TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A Project Report

On

Equation Visualizer

Submitted By:

Amrit Kandel (THA078BEI004)

Prasish Timalsina (THA078BEI026)

Sandesh Panthi (THA078BEI035)

Mingmar Gyaltsen Lama Sherpa (THA078BEI021)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

May, 2023

COPYRIGHT

Head

Department of Electronics and Computer Engineering

Thapathali Campus, Institute of Engineering

Lalitpur, Kathmandu

Nepal

ACKNOWLEDGEMENT

We would like to offer our genuine gratitude towards the Department of Electronics and Computer Engineering, Thapathali Campus for providing us with this chance to learn and carry our insight in the form of minor ventures. We are additionally grateful to our Supervisor Er. Saroj Shakya for his encouragement and enormous help for our project. We would also like to acknowledge the creators of different papers and the developers of different programming libraries which we have used as a reference for building our projects. We would like to recognize the equal contributions of every member of our project “EQUATION VISUALIZER”. Every member did a lot of work in researching and contributing to the project. Lastly, we would also like to thank every individual who is directly or indirectly related for building our projects.

Amrit Kandel (Class Roll No.: 078/BEI/004)

Prasish Timalsina (Class Roll No.: 078/BEI/026)

Sandesh Panthi (Class Roll No.: 078/BEI/035)

Mingmar Gyaltsen Lama Sherpa (Class Roll No.: 078/BEI/021)

ABSTRACT

Mathematics is a basic and important discipline. Equations are the mathematical statements that are made up of two expressions connected by an equal sign. Coordinate geometry could be a difficult chapter for students in secondary or even higher levels. Most of the time, students feel it is difficult to visualize the equations and cannot relate to real life. It is also difficult for teachers to explain the topic. This project “Equation Visualizer” can be helpful to both the students and the teachers in the teaching and learning process. With the motto of making mathematics fun, we have put forward this project. It takes the equation form the user, analyzes the equation and plots the graph of the equation.

Keywords: Co-ordinate, equation

Table of Contents

List of Figures

List of Abbreviations

CPP	C Plus Plus
IDE	Integrated Development Environment
API	Application Programming Interface
2D	Two Dimensional
GUI	Graphical User Interface

1. INTRODUCTION

1.1 Background Introduction

Equations are the foundation of modern mathematics, but many students find it quite difficult to understand. If there was a way for the students to visualize the equation in the form of the graph, it would have been easy for them to understand it. So, this project “equation visualizer” was made with the motto to provide the understanding of the equation through graph. This can clarify the properties and behaviours of the equation, as well as in solving problems involving the equation.

With the increasing need of mathematics in this modern era it is necessary for every student to learn it. Keeping that in mind, equation visualizer is a program created where students can give their equation as input and see them rendered visually in the form of graph.

An equation visualizer is a concept that plots the equation in the graphical form. It is basically the programs used to plot a wide variety of equations, including linear equation, polynomial equation and many more. Many schools, colleges and online tutoring platforms are using equation visualizer in their teaching process. Instead of plotting the equation in the graph paper a user can get it plotted by the use of equation visualizer.

1.2 Motivation

The main motivation behind this project is to make learning mathematics fun and easy. We as students have faced the problem in the past to precisely plot the equation in the graph as we didn’t have access to any tools to visualize it. With the objective of improving education by providing a tool that helps people better understand and analyze mathematical equations, this project is made. [3]

1.3 Objectives

The main objectives of our project are listed below**:**

- To make Coordinate geometry easier to learn.
- To increase the efficiency, accuracy and save time while plotting the graph.

1.4 Scope

This program can be used in schools and colleges to visualize the equation to the students in the form of a graph. It can be particularly useful for the students who struggle with traditional methods of learning. It can be used to create online tutoring platforms that allow students to receive personalized math instructions and receive visual explanation of math concepts. People wanting to learn more about mathematical equations are highly encouraged to use this project.

We can further add 3d to this program which can render the 3-dimensional shapes of the given equation which helps to learn the equation and relate it to real life.

2. LITERATURE REVIEW

There have been quite some apps and websites for various operating systems throughout the years. Graphing calculators have also been in use since long. The earliest graphing calculator is said to have been designed in 1921 by electrical engineer Edith Clarke. The calculator was used to solve problems with electrical power line transmission. The first commercial graphing calculator was launched by Casio in 1985.

These graphing calculators nevertheless come with an expense. They cost significantly more in comparison to the equation visualizers found on the web today and are bound to get broken someday. So, equation visualizers or graphing tools are preferred to graphing calculators. [2]

There are some similar applications that have already been developed. Some of them are discussed below.

