Gnuplot Helper – a new Utility for gnuplot Graph Plotting Software

Dwaipayan Deb

Ramanuj Gupta Memorial College, Ambikapatty, Silchar-788004, Assam, India

Center of Advanced Studies and Innovation Lab, Silchar, India

Abstract

The *gnuplot* is a well known software for graph plotting in the field of science and engineering. It is a command based software originally built for Linux environment, however now it's available for other operating systems too. The present paper discusses overview of a utility software namely Gnuplot Helper to be used with gnuplot. This utility can generate and run gnuplot scripts in Graphical User Interface mode very easily and can save the output as an image file which can be used in research papers or other purposes. This paper also discusses how to use the software and how it works.

Keywords: gnuplot, software, graph plotting

1.Introduction:

Showing data in graphical format is essential in any branch of science and engineering. There are various software for making graphs from data in a computer. Professional grade software like MATLAB (Yadav et al. 2014), SigmaPlot (Mingbin et al. 2010, Monk 2002), OriginPro (Yuan et al. 2015) etc. are paid softwares not free to use forever. Among freewares – gnuplot (Peng et al. 2014), LabPlot (www.labplot.kde.org), Matplotlib (Hunter 2007) are important.

Gnuplot is much popular among researchers/scientists and the students of physics as it is an open source freeware with great functionalities and can compete with any paid software. It was originally a Linux software and therefore it largely remains a command based interface, although it's now available on many platforms including Windows. Users of Windows operating system however do not

often prefer a command based software because they're habituated to Graphical User Interface (GUI) based systems and finds it difficult to memorize the command sets and their recommended order of use. A small spelling mismatch or change in order of command sequence sometimes makes it impossible to get an output, and hence users have to frequently consult help documentation which is a time consuming task. Therefore it would be nice to have a GUI based utility for Windows users which can generate and run gnuplot command scripts automatically. This paper describes in detail - the planning, designing, use and limitations of such a Windows utility named 'Gnuplot Helper' developed by me in VB.Net. This is open source software, free to use under GNU GPL v3.0 license which is available on www.sourceforge.net for free downloading. The aim of writing this paper is to elaborate the use and programming behind this software so that people, who may be interested to know the working of the program, can easily understand the codes. Apart from this – general users who are unaware of programming, will also be benefited by knowing the working of this utility.

