Data Graphics with Gnuplot

Le Yan

User Services
HPC @ LSU

Training Goals

- Produce simple interactive plots and graphs
- Create 2- and 3-d graphs from functions and data files
- Understand the automation potential of Gnuplot

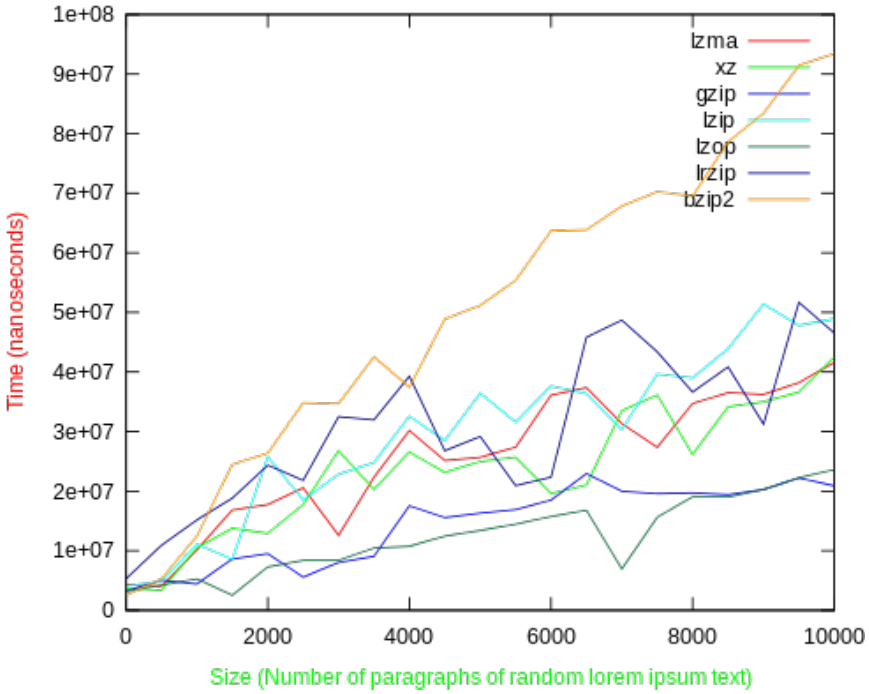
What is Gnuplot

- **Gnuplot** is a portable command-line driven graphing utility
  - for Linux, IBM AIX, MS Windows, Apple OS-X, and many other platforms
  - Open source
  - Used as the plotting engine of applications such as Octave
  - Can be used with various languages such as Perl and Python
  - Alternatives: MATLAB, Mathermatica, matploblib (python)

What is Gnuplot for?

```
1, 1, 223.04
1, 2, 220.18
1, 3, 201.26
1, 4, 199.15
1, 5, 205.6
1, 6, 213.63
1, 7, 209.51
1, 8, 196.15
1, 9, 199.64
1, 10, 198.51
1, 11, 201.32
1, 12, 204.27
1, 13, 201.88
1, 14, 189.18
1, 15, 197.51
1, 16, 183.16
1, 17, 198.2
1, 18, 198.53
1, 19, 202.36
1, 20, 186.98
1, 21, 169.01
1, 22, 194.02
1, 23, 194.02
1, 24, 194.59
1, 25, 189.92
1, 26, 184.62
1, 27, 190.85
1, 28, 184.96
1, 29, 184.96
1, 30, 183.78
1, 31, 176.46
```

Gnuplot



Gnuplot vs. Excel

- Gnuplot
  - Can be readily embedded in a program
  - Allows the batch processing of many files with simple scripting
  - Has many different terminal types

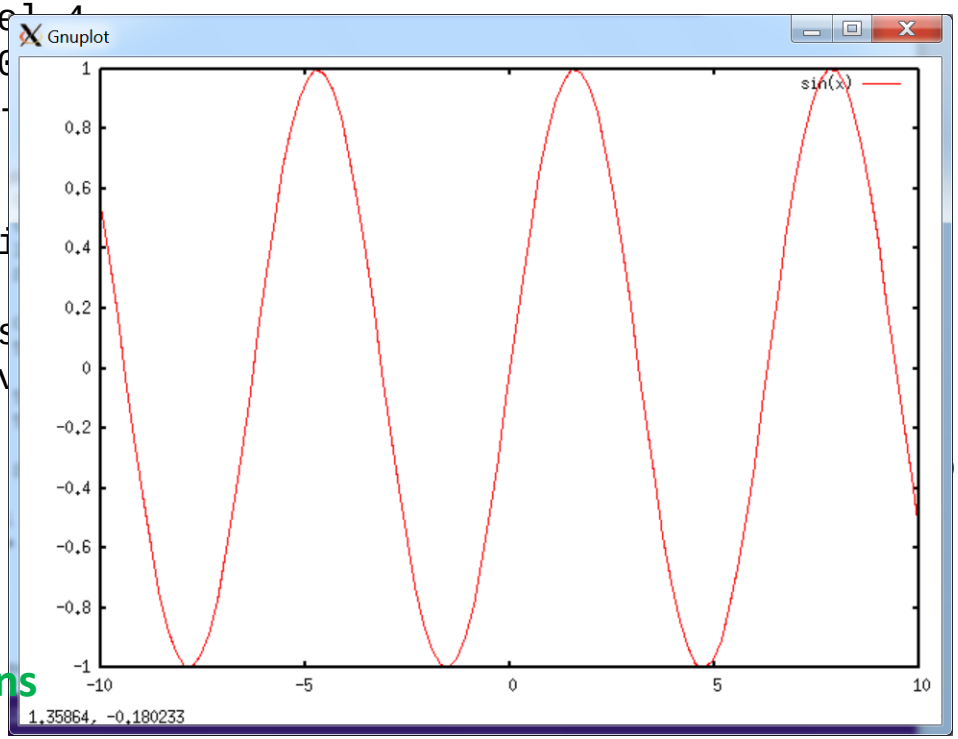
Availability on LONI and HPC Systems

- Super Mike 2: Gnuplot 4.6.0
  - Softenv key: `+gnuplot-4.6.0-gcc-4.4.6`
- Linux systems: Gnuplot 4.2.4
  - Softenv key: `+gnuplot-4.2.4-intel-11.1`
- Pandora: Gnuplot 4.4.3
  - Softenv key: `+gnuplot-4.4.3`
- You will need a working X Window system if you want to view the plot immediately
  - Linux and Mac: use "-X" option when connecting to the cluster
  - Windows: need to have a X Window server (e.g. Xming) and enable X11 forwarding in the ssh client