2.1 Microsoft Graphing Calculator

A Graphing feature, which initially was enrolled in an insider preview and later became generally available for all, provides a clean and interactive experience to allow people to visualize equations and help them learn more about subjects like linear Algebra. You can also plot graphs for trigonometric equations (equations that use sine, cosine, and tan, etc.), establish graphs for linear and quadratic inequalities, and even plot graphs for various kinds of functions like modulus, absolute values, and the greatest integer function. You can also share graphs with the contacts synced with your Microsoft account, a feature that I believe can come in handy for people who make use of linear algebra frequently. However, you have to be subscribed to windows to use the feature due to which it is not accessible to all those in need of the tool. .

2.2 Desmos Graphing Calculator

Desmos is a leading organization that has been contributing a lot in the field of geometry. It has built a powerful graphing tool with the help of which users can view the graph with any valid mathematical equation. It is very useful to the users but it has some limitations too. The Desmos Graphing Calculator is an online based web page which requires internet access. Since, many of the rural places of Nepal do not have access to the internet, we decided to make our project an offline application.

3. METHODOLOGY

This system is mainly based on three main steps. They are: 1) Taking input from the user, 2) analyzing the entered equation and 3) Showing the result in the form of a graph.[2]

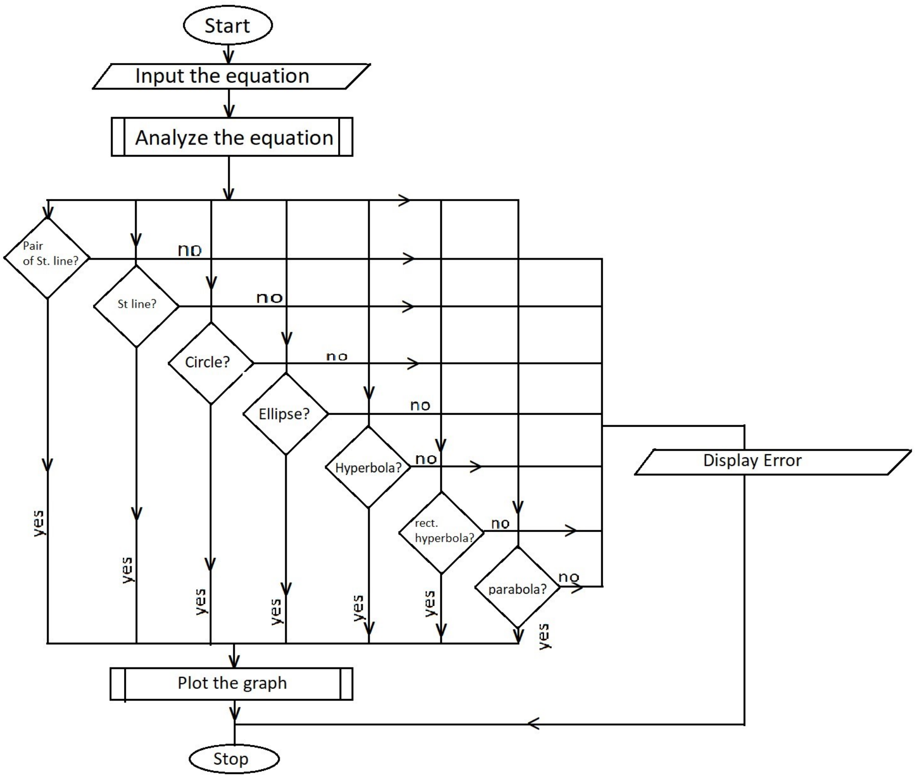


Figure 3-1: Basic flowchart of the system

3.1 Input from the user

The input is taken from the user simply from the console. It is taken in the form of a general second degree equation. Then the coefficients are arranged, stored in variables for further use.

3.2 Analyzation of the entered data

The coefficients obtained from the entered equation are then studied and analyzed to identify the particular type of curve. Many mathematical equations and formulas are used to do so. After the type of equation is known, the further process is proceeded to draw the graph of the curve as an output.

3.3 Plotting of the graph

After the type of the curve is identified, the process for plotting the graph is started. It includes various steps and methods. At first, the axis of the graph is plotted and then the boundary. After that, the remaining lines to form a graph are plotted. At last, the specific curve of the entered equation is plotted. Almost all the curves are plotted by using different algorithms due to their complexity.

4. SYSTEM DESCRIPTION

As mentioned earlier, this program is based on three main steps, but inside these three steps, many algorithms, and methods are used to draw the specific curve of the given equation. All the mechanisms will be elaborated below.

