**Sample gnuplot Scripts**

Here are some sample scripts to do some basic things with gnuplot. For more detailed examples, see the gnuplot user manual at

[*http://www.gnuplot.info/docs\_4.6/gnuplot.pdf*](http://www.gnuplot.info/docs_4.6/gnuplot.pdf)

and the plethora of examples at

<http://www.gnuplotting.org/>

See also

<http://labs.guidolin.net/2010/03/how-to-create-beautiful-gnuplot-graphs.html>

which gives examples and shows how to implement a default configuration file with macro definitions. A useful color scheme designer may be found at

<http://colorschemedesigner.com/>

It can be used to generate pleasing color palettes for gnuplot (or other applications). Here is an example of a color palette generated there:

##### Color Palette by Color Scheme Designer  
##### Palette URL: http://colorschemedesigner.com/#0u41Tw0w0w0w0  
##### Color Space: RGB;   
  
\*\*\* Primary Color:  
  
 var. 1 = #FF7400 = rgb(255,116,0)  
 var. 2 = #BF7130 = rgb(191,113,48)  
 var. 3 = #A64B00 = rgb(166,75,0)  
 var. 4 = #FF9640 = rgb(255,150,64)  
 var. 5 = #FFB273 = rgb(255,178,115)  
  
\*\*\* Secondary Color A:  
  
 var. 1 = #FFAA00 = rgb(255,170,0)  
 var. 2 = #BF8F30 = rgb(191,143,48)  
 var. 3 = #A66F00 = rgb(166,111,0)  
 var. 4 = #FFBF40 = rgb(255,191,64)  
 var. 5 = #FFD073 = rgb(255,208,115)  
  
\*\*\* Secondary Color B:  
  
 var. 1 = #1240AB = rgb(18,64,171)  
 var. 2 = #2A4480 = rgb(42,68,128)  
 var. 3 = #06266F = rgb(6,38,111)  
 var. 4 = #4671D5 = rgb(70,113,213)  
 var. 5 = #6C8CD5 = rgb(108,140,213)  
  
\*\*\* Complementary Color:  
  
 var. 1 = #009999 = rgb(0,153,153)  
 var. 2 = #1D7373 = rgb(29,115,115)  
 var. 3 = #006363 = rgb(0,99,99)  
 var. 4 = #33CCCC = rgb(51,204,204)  
 var. 5 = #5CCCCC = rgb(92,204,204)  
  
##### Generated by Color Scheme Designer (c) Petr Stanicek 2002-2010

The following assumes that gnuplot has been installed and is in your path (see the file *installing\_gnuplot* in this directory for help with installing).

**Execution Instructions**

Save one of the following scripts in a file, say with name *gnuDemo.gnu.* Then start gnuplot (this assumes that it is in your path; if it isn’t, you will have to specify the full path to the gnuplot program):

[guidry@m33new ~]$ gnuplot

G N U P L O T  
 Version 4.6 patchlevel 1 last modified 2012-09-26   
 Build System: Linux x86\_64  
  
 Copyright (C) 1986-1993, 1998, 2004, 2007-2012  
 Thomas Williams, Colin Kelley and many others  
  
 gnuplot home: http://www.gnuplot.info  
 faq, bugs, etc: type "help FAQ"  
 immediate help: type "help" (plot window: hit 'h')  
  
Terminal type set to 'wxt'  
gnuplot>

and issue the command at the gnuplot prompt

gnuplot> load “gnuDemo.gnu”

(*Note:* the quotes are mandatory. The quotes in the above line may not copy correctly if this line is copied from here and pasted into a shell window. If you get an error message, type the quotes literally.) This command should display the plot specified by the script in *gnuDemo.gnu.*

**Example of a Configuration File**

# This configuration file is called *.gnuplot* on Linux. When gnuplot starts, it searches for

# this file first in the current directory and then in the Home directory. If found, it reads

# these macro definitions and settings in as defaults. These can be overridden in the gnuplot

# file by redefining the macros or redoing the settings. Your code can't have a

# reset command, since this will clear these definitions. For more information, see

#

# http://www.gnuplotting.org/tag/macros/

# http://labs.guidolin.net/2010/03/how-to-create-beautiful-gnuplot-graphs.html

#

set macro # Enable macro definition

# Some macro definitions

# Custom colors: hex #RRGGBB in quotes

label\_color = "#867961"

tic\_color = "#383838"

title\_color = "#383838"

myblue\_color = "#5ea2c6

myred\_color = "#bb6255"

mygreen\_color = "#668874"

# Macros defining line widths and pointsizes. Reference with "@Macro"

LW1 = "1"

LW2 = "2"

