UNIVERSITY OF SURREY

Creating Graphs in Gnuplot

Rosh Sellahewa

CONTENTS

1 Basic Plots **2**

1.1 Opening Gnuplot in the Terminal 2

1.2 Plotting From a File 3

1.3 Adding Lines and Multiple Datasets 5

1.4 Manipulating Labels and Legends 8

2 3D Plots **10**

2.1 Basic 3D Plotting 10

2.2 Adding Colours and Contours 11

3 Outputting To EPS **12**

3.1 Gnuplot Scripts 12

3.2 Writing to an Output 12

3.3 Displaying Multiple Graphs 13

3.4 Inset Graphs 13

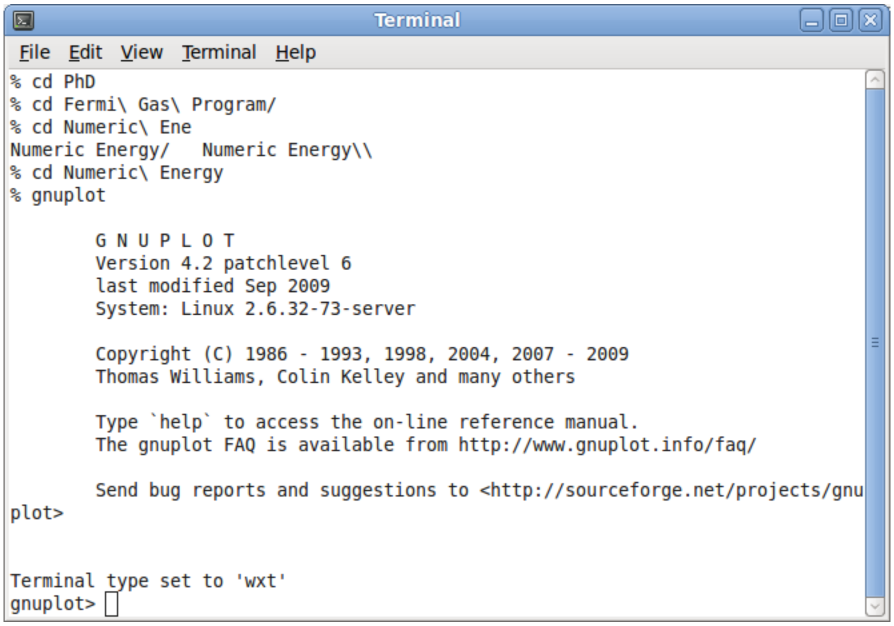
CHAPTER 1

BASIC PLOTS

1.1 OPENING GNUPLOT IN THE TERMINAL

Firstly we need to access Gnuplot. This is done by opening a terminal either through menus, or by using CTRL + SHIFT + T. Now navigate to the folder in which the data you want to plot is stored. Either do this through “cd Foldername" or open the terminal in the folder by right clicking in the file browser and selecting “Open in Terminal". Once the terminal is open in the right place simply type“gnuplot" and hit enter[1]. There is no need to add any extra information at this page and because Gnuplot is running through the terminal an & is not needed.

[1] If you get the error“ Gnuplot not found" and you have admin rights (using your own computer) then type “sudo apt-get install gnuplot" and Gnuplot will install.