The Very First Plot

```
[lyan1@philip1 ~]$ gnuplot
        G N U P L O T
        Version 4.2 patchlevel 4
        last modified Sep 200
        System: Linux 2.6.18-

        Copyright (C) 1986 -
        Thomas Williams, Coli

        Type `help` to access
        The gnuplot FAQ is av

        Send bug reports and                                    jects/gnu

Terminal type set to 'x11'
gnuplot> plot sin(x)
```

Default plot key

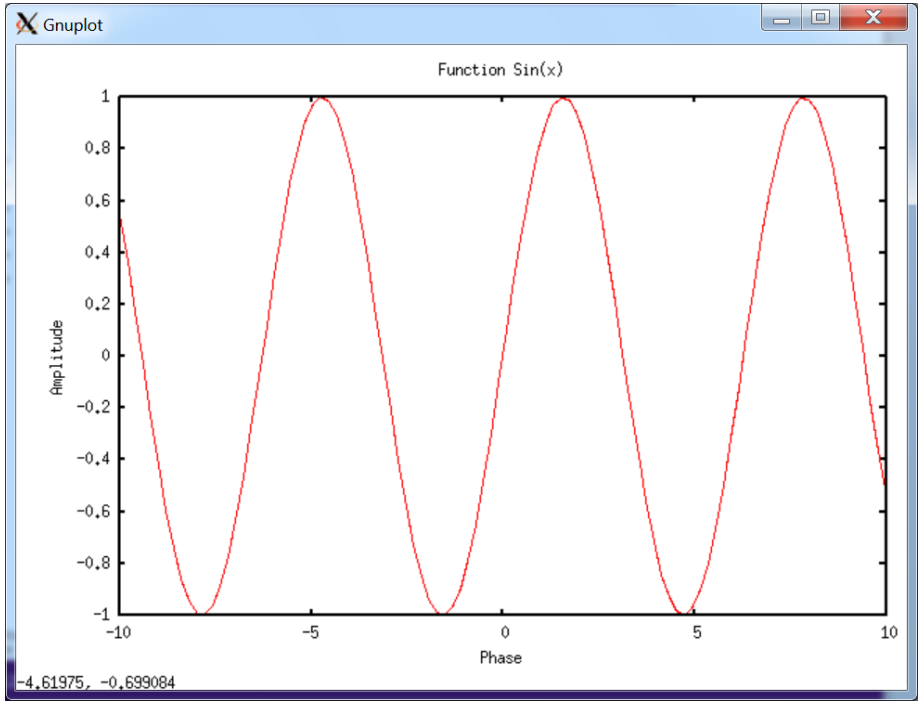


Mathematical functions

Default x- and y- axis ranges and ticks

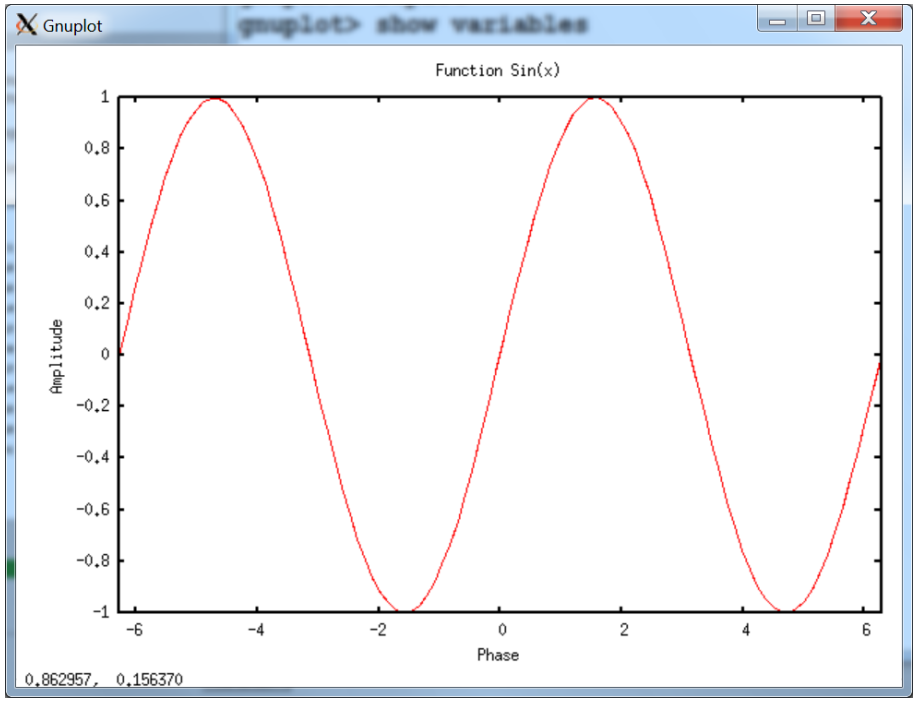
Refine the Plot (1)

```
gnuplot> plot sin(x)
gnuplot> unset key # remove plot key
gnuplot> set title "Function Sin(x)“ # Add a title above the plot area
gnuplot> set xlabel "Phase“ # Add a label for x-axis
gnuplot> set ylabel "Amplitude“ # Add a label for y-axis
gnuplot> replot
```