# Set the default tic colors

set xtics textcolor rgb tic\_color

set ytics textcolor rgb tic\_color

# Set the default point size if points are plotted

set pointsize 0.5 # Size of plotted points

# Line styles

set style line 1 lt rgb myblue\_color lw @LW1 # Define linestyle 1

set style line 2 lt rgb myred\_color lw @LW1 # Define linestyle 2

set style line 3 lt rgb mygreen\_color lw @LW1 # Define linestyle 3

set style line 4 lt rgb "black" lw @LW1 # Define linestyle 4

set style line 5 lt rgb "purple" lw @LW1 # Define linestyle 5

set style line 6 lt rgb "red" lw @LW1 # Define linestyle 6

set style line 7 lt rgb "royalblue" lw @LW1 # Define linestyle 7

set style line 8 lt rgb "goldenrod" lw @LW1 # Define linestyle 8

set style line 9 lt rgb "green" lw @LW1 # Define linestyle 9

set style line 10 lt rgb "orchid" lw @LW1 # Define linestyle 10

set style line 11 lt rgb "gold" lw @LW1 # Define linestyle 11

set style line 12 lt rgb "navy" lw @LW1 # Define linestyle 12

set style line 13 lt rgb "light-red" lw @LW1 # Define linestyle 13

set style line 14 lt rgb "magenta" lw @LW1 # Define linestyle 14

set style line 15 lt rgb "orange-red" lw @LW1 # Define linestyle 15

set style line 16 lt rgb "olive" lw @LW1 # Define linestyle 16

set style line 17 lt rgb "violet" lw @LW1 # Define linestyle 17

set style line 18 lt rgb "gray40" lw @LW1 # Define linestyle 18

set style line 19 lt rgb "yellow4" lw @LW1 # Define linestyle 19

set style line 20 lt rgb "dark-orange" lw @LW1 # Define linestyle 20

# Point styles

set style line 21 lc rgb "#5ea2c6" pt 5 # square

set style line 22 lc rgb "#5ea2c6" pt 7 # circle

set style line 23 lc rgb 'dark-orange' pt 9 # triangle

**Multiple Instances of 2 Functions (Save in File and execute as described above):**

***(In the following code examples, all lines beginning with # are comments. Gnuplot is very finicky about commands all being on one line, so use a \ to continue a line if necessary.)***

set macro # Enable macro definition

# Some macro definitions

# Custom colors: hex #RRGGBB in quotes

label\_color = "#867961"

tic\_color = "#383838"

title\_color = "#383838"

myblue\_color = "#5ea2c6

myred\_color = "#bb6255"

mygreen\_color = "#668874"

# Width and height of postscript figure in inches

width = 8

height = 5

# Line styles

set style line 1 lt rgb myblue\_color lw 1 # Define linestyle 1

set style line 2 lt rgb myred\_color lw 1 # Define linestyle 2

set style line 3 lt rgb mygreen\_color lw 1 # Define linestyle 3

set style line 4 lt rgb "black" lw 1 # Define linestyle 4

set style line 5 lt rgb "purple" lw 1 # Define linestyle 5

set style line 6 lt rgb "red" lw 1 # Define linestyle 6

set style line 7 lt rgb "royalblue" lw 1 # Define linestyle 7

set style line 8 lt rgb "goldenrod" lw 1 # Define linestyle 8

set style line 9 lt rgb "green" lw 1 # Define linestyle 9

set style line 10 lt rgb "orchid" lw 1 # Define linestyle 10

set style line 11 lt rgb "brown" lw 1 # Define linestyle 11

# Point styles

set style line 12 lc rgb "#5ea2c6" pt 5 # square

set style line 13 lc rgb "#5ea2c6" pt 7 # circle

set style line 14 lc rgb 'dark-orange' pt 9 # triangle

#set xtics rotate # Rotates x tic numbers by 90 degrees

#set ytics rotate # Rotates y tic numbers by 90 degrees

# Set tic labeling with color

set xtics textcolor rgb tic\_color

set ytics textcolor rgb tic\_color

set bmargin 4 # Bottom margin

# Set screen display to same aspect ratio as postscript plot

set size ratio height/width

set title 'Landau Dirac Dispersion with B' textcolor rgb title\_color

set xlabel 'B' textcolor rgb tic\_color

set ylabel 'E' textcolor rgb tic\_color

# Uncomment following to set log or log-log plots

#set logscale x

#set logscale y

set xrange [0:1]

set yrange[-10:10]

set pointsize 1.0 # Size of plotted points

#unset key # Don't show legend

set key out # Put legend outside plot

set key vert # vert for vertical; horiz for horizontal

set key top right # Move legend to upper right

#set timestamp # Date/time

Omega = 1.0

# Define functions to plot

scale = 1.5

plusE(B,n) = scale\*sqrt(n\*B)

minusE(B,n) = -scale\*sqrt(n\*B)

# Make plots

plot[B=0:10] plusE(B,0) ls 1, plusE(B,1) ls 1, plusE(B,2) ls 1, plusE(B,3) ls 1

replot minusE(B,1) ls 1, minusE(B,2) ls 1, minusE(B,3) ls 1

# Plot to postscript file

set out "LandauDiracDispersion.eps" # Output file

set terminal postscript eps size width, height enhanced color solid lw 2 "Arial" 32

replot # Plot to postscript file

# Plot to PNG file

set out "LandauDiracDispersion.png"

# Assume 72 pixels/inch and make bitmap twice as large for display resolution

set terminal pngcairo transparent size 2\*width\*72, 2\*height\*72 lw 2 enhanced font 'Arial,28'

replot

quit

**Sample 2D Plotting Script (Save in File and execute as described above):**

***(In the following code examples, all lines beginning with # are comments. Gnuplot is very finicky about commands all being on one line, so use a \ to continue a line if necessary.)***

# Note: next 2 lines of set key are part of single line

set key inside left top vertical Right noreverse enhanced autotitles box linetype -1 linewidth 1.000

set samples 600, 600 # x and y resolution of plot; larger is better  
  
set style line 1 lt rgb "#5ea2c6" lw 1 # Define linestyle 1; rgb is hex #RRGGBB  
set style line 2 lt rgb "#bb6255" lw 3 # Define linestyle 2; rgb is hex #RRGGBB  
  