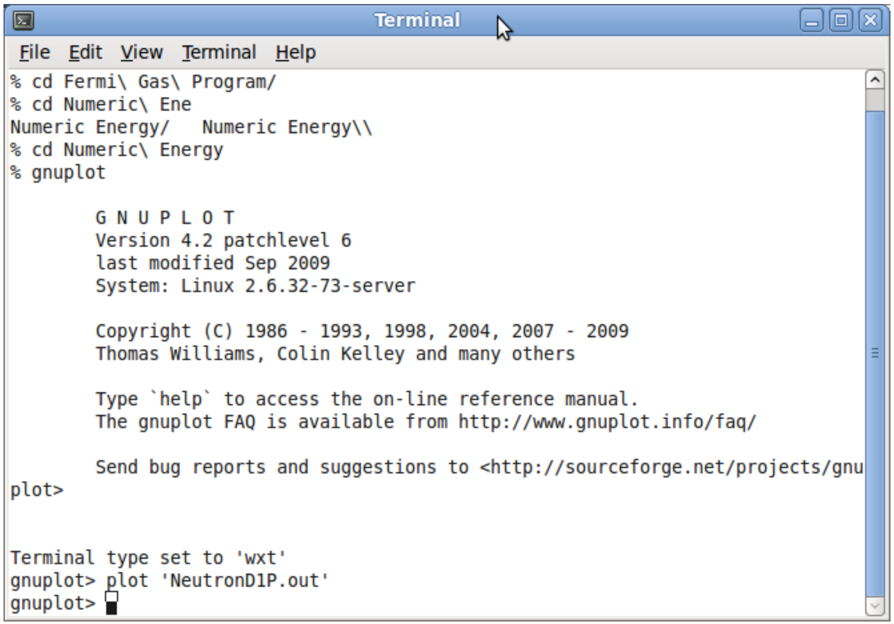


```
% cd PhD
% cd Fermi\ Gas\ Program/
% cd Numeric\ Ene
Numeric Energy/   Numeric Energy\\
% cd Numeric\ Energy
% gnuplot

        G N U P L O T
        Version 4.2 patchlevel 6
        last modified Sep 2009
        System: Linux 2.6.32-73-server

        Copyright (C) 1986 - 1993, 1998, 2004, 2007 - 2009
        Thomas Williams, Colin Kelley and many others

        Type `help` to access the on-line reference manual.
        The gnuplot FAQ is available from http://www.gnuplot.info/faq/

        Send bug reports and suggestions to <http://sourceforge.net/projects/gnu
plot>


Terminal type set to 'wxt'
gnuplot> 
```

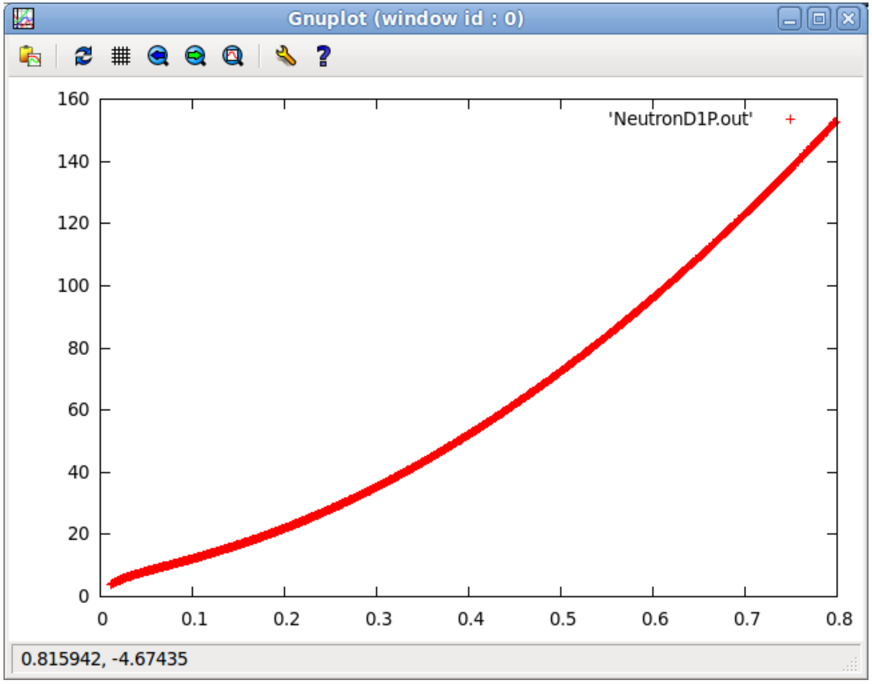
Now you should be in the Gnuplot environment which, like the Terminal, is command line based. From here we are ready to plot to a file.

1.2 Plotting From a File

Basic plotting from a file is straight forward. Start by simply typing the word “plot" and then the name of the file you’d like to plot. In Fig. we have used a file called “’NeutronD1P.out’". Note the ’ either side of the file name. To plot now, simply press enter.