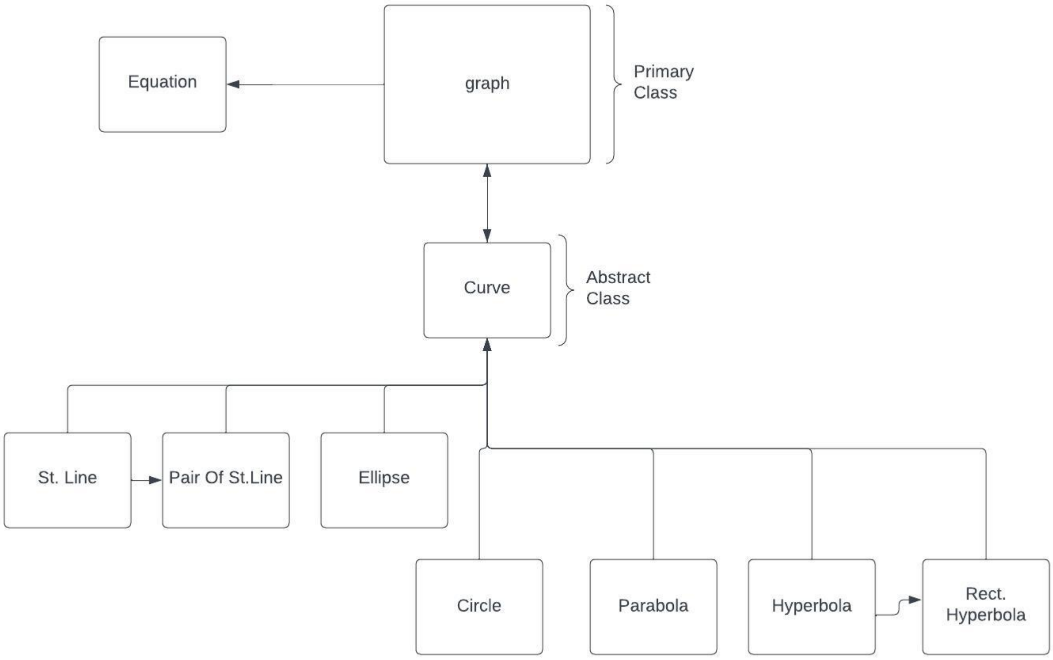


Figure 4-2: Class diagram of the system

4.1 How did we take the equation?

The input is taken in the form of an equation in the console. For this, simply iostream is used and the necessary texts are printed in the console with the help of ‘cout’ and ‘cin’ keywords. To do so, a member function of a class Equation has been used. The function takes the input from the user and initializes the data member (variables for the coefficients of the equation).

4.2 Methods to identify the type of equation

After all the coefficients of the standard second-degree equation are obtained, the curve can be analyzed and can be determined what kind of curve the equation represents. The equation can represent at most seven types of curves (i.e. straight line, pair of straight lines, circle, ellipse, hyperbola, rectangular hyperbola or parabola). We can compare the coefficients for different curves as follow:[1]

4.2.1 Straight Line

A straight line is the curve that is represented by the equation $y = mx + c$. If all the coefficients of second-degree x and y are zero then the curve formed is a straight.

For a general second degree equation to be a straight line, following conditions should be matched:
($ax^2 + by^2 +2hxy + 2gx + 2fy + c = 0$) -> a second degree equation

- $a = 0$
- $b = 0$
- $h = 0$

If the above conditions are matched, then we can confirm that the given equation is of a st. line.

4.2.2 Pair of Straight Line

For the given equation to be a pair of straight lines, it should meet the following criteria.

($ax^2 + by^2 +2hxy + 2gx + 2fy + c = 0$) -> a second degree equation

Let $\nabla = abc + 2fgh – af^2 -bg^2 -ch^2$

- a, b must not be 0
- $\nabla = \quad 0$

4.2.3 Ellipse

For the given equation to be an ellipse, it should meet the following criteria.

($ax^2 + by^2 +2hxy + 2gx + 2fy + c = 0$) -> a second degree equation

Let $\nabla = abc + 2fgh – af^2 -bg^2 -ch^2$

- $\nabla \neq 0$
- $a, b \neq \quad 0$
- $h^2 - ab < 0$

4.2.4 Circle

For the given equation to be a circle, it should meet the following criteria.

($ax^2 + by^2 +2hxy + 2gx + 2fy + c = 0$) -> a second degree equation

Let $\nabla = abc + 2fgh – af^2 -bg^2 -ch^2$

- $a = b$
- $h = 0$
- ∇ must not be zero

4.2.5 Parabola

For the given equation to be a parabola, it should meet the following criteria.

($ax^2 + by^2 +2hxy + 2gx + 2fy + c = 0$) -> a second degree equation

Let $\nabla = abc + 2fgh – af^2 -bg^2 -ch^2$

- $\nabla \neq 0$
- Either a or b should be 0
- $h^2 - ab = 0$

4.2.6 Hyperbola

For the given equation to be a hyperbola, it should meet the following criteria.

($ax^2 + by^2 +2hxy + 2gx + 2fy + c = 0$) -> a second degree equation

Let $\nabla = abc + 2fgh – af^2 -bg^2 -ch^2$

- $\nabla \neq 0$
- $a, b \neq 0$
- $h^2 - ab > 0$

4.2.7 Rectangular Hyperbola

For the given equation to be a rectangular hyperbola, it should meet the following criteria.