# Define a function to plot  
gl0 = 0.05  
a = -1.0  
b = 1.0  
gl(x) = gl0 + a\*x\*\*2 + b\*x\*\*4  
gl2(x) = -a\*x\*\*2  
  
set xlabel 'Order Parameter'   
set ylabel 'Value'  
set xrange [-1.2:1.2]  
set yrange [-0.25:0.3]  
  
plot gl(x) ls 1, gl2(x) ls 2 # Plot gl w/linestyle 1 and gl2 w/linestyle 2

**Sample 3D Plotting Script (Save in File):**

set xlabel 'Re x'

set ylabel 'Im x'

set zlabel 'f(x)'

set xrange [-1.5:1.5]

set yrange [-1.5:1.5]

set zrange [-6.0:-0.0]

set isosamples 300 # Number surface mesh lines; larger -> larger .ps

#set isosamples 500 # Production quality mesh

set hidden3d # Remove lines that would be hidden if surface opaque

set key outside # Move the legend outside the plot

#set palette model RGB # or CMY (RGB default); 'help palette' other options

# Following define Palette interpolating between user-defined colors

#set palette defined (0 '#ff0000', 1 '#dddddd')

set palette defined (0 'black', 1 'gold')

set pm3d # Shaded color surface

set view 20, 33, 1, 2.0 # Viewing angles and scale factors

#set contour base # Contours on base, surface, or both

#set cntrparam levels 10 # Target number of contour levels

a = -10 # Negative for mexican hat

b = 9

splot a\*(x\*\*2+y\*\*2) + 0.5\*b\*(x\*\*2+y\*\*2)\*\*2 notitle

**Sample Multiplot Script (Save in File)**

# NOTE: the usual conversion to postscript seems only to

# pick up the final figure of the three plotted here, so

# maybe not so useful.

set multiplot layout 1,3 title "Multiplot layout for 3 plots"

#set xtics rotate # Rotates x tic numbers by 90 degrees

set bmargin 2

#

set title "Plot 1"

unset key # Suppresses inset key box for curves

plot sin(x)

#

set title "Plot 2"

unset key

plot cos(x)

#

set title "Plot 3"

unset key

plot x\*\*2

#

unset multiplot

**Sample 2D Vector Field (Save in File)**

# Plot (Irrotational) Vector Field

# See http://gnuplot.10905.n7.nabble.com/Vector-Fields-td3627.html

# See http://lavica.fesb.hr/cgi-bin/info2html?(gnuplot)arrow for arrowheads

set key inside left top vertical Right noreverse enhanced autotitles box linetype -1 linewidth 1.000

set samples 600, 600 # x and y resolution of plot; larger is better

#unset xtics

#unset ytics

set style line 1 lt rgb "#5ea2c6" lw 1 # Define linestyle 1; rgb is hex #RRGGBB

set style line 2 lt rgb "#867961" lw 1 # Define linestyle 2; rgb is hex #RRGGBB

# Plot control parameters

hr = 5 # ranges

ss = 31 # points to evaluate vectors

vl = 0.25 # normalization factor for length

# X-Y range for plot

set xrange [-hr:hr]

set yrange [-hr:hr]

# Integer x-cordinates

set samples ss

# Integer y-cordinates

set isosamples ss

# The vector plot for gnuplot was originally designed for plotting discrete data from a file.

# The special filename "++" can be used to make vector plots from analytical expressions

# without actually having to produce the files.

# If at coordinates (x,y) the vector has x-projection dx and y-projection dy

# the appropriate plot command is of the form

# plot using "++" using 1:2:(expression for dx):(expression for dy) with vectors

# where in the expressions for dx and dy one references the variable x by $1 and the variable

# y by $2. This acts as if it were plotting a four-column data file, with the values in the

# four columns given by the four quantities separated by colons in the above plot command

# Simplest plot (no normalizations)

#plot "++" using 1:2:(-$2/($1\*\*2+$2\*\*2)):($1/($1\*\*2+$2\*\*2)) with vectors

# Plot normalized to unit length and the normalized length scaled by a factor of vl (the backslash

# is a line continuation, since a gnuplot command must be on a single line). The size and nature of

# the arrowhead on the vectors is controlled by "head filled size 0.10,20", which specifies a filled

# arrowhead with a length 0.10 that is proportional to the size of the x-axis (by default; it can

# be overridden) and 20 specifies that the arrow edges make a 20-degree angle with the line

# defining the vector. The length of the vector line is set by the scale factor vl defined above.

# Counterclockwise flow

plot "++" using 1:2:(-$2\*sqrt($1\*\*2+$2\*\*2)/($1\*\*2+$2\*\*2)\*vl):($1\*sqrt($1\*\*2+$2\*\*2)/($1\*\*2+$2\*\*2)\*vl)\

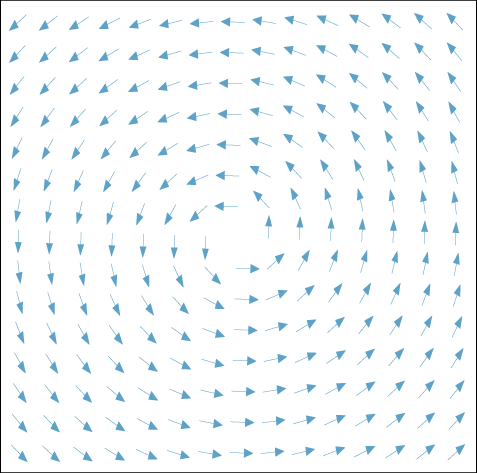
with vectors head filled size 0.10,20 ls 1