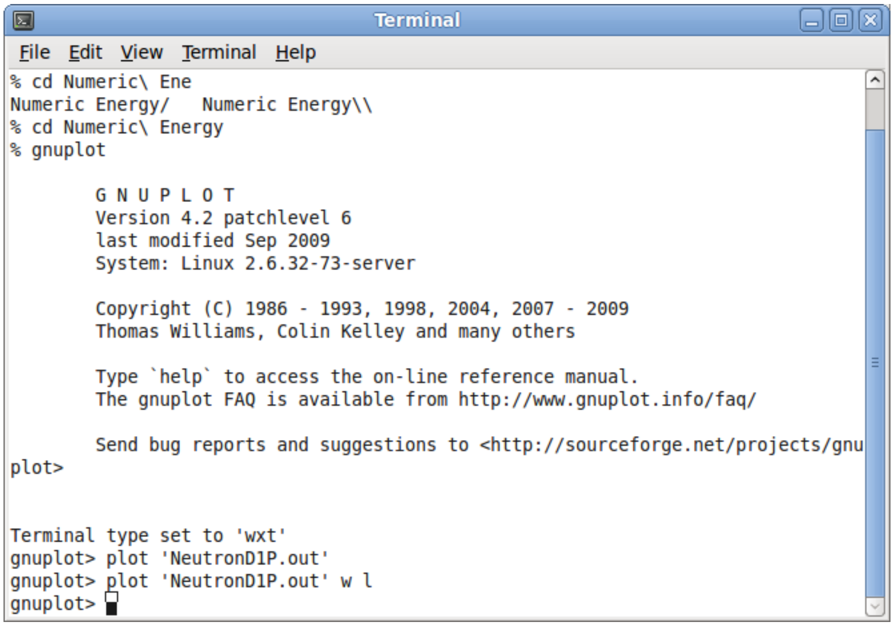
```
% cd Fermi\ Gas\ Program/
% cd Numeric\ Ene
Numeric Energy/   Numeric Energy\\
% cd Numeric\ Energy
% gnuplot

        G N U P L O T
        Version 4.2 patchlevel 6
        last modified Sep 2009
        System: Linux 2.6.32-73-server

        Copyright (C) 1986 - 1993, 1998, 2004, 2007 - 2009
        Thomas Williams, Colin Kelley and many others

        Type `help` to access the on-line reference manual.
        The gnuplot FAQ is available from http://www.gnuplot.info/faq/

        Send bug reports and suggestions to <http://sourceforge.net/projects/gnu
plot>


Terminal type set to 'wxt'
gnuplot> plot 'NeutronD1P.out'
gnuplot> 
```



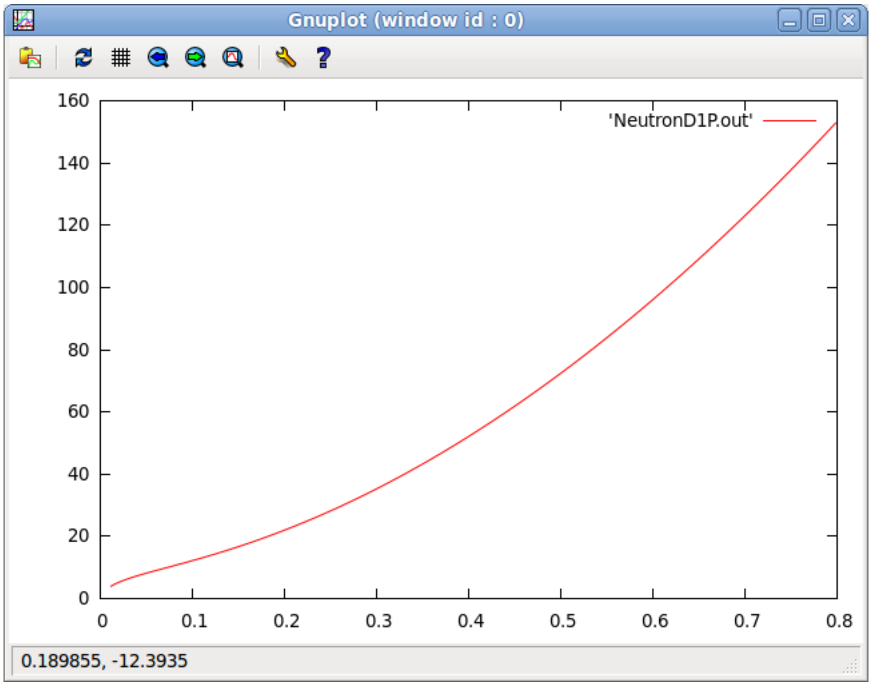
The plot is rudimentary, with only the points being plotted. Sometimes this is enough, but often we require a little more informative plot. Also note that if your data file has several columns then by default x is the first column and y is the second column.