($ax^2 + by^2 +2hxy + 2gx + 2fy + c = 0$) -> a second degree equation

Let $\nabla = abc + 2fgh – af^2 -bg^2 -ch^2$

- $\nabla \neq 0$
- $a + b = 0$
- $h^2 - ab >0$

4.3 Plotting

The plotting consists of three main steps: drawing the axis, drawing the boundary and then the main curve of the equation. Since we have used the SFML library of the C++ language, various functions and data members of the library were used. At first, we rendered a basic window and started plotting everything mentioned above in that window.`

At first, the maximum width and height of the window is obtained by using the getSize() function of the sfml library. Then the center of the graph was calculated. Then, a horizontal and a vertical line was drawn passing through the center of the graph.

Like the axis, since we have the maximum height and width of the window, we can draw four lines to form a square workspace between which the curve is to be plotted. Let xmax and ymax be the maximum width and height of the window. Then the first line will be from (0, 0) to (0, ymax). Second line will be from (0, 0) to (xmax, 0). Similarly, the third line will be from (0, ymax) to (xmax, ymax) and the fourth line will be from (xmax, 0) to (xmax, ymax).

After that, the horizontal and vertical lines that form multiple units or boxes are displayed. For that, the square workspace is divided into several units as per specific ratio. At first, the x-axis is divided into several points. Then the vertical lines are drawn passing from upper horizontal boundary to lower horizontal boundary to obtain a series of vertical lines. Similarly, the horizontal lines are also obtained.

After the axis, boundary and other lines of graphs are drawn, its time for the main curve of the equation. We have already identified the type of the curve of the equation by analyzing the coefficients of the equation.

- If the equation is of a straight line, then the slope and the y-intercept of the equation is calculated by using the formula slope (m) = -(a/b) and y-intercept = -(c/b). Then the points that lie in the boundary of the workspace and satisfy the equation y = mx + c are obtained. There are only two of such coordinates. We obtain the straight line by joining those points.
- If the equation is of a pair of straight lines, then the second degree equation is separated to two linear equations. It means we obtain two equations of straight lines from a pair of straight lines. After that we apply the same methods as of a straight for those two equations separately.
- If the equation is of a parabola, we know that only two forms of the equations are possible, either $y = ax^2 + bx +c$ or $x = ay^2 + by + c$. If it is in the form $y = ax^2 + bx +c$, we take some points (say 100 points) on the x-axis at uniform distance and we put the values of x in the equation to find the value of y. Then we collect all the points. Similarly, if it is of the form $x = ay^2 + by + c$, then we do the same but for the y-axis. We put the values of y to find the values of x and collect the points. After we collect all the points that satisfy the equation of parabola, then we join the points serially to form a parabola.
- If the equation is of an ellipse, then we divide equally divide the complete angle (360^0) to a number of angles. We also have the length of major axis (a) and length of minor axis (b) from the equation. Then, we can find the coordinates satisfying the equation by using the parametric equation of ellipse as: x = a cos(angle) and y = b sin(angle). After enough points are collected, we join them to form an ellipse.
- If the equation is of a circle, we do the following steps. We know the equation of a circle is in the form $x^2 + y^2 + 2gx + 2fy + c = 0$. We compare the equation to find the value of g, f and c. Then we can calculate the radius and center of

the circle as: radius = sqrt(g^2 + f^2 -c) and center (-g, -f) and easily draw a circle.

- If the equation is of a hyperbola, Then as in the ellipse, we divide the 2D-plane to many angles and use the angles to find the point satisfying the hyperbola by using the parametric equation of hyperbola as: x = a sec(angle) and y = b tan(angle) and then join the points to form a hyperbola.

4.4 Tools And Environment

4.4.1 Tools

- Visual Studio Code (IDE)
- Visual Studio Community (IDE)
- GitHub (Version Control System)

4.4.2 Libraries

- SFML (GUI)

5. RESULTS AND ANALYSIS

5.1 Results

The Equation Visualizer project achieved the following results.

5.1.1 User Interface

A visually appealing and user-friendly interface was developed, allowing users to easily input mathematical equations and interact with the visualizations.

5.1.2 Equation Parsing

The equation parsing algorithm was successfully implemented, enabling the tool to handle a wide range of mathematical expressions and equations.

5.1.3 Visualization

The project implemented various visualization techniques, including graph plots, 2D graphs based on the equation input. The visualizations accurately represented the mathematical relationships depicted by the equations.

5.1.4 Performance

The equation visualizer demonstrated satisfactory performance, delivering real-time visualizations even for complex equations and large datasets.

5.1.5 Usability

User feedback indicated a high level of usability, with users finding the tool intuitive and easy to navigate.

5.2 Analysis

Based on the project’s result, the following analysis can be made.

5.2.1 Accuracy

The Equation Visualizer successfully translated mathematical equations into visual representations, ensuring accuracy in conveying mathematical relationships.

5.2.2 Flexibility

The tool's ability to handle a wide range of equations allowed users to explore various mathematical concepts and visualize them effectively.

