PlasmaGraph

Created by Plasma Visualization Group
Client: Dr. Angel Gonzalez-Lizardo

Polytechnic University of Puerto Rico
Electrical & Computer Engineering and Computer Science Department
CS 4800 - Computer Science Senior Project
Spring 2014

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 provide some of the customization functionality that Matlab currently provides.

FEATURE REQUIREMENTS

The data-graphing program PlasmaGraph must 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
  - Axes and grouping 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.

CONSTRAINTS

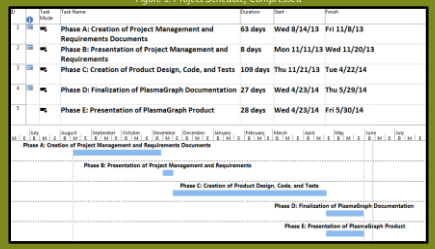
These are the constraints that limit PlasmaGraph:

- 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 1: Project Schedule, Compressed



	Task Mode	Task Name	Duration	Start	Finish
1		Phase A: Creation of Project Management and Requirements Documents	63 days	Wed 8/14/13	Fri 11/8/13
2		Phase B: Presentation of Project Management and Requirements	8 days	Mon 11/11/13	Wed 11/20/13
3		Phase C: Creation of Product Design, Code, and Tests	109 days	Thu 11/21/13	Tue 4/22/14
4		Phase D: Finalization of PlasmaGraph Documentation	27 days	Wed 4/23/14	Thu 5/29/14
5		Phase E: Presentation of PlasmaGraph Product	28 days	Wed 4/23/14	Fri 5/30/14

BUDGET

The following table provides a compressed estimate of the costs associated with the development of the PlasmaGraph program if the client would have contracted a software development company:

Table 1: PlasmaGraph Budget, Compressed Estimate

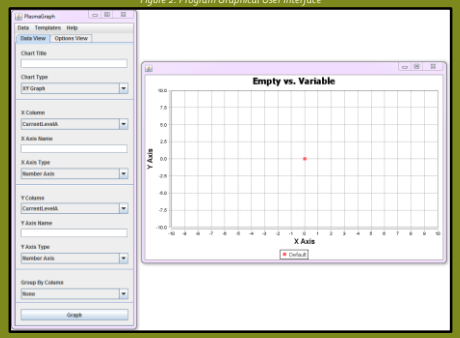
PlasmaGraph Compressed Budget, Estimate	
Component	**Cost**
Hardware	$1,955.99
Software	$1,989.95
Personnel and Locations	$74,000.00

GRAPHICAL USER INTERFACE

The Graphical User Interface (GUI) provides the tools to manipulate a graph 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.

Figure 2: Program Graphical User Interface



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.

The following figures illustrate the sequence diagram that describes the graph-creation process (see Figure 3) and sample graphs generated by PlasmaGraph (see Figure 4, Figure 5, and Figure 6)

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

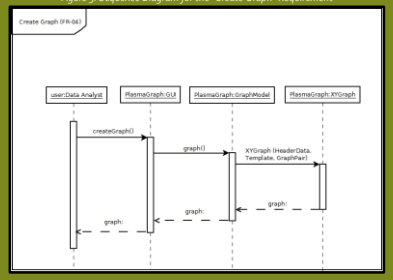


Figure 4: Graph of Microwave Power versus Vacuum Pressure

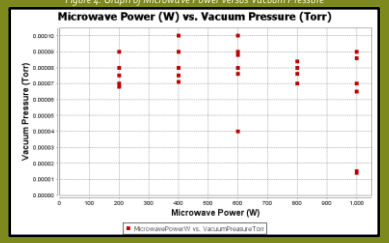


Figure 1: Graph of Plasma Potential versus Temperature and its Linear Interpolation

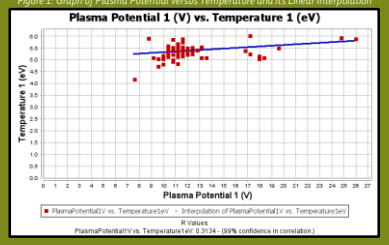
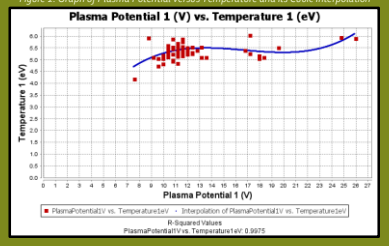


Figure 2: Graph of Plasma Potential versus Temperature and its Cubic Interpolation



TESTING AND CONCLUSION

Multiple types of tests were used to validate PlasmaGraph's proper functionality and acceptance by the PUPR Plasma Laboratory. Functionality tests were divided in two categories: tests performed by a computer and those performed by a Plasma Laboratory volunteer. Tests performed by a computer were designed using the white box method, whereas tests performed by a volunteer were designed using the black box method. Computer-performed tests were run whenever changes were made to the code, and were used to verify the program's stability; whenever one would not pass, the resulting error would be corrected. Volunteer-performed tests were performed during the finalization of the PlasmaGraph documentation, and all passed successfully on the first attempt. Functionality tests were scheduled and led by the Design Manager, Daniel E. Quintini Greco.

Program acceptance was gauged by the volunteers before performing the functionality tests by allowing them to use the program freely. Volunteers commented that PlasmaGraph was a definite improvement over Matlab in ease of use and provided a useful subset of the data-analysis tools that Matlab provides. Program acceptance tests were scheduled and led by the Design Manager, Daniel E. Quintini Greco.

These tests are detailed in the project's Software Test Documentation (STD), and allow the Plasma Visualization Group to be confident with the claim that PlasmaGraph will be a valuable tool for the PUPR Plasma Laboratory. In other words, all the requirements and constraints were met.

FUTURE WORK

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.
- The ability to select the colors used for a graph's data points.
- The ability to automatically select each axis' correct scale.

DESIGN TEAM

The following people were involved in the development of PlasmaGraph:

Table 2: 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