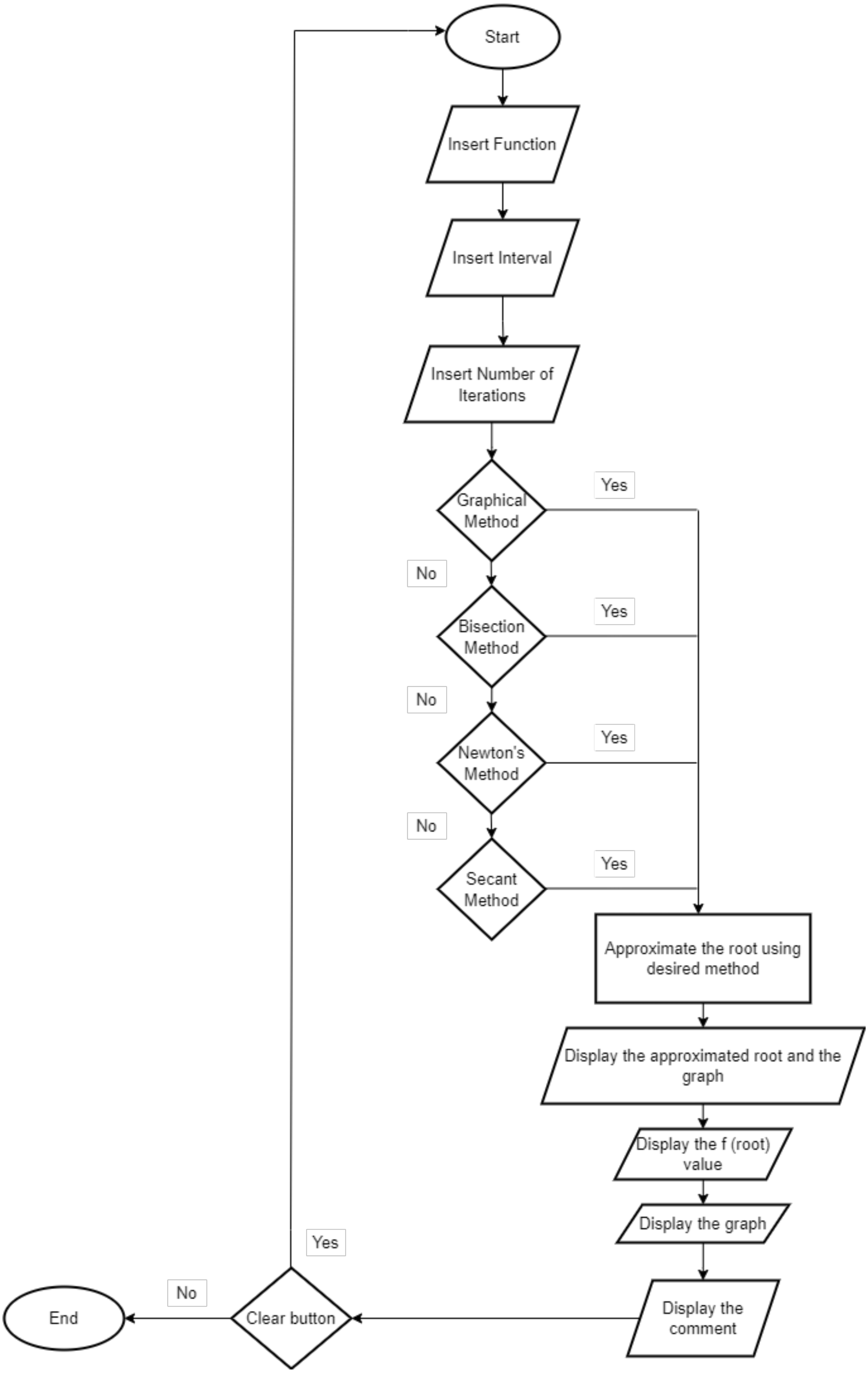


Fig. 2. Flowchart for the Simulator of Root Calculation

3. Results and Discussion

The result of the testing phase will be shown in this section. The phase is a critical part of the project to determine whether the objectives of the project are achieved or not, besides testing the functionality of the system. The result obtained by the system will be compared with the result done by calculating it manually.