Refine the Plot (2)

```
gnuplot> set xrange [-2.0*pi:2.0*pi] # You can use constants, variables or
mathematical expressions
gnuplot> replot
```



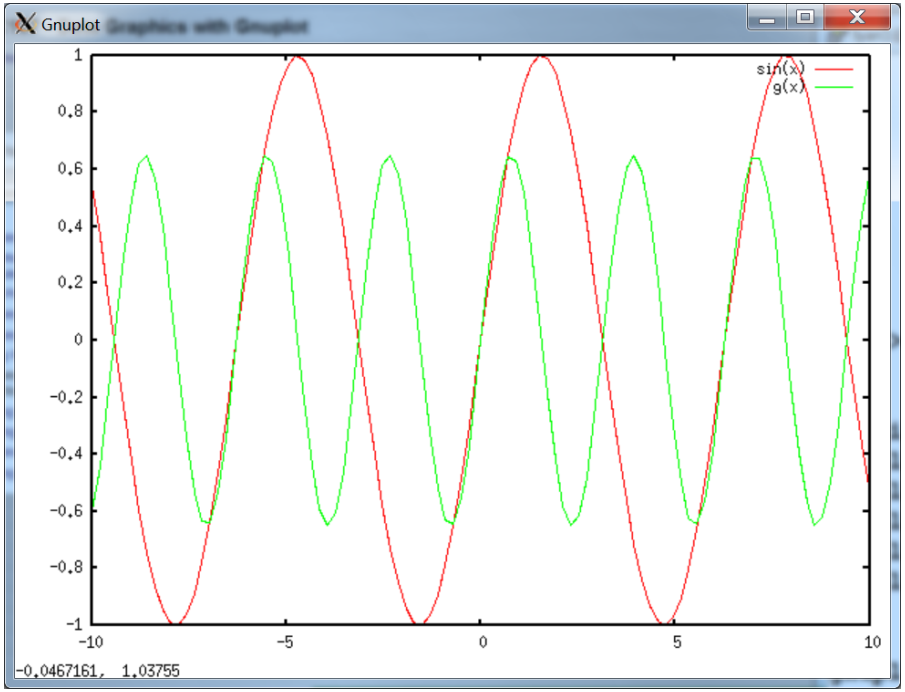
Commands: plot and replot

- **Plot** generates a new plot
  - Gnuplot records settings and uses them until they are changed
- **Replot** repeats the last **plot** or **splot** command, using remembered settings
  - Capable of adding a plot specification to what has already been display

Defining Functions and Variables

- Syntax
  - <varname> = <value>
  - <funcname> (<var1>{, <var2>, ...}) = <expression>

```
gnuplot> plot sin(x)
gnuplot> A = 1.3
gnuplot> g(x)=A*cos(x)*sin(x)
gnuplot> replot g(x)
```



Saving and Loading Work

- Gnuplot allows you to save your efforts in a file, and reuse it at a later time
  - Particularly useful with standarized plots for which data changes periodically
  - The files are plain text and can be edited using any text editor
  - Files names need to be quoted

```
gnuplot> save "my_gnuplot_file"
gnuplot> load "my_gnuplot_file"
```


Ending a Gnuplot Session

- To end a Gnuplot session, use the **quit** or **exit** command

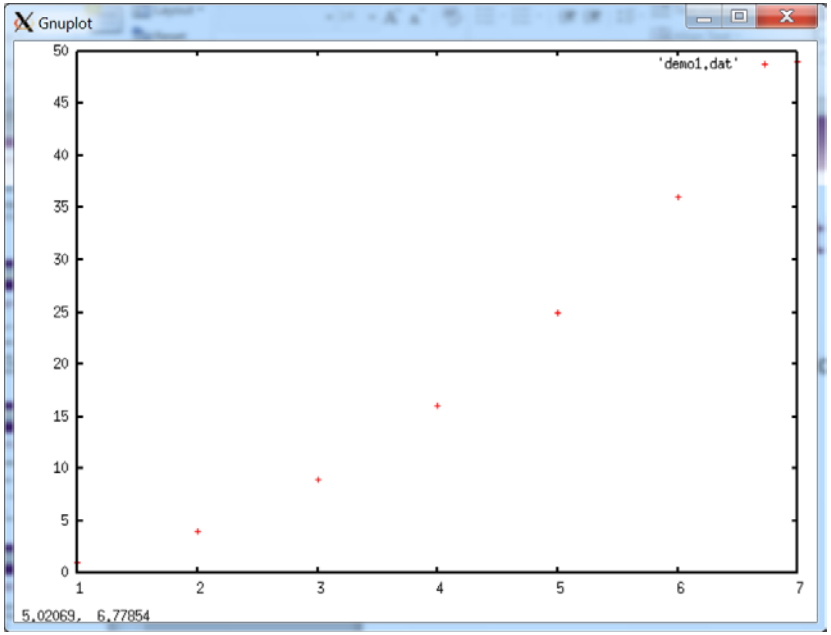
Getting Help

- Syntax
  - Help <topic> <subtopic> …

```
gnuplot> help functions abs
 The `abs(x)` function returns the absolute value of its argument.  The
 returned value is of the same type as the argument.

 For complex arguments, abs(x) is defined as the length of x in the complex
 plane [i.e.,  sqrt(real(x)**2 + imag(x)**2) ].
```

Working with Data Files

- Syntax
  - Plot “data file name” <options>
- Works with both text and binary data, but we will focus on text data




```
gnuplot> plot 'demo0.dat'
gnuplot> !cat 'demo0.dat'
1 1
2 4
3 9
4 16
5 25
6 36
7 49
```

Files for Exercises

- Files can be found under `/home/lyan1/traininglab/gnuplot` on all Linux clusters