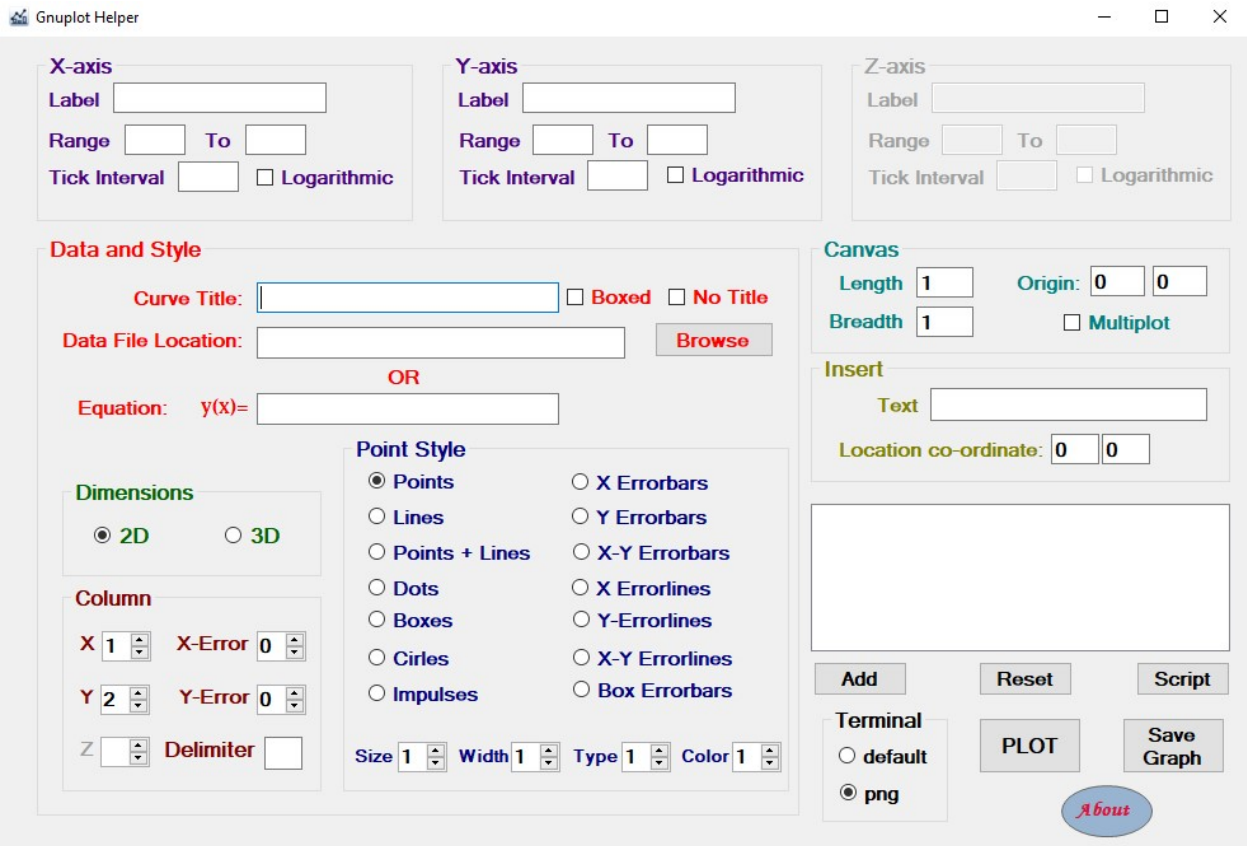

Fig.1 Graphical User Interface of Gnuplot Helper

2.Design and Features: The basic idea behind this software is that, it creates a script file (a text file with .plt extension) where commands needed to give an output graph as desired by the user is written by the software. When the user seeks for output as a graph, this script file is automatically executed with gnuplot. Fig.1 shows the GUI of Gnuplot Helper. As it can be seen that there are various fields for giving necessary inputs required by the graph, utmost care has been taken so that a wrong input is not entered by the user leading to a null output. For example, original *gnuplot* does not allow the end points of X-axis or Y-axis range (i.e. minimum or maximum value) to be set zero when logarithmic scale is chosen. So, if by mistake a user enters 0 in the range box, then the program not only makes it impossible to choose logarithmic scale option with this entry, but also warns the user to correct it. Therefore command-line error possibility – which occurs frequently in *gnuplot*, is almost not possible in this interface.

2.1Plotting of Graphs: To use this software one must first install gnuplot v4.6 or later (earlier versions are not tested). If gnuplot is not installed, the Gnuplot Helper will detect this before starting and ask user to install it before using. After gnuplot is installed, a person who is completely unaware with gnuplot script, may use this software with great experience. Users can start plotting graphs either from a data file or an equation in the respective fields. Both 2D and 3D graphs can be drawn with this utility. When plotting from data file, one must specify the column numbers of abscissa(s) and ordinate in the respective number selectors in the Column group at the bottom-left corner of the interface. Columns of error-bars can also be specified here for 2D graph (gnuplot does not have error-bar option for 3D plotting). If the column separator in the data file is other than 'space' (' ') such as commas as in csv files, one can specify it in the 'delimiter' box. When plotting an equation, care must be taken to write the equation in the gnuplot format. That is, for 2D equation the independent variable must be x(lower case) and for 3D the variables are x and y in lower case. Addition, subtraction, multiplication and division operators can be denoted by +, -, * and / as usual, but powers(exponents) are given by **(e.g. x^2 can be written as x**2). Basic trigonometric functions sine, cosine and tangent can be denoted by sin(x), cos(x) and tan(x). Other trigonometric functions are not allowed (as these can be obtained from basic functions). Natural logarithms ($\log_e$ or ln) can be written as log(x) whereas Napierian logs ($\log_{10}$) are not allowed (however they can be easily implemented by dividing Natural log by

2.3026). To assist user to write equation in the correct form, a tooltip message has been added which becomes visible for a few seconds when mouse is hovered over the equation box. After inserting a data file or an equation, one can select the point style and its size, width, type and color. Finally, after giving a curve title one can press on the PLOT button situated at the bottom-right to plot the graph. If range boxes are kept blank, gnuplot automatically selects a suitable range and plots the graph. For desired range of values to be plotted, one can fill-up these boxes and may give appropriate labels for the respective axes.