3.1 Functionality Testing

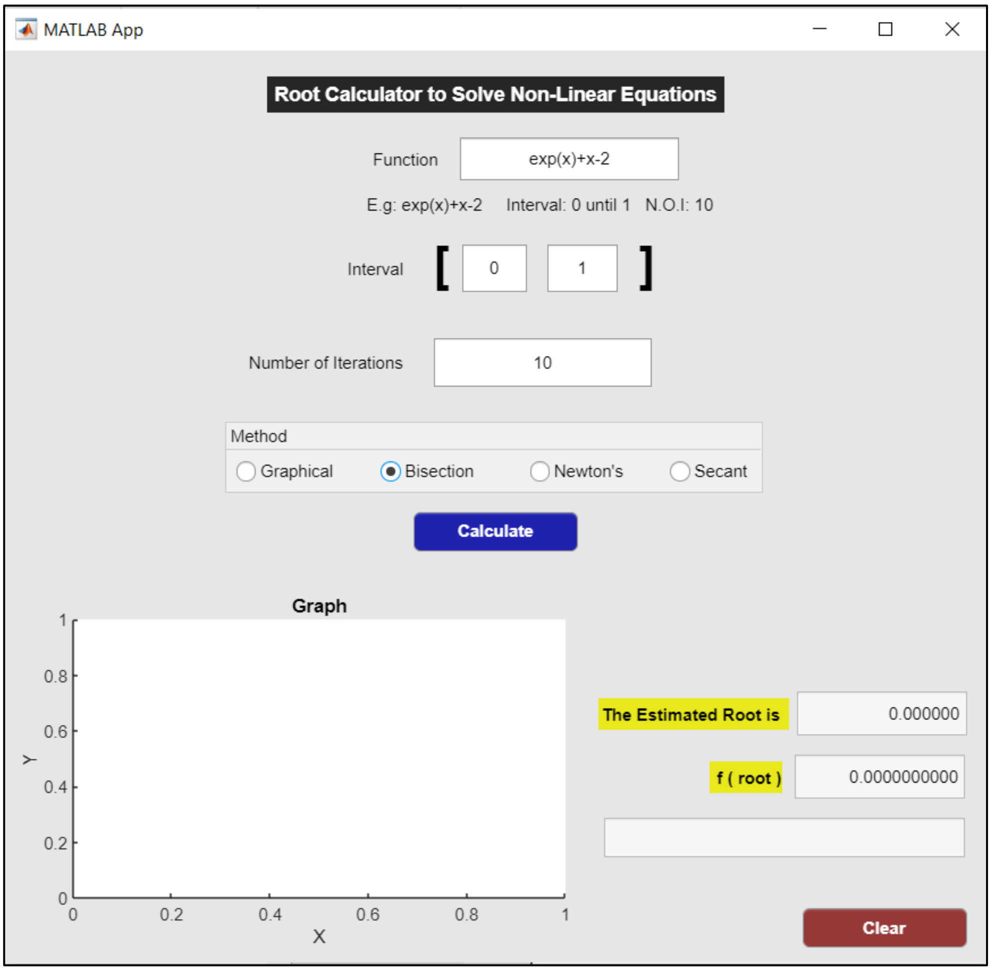
We tested the functionality of each button and component in the system to make sure they function well. We used a sample equation and inputs to see if the system will display the output we desire.

3.1.1 Function Equation

To test the system, we used the equation as shown below

$$f(x) = \mathrm{e}^{x} + x - 2 \tag{1}$$

The root of the equation is calculated within the interval from 0 to 1 with 10 iterations. For the first test, we will be using the Bisection method. Figure 3 shows the interface after inputting the details.



MATLAB App

Root Calculator to Solve Non-Linear Equations

Function: exp(x)+x-2

E.g: exp(x)+x-2 Interval: 0 until 1 N.O.I: 10

Interval [0 1]

