

# **Com S 319 Portfolio 3**

## **Graph**

**A graphing tool for drawing graph according to different functions**

**D\_319Group 18**  
**Group member: Zijie Deng**

## Table of Contents

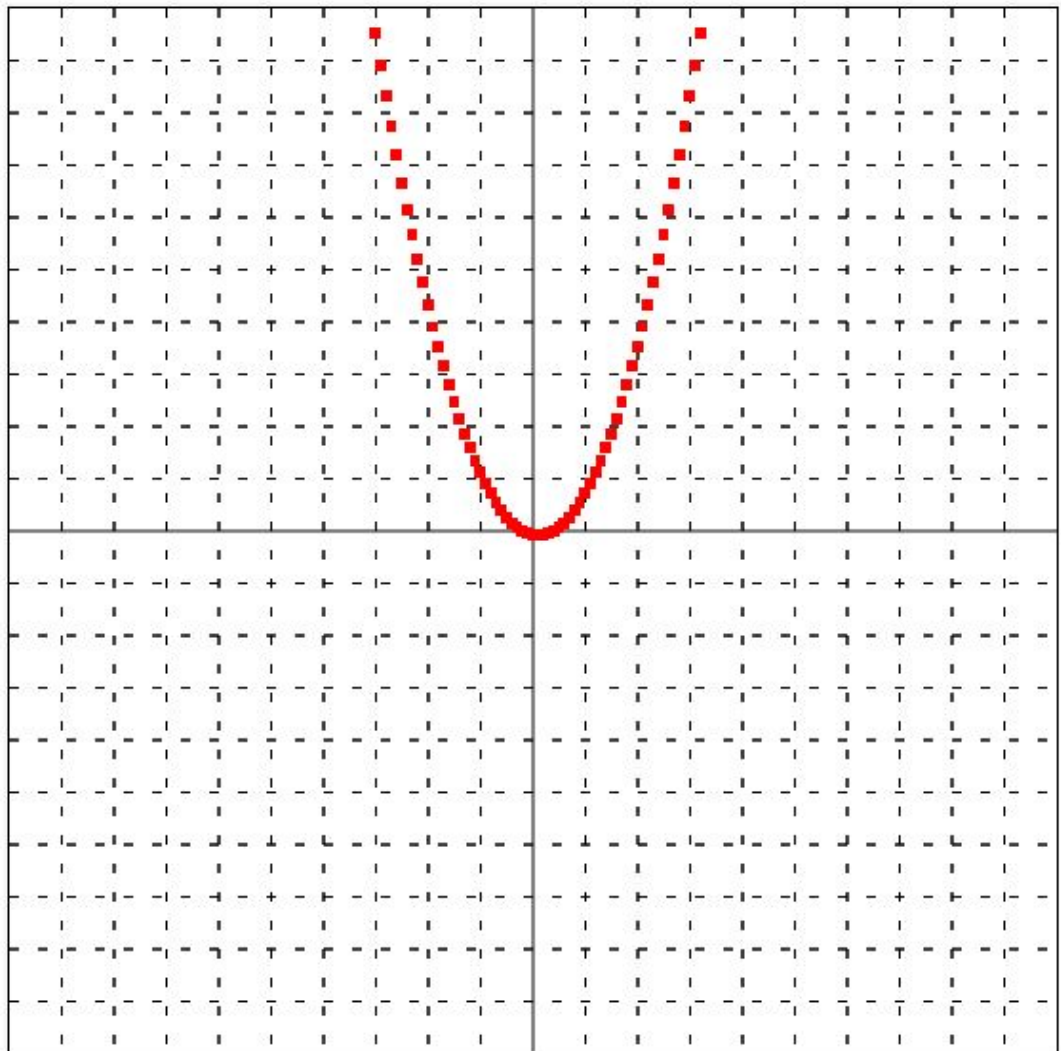
- **Introduction ----- Page 3 - 5**
- **New & Complex ----- Page 6 - 7**
  - Zoom in and Zoom out (Change the range)**
  - Animate lines**
  - Turn math equation into a graph, including math.min.js**
- **Bloom's Taxonomy ----- Page 8 - 11**
  - Plot function**
  - Draw init grid.**
  - From Cartesian**
  - Animation**

## Introduction

This is a project about drawing and showing the graph according to an inserted math equation. The graph will be drawn on a 550\*550 canvas, and users can zoom in and zoom out by using the setting section. Those math functions and variables can be entered in Plot section, also they will be shown in a list as history in Functions section. Other than that, users can also change the style, size, and color of the lines in Design section.

