Rocco Vulpis

April 25, 2024

Plotting Functions using Java and Excel

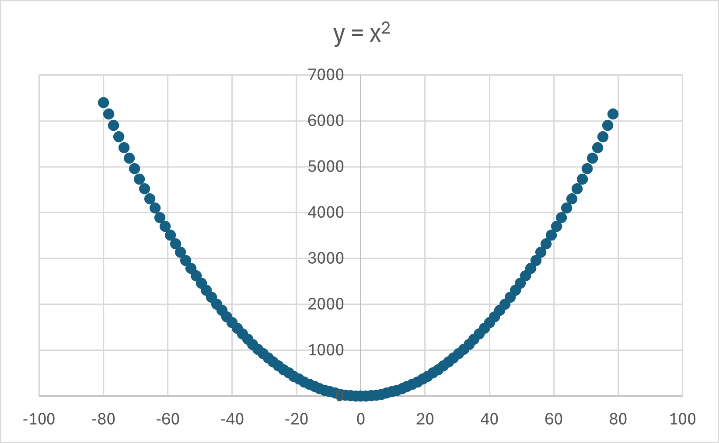
In the plotting portion of this project, a mathematical function was created and adjusted by user specified parameters. Its data points were then exported to a CSV file. From there, a graph of the data was created within Microsoft Excel displaying the results of the plotted data points. The graph of the data resembles the graph of the function based on the parameters input by the user. In this case, the function that was used was the equation of a parabola expressed in quadratic form which is *y = ax2 + bx + c.*

Each value of *a*, *b*, and *c* plays a specific role in manipulating the graph of the quadratic function. The coefficient *a* determines the width and direction of the parabola. If positive, *a* will causes the parabola to open upwards, while a negative *a* will cause it to open downwards. The coefficient *b* affects the symmetry of the parabola. It shifts the position of the vertex along the x-axis, shifting it left or right depending on the value and sign of *b*. Lastly, the constant *c* is responsible for the vertical shift of the parabola. The larger the value, the higher the graph will shift upwards.

The first step is to create the function from the ExcelPlotterTester class. This is done by creating an ExcelPlotter object and calling and passing parameters to its **run** method. Inside the **run** method in the ExcelPlotter class, it calls the **createParabolaFunction** method and passes in the parameters. Its job is to calculate the y values of the quadratic function for a set of x values within a specified range and interval. The necessary parameters are *a* (the coefficient of *x2*), *b* (the coefficient of *x*), *c* (a constant), *xRangeBeginning* (the starting value of the x range), *xRangeEnd* (the ending value of the x range), and *numberOfPoints* (the number of points to calculate in the specified range).

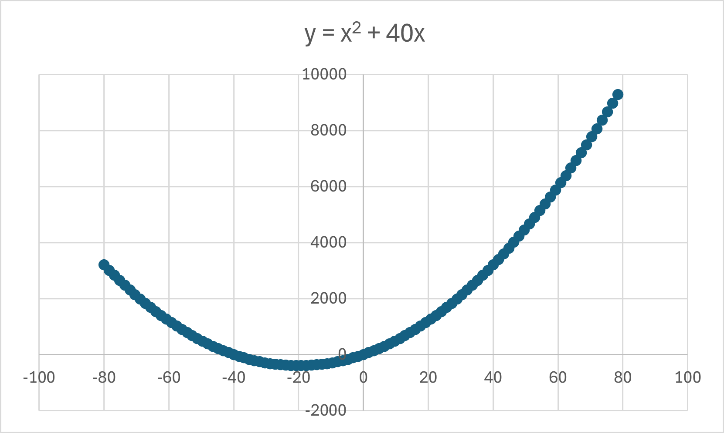
After x and y values are calculated and stored in their respective arrays, they will then be passed to the **writeToFile** method. This method takes in two parameters, *xValues* (the array of x values) and *yValues* (the array of y values). It uses the BufferedWriter class to write both the x and y values to a CSV file named “data.csv”. When the program runs, this file is generated and stored in the default workspace directory. Using excel to open the CSV, the list of x values and y values that were written are stored in columns A and B. Using Excel’s built in charts, a scatter plot can be graphed from the given data. The graph of the data should resemble the graph of the function just as it would on a graphing calculator.

The parameters of the **createParabolaFunction** method allow for modifications to the graph of the quadratic function. In this example, the function being plotted is simply y =x2, so the parameters are *a* = 1, *b* = 0, and *c* = 0, with an x-range beginning at -80 and ending at 80, and a total of 100 points plotted. Below is a screenshot displaying the graph using these specified parameters, created in Excel.

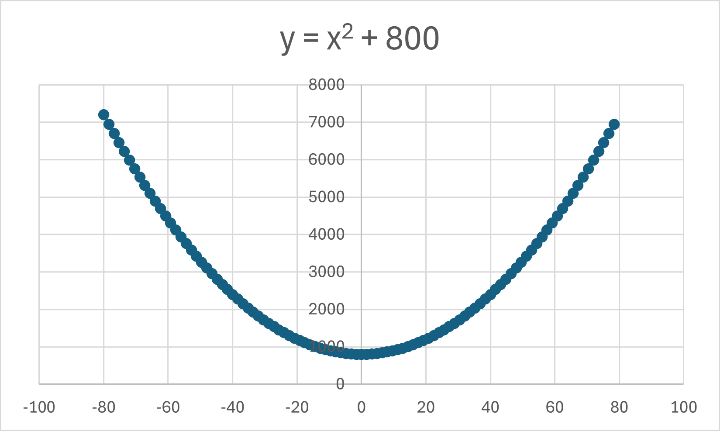


y = x2 where x is bounded between -80 and 80 with 100 points plotted.