# Clockwise flow

#plot "++" using 1:2:($2\*sqrt($1\*\*2+$2\*\*2)/($1\*\*2+$2\*\*2)\*vl):(-$1\*sqrt($1\*\*2+$2\*\*2)/($1\*\*2+$2\*\*2)\*vl)\

#with vectors head filled size 0.10,20 ls 1

Plot made with the above script:



**Plotting from a Data File with Auto PS and PNG Generation**

set macro # Enable macro definition

# Some macro definitions

# Colors: hex #RRGGBB

label\_color = "#867961"

tic\_color = "#383838"

title\_color = "#383838"

myblue\_color = "#5ea2c6

myred\_color = "#bb6255"

mygreen\_color = "#668874"

# Width and height of postscript figure in inches

width = 8

height = 5

# Line styles

set style line 1 lt rgb myblue\_color lw 1 # Define linestyle 1

set style line 2 lt rgb myred\_color lw 1 # Define linestyle 2

set style line 3 lt rgb mygreen\_color lw 1 # Define linestyle 3

# Point styles. See

# <http://www.gnuplotting.org/doc/ps_symbols.pdf>

# for point style options

set style line 1 lc rgb "#5ea2c6" pt 5 # square

set style line 2 lc rgb "#5ea2c6" pt 7 # circle

set style line 3 lc rgb 'dark-orange' pt 9 # triangle

#set xtics rotate # Rotates x tic numbers by 90 degrees

#set ytics rotate # Rotates y tic numbers by 90 degrees

# Set tic labeling with color

set xtics textcolor rgb tic\_color

set ytics textcolor rgb tic\_color

set bmargin 4 # Bottom margin

# Set screen display to same aspect ratio as postscript plot

set size ratio height/width

#set title 'Time for one GPU network integration step' textcolor rgb title\_color

set xlabel 'Number networks' textcolor rgb tic\_color

set ylabel 'Time (s)' textcolor rgb tic\_color

# Uncomment following to set log or log-log plots

#set logscale x

#set logscale y

set xrange [0:100]

set yrange[0.000:0.002]

set pointsize 0.5 # Size of the plotted points

#set key top left # Move legend to upper left

unset key # Don't show legend

#set timestamp # Date/time

# Read data from file and plot to screen

plot "async\_150\_full\_perTimestep.dat" using 1:3 ls 2 title 'Integration time, one step'

#plot "stackedNetworkTiming.dat" using 1:3 ls 2 title 'Integration time, one step'

# Plot to postscript file

set out "stackedNetworkTiming.eps" # Output file

set terminal postscript eps size width, height enhanced color solid lw 2 "Arial" 32

replot # Plot to postscript file

# Plot to PNG file

set out "stackedNetworkTiming.png"

# Assume 72 pixels/inch and make bitmap twice as large for display resolution

set terminal pngcairo transparent size 2\*width\*72, 2\*height\*72 lw 2 enhanced font 'Arial,28'

replot

quit

**Point Styles for Data in gnuplot**

The list of 75 possible point styles may be found here:

<http://www.gnuplotting.org/doc/ps_symbols.pdf>

**Plotting Functions with Auto EPS and PNG Generation**

set macro # Enable macro definition

# Some macro definitions

# Colors: hex #RRGGBB

label\_color = "#867961"

tic\_color = "#383838"

title\_color = "#383838"

myblue\_color = "#5ea2c6

myred\_color = "#bb6255"

mygreen\_color = "#668874"

# Width and height of postscript figure in inches

width = 8

height = 5

# Line styles

set style line 1 lt rgb myblue\_color lw 1 # Define linestyle 1

set style line 2 lt rgb myred\_color lw 1 # Define linestyle 2

set style line 3 lt rgb mygreen\_color lw 1 # Define linestyle 3

# Point styles

set style line 1 lc rgb "#5ea2c6" pt 5 # square

set style line 2 lc rgb "#5ea2c6" pt 7 # circle

set style line 3 lc rgb 'dark-orange' pt 9 # triangle

#set xtics rotate # Rotates x tic numbers by 90 degrees

#set ytics rotate # Rotates y tic numbers by 90 degrees

# Set tic labeling with color

set xtics textcolor rgb tic\_color

set ytics textcolor rgb tic\_color

set bmargin 4 # Bottom margin

# Set screen display to same aspect ratio as postscript plot

set size ratio height/width

#set title 'Time for one GPU network integration step' textcolor rgb title\_color

set xlabel 'Number networks' textcolor rgb tic\_color

set ylabel 'Time (s)' textcolor rgb tic\_color

# Uncomment following to set log or log-log plots

#set logscale x

#set logscale y

set xrange [0:100]

set yrange[0.000:0.002]

set pointsize 0.5 # Size of the plotted points

#set key top left # Move legend to upper left

unset key # Don't show legend

#set timestamp # Date/time

# Define a function to plot

gl0 = 0.05

a = -1.0

b = 1.0

gl(x) = gl0 + a\*x\*\*2 + b\*x\*\*4

gl2(x) = -a\*x\*\*2

set xlabel 'Order Parameter'

set ylabel 'Value'

set xrange [-1.2:1.2]

set yrange [-0.25:0.3]

plot gl(x) ls 1, gl2(x) ls 2 # Plot gl w/linestyle 1 and gl2 w/linestyle 2

# Plot to postscript file

set out "DeltaVsQ.eps" # Output file

set terminal postscript eps size width, height enhanced color solid lw 2 "Arial" 32

replot # Plot to postscript file

# Plot to PNG file

set out "DeltaVsQ.png"