1.3 Adding Lines and Multiple Datasets

To plot the graph with lines rather than just points is very simple. We type the same command as before but this time we add “with lines" at the end. In fact we can just say “w l", no need to type the whole word. This now gives us a more informative plot. The lines are simply linking the points so it is not correct to treat this as a line of best fit.



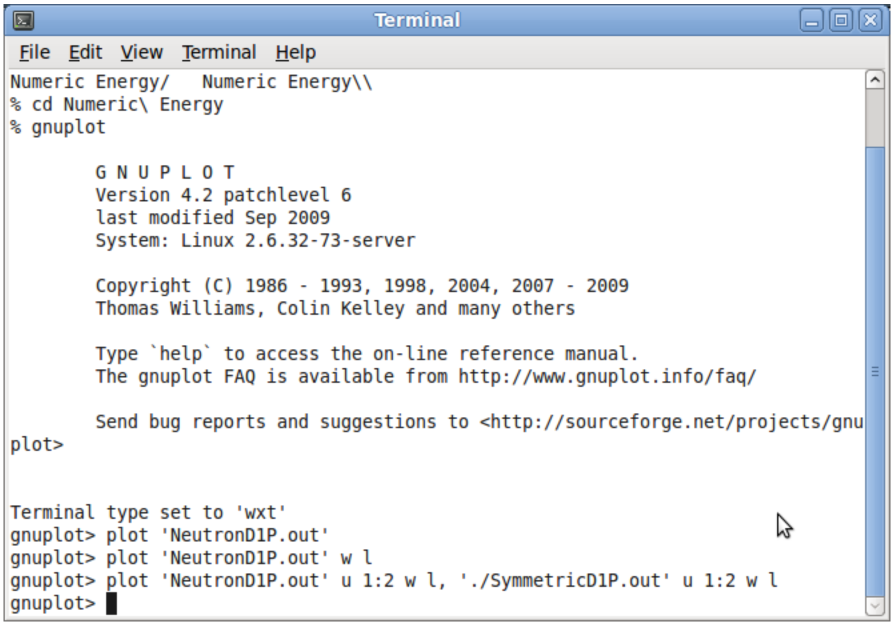
```
% cd Numeric\ Ene
Numeric Energy/   Numeric Energy\\
% cd Numeric\ Energy
% gnuplot

        G N U P L O T
        Version 4.2 patchlevel 6
        last modified Sep 2009
        System: Linux 2.6.32-73-server

        Copyright (C) 1986 - 1993, 1998, 2004, 2007 - 2009
        Thomas Williams, Colin Kelley and many others

        Type `help` to access the on-line reference manual.
        The gnuplot FAQ is available from http://www.gnuplot.info/faq/

        Send bug reports and suggestions to <http://sourceforge.net/projects/gnu
plot>


Terminal type set to 'wxt'
gnuplot> plot 'NeutronD1P.out'
gnuplot> plot 'NeutronD1P.out' w l
gnuplot> 
```



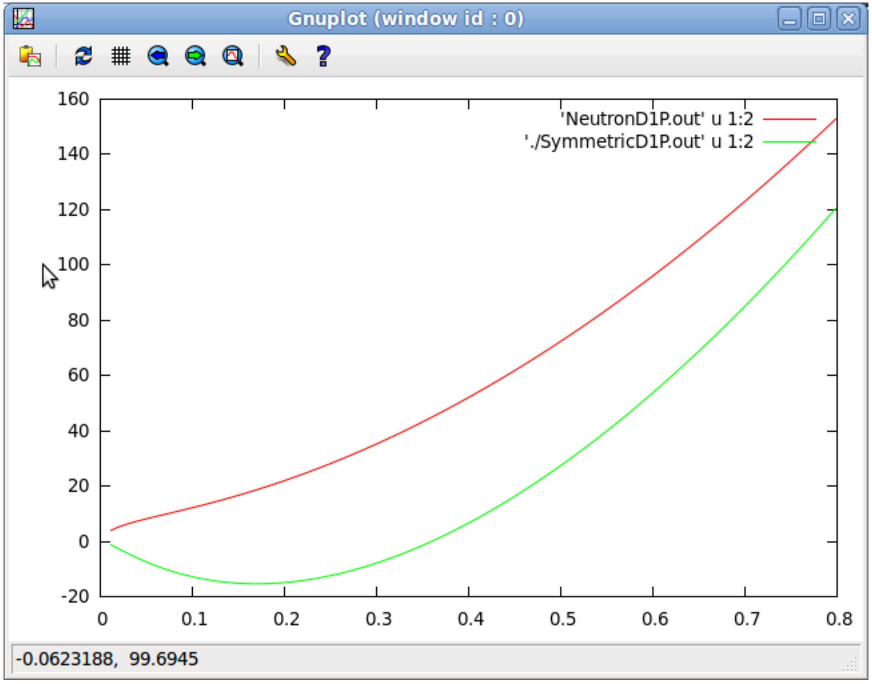
If we want to plot from columns other than the first and second then we use the term “using" or simply “u" followed by column1:column2. It is not important which order you specify "using" or "with lines" commands. The full command now would be “plot FILENAME u 2:3 w l" to plot columns 2 and 3 of file FILENAME, with lines.

