A summary
Latex
Gnuplot
Git and Probabiriry

by Praphasri Bidasak,

May 20, 2016

Contents

A simple guide to LaTeX

1.The basic layout of a LaTeX

Creating documents with LaTeX is simple. In contrast to Word, you start off with a plain text file (.tex file) which contains LaTeX code and the actual content. The control statements tell LaTeX how your content should be formatted. Before the settings become apparent, the .tex file has to be compiled (translated) into a .pdf file. A basic example document can be created with the following code:

\ documentclass {article}
\ begin{document}
\ end{document}

Table. 1

Once translated this code into a PDF document \ documentclass {article}
This specifies what sort of document you intend to write. After that, add commands to influence the style of the whole document, or load packages that add new features to the LATEX system. To load such a package you use the command

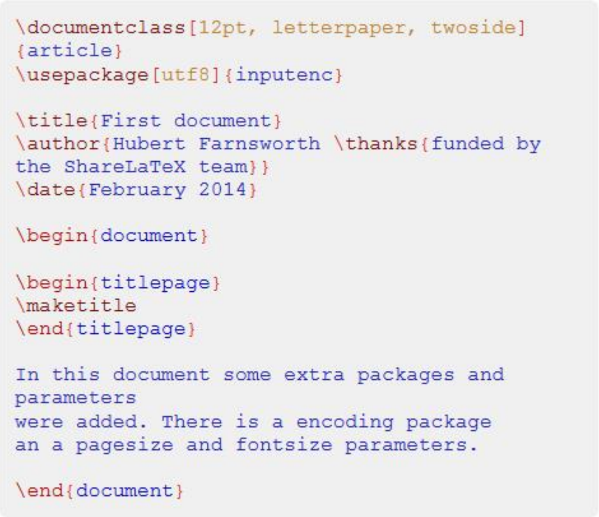
\ begin{document}
Now you enter the text mixed with some useful LATEX commands. At the end of the document you add the

\ end{document}
It is invaluable for debugging. Sometimes some complicated bit of TeX code that's going badly wrong but can't figure out exactly where the error is (for example, beamer tends to report errors at the end of the frame - since that's when it encounters them - but the actual error is buried deep within it). Then putting \ end {document}

2.Title page

Displaying the title of document

To display the title of your document you have to declare its components in the preamble and then use some additional code:



```
\documentclass[12pt, letterpaper, twoside]
{article}
\usepackage[utf8]{inputenc}

\title{First document}
\author{Hubert Farnsworth \thanks{funded by
the ShareLaTeX team}}
\date{February 2014}

\begin{document}

\begin{titlepage}
\maketitle
\end{titlepage}

In this document some extra packages and
parameters
were added. There is a encoding package
an a pagesize and fontsize parameters.

\end{document}
```

Figure.1

There is a block with three lines in the preamble that define what information is going to be included in the title page.
\ title{First document}
This is the title.

\ author{Hubert Farnsworth}
Here you put the name of the Author(s) and, as a optional parameter, you can add the next command:

\ thanks {funded by the ShareLaTeX team} This can be added after the name of the autor, inside the braces of the title command. It will add a superscript and a footnote with the text inside the braces. Useful if you need to thank an institution in your article.

\ date{February 2014} You can enter the date manually or use the command May 20, 2016so the date will be updated automatically at the time you compile your document.
Once you have that in the preamble now in the body of your document you can use the next commands for the information to be printed.

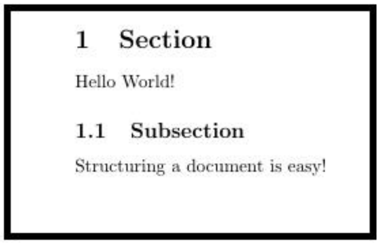
\ begin{titlepage} \ end {titlepage} This declares an environment, a block of code with a specific behaviour depending on its type. In this case whatever you include in this titlepage environment will appear in the first page of your document.

\ maketitle
This command will print the title, the author and the date in the format shown in the example. If it's not enclosed in a titlepage environment, it will be shown at the beginning of the document, above the first line.