# Assume 72 pixels/inch and make bitmap twice as large for display resolution

set terminal pngcairo transparent size 2\*width\*72, 2\*height\*72 lw 2 enhanced font 'Arial,28'

replot

quit

**Plotting Multiple Functions with Conditional Plotting Ranges on Each Curve**

set macro # Enable macro definition

# Some macro definitions

# Custom colors: hex #RRGGBB in quotes

label\_color = "#867961"

tic\_color = "#383838"

title\_color = "#383838"

myblue\_color = "#5ea2c6

myred\_color = "#bb6255"

mygreen\_color = "#668874"

# Width and height of postscript figure in inches

width = 8

height = 5

# Line styles

set style line 1 lt rgb myblue\_color lw 1 # Define linestyle 1

set style line 2 lt rgb myred\_color lw 1 # Define linestyle 2

set style line 3 lt rgb mygreen\_color lw 1 # Define linestyle 3

set style line 4 lt rgb "black" lw 1 # Define linestyle 4

set style line 5 lt rgb "purple" lw 1 # Define linestyle 5

set style line 6 lt rgb "red" lw 1 # Define linestyle 6

set style line 7 lt rgb "royalblue" lw 1 # Define linestyle 7

set style line 8 lt rgb "goldenrod" lw 1 # Define linestyle 8

set style line 9 lt rgb "green" lw 1 # Define linestyle 9

set style line 10 lt rgb "orchid" lw 1 # Define linestyle 10

set style line 11 lt rgb "brown" lw 1 # Define linestyle 11

# Point styles

set style line 12 lc rgb "#5ea2c6" pt 5 # square

set style line 13 lc rgb "#5ea2c6" pt 7 # circle

set style line 14 lc rgb 'dark-orange' pt 9 # triangle

#set xtics rotate # Rotates x tic numbers by 90 degrees

#set ytics rotate # Rotates y tic numbers by 90 degrees

# Set tic labeling with color

set xtics textcolor rgb tic\_color

set ytics textcolor rgb tic\_color

set bmargin 4 # Bottom margin

# Set screen display to same aspect ratio as postscript plot

set size ratio height/width

set title 'SU(4) Relation between AF and SC order' textcolor rgb title\_color

set xlabel 'AF order beta' textcolor rgb tic\_color

set ylabel 'Pairing Order Delta (beta, x)' textcolor rgb tic\_color

# Uncomment following to set log or log-log plots

#set logscale x

#set logscale y

set xrange [-0.5:0.5]

set yrange[0:0.5]

set pointsize 0.5 # Size of plotted points

#unset key # Don't show legend

set key out # Put legend outside plot

set key vert # vert for vertical; horiz for horizontal

set key top right # Move legend to upper right

#set timestamp # Date/time

Omega = 1.0

# Define function to plot. The conditional A = (condition ? option 1 : option 2)

# sets A to option 1 if condition is satisfied and to option 2 if it is not. The

# construction 1/0 for one of the options effectively means don't plot.

# Physical ranges for beta

blow(x) = -0.5\*sqrt(1-x)

bhigh(x) = 0.5\*sqrt(1-x)

Q(beta,x) = (beta >= blow(x) && beta <= bhigh(x) ? 2\*Omega\*beta\*sqrt(0.5\*(1-x) - beta\*\*2) : 1/0)

Del(beta,x) = (beta >= blow(x) && beta <= bhigh(x) ? 0.5\*sqrt(0.25-(Q(beta,x)-x/2)\*\*2) \

+ 0.5\*sqrt(0.25-(Q(beta,x)+x/2)\*\*2) : 1/0)

# Set range on independent variable to change it from the default x to beta using plot[beta=low:high]

plot[beta=-0.5:0.5] Del(beta,0.0) ls 1, Del(beta,0.1) ls 2, Del(beta,0.2) ls 3, Del(beta,0.3) ls 4, \

Del(beta,0.4) ls 5, Del(beta,0.5) ls 6, Del(beta,0.6) ls 7, Del(beta,0.7) ls 8, Del(beta,0.8) ls 9, \

Del(beta,0.9) ls 10, Del(beta,0.99) ls 11

# Plot to postscript file

set out "DeltaVsBeta.eps" # Output file

set terminal postscript eps size width, height enhanced color solid lw 2 "Arial" 32