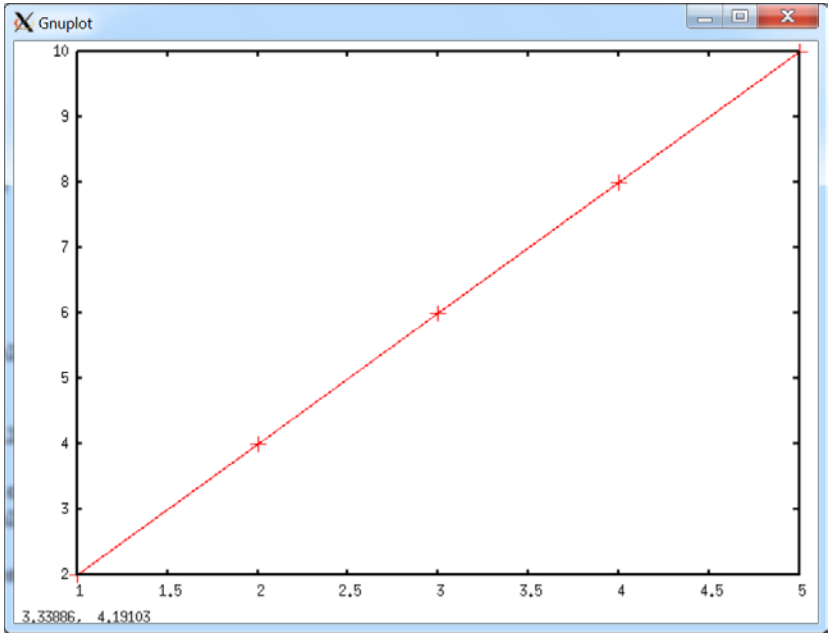
Date File Structure

- A data file may contain
  - Comment lines
  - Point data provided in column format
    - White space is the default delimiter unless otherwise specified
  - Breaks to separate unique lines within a data set
  - Multiple data sets
    - The index starts from 0

```
# This is a comment
# Data records starts here
1 1
2 4
3 9

# One blank marks discontinuity
# in a plot
4 16
5 25


# Two blanks mark end of dataset
# The data below belongs to
# dataset 1
1 2 4
2 4 8
3 6 12
```

INDEX option

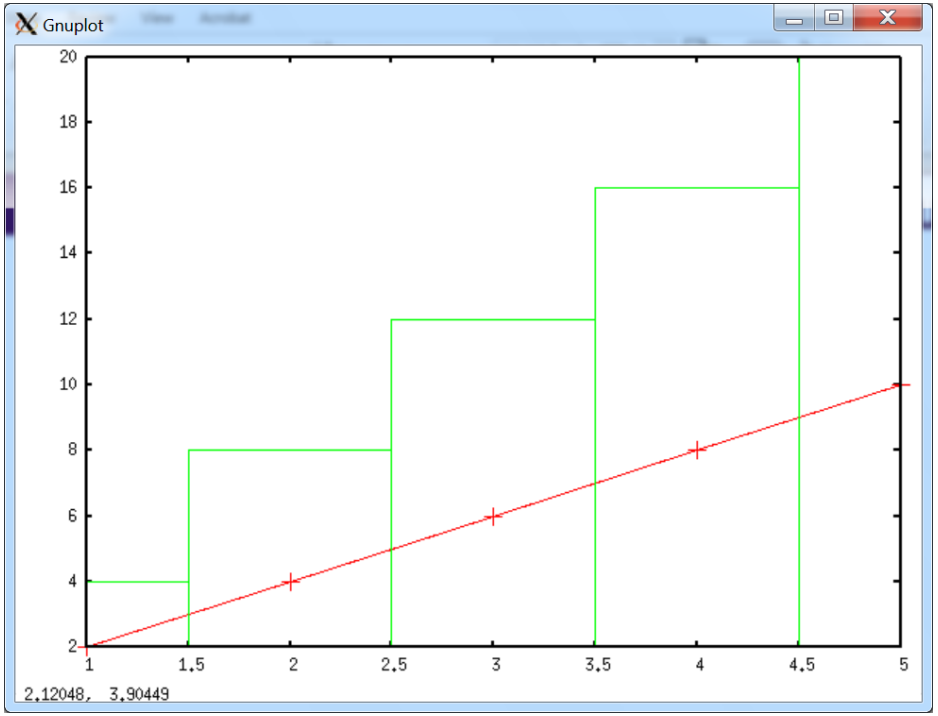
- The **INDEX** option allows users to choose datasets to plot from a data file




```
gnuplot> unset key
gnuplot> set pointsize 3
gnuplot> plot 'demo1.dat' index 1 using 1:2 with lp
```

USING option

- The **USING** keyword allows user to
  - Select columns from a multi-column data set
  - Manipulate the elements of a column to change the values that are plot




```
gnuplot> plot 'demo1.dat' index 1 using 1:2 with lp, \
> 'demo1.dat' index 1 using 1:3 with boxes
```