The next useful step is to be able to plot or more two different files on the same graph. This is especially useful for comparing two different models. Gnuplot is built for exactly this kind of thing and so the syntax is very straight forward: we finish the first plot command, then insert a , and then add the second command[2]. So to plot two files the command becomes “plot ’NeutronD1P.out’ u 1:2 w l, ’SymmetricD1P.out’ u 1:2 w l".

[2] a plot command is not needed at this point



```
Numeric Energy/   Numeric Energy\\
% cd Numeric\ Energy
% gnuplot

        G N U P L O T
        Version 4.2 patchlevel 6
        last modified Sep 2009
        System: Linux 2.6.32-73-server

        Copyright (C) 1986 - 1993, 1998, 2004, 2007 - 2009
        Thomas Williams, Colin Kelley and many others

        Type `help` to access the on-line reference manual.
        The gnuplot FAQ is available from http://www.gnuplot.info/faq/

        Send bug reports and suggestions to <http://sourceforge.net/projects/gnu
plot>


Terminal type set to 'wxt'
gnuplot> plot 'NeutronD1P.out'
gnuplot> plot 'NeutronD1P.out' w l
gnuplot> plot 'NeutronD1P.out' u 1:2 w l, './SymmetricD1P.out' u 1:2 w l
gnuplot> 
```



1.4 Manipulating Labels and Legends

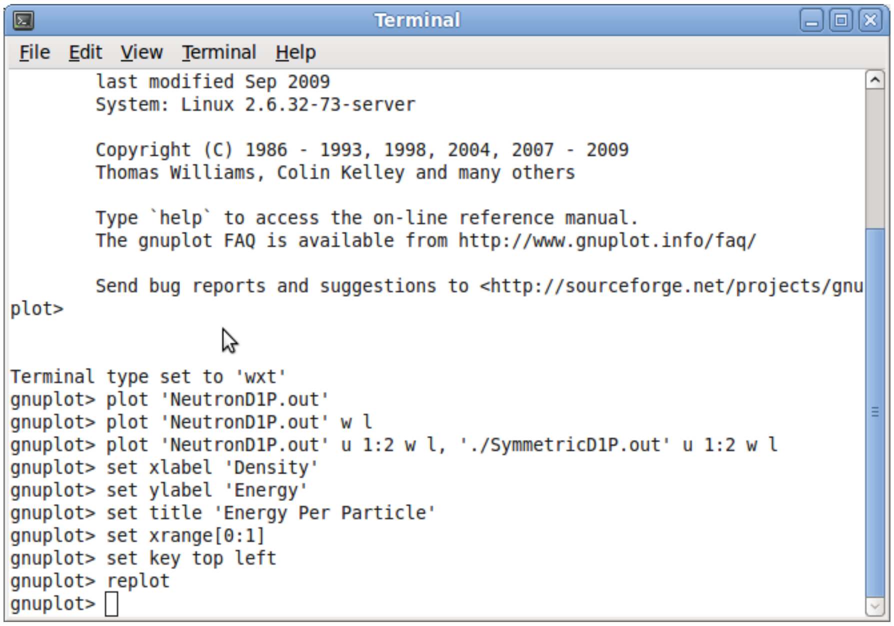
Now that we have our graph plotted it would be nice to make it a little more informative. To do this we can add some labels to the graph. The commands are very simple:
Setting x Label to “$Density$" → set xlabel ’Density’
Setting y Label to “$Energy$" → set ylabel ’Energy’
Setting the Title to “$EnergyPerParticle$" → set title ’Energy Per Particle’