Number of Iterations: 10

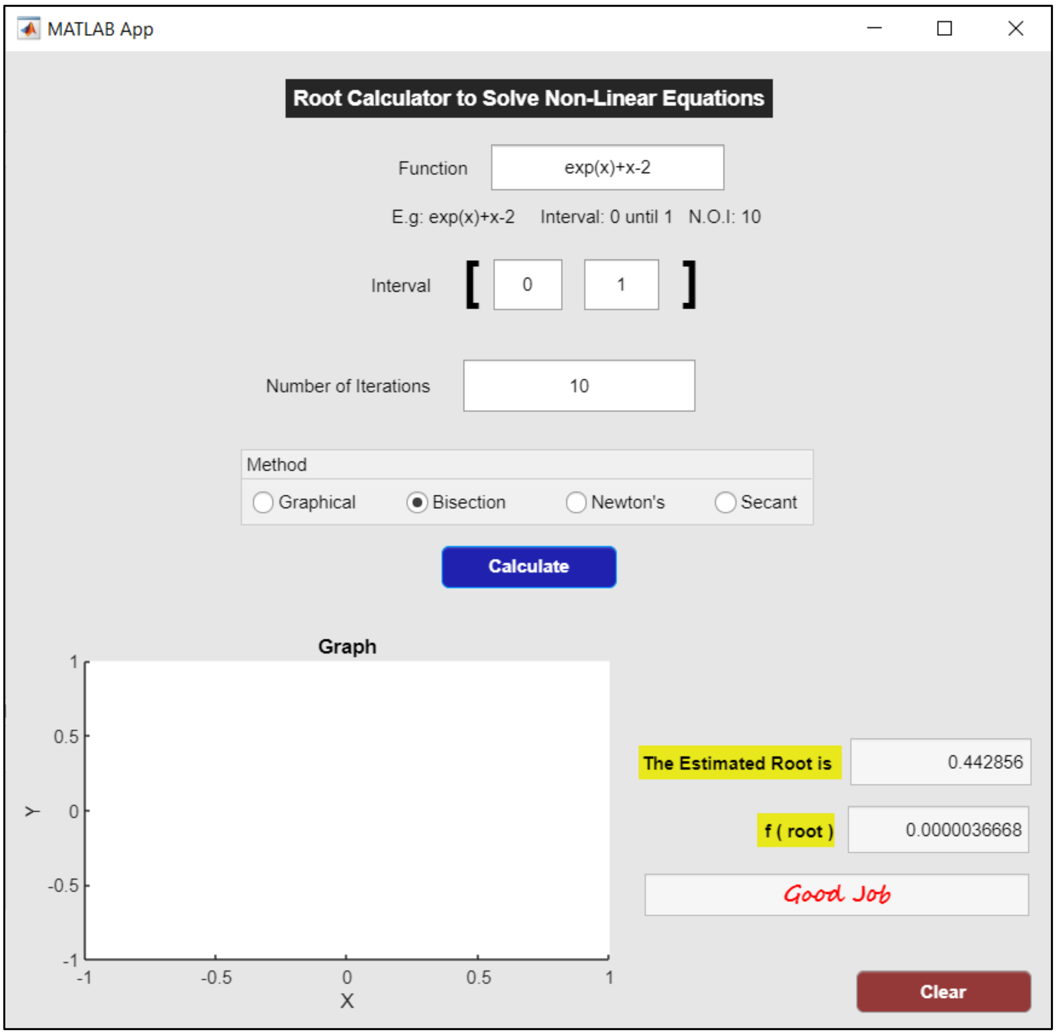
Method: ○ Graphical ● Bisection ○ Newton's ○ Secant

The Estimated Root is: 0.000000

f (root): 0.0000000000

Fig. 3. Inputting function in the GUI

The next step is to press the Calculate button. Figure 4 illustrates the output displayed after the Calculate button is pressed.



MATLAB App

Root Calculator to Solve Non-Linear Equations

Function: exp(x)+x-2

E.g: exp(x)+x-2 Interval: 0 until 1 N.O.I: 10

Interval [0 1]