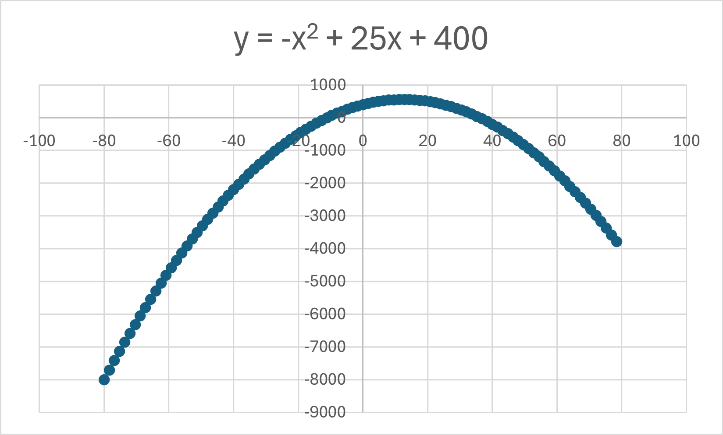
Some more examples of graphs using different parameters.



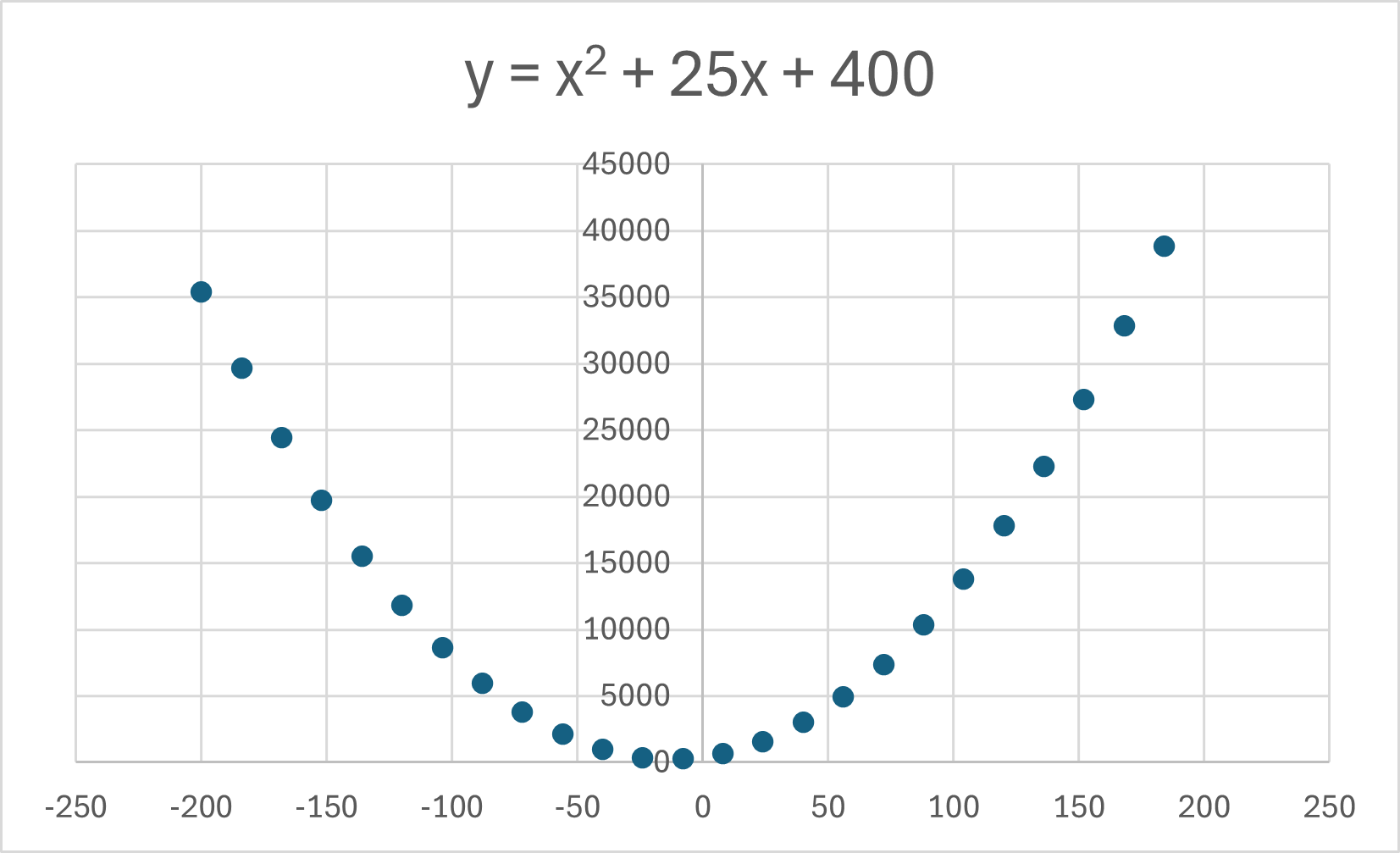
y = x2 + 35x where x is bounded between -80 and 80 with 100 plotted points.



y = x2 + 800 where x is bounded between -80 and 80 with 100 plotted points.



y = -x2 + 25x + 400 where x is bounded between -80 and 80 with 100 plotted points.



y = -x2 + 25x + 400 where x is bounded between -200 and 200 with 25 plotted points.