3. Structuring document - Adding paragraphs and sections

Ccreated a very basic document in the previous lesson, but when writing a paper, it's necessary to structure the content into logic units. To achieve this, LaTeX offers us commands to generate section headings and number them automatically. The commands to create section headings are straightforward

Example output for section and subsection The section commands are numbered and will appear in the table of contents of your document.
Paragraphs aren't numbered and won't show in the table of contents. Here an example output using sections: In order to get this output, we just have to add a few lines to our program



1 Section

Hello World!

1.1 Subsection

Structuring a document is easy!

Figure.2

Struturing a document LaTeX uses the commands

LaTeX uses the commands \ section \ subsection and textbackslash subsubsection to define sections in document
The sections will have successive numbers and appear in the table of contents . Paragraphs are not numbered and thus don't appear in the table of contents

\ begin{document}
\ maketitle
\ pagenumbering{gobble}
\ newpage
\ pagenumbering{arabic}
\ pagenumbering section{Section}
Hello World!
\ pagenumbering subsection{Subsection}
Structuring a document is easy!
\ pagenumbering pagenumbering
\ end{document}

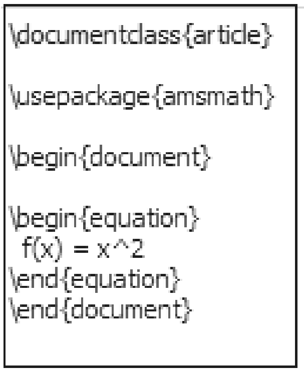
Table. 2

4. Use packages in LaTeX to add more functions.

There are countless packages, all for different purposes in my tutorials I will explain some of the most useful. To typeset math, LaTeX offers (among others) an environment called equation. Everything inside this environment will be printed in math mode, a special typesetting environment for math. LaTeX also takes care of equation numbers for us:

Including a package

The automatic numbering is a useful feature, but sometimes it's necessary to remove them for auxiliary calculations. LaTeX doesn't allow this by default, now we want to include a package that :



```
\documentclass{article}

\usepackage{amsmath}

\begin{document}

\begin{equation}
  f(x) = x^2
\end{equation}
\end{document}
```

Table.3

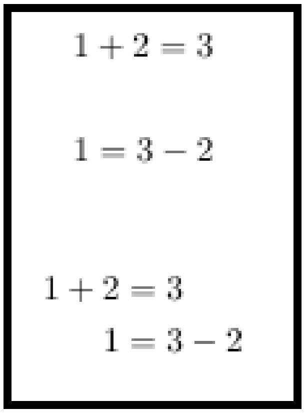
Output as before, only the equation number is removed: $f(x) = x^2$

5. Using inline math - embed formulas in your text

The equation and align environment

The most useful math envorinments are the equation environment for typesetting single equationsand the align environment for multiple equations and automatic alignment

The align environment will align the equations at the ampersand &. Single equations have to be seperated by a linebreak There is no alignment when using the simple equation environment. Furthermore it is not even possible to enter two equations in that environment, it will result in a compilation error. The asterisk (e.g. equation*) only indicates, that I don't want the equations to be numbered



$$1 + 2 = 3$$

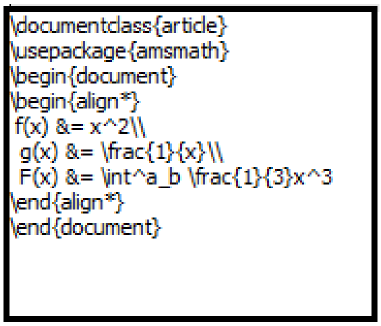
$$1 = 3 - 2$$

$$\begin{aligned} 1 + 2 &= 3 \\ 1 &= 3 - 2 \end{aligned}$$

Figure.3

Fractions and more

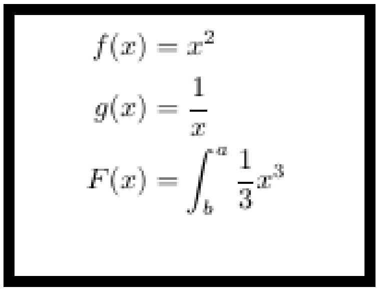
LaTeX is capable of displaying any mathematical notation. It's possible to typeset integrals, fractions and more. Every command has a specific syntax to use. I will demonstrate some of the most common LaTeX math features:



```
\documentclass{article}
\usepackage{amsmath}
\begin{document}
\begin{align*}
 f(x) &= x^2\\
  g(x) &= \frac{1}{x}\\
  F(x) &= \int^a_b \frac{1}{3}x^3
\end{align*}
\end{document}
```