Number of Iterations: 10

Method: ○ Graphical ● Bisection ○ Newton's ○ Secant

The Estimated Root is: 0.442856

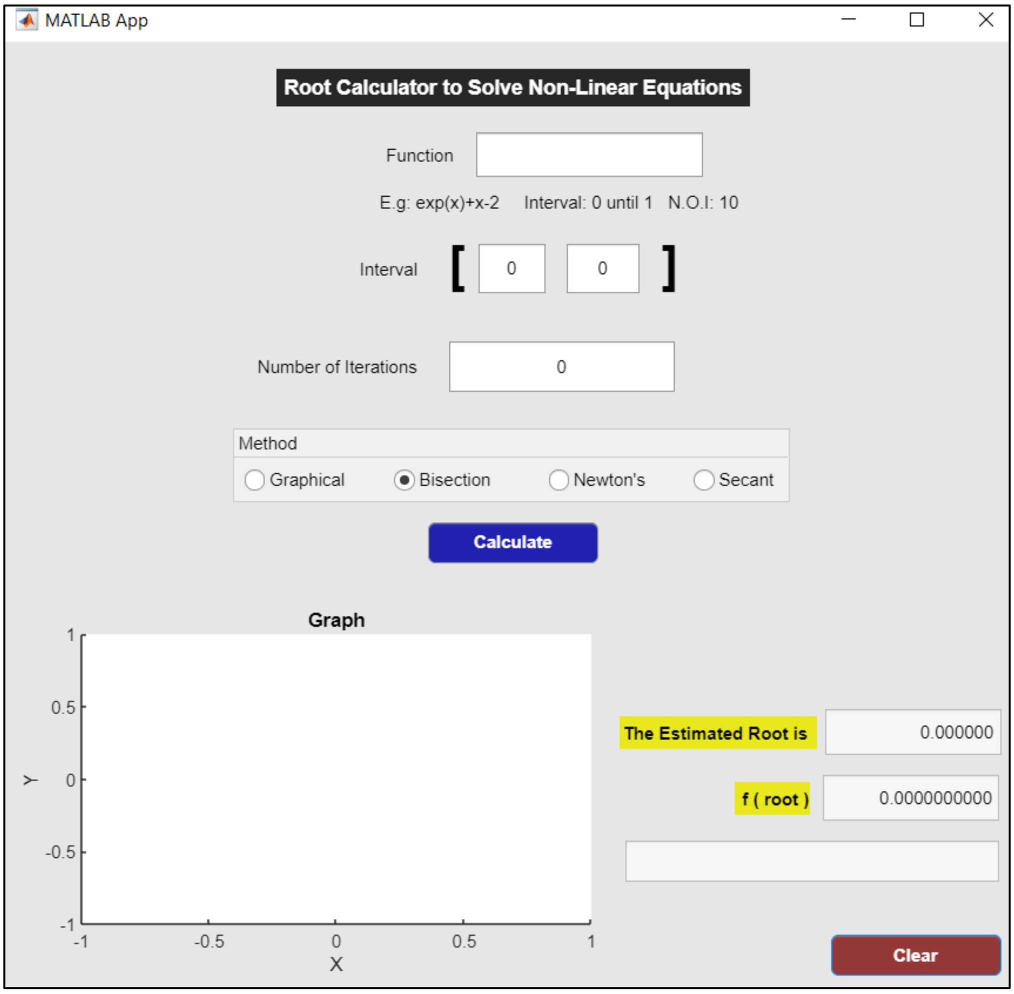
f (root): 0.0000036668

Good Job

Fig. 4. Output display after pressing the Calculate button

The Calculate button is deemed to be functioning well when the estimated root, the value of the function, and the graph are displayed. From the above figure, we can see that all of the expected outputs are displayed clearly which concludes that the button is functioning well.

Following this button is the test for the Clear button which is expected to clear all the output and input that was keyed in previously. The result after pressing the Clear button is shown in Figure 5



MATLAB App

Root Calculator to Solve Non-Linear Equations

Function

E.g: exp(x)+x-2   Interval: 0 until 1   N.O.I: 10

Interval [0 0]

Number of Iterations 0

Method
○ Graphical ● Bisection ○ Newton's ○ Secant

The Estimated Root is 0.000000

f (root) 0.0000000000

Fig. 5. Testing the Clear button

It is shown that all the fields have been cleared after pressing the Clear button. This proves that the Clear button is also functioning well.

The functionality testing concludes that all the buttons and components are operating well and the coding in the system are giving the desired output. Now that we are sure that every component is usable, we will do the accuracy test to check how accurate the result of the output is.

3.2 Performance Testing

The performance of the output displayed by the system is tested during this phase. The test is performed to ensure that the system is both accurate and reliable to use. To confirm the accuracy, a comparison between the output displayed by the system and the manual calculation will be carried out.

3.2.1 Function Equation

To test the accuracy of the system, several types of functions are used as the input. The accuracy will be tested by comparing the result obtained by the system with the result calculated manually.