5.2.3 Impact

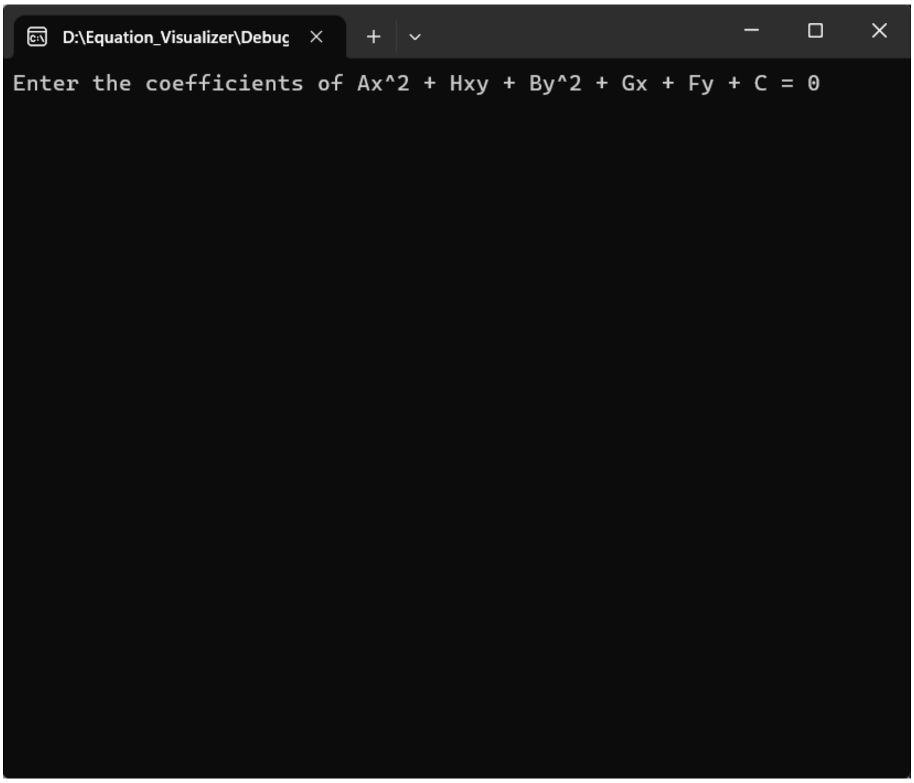
The equation visualizer project has the potential to benefit a wide range of users, including students, researchers, and professionals in fields like mathematics, engineering, and physics.

5.2.4 Further Improvements

While the project achieved its objectives, there are opportunities for further enhancements. These may include expanding the supported equation types, incorporating advanced visualization techniques, and refining the user interface based on user feedback. This project can be made able to view more than one equation at the same time so that users can compare the equation and learn more. Similarly, the characteristics of each curve can also be included in the output.

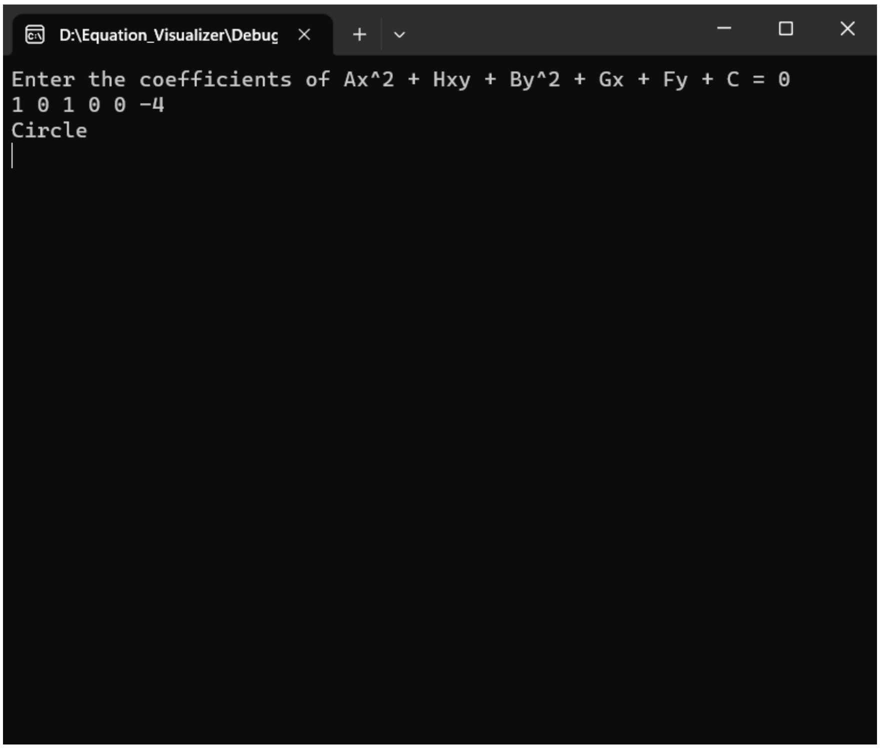
5.3 User Interface

The user interface of our project can be seen in the figures below. Our project runs as our plan and architecture. The different examples of output and graph obtained from our project are included below.



```
Enter the coefficients of Ax^2 + Hxy + By^2 + Gx + Fy + C = 0
```