Table.4

Output:

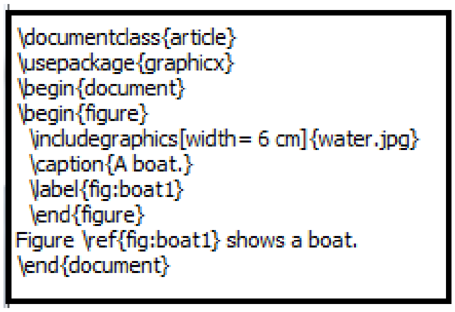


$$f(x) = x^2$$
$$g(x) = \frac{1}{x}$$
$$F(x) = \int_b^a \frac{1}{3}x^3$$

Figuer.4

6. Adding figures in LaTeX

From time to time, it's necessary to add pictures to your documents. Using LaTeX all pictures will be indexed automatically and tagged with successive numbers when using the figure environment and the graphicx package.



```
\documentclass{article}
\usepackage{graphicx}
\begin{document}
\begin{figure}
  \includegraphics[width= 6 cm]{water.jpg}
  \caption{A boat.}
  \label{fig:boat1}
  \end{figure}
Figure \ref{fig:boat1} shows a boat.
\end{document}
```

Table.5

The code above will create the following pdf:

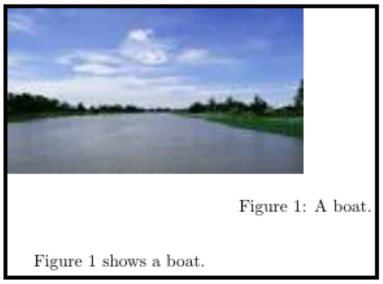


Figure 1: A boat.

Figure 1 shows a boat.

Figure.5

The figure environment takes care of the numbering and positioning of the image within the document. In order to include a figure, you must use the — includegraphics command. It takes the image width as an option in brackets and the path to your image file.can see, I put — linewidth into the brackets, which means the picture will be scaled to fit the width of the document. As a result smaller pictures are upscaled and larger pictures downscaled respectively. As I mentioned before the brackets contain the path to the image. In this case the image is stored in the same directory as my .tex file, so I simply put boat.jpg here to include it. For large documents, you probably want to store image files in a different folder, say we created a folder images, then we would simply write images/boat.jpg into the braces. In the next command we set a — caption, which is the text shown below the image and a hich is invisible, but useful if we want to refer to our figure in our document. You can use the **??**ommand to refer to the figure (marked by label) in your text and it will then be replaced by the correct number. LaTeX is smart enough to retrieve the correct numbers for all your images automatically. Note that you will need to include the graphicx package in order to use this code.

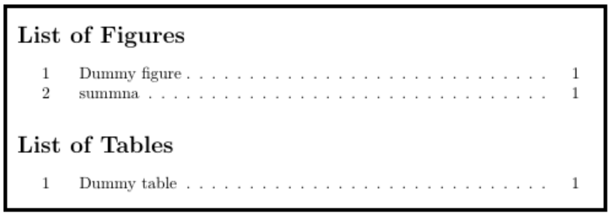
7. Generate a table of contents in LaTeX

Generating a table of contents can be done with a few simple commands. LaTeX will use the section headings to create the table of contents and there are commands to create a list of figures and a list of tables as well
The generation of a list of figures and tables works the same way. I added a dummy figure and table and put the lists in the appendix of my document:

```
\ begin{document}
\ begin{figure}
\ caption{Dummy figure}
\ end {figure}
\ begin{table}
\ end{table}
\ begin{appendix}
\ listoffigures
\ listoftables
\ end{appendix}
\ end{document}
```