replot # Plot to postscript file

# Plot to PNG file

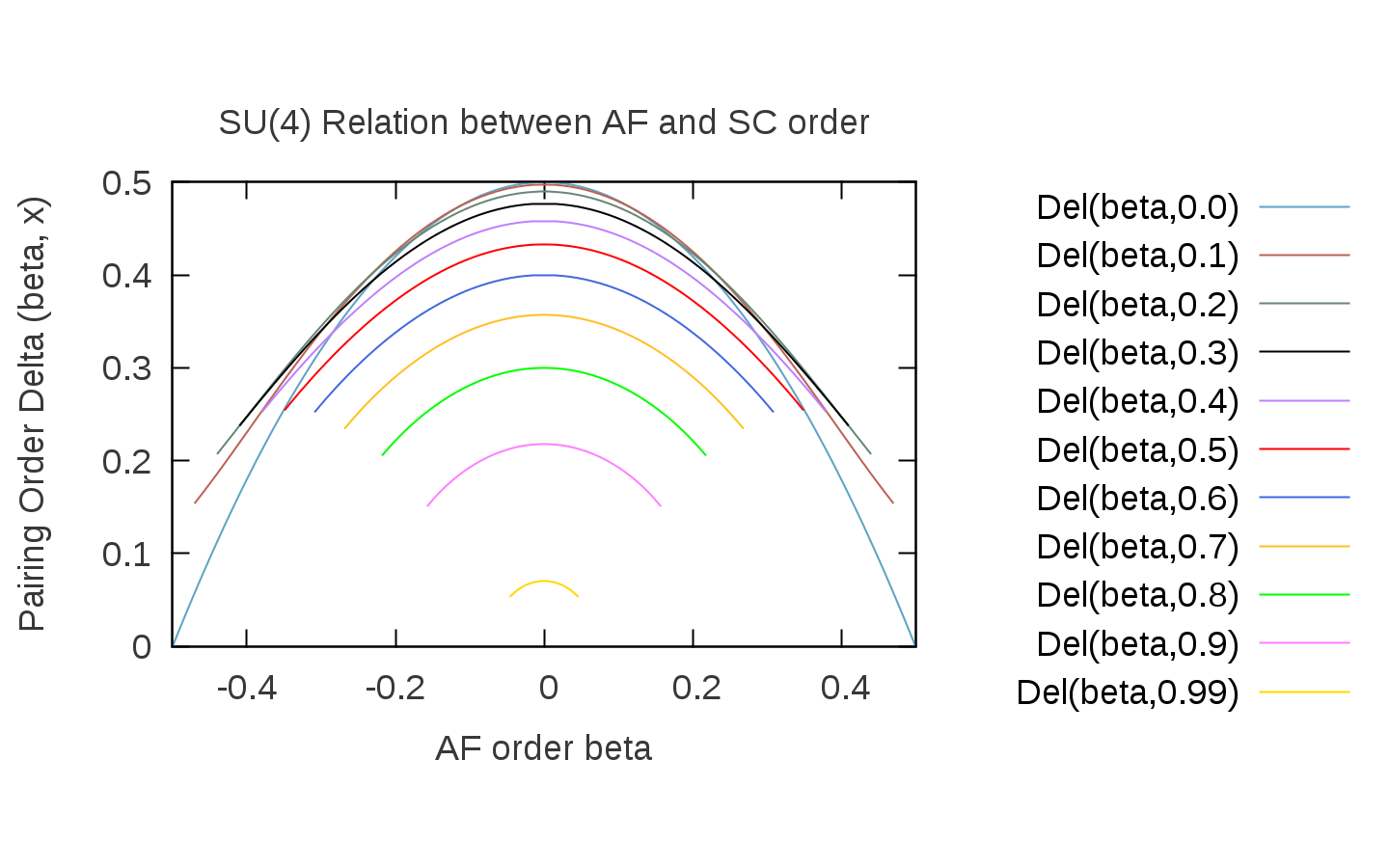
set out "DeltaVsBeta.png"

# Assume 72 pixels/inch and make bitmap twice as large for display resolution

set terminal pngcairo transparent size 2\*width\*72, 2\*height\*72 lw 2 enhanced font 'Arial,28'

replot

quit



**Saving Plot as Postscript**

(Note that several examples above show how to generate both PS and PNG directly, without having to do this.) Once a plot is displayed on a screen, it can be saved as a postscript file by the following procedure.

1. Place the following script in a file in the directory where you are working:

# File name: save.plt - save a Gnuplot plot as a PostScript file

# to save the current plot as a postscript file issue the commands:

# gnuplot> load 'saveplotPS.gnu'

# gnuplot> !mv myplot.ps another-file.ps

set size 1.0, 0.45 # Relative vertical and horizontal size

set terminal postscript portrait enhanced color solid lw 1 "Arial" 16

set output "myplot.ps"

replot

set terminal x11

set size 1,1

# The meaning of the arguments for postscript:

# set terminal postscript {<mode>} {enhanced | noenhanced}

# {color | colour | monochrome}

# {blacktext | colortext | colourtext}

# {solid | dashed} {dashlength | dl <DL>}

# {linewidth | lw <LW>}

# {<duplexing>}

# {"<fontname>"} {<fontsize>}

# The meaning of the arguments for GIF output

# set terminal gif {transparent} {interlace}

# {tiny | small | medium | large | giant}

# {size <x>,<y>}

# {<color0> <color1> <color2> ...}

# The meaning of the arguments for PNG output

# set terminal png

# {{no}transparent} {{no}interlace}

# {tiny | small | medium | large | giant}

# {font <face> {<pointsize>}}

# {size <x>,<y>} {{no}crop}

# {{no}enhanced}

# {<color0> <color1> <color2> ...}

2. Then issue the following command at the gnuplot prompt:

gnuplot> load 'saveplotPS.gnu'

This will produce a postscript file of the currently displayed plot with the default name “myplot.ps” (this name is set in the script).

