

Git, Gnuplot and Vim Tutorial

Oliver Thomas

Quantum Engineering CDT
University of Bristol

September 2, 2018

Overview

Git, Gnuplot
and Vim
Tutorial

Oliver Thomas

- Version control using Git
- Plotting using Gnuplot
- Text editing using Vim

Why you should use version control

- Does this seem familiar?

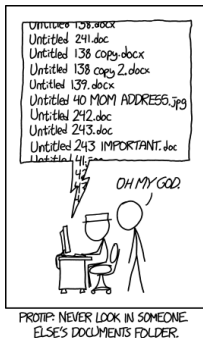


Figure: Bad version control¹

¹<https://xkcd.com/1459/>

What is Git?

- Git is one of most used version control software in the world
- Git is cross-platform and easy to use²

²<https://try.github.io/levels/1/challenges/1>

What is GitHub?

- Github is a cloud service for git which lets you store your repository online
- Why would you store your repository online?
 - Working remotely
 - Collaborative work
 - Hard drive failure!

Making a repository

- You can do this online on the Github website ³
- Create a new repository
- Then click clone to get the url, open git on your computer and type:
`git clone url`

³<https://github.com/>

Making a repository

- Go to the folder and right click git with bash
- You are now able to use bash for the rest of the talk!

Basic Git commands

- There are four⁴ important commands you will need for git:
- `git pull`
- `git add *`
- `git commit -a`
- `git push`

⁴I cheat here and write a bash script which does these in order so I only have to run a single command.

Adding your first commit

Every repository should contain a readme, make one now then run:

- `git add *`
- `git commit -a`
- `git push`

Or use the windows GUI version and commit them to your repository.

Branching

- Branching is useful, it lets you test something out separately to the main branch.
- To make a new branch called test
`git branch test`
- You can check all of the current branches and which branch you are on with
`git branch`

Branching

- To switch to the test branch type:
`git checkout test`

Adding Collaborators

- Go to a repository and on the settings tab click collaborators, you can then search using a github username

Advanced Git commands

- One of the great things about Git is that you can get by with just the four (main) commands mentioned earlier.
- The git man page is very useful, especially,
`man gittutorial`
`man giteveryday`
- `giteveryday` is a super useful collection of the 20 commands you will need regularly.

Gnuplot

Git, Gnuplot
and Vim
Tutorial

Oliver Thomas

- Gnuplot is popular, multi-platform and standard software on computing clusters⁵
- <https://sourceforge.net/projects/gnuplot/files/gnuplot/5.2.4>

⁵standard on most of the popular linux distributions

Example 0 Quick plotting

- Go to the src folder
- open gnuplot and type
`plot 'data.txt'`

Example 0 Quick plotting

Git, Gnuplot
and Vim
Tutorial

Oliver Thomas

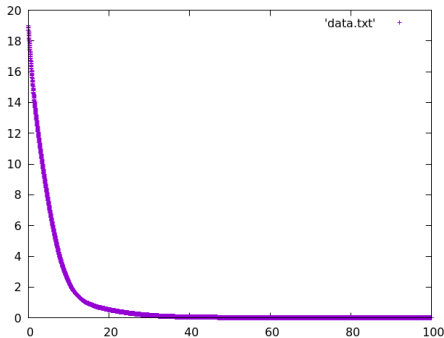


Figure: Plotting data

Lets you very quickly see what the data is doing

Example 1 Plotting functions

- Go to the src folder
- open gnuplot and type
load 'ex1_basic.p'

```
set title "ex1 Simple Plots"  
set xlabel "x"  
set ylabel "y"
```

```
plot [-10:10] sin(x), \  
      atan(x)
```

Example 2 Saving plots

- Go to the src folder
- open gnuplot and type
load 'ex2_saving.p'

```
set terminal pngcairo  
set output 'ex2.png'
```

```
set title "ex2 Saving Simple Plots"  
set xlabel "x"  
set ylabel "y"
```

```
plot [-10:10] sin(x), \  
      atan(x)
```

Example 2 Plotting functions

Git, Gnuplot
and Vim
Tutorial

Oliver Thomas

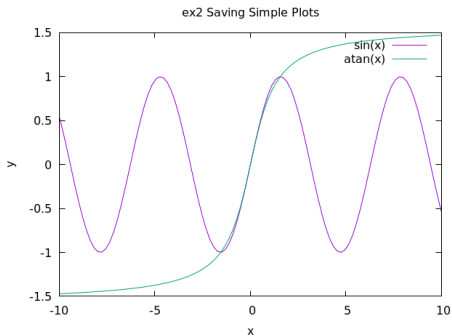


Figure: Plotting functions

Produces a png

Example 3 Bar charts

- open gnuplot and type
load 'ex3_barchart.p'

```
set boxwidth 1.0
```

```
set style fill solid
```

```
plot "schmidtout.dat" with boxes
```