Table. 6

Autogenerate a table of content using



List of Figures

1	Dummy figure	1
2	summna	1

List of Tables

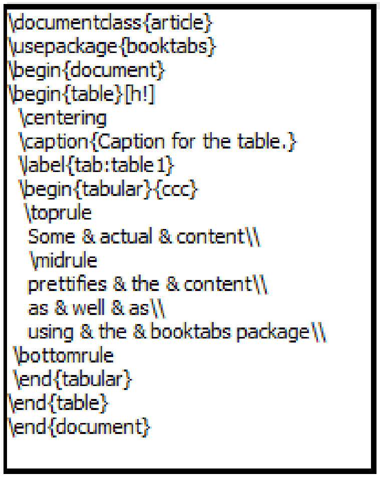
1	Dummy table	1

Figure.6

Autogenerate a table of content using —tableofcontents
Create lists of your figures and tables with — listoffigures and — listoftables
Always compile twice to see the changes

8.Tables in LaTeX

Try out all the features of LaTeX tables yourself. There’s a huge list of code examples for tables somewhere on Wikibooks, which vast amount is almost overwhelming. As I mentioned before, it’s usually faster to use a tool like excel to format your content and then let the pgfplots package autogenerate a table .



```
\documentclass{article}
\usepackage{booktabs}
\begin{document}
\begin{table}[h!]
  \centering
  \caption{Caption for the table.}
  \label{tab:table1}
  \begin{tabular}{ccc}
   \toprule
   Some & actual & content\\
    \midrule
   prettifies & the & content\\
   as & well & as\\
   using & the & booktabs package\\
 \bottomrule
 \end{tabular}
\end{table}
\end{document}
```

Table. 7

There are two disadvantages of writing tablesWhile it works for small tables similar to the one in our example, it can take a long time to enter a large amount of data by hand. Most of the time the data will be collected in form of a spreadsheet and we don't want to enter the data twice. Furthermore once put into LaTeX tables, the data can not be plotted anymore and is not in a useful form in generalBut let's stick with writing tables by hand for now. The table above does not look very appealing. For this reason, we're now going to use the booktabs package to change this pitiful situation. The table we're going to get will be much nicer than before. Have a look at this:

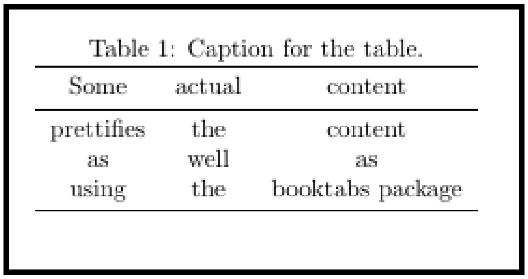


Table 1: Caption for the table.

Some	actual	content
prettifies	the	content
as	well	as
using	the	booktabs package

Figure.7

The code requires just a few changes. used the following code to get this result:
LaTeX offers the table and tabular environment for table creation
The table environment acts like a wrapper for the tabular similar to the figure environment
Alignment and vertical separators are passed as an argument to the tabular environment (begintabular)
It's possible to align the content left (l), centered (c) and right (r), where the number of alignment operators has to match the desired number of columns
The columns can be seperated by adding — in between the alignment operators

Rows can be seperated using the — hline command and columns using the ampersand symbol

Gnuplot Short Course

1.Introduction

Gnuplot was originally developed by Colin Kelley and Thomas Williams in 1986 to plot functions and data
les on a variety of terminals. In 1988 and 1989 I created an alternate version, known as GnuTEX, that supported a new —terminal type called latex, so gnuplot would output LATEX code. The plot could then be included in a LATEX document. I added a number of embellishments, supported only by the latex terminal, allowing the user to produce publication-quality plots.

Gnuplot is a powerful frccware program for plotting both 2D and 3D data Gnuplot will run under a variety of environments including Linux L,IRLX, Solaris, Windows. and DOS.
Should use this document in conjunction with ”gnuplotcourse.tar.gz, an archive file that contains all of the scripts and data in this document (as well as the document itsilf) Form MI net machines.