3. Since doing this again will overwrite myplot.ps, copy it to another file name to save.

**Colors, LineStyles, and Pointstyles**

Examples for setting linestyles:

set style line 1 lt rgb "#5ea2c6" lw 1 # Define linestyle 1; rgb is hex #RRGGBB

set style line 2 lt rgb "purple" lw 1 # Define linestyle 2

where "purple" is one of the predefined color names in the list given below. Then

these styles can be used, e.g., with

plot sin(x) ls 1, cos(x) ls 2

Examples for setting pointstyles for plotting data:

# Point styles

set style line 12 lc rgb "#5ea2c6" pt 5 # square

set style line 13 lc rgb "#5ea2c6" pt 7 # circle

set style line 14 lc rgb 'dark-orange' pt 9 # triangle

Example for defining custom colors:

set macro # Enable macro definition

# Some macro definitions

# Custom colors: hex #RRGGBB in quotes

label\_color = "#867961"

tic\_color = "#383838"

title\_color = "#383838"

myblue\_color = "#5ea2c6

myred\_color = "#bb6255"

mygreen\_color = "#668874"

Then they can be used in, for example, a linestyle definition:

set style line 1 lt rgb myblue\_color lw 1 # Define linestyle 1

***gnuplot: 112 predefined color names:***

white #ffffff = 255 255 255

black #000000 = 0 0 0

dark-grey #a0a0a0 = 160 160 160

red #ff0000 = 255 0 0

web-green #00c000 = 0 192 0

web-blue #0080ff = 0 128 255

dark-magenta #c000ff = 192 0 255

dark-cyan #00eeee = 0 238 238

dark-orange #c04000 = 192 64 0

dark-yellow #c8c800 = 200 200 0

royalblue #4169e1 = 65 105 225

goldenrod #ffc020 = 255 192 32

dark-spring-green #008040 = 0 128 64

purple #c080ff = 192 128 255

steelblue #306080 = 48 96 128

dark-red #8b0000 = 139 0 0

dark-chartreuse #408000 = 64 128 0

orchid #ff80ff = 255 128 255

aquamarine #7fffd4 = 127 255 212

brown #a52a2a = 165 42 42

yellow #ffff00 = 255 255 0

turquoise #40e0d0 = 64 224 208

grey0 #000000 = 0 0 0

grey10 #1a1a1a = 26 26 26

grey20 #333333 = 51 51 51

grey30 #4d4d4d = 77 77 77

grey40 #666666 = 102 102 102

grey50 #7f7f7f = 127 127 127

grey60 #999999 = 153 153 153

grey70 #b3b3b3 = 179 179 179

grey #c0c0c0 = 192 192 192

grey80 #cccccc = 204 204 204

grey90 #e5e5e5 = 229 229 229

grey100 #ffffff = 255 255 255

light-red #f03232 = 240 50 50

light-green #90ee90 = 144 238 144

light-blue #add8e6 = 173 216 230

light-magenta #f055f0 = 240 85 240

light-cyan #e0ffff = 224 255 255

light-goldenrod #eedd82 = 238 221 130

light-pink #ffb6c1 = 255 182 193

light-turquoise #afeeee = 175 238 238

gold #ffd700 = 255 215 0

green #00ff00 = 0 255 0

dark-green #006400 = 0 100 0

spring-green #00ff7f = 0 255 127

forest-green #228b22 = 34 139 34

sea-green #2e8b57 = 46 139 87

blue #0000ff = 0 0 255

dark-blue #00008b = 0 0 139

midnight-blue #191970 = 25 25 112

navy #000080 = 0 0 128

medium-blue #0000cd = 0 0 205

skyblue #87ceeb = 135 206 235

cyan #00ffff = 0 255 255

magenta #ff00ff = 255 0 255

dark-turquoise #00ced1 = 0 206 209

dark-pink #ff1493 = 255 20 147

coral #ff7f50 = 255 127 80

light-coral #f08080 = 240 128 128

orange-red #ff4500 = 255 69 0

salmon #fa8072 = 250 128 114

dark-salmon #e9967a = 233 150 122

khaki #f0e68c = 240 230 140

dark-khaki #bdb76b = 189 183 107

dark-goldenrod #b8860b = 184 134 11

beige #f5f5dc = 245 245 220

olive #a08020 = 160 128 32

orange #ffa500 = 255 165 0

violet #ee82ee = 238 130 238

dark-violet #9400d3 = 148 0 211

plum #dda0dd = 221 160 221

dark-plum #905040 = 144 80 64

dark-olivegreen #556b2f = 85 107 47

orangered4 #801400 = 128 20 0

brown4 #801414 = 128 20 20

sienna4 #804014 = 128 64 20

orchid4 #804080 = 128 64 128

mediumpurple3 #8060c0 = 128 96 192

slateblue1 #8060ff = 128 96 255

yellow4 #808000 = 128 128 0

sienna1 #ff8040 = 255 128 64

tan1 #ffa040 = 255 160 64

sandybrown #ffa060 = 255 160 96

light-salmon #ffa070 = 255 160 112

pink #ffc0c0 = 255 192 192

khaki1 #ffff80 = 255 255 128

lemonchiffon #ffffc0 = 255 255 192

bisque #cdb79e = 205 183 158

honeydew #f0fff0 = 240 255 240

slategrey #a0b6cd = 160 182 205

seagreen #c1ffc1 = 193 255 193

antiquewhite #cdc0b0 = 205 192 176

chartreuse #7cff40 = 124 255 64

greenyellow #a0ff20 = 160 255 32

gray #bebebe = 190 190 190

light-gray #d3d3d3 = 211 211 211

light-grey #d3d3d3 = 211 211 211

dark-gray #a0a0a0 = 160 160 160

slategray #a0b6cd = 160 182 205

gray0 #000000 = 0 0 0

gray10 #1a1a1a = 26 26 26

gray20 #333333 = 51 51 51

gray30 #4d4d4d = 77 77 77

gray40 #666666 = 102 102 102

gray50 #7f7f7f = 127 127 127

gray60 #999999 = 153 153 153

gray70 #b3b3b3 = 179 179 179

gray80 #cccccc = 204 204 204

gray90 #e5e5e5 = 229 229 229

gray100 #ffffff = 255 255 255

**Example 3D Surface Plot**

set key inside center top vertical Right noreverse enhanced autotitles box \

linetype -1 linewidth 0.5

set xlabel 'k\_x'

set ylabel 'k\_y'

set zlabel 'E(k\_x, k\_y)'