2.2 Drawing Multiple Graphs: In many cases we need to draw more than one data sets or equations in one graph plane. There are three ways of doing this in gnuplot which becomes easier to implement with Gnuplot Helper. These are discussed below-

(1) Discrete curves: This method is an inherent property of gnuplot and directly not related to Gnuplot Helper. But it is being discussed here since this is an appropriate place to discuss this topic and many people who are comfortable with gnuplot commands may not know this method. This is applicable only for data sets joined by lines with same color or type. Suppose user wants to plot two curves of same color and type (i.e. dashed, solid etc.) on the same plane. First thing that comes to user's mind is to make two separate data files containing two sets of data and plotting this on gnuplot with same line type. This is Ok, but there is an easier technique. Let the two sets of data be in a single file but having one or two blank lines between them. Now if the data is plotted from that single file joined by lines, the output will be two discrete curves having a disconnection at the location where the bank lines in the data file exists. Fig.2 illustrates the picture.

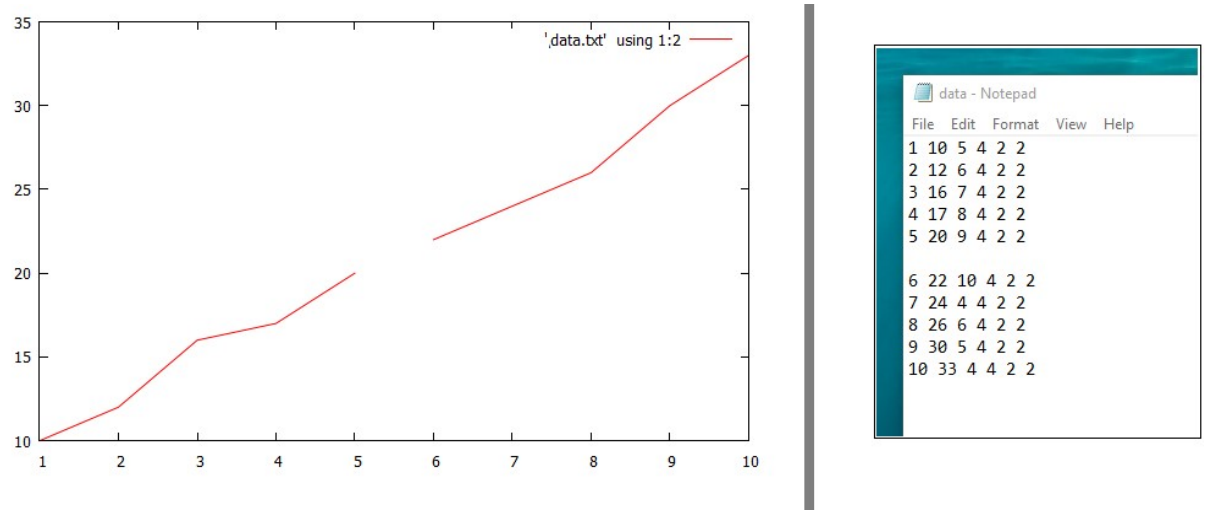

Fig.2 Plotting of discrete data on same plane.

To implement this in Gnuplot Helper user need not do anything extra, but follow the plotting of curve from data file by using lines as done normally, only the inclusion of blank line in the data file as depicted above is necessary.

(*2) Many data sets on the same space:* Whether in two dimensions or in three dimensions, often we need to plot many data sets in a single space for a comparative study. This can be done very easily with Gnuplot Helper just by a button click. After user plots a data set satisfactorily by following steps of Sec2.1, user can click on the 'Add' button at the right panel to add this plot in the plotting list right above this button. Then user may choose another equation or data sets to plot a second graph and again add to the plotting list. In this manner user may add as many data sets as he/she wants and after finish, by clicking on the PLOT button gets the result. If any point the user feels that the graph is not satisfactory, by clicking on 'Reset' button the list can be cleared and a fresh plot can be initiated.