1.Basic 2D Plots

To start guplot (make sure you are in the ”data” directory first)’ simply type unixSay you have a single column of data in a file called ”1ch.dat” that you woule liked to plot
28.062
52.172
55.703
64.281
43.438, etc.
This is simple to plot
gnuplot will autoscale axes to include all the data. Bydefault, gnuplot will plot the data using points. The file ’1ch.dat’ must be in the current working directory that you ran gnuplot from, else you will have to specify the path to the file .../ and backslash are allowed
gnuplot >plot ’1ch.dat’

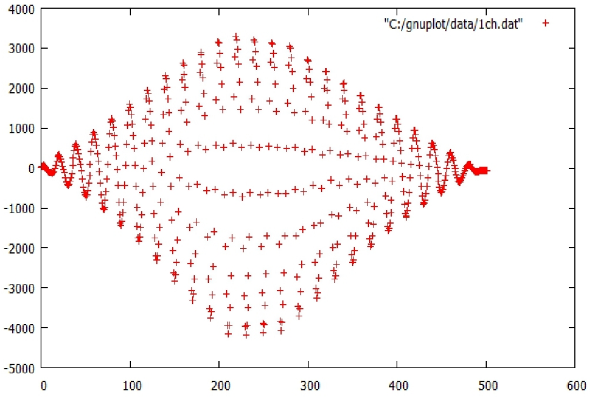


Figure.8

Let’s plot the data using lines, add more divisions to the xaxis, and rescale using set xrange.
gnuplot >set data style lines
gnuplot >set xtics 0,50,1000
gnuplot >set xrange[0:500]
gnuplot >plot ’../ data /textslash 1ch.dat’

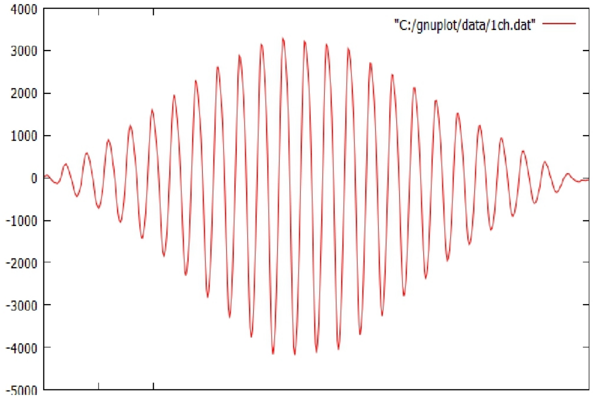


Figure.9

Ther are several choices for data style. including lines, points, linespoints, or dots. The set xtics command takes three arguments,
<start >, <increment >, <end >,and set xranger, two, [<start >: <end >] . But what if you have multicolumn data? Is it possible to plot one column data? Of course! Gere we plot the second column of data from the file ”3ch.dat”

```
gnuplot >plot ’3ch.dat’ using 2
```

The using command specifics a certain column of data, or cross-plots between coluns. Here we’ll plot column l on the x-axis aand column 2 on the y.

```
gnuplot >plot ’3ch.dat’ using 1:2
```

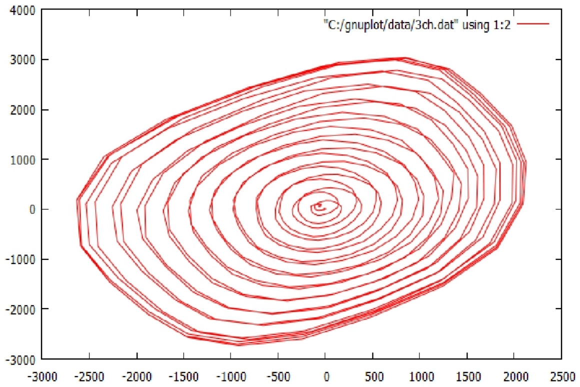


Figure.10

Operators, Constants, and Functions

Gnuplot not only reads data fro files, it can also plot analytical functions. Toward thes cnd, gnuplot provides the usual list of operatiors +, -,*,/, **, etc, and functions,sin(), cos(), log(), exp(), etc, A summary of operators and functions is available at the end of this text. Lext. Let’s plot a simple function based on the above (note that ** is exponentiation a la FDRTRAN).

```
gnuplot >set xrange [0:250]
gnuplot >set(x) * (x**2)
```

Gnuplot assumes that x is the independent variable.