WITH option

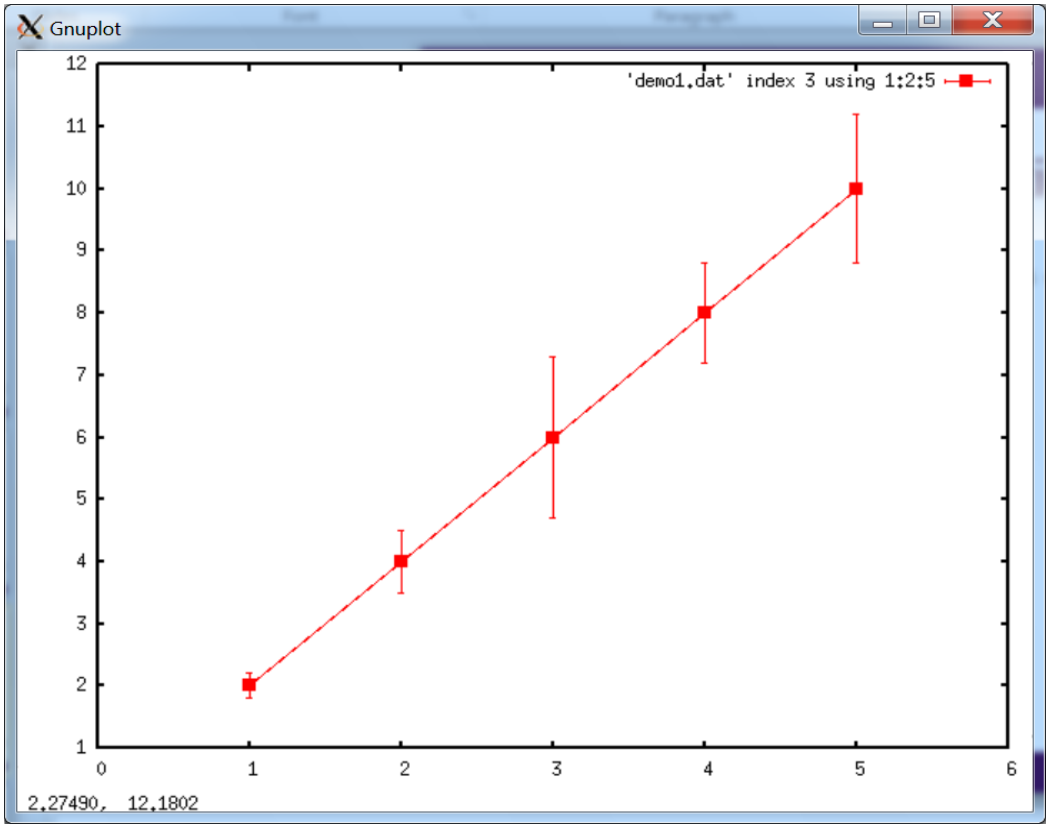
- The **WITH** keyword allows specification of the style of the data display
- Possible styles

```
Subtopics available for plotting styles:
    boxerrorbars      boxes             boxxyerrorbars    candlesticks
    dots              errorbars         errorlines        filledcurves
    financebars       fsteps            histeps           histograms
    image             impulses          labels            lines
    linespoints       points            rgbimage          steps
    vectors           xerrorbars        xerrorlines       xyerrorbars
    xyerrorlines      yerrorbars        yerrorlines
```

- Use the **help plotting styles** command to find more

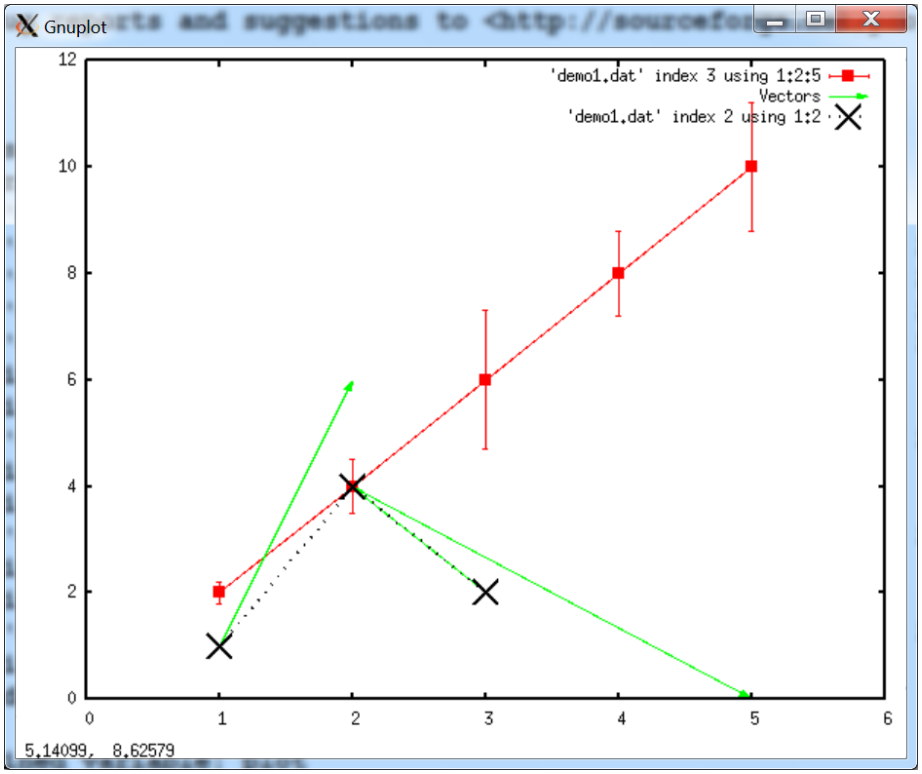
WITH option

```
gnuplot> set xrange [0:6]
gnuplot> plot 'demo1.dat' index 3 using 1:2:5 with yerrorlines pt 5 ps 2
```



Multiple Plots with Single Plot Command

```
gnuplot> plot 'demo1.dat' i 3 u 1:2:5 w yerrorlines pt 5 ps 2, \
> 'demo1.dat' i 2 w vectors head filled title "Vectors", \
> 'demo1.dat' i 2 u 1:2 w lp lt 0 lw 2 ps 4
```