Figure 5-3: User Interface to enter the equation



```
Enter the coefficients of Ax^2 + Hxy + By^2 + Gx + Fy + C = 0
1 0 1 0 0 -4
Circle
```

Figure 5-4: Example of entering equation

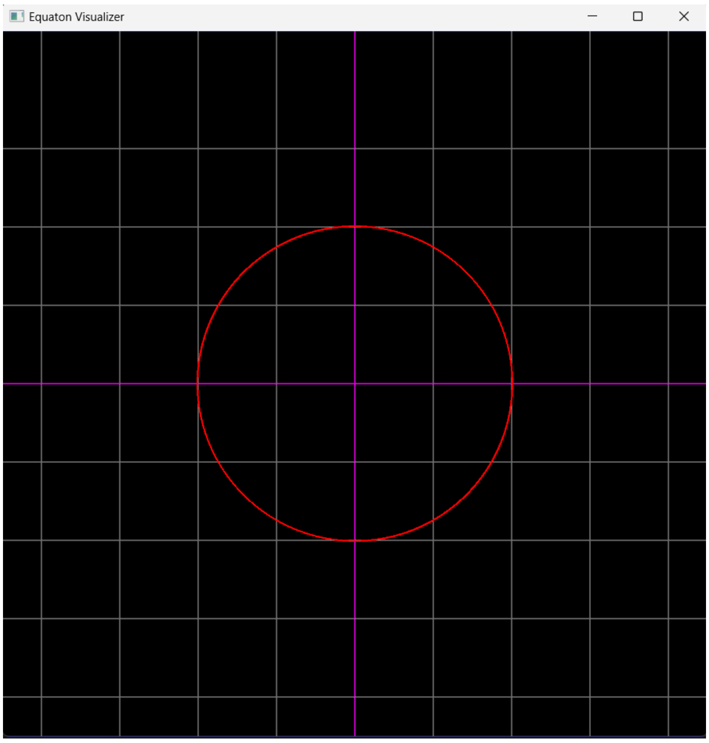


Figure 5-5: Graph of a circle

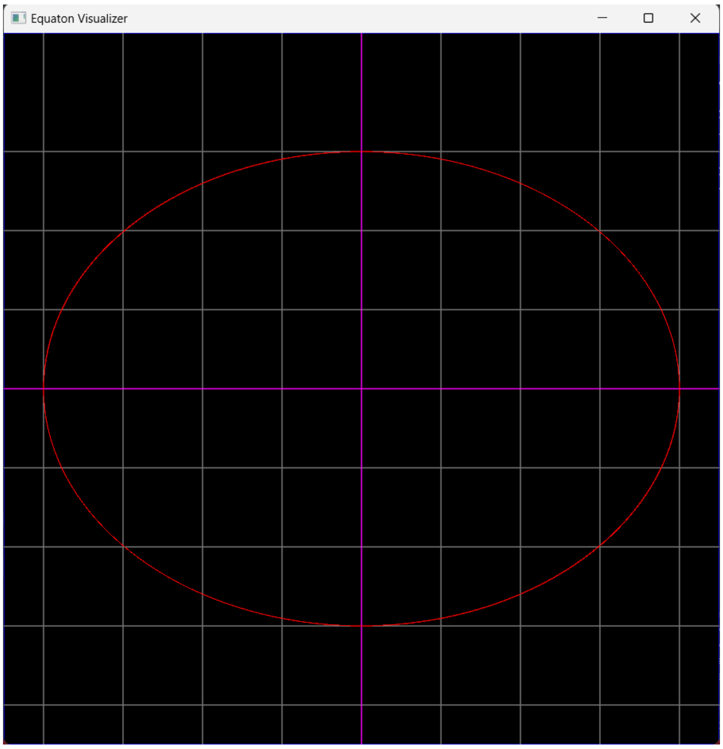


Figure 5-6: Graph of an ellipse

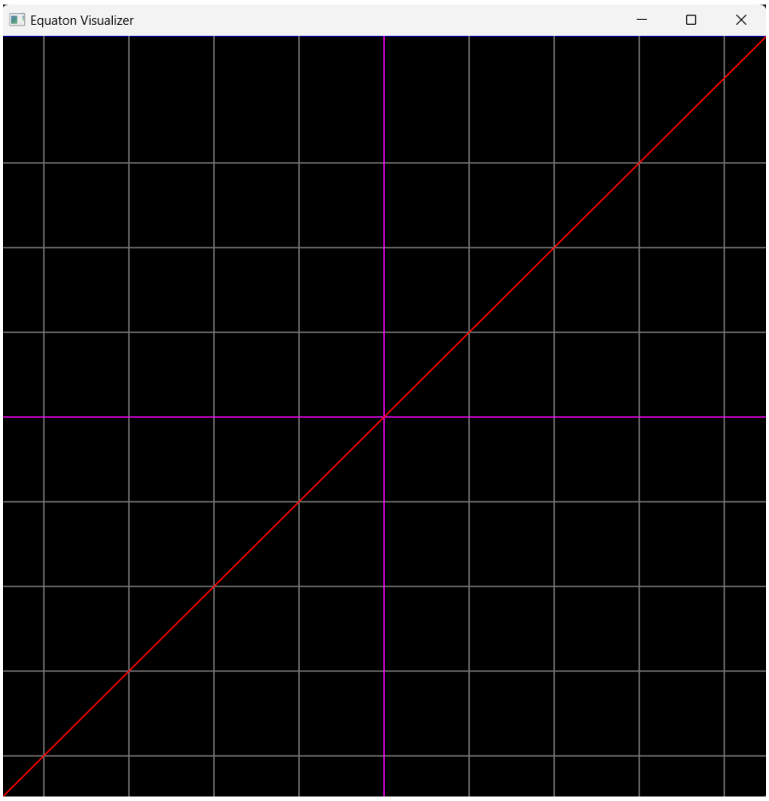


Figure 5-7: Graph of a Straight Line

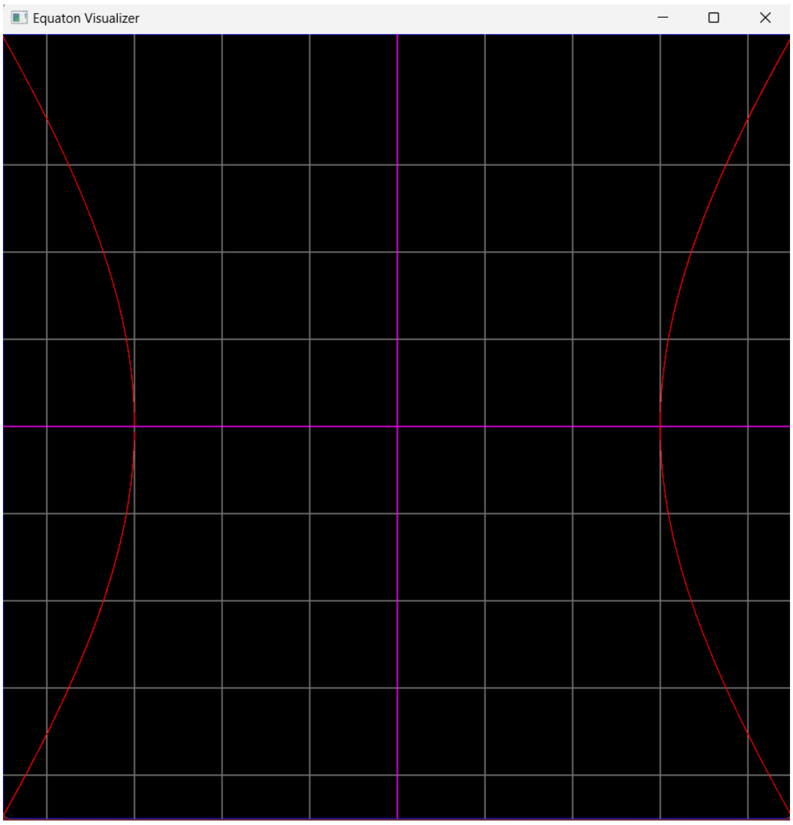


Figure 5-8: Graph of a hyperbola

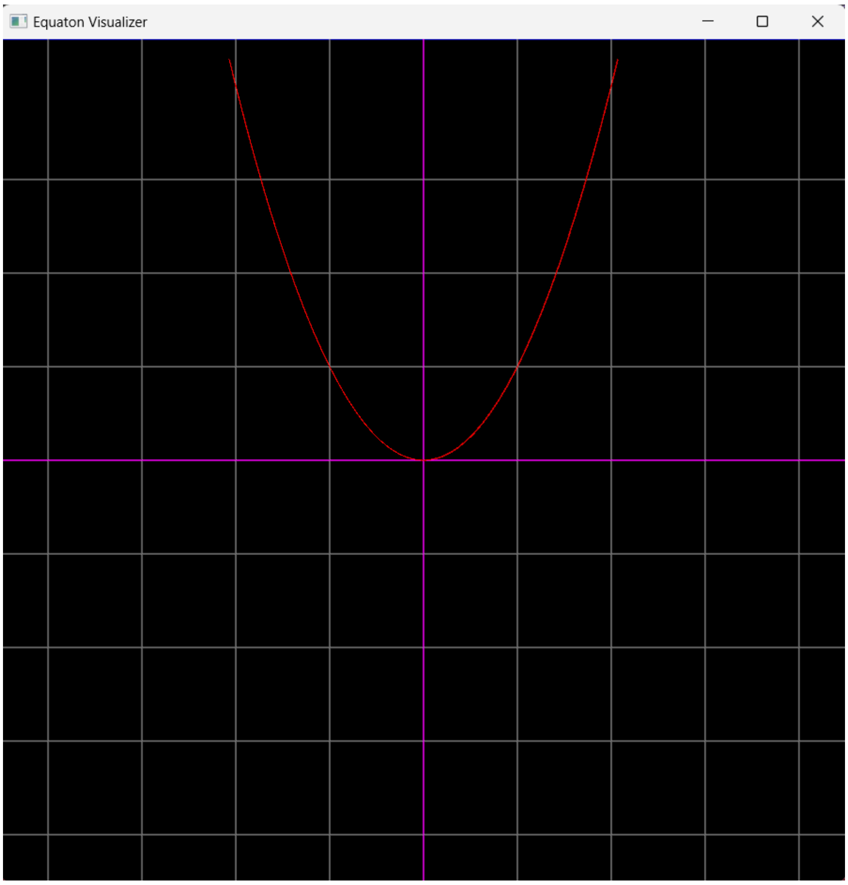


Figure 5-9: Graph of a parabola

6. CONCLUSION AND FUTURE ENHANCEMENT

6.1 Conclusion

The Equation Visualizer project successfully developed a tool capable of parsing mathematical equations and generating accurate visual representations. The project's results demonstrate the effectiveness and usability of the tool, with potential applications across various domains. Continued improvements and enhancements could further enhance the tool's functionality and impact.