(3) Multiplot method: The gnuplot has a method to plot many separate graphs on the same plot area. That is, one can plot different data sets in different boxes lying adjacent to each other. The script/command set of this technique is a bit difficult to remember but Gnuplot Helper makes it very easy to implement. There is a panel 'Canvas' at the right side of GUI. First of all the Checkbox namely Multiplot must be checked. Then user has to set the size and origin of the box within the canvas area as a fraction of maximum. The maximum canvas size is 1 for both length and breadth for a perfect fit with the plotting area (and therefore for a normal graph it is

set to 1). If length and breadth is chosen to be 0.5, a half sized graph will be plotted within a box. The origin determines the position of the box on the plotting area. For a normal graph origin is set to (0,0). If origin is chosen to be (0.5,0.5), the graph will start from half way of the length and breadth of the plotting area (Fig.3 illustrates this case). After choosing proper canvas, and then filling other required fields, one has to add the plot to the list before getting output by clicking on PLOT button. It is to be remembered that Multiplot is only applicable to 2D plot.

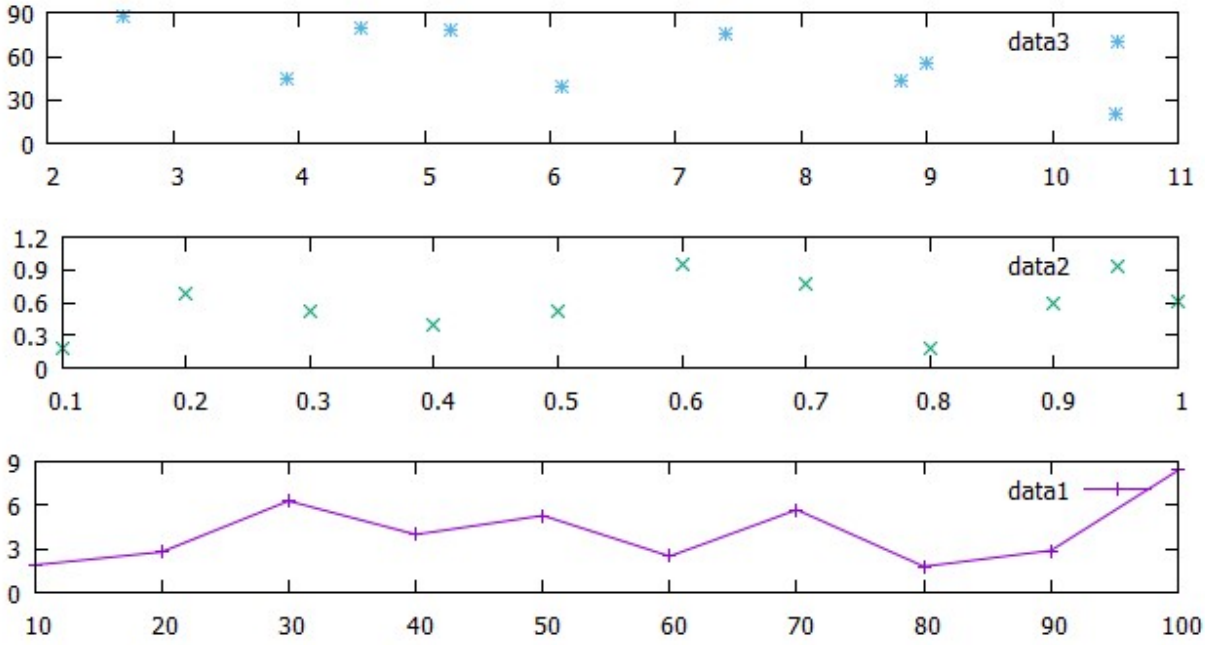

Fig.3 Example of multiplot

2.3 Saving Files: After a graph is obtained with satisfaction, one might be interested to save the file either in image format, or in script format. There is a button 'Script' below the plot list which opens the script file on Notepad when clicked. One may use the 'Save As' method of Notepad to save the file somewhere else for external use. To save the graph directly in the image format, one can click on 'Save Graph' button and select one of the available image formats to save the graph as an image file. Presently supported image formats are – ps,eps,png and jpeg.

2.4 Choosing Terminal: Terminal is that interface of gnuplot where the image is being displayed when someone runs a script on gnuplot. The default terminal is wxt terminal, but often this terminal does not work properly on double click of a plt file and the graph is closed immediately after it is produced. Generally this happens due to some sort of registry error which may take place due to improper installation of gnuplot software. In that case, on Gnuplot Helper one may change the terminal type to png in order to get a stable output.