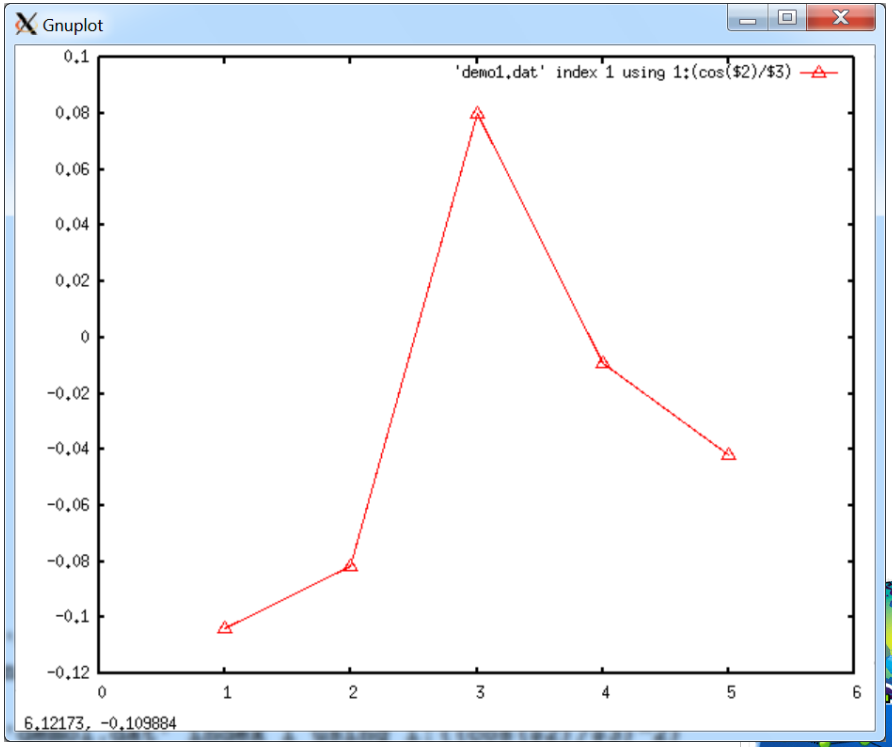


Separate plot specifications by comma

Processing Data Before Plotting

- Gnuplot allows one to process the data in a column before it is used to plot
  - Use the **using** option with a function defined with **$N** representing values from the N'th column.
  - This feature helps plot transformed data without writing a program to create a transformed data file.

```
gnuplot> plot 'demo1.dat' i 1 \
> u 1:(cos($2)/$3) \
> w lp ps 2 pt 8
```



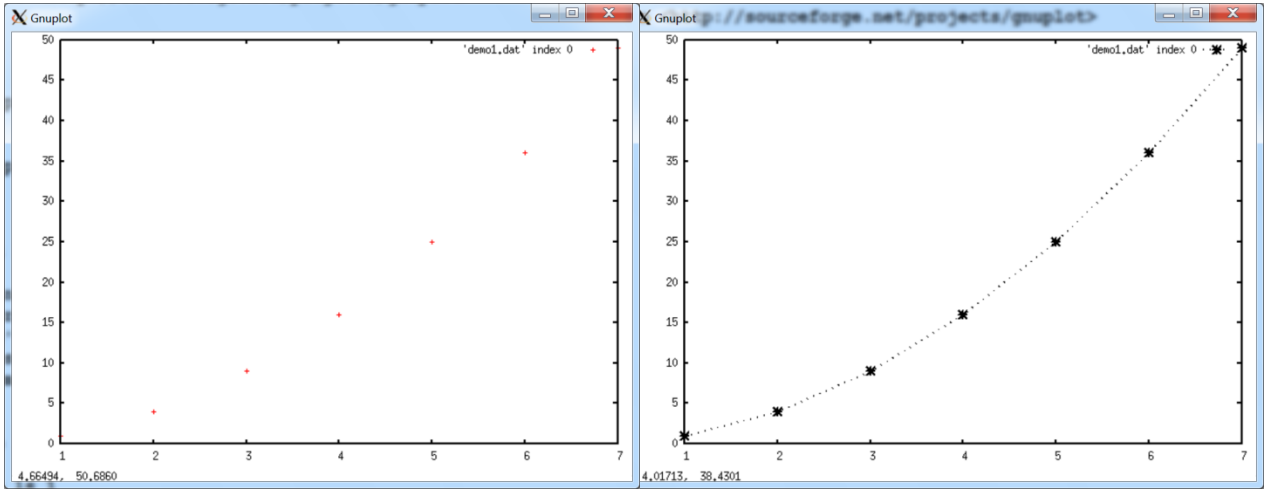
SET Command

- One can use the **set** command to set LOTS of options
  - Try **help set** to see all the option names
- The option value can be shown by the **show** command
  - Syntax: show <option name>
  - **show all** shows all the options (which will be a very long list)

Setting Line Styles

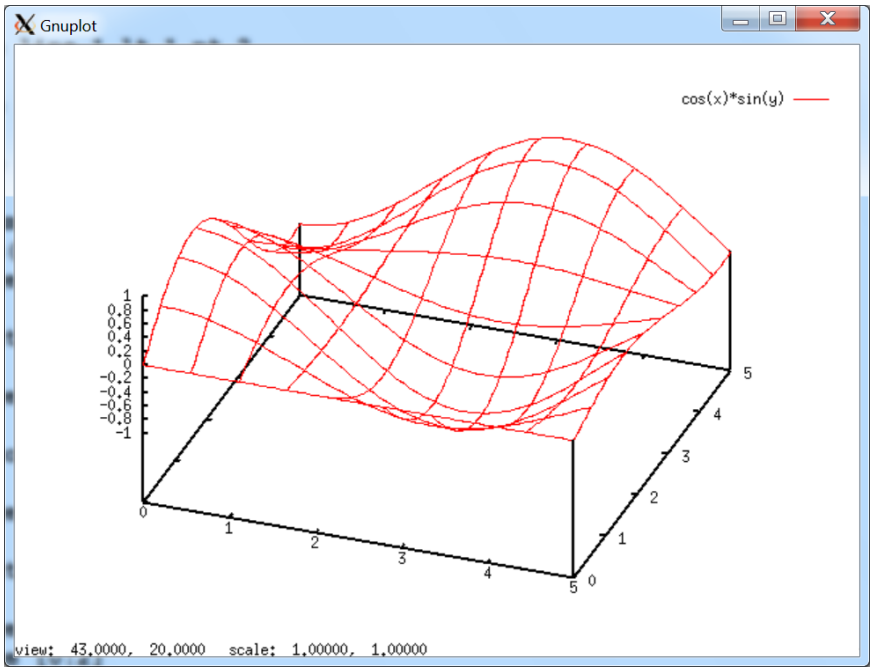
- Gnuplot allows users to define line styles (line weight, line type, line color, point size, point type etc.) and use them later by referring to the index

```
gnuplot> plot 'demo1.dat' index 0
gnuplot> set style line 1 lt 0 lw 2 pt 3 ps 2
gnuplot> plot 'demo1.dat' i 0 w lp ls 1
```