Finally, with the example files I’ve used the legend is in an inconvenient place, so lets expand the range in x and move the legend to the left. Setting x or y ranges is very straight forward:

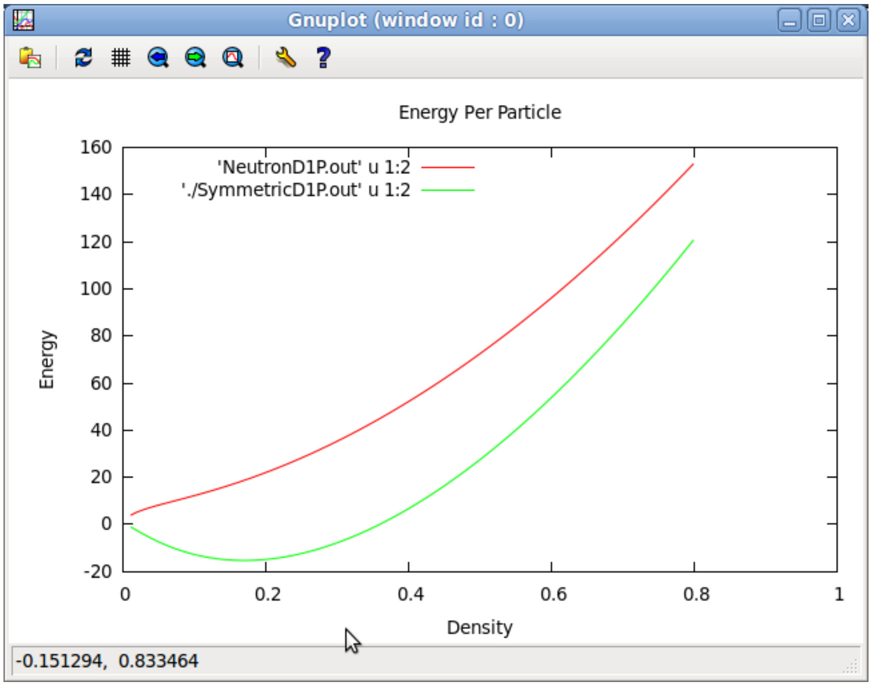
“set xrange[0:1]"

and moving the legend is also quite intuitive:

“set key top left"



```
        last modified Sep 2009
        System: Linux 2.6.32-73-server

        Copyright (C) 1986 - 1993, 1998, 2004, 2007 - 2009
        Thomas Williams, Colin Kelley and many others

        Type `help` to access the on-line reference manual.
        The gnuplot FAQ is available from http://www.gnuplot.info/faq/

        Send bug reports and suggestions to <http://sourceforge.net/projects/gnu
plot>


Terminal type set to 'wxt'
gnuplot> plot 'NeutronD1P.out'
gnuplot> plot 'NeutronD1P.out' w l
gnuplot> plot 'NeutronD1P.out' u 1:2 w l, './SymmetricD1P.out' u 1:2 w l
gnuplot> set xlabel 'Density'
gnuplot> set ylabel 'Energy'
gnuplot> set title 'Energy Per Particle'
gnuplot> set xrange[0:1]
gnuplot> set key top left
gnuplot> replot
gnuplot> 
```