3. Technical Background: This utility exploits an inherent feature of gnuplot, to run script files with .plt extension. That means, if a file with .plt extension containing gnuplot commands written in text mode, is opened by double clicking on that file, it directly gives an output in the form of a graph in the default terminal of gnuplot, provided there is no error in the script file. If there is an error in the script, then nothing will be shown as output. Gnuplot Helper makes a script file with .plt extension according to the information entered by the user, and then automatically executes the file from the program itself. In doing this, it was important to take care about the correctness of the script and prevent the user to enter such options which would lead to error in the script and hence no output.

The software has been developed in VB.net. It is a form application and the project contains four forms. Form-1 is the main form where all the necessary codes are written, whereas other forms are for information to the user. A complete description of the program would make this paper too large and therefore a brief description of various useful *methods* in Form1 is given below

- *Form1_Load* section is executed during the starting of every VB.net program. Here, it starts the initial Welcome Note (Form2) and checks for existence of various files to be used by this program. If the files are not found (which will occur during the first run), then they are created. This stage also checks whether the user has already installed gnuplot.exe or not and prompts user to install it first before starting to use this utility. Various tooltips are also defined here which assists the user by showing information in a balloon when mouse is hovered over a control.
- There are four *functions* - FileWriter, pngSaver, multiplotSaver and tempoutSaver. Basically all of them do the same job by writing gnuplot command texts in a text file, but for different purposes. All of them take

two arguments – first one is the text to be written in String format and second one is a Boolean true/false where true is for append the text to what already exists in the text file and false is for deleting all existing text before writing. These functions are used to write gnuplot scripts in the files with plt extension.

- TextBox7 and TextBox8 are used for input data file location and entering equation respectively. Since only any one of them is acceptable at a time, TextChanged event of one clears everything within the other box. This also avoids any confusion that may arise in the user's mind intending to plot a data set and an equation on a single plot. TextBox8_TextChanged event also makes GroupBox5 (containing the column inputs for data file option) disabled when TextBox8 is not empty so that user is not confused.
- RadioButtons 11 and 12 respectively designates the 2D and 3D options at the left side of GUI. By default BadioButton11 is in Checked status. All the plot options are not available for each of them and hence non-relevant options are made disabled on CheckedChanged event. This reduces error possibility during runtime. Apart from that, during this event the necessary command for plotting 2D graph ("plot " or "p") or 3D graph ("splot" or "sp") in gnuplot is determined and stored in a String variable for writing in the script file in a later stage.
- CheckBoxes 1, 2 and 3 are used to determine whether the scales to be set liner or logarithmic along the X, Y and Z direction. In gnuplot there are certain restrictions on using log scale. In case of a linear scale if one wishes to leave the minimum and maximum fields empty that will not lead to any error, gnuplot automatically chooses a suitable range by itself. But one should clearly define the minimum and maximum of the range and none of them can be zero, otherwise error will occur. Therefore when the Logscale option is chosen on the Gnuplot Helper by checking the checkboxes, CheckedChanged event is called and here the codes in this event prevents user to leave the range boxes empty or zero value. Thus any kind of error in script due to this is avoided.
- Logscale.vb is another form which is displayed if Logscale CheckBoxes are successfully checked (as described in the previous step). This form allows user to select the base of logscale to be *e* or 10.

- The range values, ticks, length, breadth and origin of canvas, location co-ordinates of label texts can only be numbers, any non numeric text entered in these fields by mistake would lead to script error and hence no output. Therefore at the TextChanged or Validated event of the relevant TextBoxes, codes have been written to prevent user to enter any non-numeric text (only decimal point is allowed) by showing error message and also deleting the text.
- Button2 is the PLOT button in the GUI of Gnuplot Helper. At the Button2_Click event, first it checks whether those fields which are mandatory to give some values (such as canvas size, origin etc.) are empty or not. If those are found empty, then prompts user to fill these fields first before plotting, otherwise this may lead to script error. After that, it checks whether Multiplot checkbox (CheckBox4) is checked or not. If it is found unchecked, first it calls a function FinalPlot which writes all the necessary gnuplot commands in a temporary text file based on the information entered by the user and then it executes this script by using command Process.Start("<Script File Path>"). If CheckBox4 is found checked then it checks whether ListBox1 (which displays the added plots) is empty or not. If it is found empty, it asks user to add the plot before plotting a multiplot graph and exits the method. If this stage is passed, the script file is executed like the previous case. Actually, for multiplot, when a plot is added to the list, it is this stage when the script is written in a text file. When button2 is clicked this file is only executed.
- Button4 saves the output graph in various image format. On click, it displays a dialog to choose the location and file name of the output file. The two functions – tempoutSaver and multiplotSaver are used to generate a temporary .plt file with commands of gnuplot for saving an output file, along with all the other commands previously set by the user for plotting a graph. Now, when these files are executed by using Process.Start("<Script File Path>") , the image files are automatically generated by the gnuplot itself.