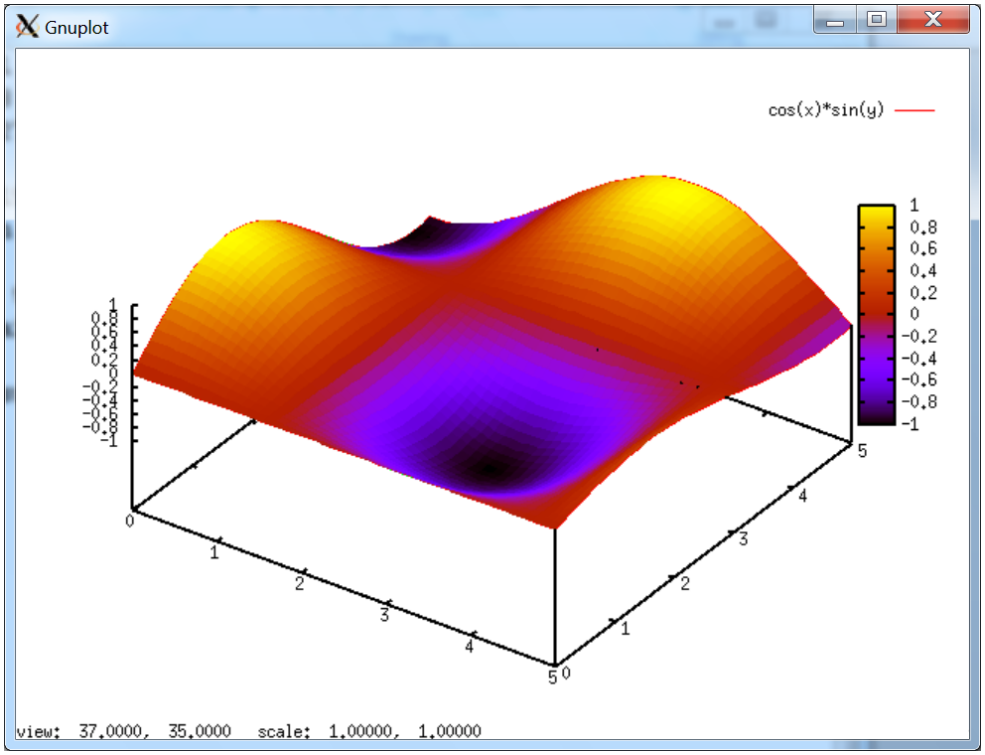
3D Plot

- The **splot** command generates surface plot
  - Functions with 2 variables, matrix data, or x-y-z data
  - Works in the same fashion with the **plot** command
- Multiple styles are available
- Can add contour to it




```
gnuplot> set xrange [0:5]
gnuplot> set yrange [0:5]
gnuplot> splot cos(x)*sin(y)
```

Refining 3D Plot

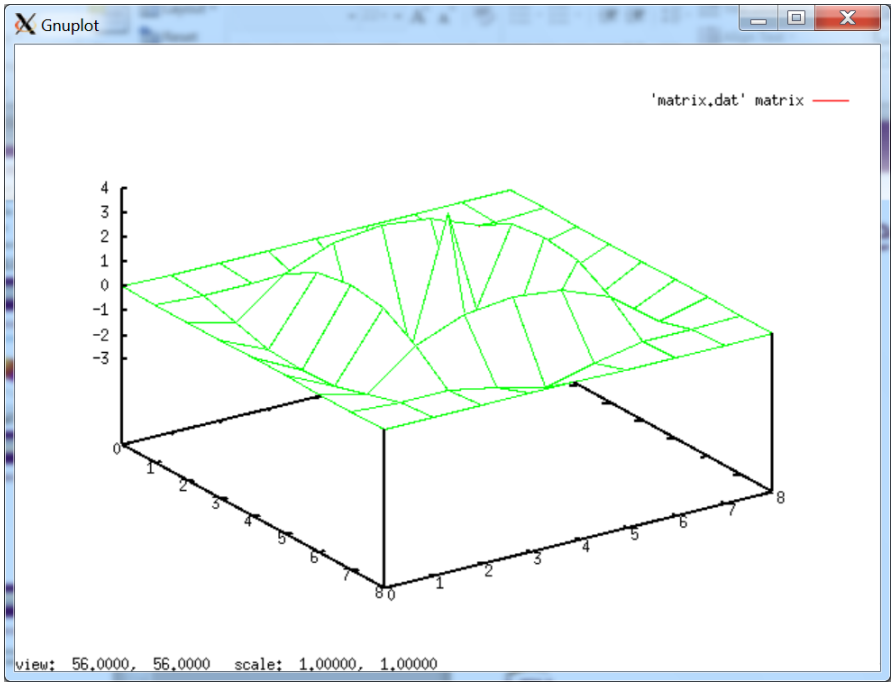



```
gnuplot> set pm3d # Add pallete map
gnuplot> set isosamples 50,50 # Increase sample points
gnuplot> set hidden3d # Enable hidden line removal
gnuplot> replot
```