Example 3 Bar charts

Git, Gnuplot
and Vim
Tutorial

Oliver Thomas

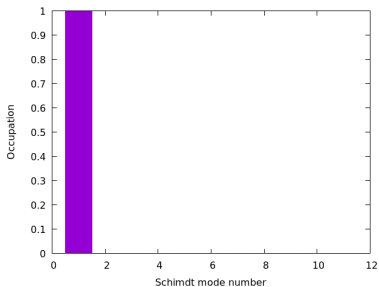


Figure: Plotting a bar chart

Example 4 Subplots

- open gnuplot
 - load 'ex4_multiplot.p'

```
set multiplot layout 1,2

set yrange [-0.3:0.35]
set xrange [0:100]

set key box opaque

set style line 1 lw 3 lc 1
set style line 2 lw 3 lc 2

plot 'signalfreq1.dat' using 2:3 with lines ls 1
plot 'idlerfreq1.dat' using 2:3 with lines ls 2
```

Example 4 Subplots

Git, Gnuplot
and Vim
Tutorial

Oliver Thomas

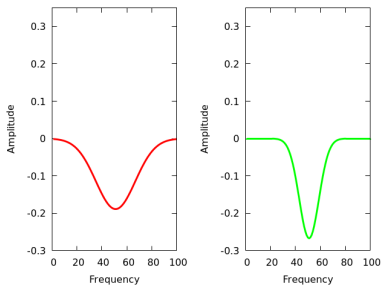


Figure: Plotting multiple subplots

Example 5 Surface plots

- open gnuplot
load 'ex5_splot.p

```
set terminal pngcairo
```

```
set output "ex5.png"
```

```
unset key
```

```
set hidden3d
```

```
set palette model CMY rgbformulae 7,5,15
```

```
set xlabel "x axis"
```

```
set ylabel "y axis"
```

```
set zlabel "z axis"
```

```
splot 'fplotw1w2.dat' using 1:2:3 with linespoints
```

```
↪ palette
```


Example 5 Surface plots

Git, Gnuplot
and Vim
Tutorial

Oliver Thomas

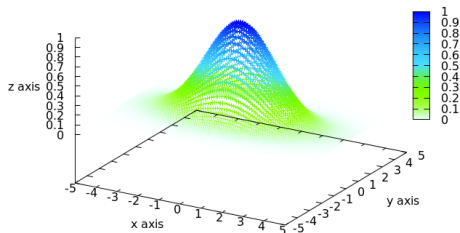


Figure: Plotting 3D data

Gnuplot summary & features

- The documentation is very good, there will be an example of whatever you want to do somewhere
- You can set *pointstyle*, *linestyle*, and colours
- Very easy to generate quick plots
- Scripts makes it easy to generate nice figures
- You can make GIFs

A brief note on text editors: Vim

- Vim is a powerful cross-platform text editor, released in 1991 and is still regarded as one of the most popular editors ⁶.
- Flexible with thousands of plugins available e.g. I use Vim to compile latex documents, this presentation was written in Vim.
- Computing clusters normally only have CLI so if you are running high performance code you will need to be familiar with Vim, Emacs or Nano.
- Overleaf supports Vim keybindings
- You can feel like a Hacker.

⁶Along with Emacs

Vim commands

- The most important thing to remember is that Vim has two main modes, *NORMAL*, ESC and *INSERT*, i
- All commands are run from *NORMAL* mode using :
- to quit use, ESC:q (meaning go to *NORMAL* mode, : means command and q is quit without saving)
- to save and quit use, ESC:wq (w stands for write)

In case everything goes wrong, :q! is force-quit without saving

Vim commands continued

All of these commands are case-sensitive and must be run in *NORMAL* mode not *INSERT*

`v` puts you in visual mode, useful for highlighting a block of text to copy or cut and paste

- `y` -yank (copy), `yy` -yank (copy) whole line
- `d` -delete (cut), `dd` -delete (cut) whole line
- `p` -paste after cursor, `P` -paste before cursor
- `x` -delete character
- `u` -undo
- `CTRL R` -redo

Movement commands

- a -append at the end of the next word, A -append at the end of the line
- o -open line below in *INSERT* mode, O -opens line above in *INSERT* mode
- 0 -go to start of line
- \$ -go to end of line
- { -go to previous paragraph
- } -go to next paragraph

Searching and editing in Vim

Searching

- `fx` -find next occurrence of `x` in text,
e.g. `fb` finds the next letter `b` in a line
- `/x` -search the whole document for `x`,
e.g. `/b` finds all letter `bs`

use `n` to go to next occurrence, `N` to go to previous.

Editing


- `r` -replace character,
e.g. `ra` replaces character with `a`
- `~` -Changes the CASE of character,
e.g. `~` when the cursor is over `a` will change it to a capital,
`A`
can be used with *VISUAL* mode to block capitalise or
block lower-case

Using SED in Vim

Sed stands for Stream EDitor and can be used directly from vim ⁷

Probably the most regularly used sed commands you will need are,

- `:%s/foo/bar/g` -replaces all instances of foo with bar globally
- `:4,31s/foo/bar/g` -replace instances of foo with bar in lines 4-31

⁷http://vim.wikia.com/wiki/Search_and_replace 

Vim summary

- You should try Vim, it is available in overleaf
- It takes some getting used to, but I and many others think it is worth it.
- Cross-platform and powerful
- Very good documentation

Conclusion

- You should use version control
- I recomend gnuplot as it is easy to use
- Vim is a fantastic editor but does require a small amount of effort to learn

Thanks for listening!



Figure: If it all goes wrong ...⁸

¹<https://xkcd.com/1597/>