Writing a Thesis with LaTeX

Bhavesh Shrimali

Department of Civil and Environmental Engineering [1]

March 3, 2017

Abstract

The following work is intended to demonstrate typesetting using LaTeX. For the purposes of this work, and this work only, it shall be assumed that the reader has no prior knowledge of LaTeX. However at no point during this work, or the publications that have resulted out of this work, it would be implied that the author(s) is(are) experts in the field. Additionally at all points, whether intentional or unintentional, snide references/remarks to MS-Word shall be made. We hope not to offend any MS-Word fans, but if you get offended we can't help!

LaTeX is AWESOME!!

Contents

List of Figures

List of Tables

Chapter 1

LaTeX Constructs: Basics

Having stated the important notions relevant to writing a LaTeX script, we are well positioned to actually start writing a document. The reference for the same is [1]. Note that although this document is written in a way to mimic a thesis, but it is equally valid in context of writing a research article. It is imperative that we tell LaTeX kernel the logical and semantic structure of our text. To this end, we consider the most basic work-flow of a LaTeX script.

1.1 Document Class

The first thing that you want to tell the kernel is the format of the document you wish to generate. For this document, we specify it to be a "`report`". The syntax of the command is as follows :

```
                 \documentclass[Substyle]{sty}
for e.g.         \documentclass[11pt]{report}
```

Some important classes (with options) rendered by the `documentclass` command are as follows:

- `[11pt]article`, `[11pt, twocolumn]article`

- `book`

- `report`

1.2 Preamble

The space between `documentclass` and `begin{document}` is called the *Preamble*. It is used to include any packages that might be useful in putting together the document and/or defining any commands to ease our task(s). For a long document, usually, it is beneficial to declare preamble in a separate file, `commands.tex` in this case, to facilitate easy navigation. A sample of a preamble is as follows:

```
            \usepackage{epstopdf}
            \usepackage{amsmath}
            \usepackage{epsf}
            \usepackage{amsfonts}
            \usepackage{amssymb}
            \usepackage{color}
            \usepackage{xspace}
            \usepackage{hyperref}
            \newcommand{\latex}{\LaTeX\xspace}
            \newcommand{\tex}{\TeX\xspace}
```

1.3 Title and Author

It is sometimes beneficial to declare the `title` and `author` or the document before actually proceeding to write the body. A sample layout is as follows:

```
\title{Writing a Thesis in \latex}
\author{Bhavesh Shrimali}
```

1.4 Document: Main Body

Once you have declared everything in the preamble, and assigned the `title` and `author` we proceed with writing the main body of the document. A sample of the main body is:

```
\begin{document}
        {Main Body}
\end{document}
```

1.4.1 Section

Sections and Subsections : The main body of the document is usually divided in to `section`, `subsection`, `subsubsection` etc. Each of these may further contain text, equations, paragraphs, figures, tables, code-snippets etc. We will now try to go over these in the listed order. A sample of how a section would look like is as follows:

```
\section{DocumentClass}
The first thing that you want to tell the kernel
is the format of the document you wish to generate.
For this document, we specify it to be a
"\texttt{report}". The syntax of the command is as
follows : \\
```

A few important pointers while writing the main body:

- Keep a count of the `brackets` "{", " [", " (" . Each corresponding bracket should be closed with its dual.

- You can include include new lines in the script, however it does not introduce new lines in your document. In order to introduce a newline you have to use a `DoubleBackSlash`.

- However when in Math mode, such as `align` or `equation`, avoid including new lines as these may result in compilation issues.