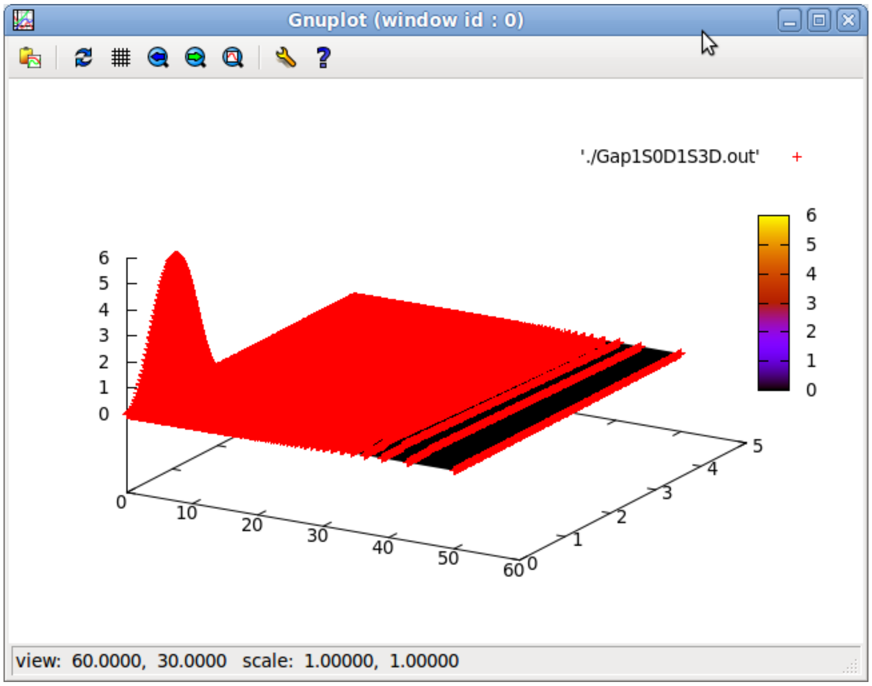

Gnuplot (window id : 0)
Energy Per Particle
'NeutronD1P.out' u 1:2
'./SymmetricD1P.out' u 1:2
Energy
160
140
120
100
80
60
40
20
0
-20
0
0.2
0.4
0.6
0.8
1
Density
-0.151294, 0.833464

CHAPTER 2

3D PLOTS

2.1 BASIC 3D PLOTTING

Sometimes it is useful to plot a function dependent on two terms. For example, the superfluid gap can depend on two different momenta, k_1 and k_2. Plotting this is almost identical to the 2D case except instead of “plot" we use “splot". We must also define a 3D surface and for that we input the command “set pm3D" and hit enter before splotting[1]. Plotting the file we now get a 3D plot of our values.

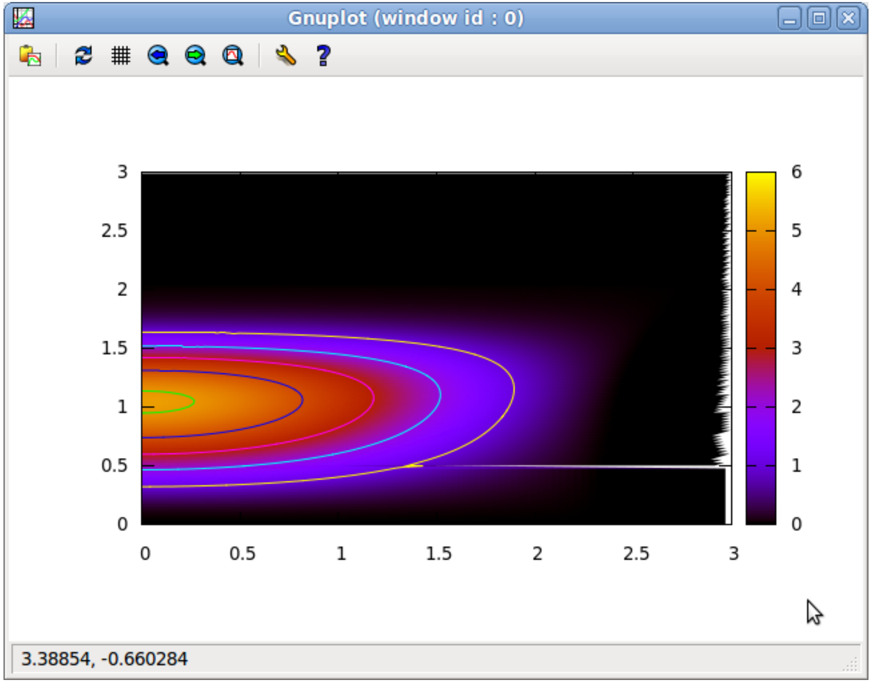


view: 60.0000, 30.0000 scale: 1.00000, 1.00000

[1] When using pm3D make sure that after each set of rows in the data table we insert a blank line

2.2 Adding Colours and Contours

Although the 3D plot is accurate it isn't overly informative in its raw state and so we can manipulate the plot to make it more manageable. First we "unset surface" to get rid of the 3D mesh. "set view map" will give us an aerial view: slightly less fun to move around but generally more informative when writing papers and reports. Finally we can "set contours" to give contour lines, just like on a standard map. These can be very useful. To replot we can actually just type "rep" instead the entire splot command. This also works with normal plots.



CHAPTER 3

OUTPUTTING TO EPS

3.1 GNUPLOT SCRIPTS

Whilst plotting a graph to the screen can be a great way to check your results, ultimately you are likely to want to include these plots in a paper or report. The best way to include these plots is by making Gnuplot send the graph directly into a file. In this example we will focus on eps files.

Firstly, Gnuplot is able to run a list of commands from a “script" file, with the suffix “.gpl". Gnuplot will read in this file and actuate the commands line-by-line as if they had been run in the Terminal. For instance we can write the same things as in the examples above. To run this script do not enter the Gnuplot environment. If you are already in the environment then type “exit" or simply “q" into gnuplot and press enter to return to the Terminal. To run the script we type “gnuplot ’SCRIPTNAME.gpl’ " and press enter. The graph will run on the screen but it does not persist, it flashes only for a second. This is because once the script has plotted the graph it finishes and so exits.

3.2 WRITING TO AN OUTPUT

To write directly to a file you can add some extra lines into the script[1]. To set an output, start by adding the line “set terminal eps color enhanced font ’14’". This will tell Gnuplot to output to an eps file, in colour[2], with the font size of 14pt for labels and titles.

[1] This technique will work if input directly into the terminal but in this case we are focussing on the script

[2] You have to use the American spelling

Now add the line “set output ’FILENAME.eps’". This is the target location for the file. At the end of the script simply write “unset output" and you are ready to run the script.

3.3 Displaying Multiple Graphs

Displaying multiple graphs is slightly more complex but learning how to do it will help you to produce some great looking plots. Once again we are going to be using a script to create these graphs.

In the script, just below setting the output, type “set multiplot". There are mutiple ways to go from here but the method that we will be using is defining the position of the four edges of each graph by their relative position on the board. For this we will use 6 variables: DX, DY, SX, SY, SDX, SDY. These represent as follows:

DX =Distance from the left edge of the eps to the leftmost axis line

DY =Distance from the lower edge of the eps to the bottommost axis line

SX =Width of the graph

SY =Height of the graph

SDX =Horizontal distance between graphs

SDY =Vertical distance between graphs

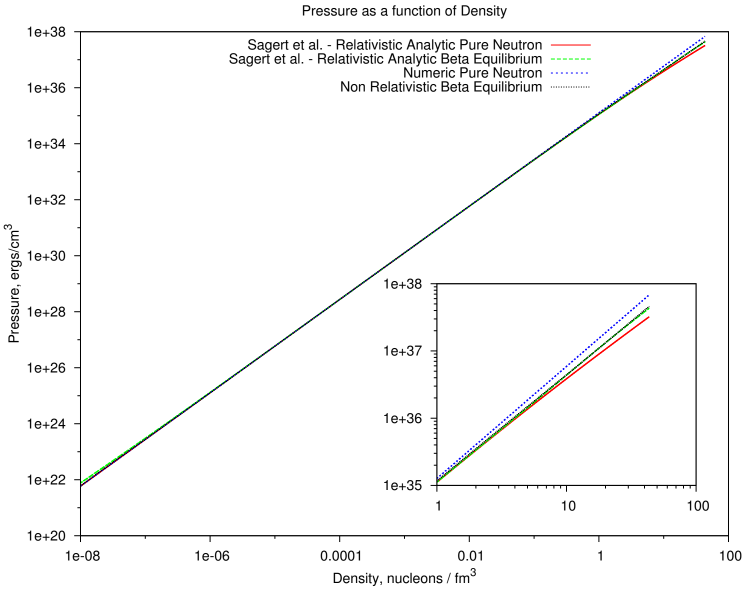
We can now set the axis positions of our first graph using the following:

```
set lmargin at screen DX
set rmargin at screen DX + SX
set tmargin at screen DY + SY
set bmargin at screen DY
```

Then actuate the plot command. Now that this has been plotted we can define a new set of axis positions, for our second graph. Again, we run the plot. In the example we have set the key to be hidden for the second graph. Finally we “unset multiplot".

3.4 Inset Graphs

The ability to define the coordinates of the graph axes allows us to inset graphs within graphs. This is done easily by setting the position of the second graph to be within the area of the first.



The number of graphs is not limited to two. Using this simple program we have now begun making publication grade plots, I encourage you to be ambitious with Gnuplot: There is a way of doing most things, this tutorial has been designed to be introductory rather than exhaustive. Happy plotting!