Overall, the Equation Visualizer project has been a valuable endeavor, contributing to the field of mathematical visualization and providing users with a powerful tool to explore and understand mathematical concepts.

6.2 Limitations

Nothing is perfect. This project has some limitations too. Some of them are listed below.

- The zoom in and zoom out feature is limited to a certain range only.
- The graph does not have numbering for now. So it may be difficult to read the data from the graph.
- Sometimes, the graph takes some time to render.
- The project is unable to display the curves that are rotated by some angle i.e. the equation obtained after transformation can not be shown.

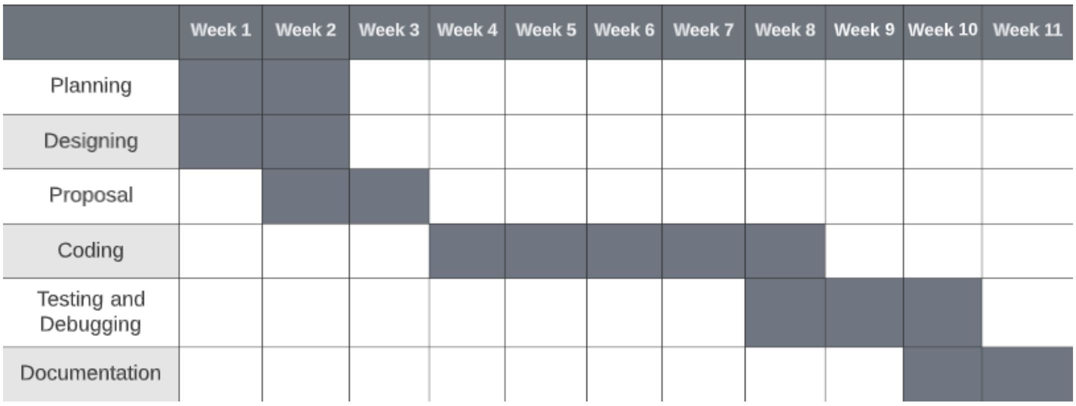
6.3 Future Enhancement

There is still something in the project in which we can work more and make it more user-friendly, reliable and efficient. Some of these are listed below.

- The feature of unlimited zoom can be added to the project.
- The basic characteristics of the curve can be displayed along with the curve as well.
- The output that is being taken through the console for now can be completely done with the help of GUI.

7. APPENDICES

Appendix A: Gantt Chart



	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11
Planning	■	■									
Designing	■	■									
Proposal		■	■								
Coding				■	■	■	■	■			
Testing and Debugging								■	■	■	
Documentation										■	■

References

[1] D. R. Bajracharya, R. M. Shrestha, M. B. Singh, Y. R. Sthapit, and B. C. Bajracharya, *Basic Mathematics - Grade XII*. Sukunda Pustak Bhawan, 2078.

[2] R. D. Bhatta, *A Textbook of Object Oriented Programming in C++*. Kathmandu: Vidyarthi Pustak Bhandar, 2072.

[3] E. Milou, “The Graphing Calculator: A Survey of Classroom Usage,” *School Science and Mathematics*, vol. 99, no. 3, pp. 133–140, Mar. 1999, doi: 10.1111/j.1949-8594.1999.tb17461.x.