1.4.2 Subsection

A sample of how a subsection looks like, is as follows. Note that the syntax is exactly the same as a section. :

```
\subsection{Section}
The main body of the document is usually divided
in to \tet{section}, \tet{subsection},
\tet{subsubsection} etc. Each of these may further
contain text, equations, paragraphs, figures, tables,
code-snippets etc. We will now try to go over these in
the listed order. A sample of how a section would look
like is as follows: \\
\begin{lstlisting}[frame=single]
\section{DocumentClass}
The first thing that you want to tell the kernel
is the format of the document you wish to generate.
For this document, we specify it to be a
"\texttt{report}". The syntax of the command is as
follows : \\
\end {lstlisting}
```

1.4.3 Paragraphs

In line with the previous constructs, paragraphs work in a similar way. A sample layout is as shown below:

```
\subsection{Section}
\paragraph{Sections and Subsections}:
The main body of the document is usually divided
into \tet{section}, \tet{subsection},    \tet{subsubsection}
etc. Each of these may further contain text, equations, paragraphs,
figures, tables, code-snippets etc.
We will now try to go over these in the listed order.
A sample of how a section would look like is as follows:
```

Some general remarks about Paragraphs in particular, and sections/subsections in general:

- Paragraphs, and/or the starting of a new section are generally indented.

- In order to remove the indent you should use the `setindent` command; note the other uses also:

```
                     \setlength{\parindent}{0}
                     \setlength{\parskip}{2cm plus4mm minus3mm}
or                   \noindent {...Paragraph...}
```

- If you want to indent an unindented paragraph use `indent` command:

```
                     \indent{}
```

For customized paragraphs we use `verbatim` command.

```
                     \begin{verbatim}
                            {Your text.... as it is .. \\ }
                     \end{verbatim}
```

1.4.4 URL and HYPERREF

For typesetting of urls/hyperlinks it is recommended that you use either of the `url` and/or `hyperref` packages. The syntax of both is as follows:

```
\url{<link >}{...text ...}
\hyperref{<link >}{...text ...}
```

1.4.5 Lists

LaTeX also allows for the typesetting of lists using a number of inbuilt structures, each of which is marked briefly below. In most cases, for thesis or articles, only `enumerate` and `itemize` are used:

```
\begin{itemize}
\item Item 1
\item Item 2
\end{itemize}


\begin{enumerate}
\item Item 1
\item Item 2
\end{enumerate}
```

Note: Additional packages such as { `enumerate *` } are useful in having inline lists in your document. The syntax for these is exactly same as described above for `itemize`.

Chapter 2

Graphics

The following chapter is intended to provide a brief overview of the graphics' packages in LaTeX and their usage in appropriate environments.

2.1 Floats

Floats are basically things that cannot be broken over a page. Some common examples of floats can be, `table`, `figure` etc.

2.1.1 Figure

The syntax for `figure` is given below. The `subfigure` package, with the `subfigure`, works similarly:
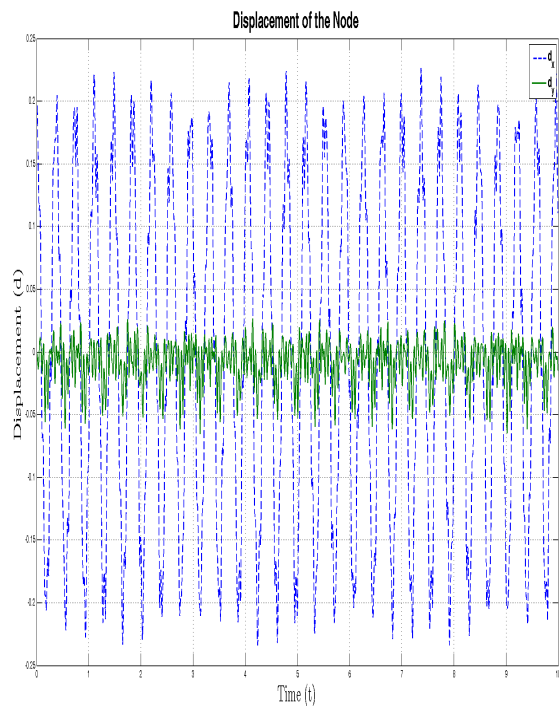
```
\begin{figure}[placement specifier]
        <figure details>
\end{figure}
```
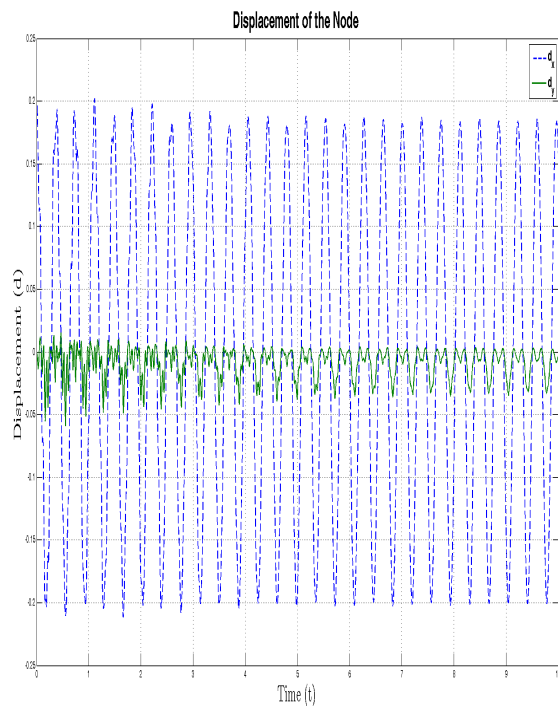
For specifiers we, most commonly use:

- `h` - "here", places figure almost at the same point (Not SPOT!)

- `t` - "top", places figure at the top of the current page

- `b` - "bottom", places the figure at the bottom of the current page.

- `p` - "page", float-only page

- `!` - "good", determines the best possible location for the float.

A sample syntax is as follows:

```
\begin{figure}[htbp]
\centering
\begin{subfigure}{.5\textwidth}
        \centering
        \includegraphics[width=3.5in,height = 4in]{dis4ci}
\label{fig:sub1}
\end{subfigure}%
\begin{subfigure}{.5\textwidth}
\centering
\includegraphics[width=3.5in,height = 4in]{dis4cii}
\caption{$\Delta t$ = 0.005, $\alpha = -1/3$}
\label{fig:sub2}
```

(a) $\Delta t = 0.005$, $\alpha = 0$

(b) $\Delta t = 0.005$, $\alpha = -1/3$

Figure 2.1: Displacement-Time Plots for Shear Loading at Node (3)

```
\end{subfigure}
\label{Disp-Shear}
\end{figure}
```

The following packages are also pre-installed in the preamble to facilitate the rendering of the above figure.

- `graphicx` and setting the `graphicspath` `{{}}`

- `caption`

Other variants for figures are also possible, for e.g.,

- `sidecap` package with the command `SCfigure`

- `subcaption`

2.1.2 Table

LaTeX can handle tables efficiently but the syntax can be cumbersome, particularly when large tables are being input. Note, that for a scientific report this may not be the case (sign of bad report if present!) but in general there are several ways to workaround, one of them being `excel2latex` (not efficient without Macros!) Although the floats in LaTeX don't seem to be well placed in terms of the default location, it is

Table 2.1: Custom Caption

Iter	N = 10	N = 50
1	3.614743	3.839848
2	0.089508	0.422621
3	0.001458	0.036089
4	3.99E-05	0.00397
5	6.8E-07	0.000346
6	1.79E-08	3.74E-05
7	3.17E-10	3.32E-06
8	0	3.53E-07
9		3.19E-08
10		3.34E-09
11		3.05E-10
12		0

often easy to manipulate them using macros. (i.e. you can design your own `customtable` environment)

2.1.3 Referencing

The command `listoffigures` takes care of putting together references to whenever the `figure` command is called in the document, referencing the page number and the name of the figure, at the beginning of the report. Similarly the `listoftables` command takes care of putting together all the references to the `table` command in the document, and automatically places them in the start of the document.

2.1.4 Additional Notes

Special Cases: There are additional environments to take into account special cases for floats. (for e.g. `wrapfig`).