4. Limitations: The gnuplot in itself is very rich in functionality and everything was not possible to include in this tiny utility Gnuplot Helper. Although it has been

designed to meet all commonly used features, there are still some important limitations and leaves scope of improvement in a future update. Some of these are - (i) presently single column data cannot be plotted against the serial number of data which is possible in gnuplot through command line. (ii) while plotting data from a dynamic data file (i.e. a file whose column values are constantly changing with time), real time refreshing of the output curve after a regular period is yet to be implemented. (iii) fitting of a function with arbitrary constants with a set of data is possible in gnuplot which is missing here. (iv) algebraic operations between values of different columns of a data file while plotting data, without affecting the data file can be done in gnuplot, but this is yet to be implemented in Gnuplot Helper. (v) presently, while using multiplot function, one has to add the individual graphs to the 'plotting list' after carefully choosing all the parameters without viewing the output. If one wishes to change any of the parameters (e.g. tick intervals or point type) after viewing the graph, this can be done by editing the script file generated by Gnuplot Helper. Users who are completely unaware of gnuplot commands may feel it difficult and therefore needs to be addressed in a future update so that anyone can do this without editing the script file. (vi) In case of errors, which may occur due to incorrect input information (e.g. column number entered does not exist in the data file, or suppose, an invalid arithmetic operator is used in the equation box), there will be null output and no error notification can be seen. This is also needed to be addressed.

5. Licensing: Licensing is an important step for a developer to distribute his/her program among people. By choosing appropriate licensing, the developer chooses the extent of freedom allowed to the user - to use, distribute and/or modify the program. This software has been licensed for GNU General Public Licence Version 3 (GPL v3). According to GPL, the user can use, distribute and modify the software without restriction. However, there are some guidelines for publishing a newer version of the software. Details of GPL can be obtained at https://www.gnu.org/licenses/gpl-3.0.en.html.

6. Software Publishing and Distribution: This software has been published on www.sourceforge.net for free download. The homepage of this project on SourceForge is https://sourceforge.net/projects/gnuplot-helper. There is also a main page https://www.respt.in/p/gnuplot-helper.html for detailed description and

feature updates. The source codes are available for download from SourceForge page.

7. Conclusion: This software is useful to save time and removes the hassle of remembering commands of 'gnuplot' and hence makes it easier to draw graphs of both 2D and 3D format in a user friendly manner via a graphical user interface. However, still there are scopes of improvement which has been described in Sec.3. As the software is open source under GPL v3, and source codes are available to anyone from sourceforge.net, readers who are interested in software development may be benefited from this work. Also, one may contribute to this project for improvements and thus help mass users in using all the resources of gnuplot in an easier way.

References:

Binod Prasad Yadav, CS Patil, RR Karhe, PH Patil, An automatic recognition of fake Indian paper currency note using MATLAB, Int. J. Eng. Sci. Innov. Technol 3, 560-566, 2014

Zheng Mingbin, Ja Zhenbin, Wu Xiren, Mei Jun, The Application of SigmaPlot in Drawing Phase Diagram, Guangdong Chemical Industry 1, 2010

Scott Monks, SigmaPlot 8.0, Biotech Software & Internet Report: The Computer Software Journal for Scientists 3 (5-6), 141-145, 2002

Haodong Yuan, Bin Deng, Guozhi Wang, Jian Ke, Mao Li, The application of seamless integration technology invoking OriginPro in LabVIEW on overflow valve test bench, Modern Machinery, 03, 2015

Qi Peng, Wang Diansheng, Yan Xianghong, Zhou Lixia, Chen Dongmeng, Application of Gnuplot in data processing in modern physics experiment, Experimental Technology and Management, 10, 2014

John D Hunter, Matplotlib: A 2D graphics environment, IEEE Annals of the History of Computing 9 (03), 90-95, 2007