(a) Exponential Function

We first started the test by using equation (1) and computing its root manually. The steps for the manual calculation are shown below [13].

1) Identify the existence of a root.
There exists a root in [0,1].

2) Compute $c = \frac{a+b}{2}$ and $f(c)$.

$$c = \frac{0+1}{2} = 0.5$$
$$f(0.5) = 0.1487$$

3) Determine

$$f(0) \cdot f(0.5) < 0$$
$$f(0.5) \cdot f(1) > 0$$

Then $r\epsilon(0{,}0.5)$.

4) Repeat until the desired iteration. After the 12th iteration, we will get the value of 0.4426 with the initial root being set as 1.

From the above calculation, the root for this equation is 0.4426 whereas, in the previous section, we saw that the system displayed the output of the root of 0.442856 or 0.4429 in 4 decimal places.

To see whether the system is performing well, we insert the root into the original equation and the nearer the result is to zero, the better it is.

Table 2
Comparison table for the exponential function

Platform of calculation	Root obtained	*f(root)*
System	0.442856	4.08887×10^{-6}
Manual	0.4426	$-6.5049002 \times 10^{-4}$

Based on the comparison table, we can see that the system produced a better result since it is nearer to 0 [14].

4. Conclusion

The simulator for approximating the root in solving non-linear equations using four numerical methods provided was developed and this successfully met the first objective of this project. In the first objective, the approximated root that will produce is based on the input keyed in by the user which are the function, interval, number of iterations, and the method that was chosen. The output that will produce is the estimated root of the function, the graph, the value when the root is substituted into the function which is the *f(root)* value, and the comment. The graph based on the function given will help in understanding deeper with the way of illustration and this met the second objective of this project which is to display the approximate root in a graph [15]. This simulator can be used especially by the students in their learning processes to check for the answer quickly and accurately anytime they want and this fulfills the last objective of this project.

Lastly, from this project, the developer gained much new knowledge in developing a new graphical user interface by using one of the popular mathematical software which is MATLAB R2020b.

Even though there are so many challenges that need to face throughout this project, this experience will be very useful to the developer in the future.

References

[1] "Root | mathematics | Britannica." https://www.britannica.com/science/root-mathematics (accessed Jan. 20, 2023).

[2] Z. Sharer, "CHAPTER 3 ROOTS OF EQUATION," 2018.

[3] Maheshwari, Amit Kumar. "A fourth order iterative method for solving nonlinear equations." *Applied mathematics and computation* 211, no. 2 (2009): 383-391.

[4] Grosan, Crina, and Ajith Abraham. "A new approach for solving nonlinear equations systems." *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*38, no. 3 (2008): 698-714.

[5] Grosan, Crina, and Ajith Abraham. "A new approach for solving nonlinear equations systems." *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*38, no. 3 (2008): 698-714.

[6] Epperson, James F. *An introduction to numerical methods and analysis*. John Wiley & Sons, 2021.

[7] Valentine, Daniel T., and Brian H. Hahn. *Essential MATLAB for engineers and scientists*. Academic Press, 2022.

[8] Noor, Muhammad Aslam, Faizan Ahmad, and Shumaila Javeed. "Two-step iterative methods for nonlinear equations." *Applied mathematics and computation* 181, no. 2 (2006): 1068-1075.

[9] "Computer Simulations: Definition, Examples, Uses | Built In." https://builtin.com/hardware/computer-simulation (accessed Jan. 27, 2023).

[10] Depcik, Christopher, and Dennis N. Assanis. "Graphical user interfaces in an engineering educational environment." *Computer Applications in Engineering Education* 13, no. 1 (2005): 48-59.

[11] B. J. Jansen, "The Graphical User Interface".

[12] S. Q. Sherrick, "An Introduction to Graphical User Interfaces and Their Use by CITIS".

[13] "text note - Google Drive." https://drive.google.com/drive/u/0/folders/109jIER9ZXDVezJOykQJHLbgalVYR1GZg (accessed Jan. 20, 2023).

[14] Süli, Endre, and David F. Mayers. *An introduction to numerical analysis*. Cambridge university press, 2003.

[15] Mayer, Richard E. "Using multimedia for e-learning." *Journal of computer assisted learning* 33, no. 5 (2017): 403-423.