Chapter 3

Math

The most important and perhaps the most versatile component of LaTeX is typesetting math. A number of packages such as `amsmath`, `amssymb`, `mathtools` enrich the math environment with a number of additional choices. (such as those required for special characters, in latin, greek alphabets etc.)

3.1 Inline Math

The inline math environment allows for typing in math in-place, i.e. withing your paragraph along with regular text. The syntax for this is as follows:

```
regular text ....... $type your math here $ .... regular text
```

3.2 Out of line Math

For a number of situations it is required to type in math, in the form of equations, relations etc. for which the inline math environment is unsuitable. There are a number of ways to do this:

3.2.1 Using the Square Brackets

The syntax for this is as follows:

```
\[    \alpha^2 + \beta^2 = \frac{\gamma^2}{4}    \]
```

3.2.2 Using the Equation environment

```
\begin{equation}
        type your equation here ...
\end{equation}
```

3.2.3 Using the align environment

```
\begin{align}
        type your equation here ...
\end{align}
```

Note that the `align` and `equation` environments automatically start numbering the equation, for every newline character that they encounter. For equations which need to be skipped for numbering, a has to be prefixed before the starting of the equation. If there is a good number of equations whose numbering has to be skipped, prefer using `align *` instead. For numbering a group of equations together as one, use the following

```
\begin{align}
    \begin{split}

    \end{split}
\label{ }
\end{align}
```

Note the usage of `label` command. This assigns the set of equations within `split` a single equation label. The next time you want to reference the equation, just use `ref` command with the `label`. Labeling and referencing in LaTeX follows a common convention across all the environments.

3.3 Symbols

The most common list of symbols is available `https://en.wikibooks.org/wiki/LaTeX/Mathematics`. These need not be memorized as they are available for reference in a number of standard `CTAN` documentation. Most common examples of the ones used are `int` and `sum` commands.

3.4 Matrix Environments

A number of occasions in reports demand the user to input (show) matrices and vectors. LaTeX allows for a customized input of these using the `bmatrix`, `Bmatrix` and `vmatrix` environments. The syntax for these is as follows:

```
\begin{bmatrix}
    a & b \\
    c & d
\end{bmatrix}
```

Note that the ampersand character (&) stands for the alignment. It is used in the align environment, in particular, to align the equations with respect to a particular character.(separate for each line)

Chapter 4

Referencing and BiBTeX

In all reports, research articles and theses, the list of references is a must have. LaTeX allows for incorporating this in a multiple ways. BIBTeX (or lately BibLaTeX) is a robust tool to incorporate this.

4.1 Background

BIBTeX is an auxiliary tool that works to separate bibliography information, from its corresponding presentation, in a thorough and consistent manner. A BIBTeX database is stored as a *.bib file (created using any text editor) and is referenced in the main TeX document. BIBTeX has been succeeded by BIBLATeX which follows the same syntax. For this workshop we shall mainly be focusing on BIBTeX, and its usage with LaTeX. The standard syntax followed for a *.bib file is as follows:

```
@article{thesis,
author    = "Shrimali, Bhavesh",
title     = "Writing a thesis with LaTeX",
journal   = "ACI Workshop",
year      = "2017",
}
```

BIBTeX knows almost all of the document types that occur in literature, i.e. reports, books, articles. Hence it allows you to completely forget about the formatting as well as the numbering of the references. The syntax for citing a particular document is :

```
\cite{}
```

There are additional packages such as `natbib` which allow for more flexibility in styling references. `natbib` allows for a quick and smooth transition between the *Harvard*(author-year) and *numeric* styles. For further reference it is suggested that you look at the references. :)

Note If you are using BIBTeX, the order of compiling your files is very important. For instance. Generally the order is as follows:

- `latex` or `pdflatex`
- `bibtex .aux` file(s)
- `latex` or `pdflatex`
- `latex` or `pdflatex`

Depending on the number of cross-references LaTeX should be able to list all the cross-references with two successive compilations.

Bibliography

[1] Leslie Lamport. *I1 (\ LaTeX)—A Document*, volume 410. pub-AW, 1985.