3D Plot – Matrix Data

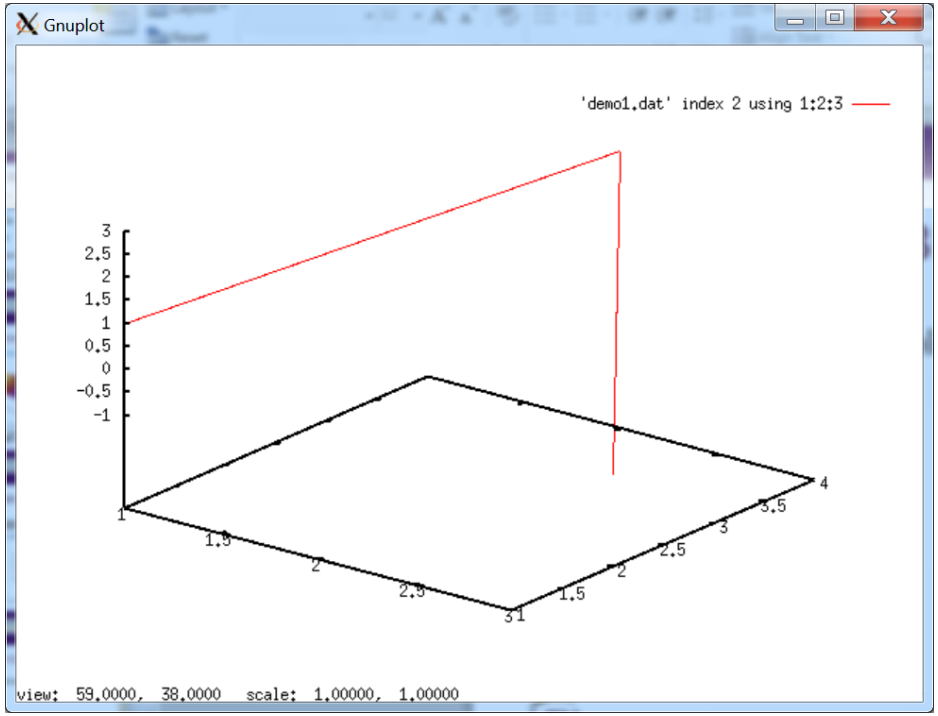
- Matrix data can be plotted using the matrix option
  - The cell index is used as the X-Y coordinates, and the cell contents as the Z value

```
gnuplot> set hidden3d
gnuplot> splot 'matrix.dat' matrix with lines
```



3D Plot – X-Y-Z data

- Can also use **splot** to plot column data




```
gnuplot> splot 'demo1.dat' index 2 using 1:2:3 with lines
```

Terminal Types

- Gnuplot supports many terminal types
  - Use **help set terminal** to see all of them
  - Mainly three types: actual terminals, printers, graphical file formats
- The sub-options vary from terminal to terminal
  - GIF terminal
    - Font settings, background settings, animation options etc.
  - PostScript terminal
    - Font settings, page orientation, print settings etc.

Saving Graphical Output

- Plots and graphs can be saved to file by the following steps
  - Set terminal type
  - Specify file name
  - Generate the plot
- If the plot has been generated, then use the replot command to save to a file after setting the terminal type and file name

```
gnuplot> set terminal jpeg
Terminal type set to 'jpeg'
Options are 'nocrop medium '
gnuplot> set output "3d_xyz.jpg“
gnuplot> splot 'demo1.dat' index 1 using 1:2:3 with lines
```

Gnuplot script

- Like shell scripts, a Gnuplot script is a text file that contains Gnuplot commands
  - We already saw the **save** and **load** command
  - We can also write one from scratch, then
  - Run it by
    - Executing `gnuplot <script name>`
    - Making it executable by adding the shebang line

Gnuplot script

```
[lyan1@philip1 gnuplot]$ cat 3d_surface.gnuplot
#!/usr/local/packages/gnuplot/4.2.4/intel-
11.1/bin/gnuplot

unset key
set xrange [0:5]
set yrange [0:5]
set hidden3d
set pm3d
set isosamples 50,50
set terminal jpeg
set output "3d_surface.jpg"

splot cos(x)*sin(y) title "3D surface plot"
[lyan1@philip1 gnuplot]$ chmod u+x 3d_surface.gnuplot
[lyan1@philip1 gnuplot]$ ./3d_surface.gnuplot
[lyan1@philip1 gnuplot]$ display 3d_surface.jpg
```

Curve Fitting

- Two step process
  - Define a function with unknown parameters
  - Fit the parameters with specified data
- A log file will be created (default name is ‘fit.log’)

```
gnuplot> f(x)=a*x**2+b*x+c
gnuplot> fit f(x) 'demo1.dat' index 0 via a,b,c
…
Final set of parameters            Asymptotic Standard Error
=======================            ==========================

a                 = 1              +/- 4.893e-15    (4.893e-13%)
b                 = -2.55713e-13   +/- 8.214e-14    (32.12%)
c                 = 3.30293e-13    +/- 1.098e-14    (3.323%)
…
```

Shell Commands

- Gnuplot supports the ability to issue shell commands
  - Shell commands start with "!"

```
gnuplot> !ls
demo1.dat  matrix.dat
!
gnuplot> !head demo1.dat
# Data set 0
1 1
2 4
3 9
4 16
5 25
6 36
7 49


!
gnuplot> !pwd
/home/lyan1/traininglab/gn
uplot
!
```

What's Next

- HPC Moodle course "Data Graphics with Gnuplot"
  - https://docs.loni.org/moodle/course/view.php?id=19
- Gnuplot website
  - Links to user documentation and tutorials
  - Demo scripts: http://www.gnuplot.info/demo_4.6/

Questions?


Shreveport
Monroe
Ruston
Alexandria
LONI
Lake Charles
Baton Rouge
Lafayette
New Orleans