- Canvas:

The canvas section provides a 550\*550 canvas for drawing and showing the graph. Also the scale of this graph will be shown on the canvas, which means that users can see the the graph clearer and in details. Note that the  $(0, 0)$  is the intersection of two solid line.



- **Setting:**

This section is in charge of the “zoom in” and “zoom out” functionality. Users can enter numbers in the text area to set the area that they want to see on the graph. This consist of the range of x and the range of y. By clicking on the “Clear/Refresh” button, the previous graph will be removed and the new graph will be shown on the canvas in the setted range.

- **Plot:**

Plot allows users to enter their variable and math function that need to be turned into graph. The default variable is x, and default function is  $x^2$ .

### Setting

X: range	<input type="text" value="-10"/>
to	<input type="text" value="10"/>
Y: range	<input type="text" value="-10"/>
to	<input type="text" value="10"/>

Clear/Refresh

### Plot

Variable:	<input type="text" value="x"/>
Function:	<input type="text" value="x^2"/>

Add Function

- **Functions:**

This is a list of history function, which means that this list would show all the functions that have been entered.

- **Design:**

The design section is the section for graph design. In details, users can change the font size, grid size, and animate speed by entering numbers in the textarea. Also users are able to switch to different color for a line.

## Functions

$x^2$

## Design

Font Size:

Grid Size:

Grid Style:

Select

⬆

⬇

⬆

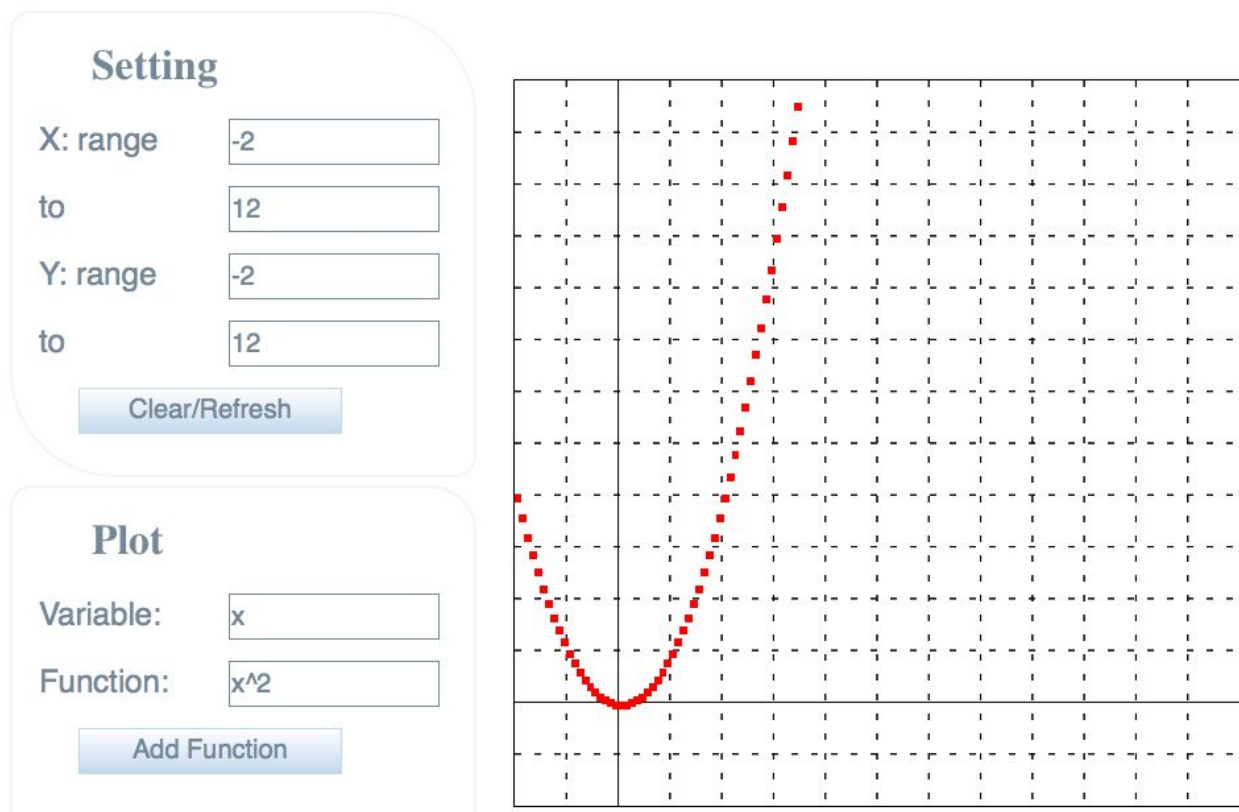
⬇

Color:

## New & Complex

### Zoom in and zoom out (Change the range):

This the default range of this graph is from -10 to 10 on the x-axis, and -10 to 10 on the y-axis, which means that the intersection is at the middle of the canvas. Users can not only view the graph generated according to functions, but also select the range of the graph that they desire to view, including zoom in and zoom out. In order to revise the range, there is a setting section that contains the range of x and the range of y, which allows users to change the lower and upper bound for both x-axis and y-axis.



## Animate lines:

Another new feature is about animating the lines. Usually the graphing tools would show the whole graph immediately. But on our website, the lines are shown gradually, which means that the line of a function would be drawn on the graph from the left side to the right side of the canvas. Users can also change the style of the line, like animate speed, and the color of the line, by making changes in design section.

## Turn math functions into graphs:

`math.js`

<https://github.com/josdejong/mathjs>

Math.js is an extensive math library for JavaScript and Node.js, It features real and complex numbers, units, matrices, a large set of mathematical functions, and a flexible expression parser.

This project included math.js, in order to help draw all kinds of math functions on the graph. It takes a String and then give us an output.

For example:

It takes function  $2+3$ , and returns 5.

It takes function  $\sqrt{1234567}$ , and returns 1111.11070556.

It can take any functions, including those really complex ones, and return the result.

## Bloom's Taxonomy

### Plot function

This method will plot function. This is the main function for plotting. It converts from Cartesian coordinates to canvas coordinates based on where the origin is on the canvas and based on max and min values of x-axis range and y-axis range.

```
// expression is of form like "x^2"
function plotfunction(expression, color, size) {
    for (x = xmin; x<=xmax; x+=0.1)
    {
        y = evalfunction(expression, x);
        var p = fromCartesian(x, y);
        // TODO font size
        drawcs(p.x, p.y, color, size);
    }
}
```

### Draw init grid.

```
function drawInitGrids()
{
    for (i = 1; i < xmax; i++)
    {
        if (i!=0)
            drawGrids(i, i);
    }
    for (i = -1; i > xmin; i--)
    {
        if (i!=0)
            drawGrids(i, i);
    }
}
```



```

    }
    for (i = 1; i < ymax; i++)
    {
        if (i!=0)
            drawGrids(i, i);
    }
    for (i = -1; i > ymin; i--)
    {
        if (i!=0)
            drawGrids(i, i);
    }
}

```

This method creates the main GUI of the graph canvas. It draws the x-axis and y-axis. It also draws the grids on the Cartesian graph to make it look more professional. Users can also configure the width of the grids, and look and feel of the dashed lines of the grids.

From Cartesian

```

function fromCartesian(x, y) {
    var p = {x:0, y:0};

    var topx = x - xmin;
    var botx = xmax - xmin;
    p.x = topx * canvaswidth / botx;

    var topy = ymax - y;
    var boty = ymax - ymin;
    p.y = topy * canvasheight / boty;

    return p;
}

```

}

This is the function that converts from Cartesian coordinates to canvas positions. For example, if x ranges from -5 to 5 and y ranges from -5 to 5 then Cartesian coordinates (0, 0) will be translated to (5, 5) in canvas coordinates, and the canvas coordinates (0, 0) will represent (-5, 5) in Cartesian coordinates because lower point should have low y-value and higher point should have high y-value and LHS point should have low x-value and RHS point should have high x-value.

This function will also scale the graph, so the length of the line on x-axis will not have the same scale as the length of the line on y-axis if the range of x-value and the range of y-value are not the same. For example, if x-max minus x-min is 10 and y-max minus y-min 20 then the same length on y-axis will represent twice as long as the length on x-axis.

## Animation

```
$("#animate").click(function() {  
  
    // clearplots();  
  
    var x = xmin;  
    var interval = setInterval(function() {  
  
        y = evalfunction($("#functionexpression").val(), x);  
  
        // TODO animate step  
        var p = fromCartesian(x, y);  
        draws(p.x, p.y, 5);  
  
        x+=0.1;
```

```
        if (x > xmax)
            clearInterval(interval);

    }, 10);

    addFunctionToList($("#functionexpression").val());
});
```

This part of code is responsible for producing the animation effect of drawing. When user clicks the draw button, this anonymous function will be called to draw from x-min value to x-max value.