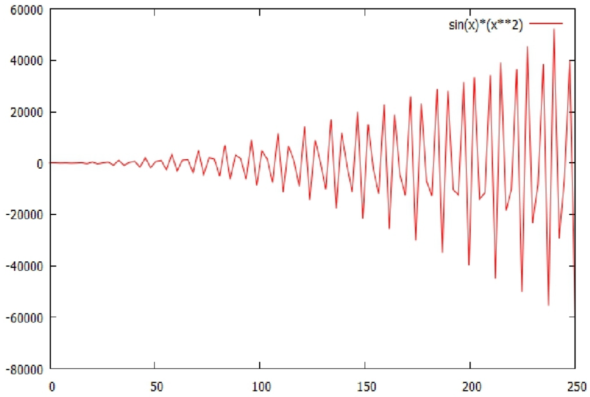


Figure.11

The graph does not look like a sooth function at all. Gnuplot evaluatcs the function at certain points only - the sample arte is too low for this function. Th change this, change

```
gnuplot >set samples 1000
gnuplot >replot
```

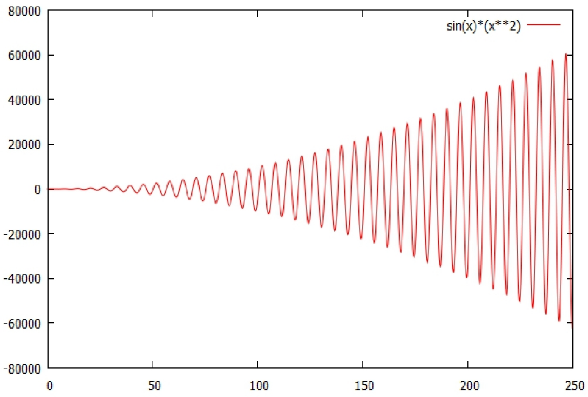


Figure.12

2. Gnuplot and LaTex

If want to include gnuplot output in a Latex documemt, there are scveral options available. One gnuplot terminal type is latex. In this case, the output is raw Latex code that can be included directly int a docuent. This has the advantage that LaTex is commands can be used in titles, etc.

gnuplot >set xrange ’ϵ (mm/mm)’
gnuplot >set ylabel ’σ (MPa)’
Gnuplot >output ’plot.tex’
gnuplot >set terminal latex
gnuplot >load ’script1.gp’
gnuplot >set term x11 ; replot

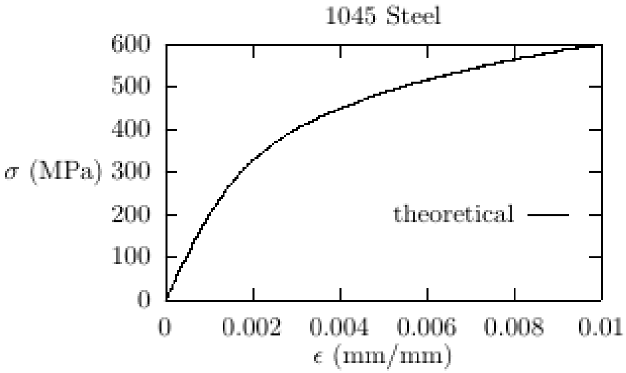
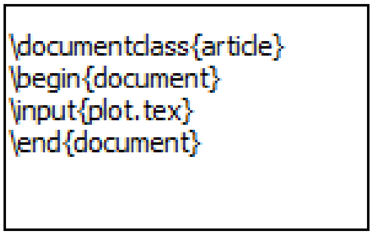


Figure.13

The simplest way to include this in a Latex cocment is



```
\documentclass{article}
\begin{document}
\input{plot.tex}
\end{document}
```

Table.8

Git Short Course