# To plot only the surfaces with no labels or axes, uncomment following

# two lines. Comment out to include those things.

unset key; unset tics; unset border; unset colorbox;

set xlabel ''; set ylabel ''; set zlabel ''

set samples 600, 600 # x and y resolution of plot; larger is better

set style line 1 lt rgb "#5ea2c6" lw 1 # Define linestyle 1; solid blue

set style line 2 lt rgb "#bb6255" lw 1 # Define linestyle 2; solid red

set style line 3 lt rgb "#668874" lw 1 # Define linestyle 3; solid green

set style line 4 lt rgb "black" lw 0.5 # Solid black line

# Change default (x,y) plot variables to (kx, ky)

set dummy kx, ky

# Set window terminal canvas size

set terminal wxt size 750, 750

# Width and height of postscript figure in inches

width = 8

height = 8

# Set screen display to same aspect ratio as postscript plot

set size ratio height/width

# Rotate z-axis label to vertical and shift it two pixels to right

set zlabel rotate by 90 offset 2, 0, 0

#unset xtics

#unset ytics

#unset ztics

# Control amount of blank space below plot

set xyplane 0

set xrange [-2.5:2.5]

set yrange [-2.5:2.5]

set zrange [-8:12]

set isosamples 150 # Number surface mesh lines; larger -> larger .ps

#set isosamples 400 # Production quality

set hidden3d # Remove lines that would be hidden if surface opaque

#set key outside

set palette model RGB # or CMY (RGB is default);

set palette defined (0 '#9c27b0', 0.4 'gold', 1 '#3f51b5') # gradient

#set palette defined (0 'dark-plum', 0.4 'gold', 1 '#3f51b5') # gradient

#set palette defined (0 '#ff0000', 1 '#0000ff') # gradient

#set palette defined (0 'dark-plum', 1 'gold') # gradient

#set palette defined (0 'dark-plum', 0.5 'gold', 1 '#2e7d32') # gradient

#set palette gray

set pm3d # Shaded color surface

set view 80, 50, 1.0, 1.0 # Viewing angles and scale factors

#set contour base # Contours on base, surface, or both

#set cntrparam levels 20 # Target number of contour levels

a = 1

factor = sqrt(3)\*a

coeff1 = 3

coeff2 = 0.3

f(kx,ky)=3+2\*cos(factor\*kx)+4\*cos(0.5\*factor\*kx)\*cos(1.5\*ky)

# Equivalent form

#f(kx,ky) = 3 + 2\*cos(factor\*kx) + 2\*cos(0.5\*factor\*(kx+sqrt(3)\*ky)) +\

#2\*cos(0.5\*factor\*(-kx+sqrt(3)\*ky))

# Dispersion lower branch

el(kx,ky) = -coeff1\*sqrt(f(kx,ky)) +coeff2\*f(kx,ky)

# Dispersion upper branch

eu(kx,ky) = coeff1\*sqrt(f(kx,ky)) +coeff2\*f(kx,ky)

splot el(kx,ky), eu(kx,ky) notitle

# Plot to postscript file

set out "grapheneDispersion3D.eps" # Output file

set terminal postscript eps size width, height enhanced color solid lw 2 "Arial" 32

replot # Plot to postscript file

# Plot to PNG file

set out "grapheneDispersion3D.png"

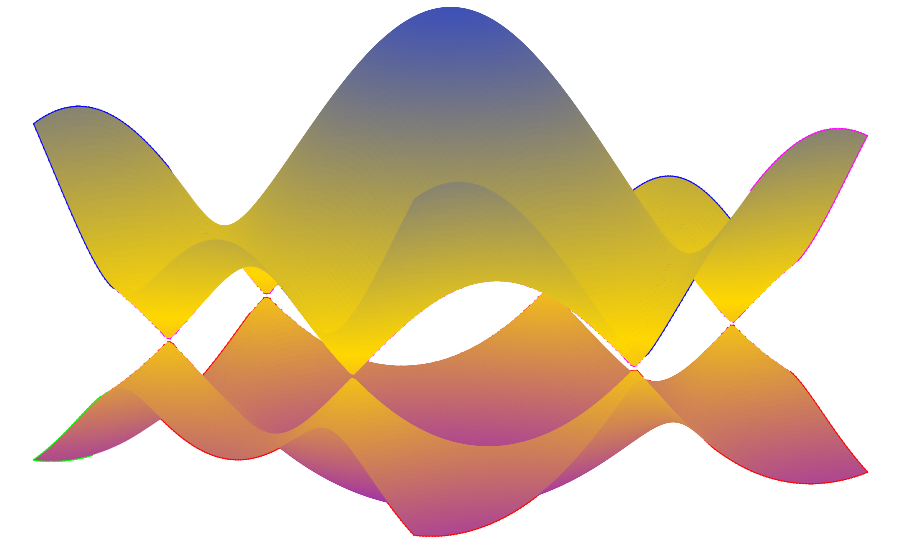
# Assume 72 pixels/inch and make bitmap twice as large for display resolution

set terminal pngcairo transparent size 2\*width\*72, 2\*height\*72 lw 2 enhanced font 'Arial,28'

replot

quit

The following image of the dispersion for graphene was plotted with the above script with the lines uncommented to suppress the axes and labels:



Here is the corresponding image with the axes and labels:

