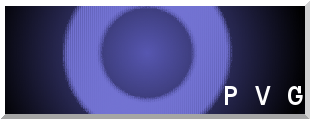
PlasmaGraph Poster Contents

# Logos





Polytechnic University of Puerto Rico

Electrical & Computer Engineering and Computer Science Department

Computer Science Senior Project

Spring 2014

# Project Information

PlasmaGraph

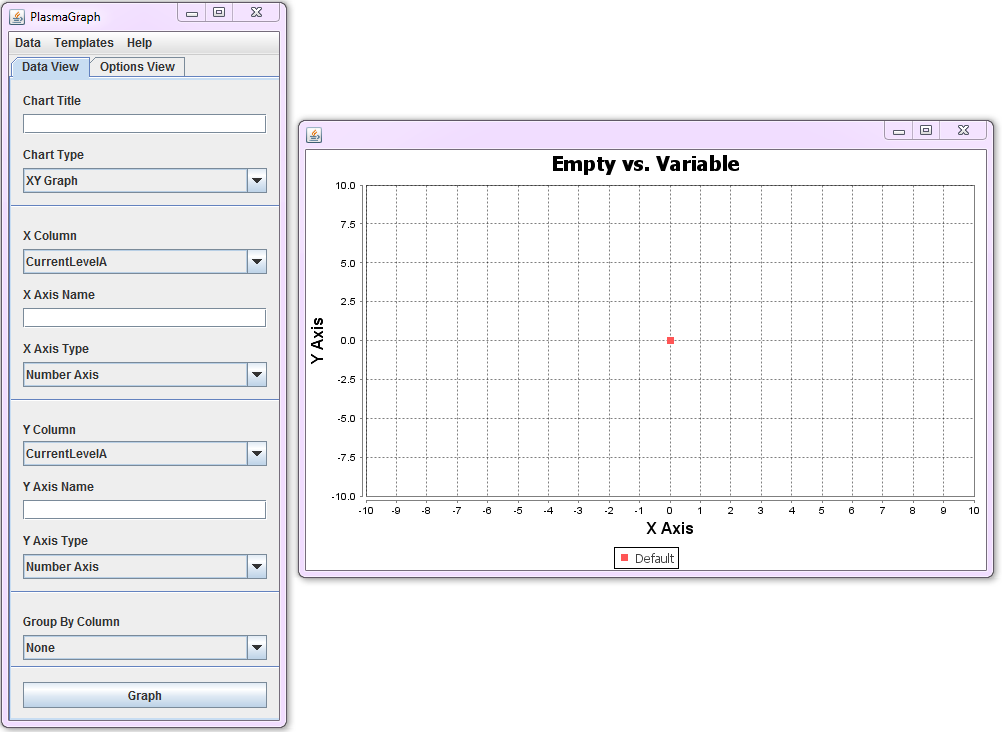
Created by Plasma Visualization Group

Client: Dr. Angel Gonzalez-Lizardo

# Abstract

The Polytechnic University of Puerto Rico runs a Plasma Laboratory wherein students perform various experiments under the supervision of Dr. Angel Gonzalez-Lizardo. The laboratory’s Mirror and Cusp Plasma Machine produces experiment data, which is then stored and graphed using the data analysis software “Matlab”. However, the students find Matlab’s user interface complex and cumbersome due to the high learning curve of using the software’s programming language to create graphs.

As a result, Dr. Angel Gonzalez-Lizardo requested the Plasma Visualization Group to develop a specialized graphing tool to substitute the usage of Matlab. This graphing tool must be easier to learn how to use and must provide some of the customization functionality that Matlab currently provides.

Figure 1: Program Screenshot

# Feature Requirements

The data-graphing program PlasmaGraph can perform the following actions:

* Read Matlab Level-5 data files generated by the Laboratory’s Mirror and Cusp Plasma Machine.
* Verify that the Matlab data files are correctly formed and have valid data based on the following rules:
  + The file must be a Matlab-produced Level-5 binary file.
  + The file must contain two or more variables.
  + Each variable must contain a vector of length N and width 1.
  + All variables in the file must contain the same number of values.
  + All variables must be of types “double” or “cell”.
    - Double variables are used to store numerical values.
    - Cell variables are used to store text values in the form of Character arrays.
  + The file cannot be larger than 36,500 KB.
* Allow the user to modify the follow graph components via the user interface’s options:
  + Title
  + Axes names and scales
  + Graphable columns
  + Appearance of interpolation lines or curves
  + Removal of possible outliers
* Produce graphs based on the data files and the user-selected options.
* Save the graph as a PNG-encoded image file.
* Provide tools with which to add functionality in the future.

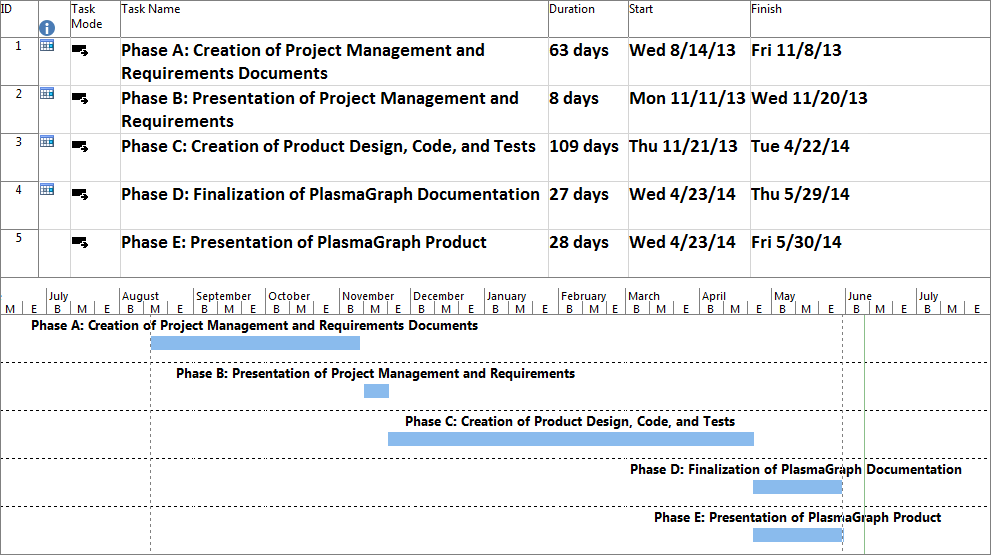
# Constraints

* The project’s duration will not exceed 10 months.
* The project will be completed with only two (2) members.
* The project will be made with the Java programming language.
* All computers using the PlasmaGraph program must have Java version 7 or later installed.

# Timeline

The team’s progress in the development of the PlasmaGraph product is depicted in the following figure:

Figure 2: Project Schedule, Compressed



# Budget

The following table provides a compressed estimate of the costs associated with the development of the PlasmaGraph program.

Table 1: PlasmaGraph Budget, Compressed Estimate

|  |  |
| --- | --- |
| **PlasmaGraph Compressed Budget, Estimate** | |
| **Section** | **Cost** |
| Hardware | $1,955.99 |
| Software | $1,989.95 |
| Personnel and Locations | $74,000.00 |

# Graphical User Interface

The Graphical User Interface (GUI), as shown in Figure 1, provides the tools to manipulate graphs into the desired shape. The interface is separated into two sections:

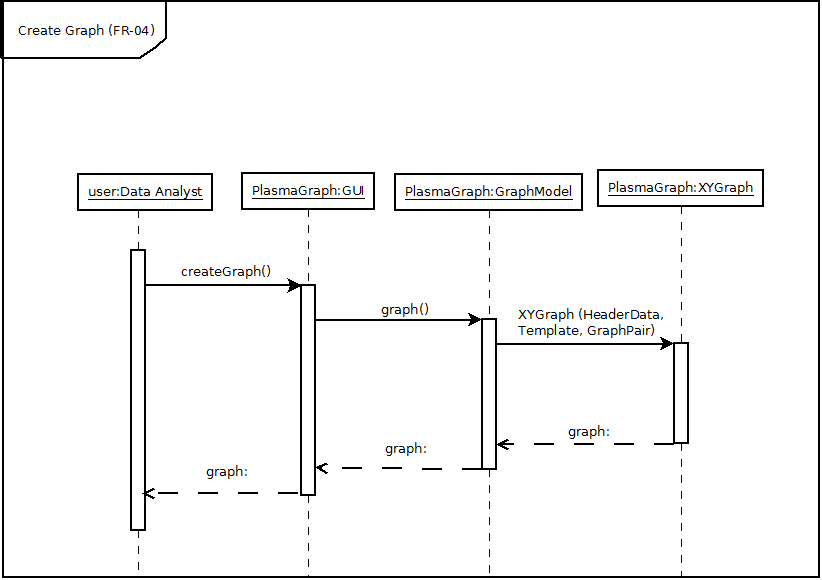
* Data Settings: This window handles the options related to the visual components of the graph’s data, such as what columns will be used to graph, column axis names, graph title, and grouping column.
* Tool Settings: This window handles the options related to the tools available in PlasmaGraph, such as the Interpolation or Outlier Search capabilities.

Furthermore, the program allows the user to view the data contained in the file via the “View Data” option on the Menu Bar.

# Graphing

PlasmaGraph utilizes the tools provided by the Java-based “JFreeChart” library in order to create graphs. The program obtains the translated data file and provide a graph representative of what the user selected.

Figure 3: Sequence Diagram for the "Create Graph" Requirement



This process creates graphs as shown on Figures 4, 5, and 6, seen below:

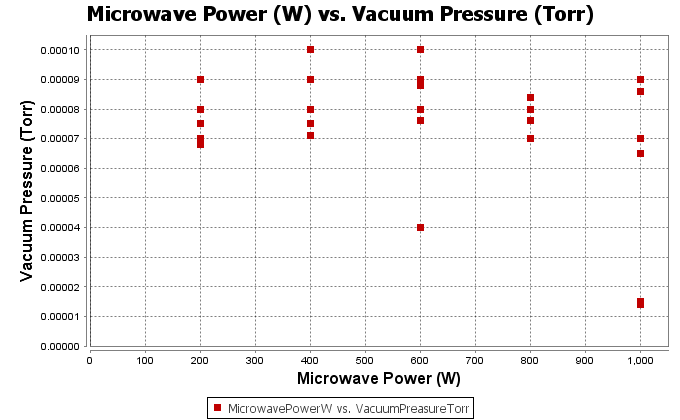
Figure 4: Graph of Microwave Power versus Vacuum Pressure

Figure 5: Graph of Plasma Potential versus Temperature and its Linear Interpolation

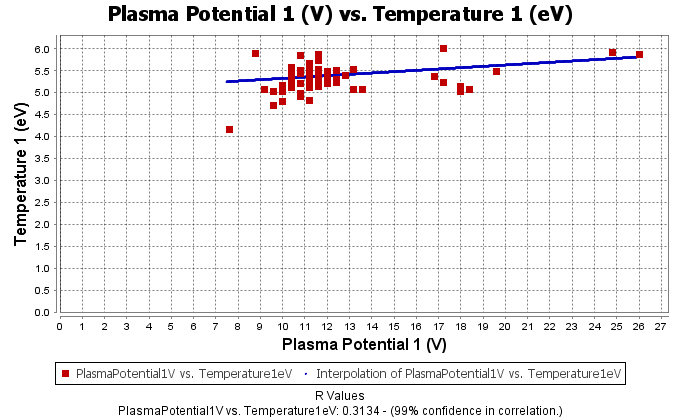
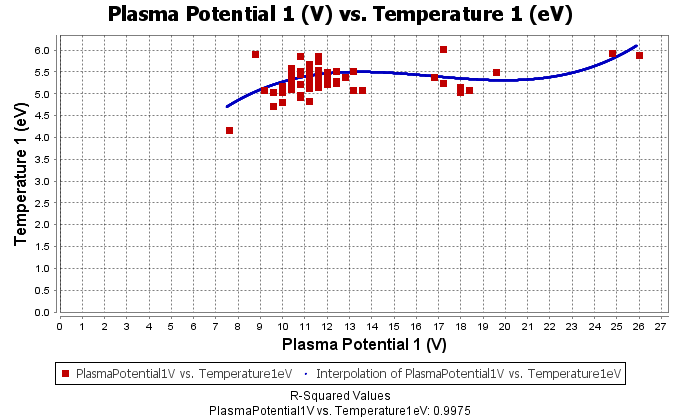


Figure 6: Graph of Plasma Potential versus Temperature and its Cubic Interpolation



# Conclusion

After a comprehensive set of tests, PlasmaGraph proved to be a powerful graphing tool. Its testing encompasses all the requirements, as well as a variety of internal functions that PlasmaGraph uses in order to support its requirements. Tests were divided between those performed by a computer and those performed by a Plasma Laboratory volunteer. Tests performed by a computer were designed using white box methods, whereas tests performed by a volunteer were designed using black box methods. Furthermore, the laboratory test volunteers were able to quickly understand how to properly use the program with little guidance and were positive that PlasmaGraph would be of great benefit to them. Therefore, we are confident this program will be a useful addition to the Plasma Laboratory’s data analysis tools.

# Future Feature Requirements

PlasmaGraph should include the following additional features:

* The ability to create Bar Charts.
* The ability to create three-dimensional graphs.
* The addition of new Interpolation and Outlier-Searching methods.

# Design Team

Table 2: Plasma Visualization Group Design Team Information

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| “Plasma Visualization Group” Design Team Information | | | |
| Name | ID | Program | Project Role |
| Gerardo A. Navas Morales | 69615 | Computer Science | Project Manager |
| Daniel E. Quintini Greco | 73749 | Computer Science | Design Manager |