LaTeX Starter Pack

190050087-190050090

August 30, 2020

Contents

1 Introduction

LaTeX (most popularly pronounced lay-tech; sometimes laa-tech) is an incredibly efficient office tool to typeset professional looking documents and reports. You will certainly find it useful to write assignments, format your resume, and more generally, to make everything you do look cooler.

LaTeX like HTML, is a **markup language.** Is's part of the TeX typesetting system created by the immortal Donald Knuth. The presentation of the content depends on the properties of the tags it is wrapped in. For more involved typesetting purposes, thius gives it a clear edge over mainstream word processors like MS-Word: in Word, *what you see is what you get,* and getting what you want can be insanely tough.

Here's how it works: you write your markup commands in the source file, which has a `.tex` extension. You need "software", or formally, a TeX distribution, to actually typeset them into a format suitable for distribution, which is generally a pdf. The most popular distribution to install on your machine is TeX Live; MikTeX is an alternative. You could also work online with Overleaf-no installations, and a ridiculously straightforward workflow. This is ideal for smaller projects. Weigh your options here. Yes, a hyperlink!

LaTeX allows us to write complex mathematical equations without much fuss; its environments save us the hassle of organising large documents manually; with LaTeX we can showcase code and render almost any scientific illustration. LaTeX is paradise for anyone who works in STEM. Once you have experience, you can typeset assignments, papers, articles and theses with unprecedented ease.

In this assignment, we will explore and demonstrate some features that often prove themselves useful for several purposes.

In this introductory section, we have also seen how we can format text. For instance, we can make text bold, italicized, colour it, or manipulate its size, and even toggle between alignments!

CARPE DIEM!

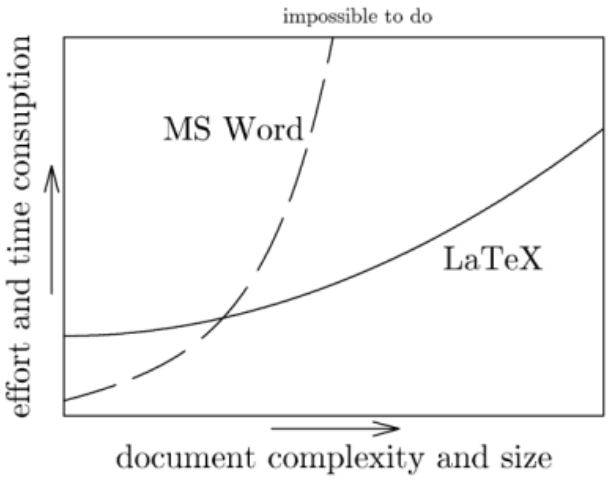

Figure 1: This graph by Marko Pinteric is spot on.

2 Basic Document Formatting

In STEM, brevity is highly valued. You want to put forth your arguments as crisply as possible. Of course, sometimes a rather long clarification may be in order[1], however, it is better to stick to the central theme and not disrupt the flow. In order to make your point, lists are often the cleanest option.

Features we demonstrate

- Making a title and table of contents
- Organising the document into sections
- Setting up the page layout
- Designing our custom header and footer
- Formatting text
- Making (nested) lists
    1. itemize
    2. enumerate
    3. description
- Footnotes
- Typesetting mathematics
- Theorem and Proof environments
- Hyperlinks and cross references within the document
- Custom environments
- Algorithms and code
- Inserting images
- Drawing tables
- Citations †

In order to make your lists appear more concise, you can specify the `itemsep` parameter as am optional argument to the environment. You will need the `enumitem` package for that.

Descriptive lists are sometimes handy:

CS 207 Discrete Structures
CS 213 Data Structures and Algorithms
CS 215 Data Analysis and Interpretation
CS 251 Software Systems Lab
CS 293 Data Structures and Algorithms Lab

The Page Layout

The paper size for this document is A4. The left and right margins are 1.2 inches each; the lower margin is 1.5 inches. The `geometry` package is very convenient to set up and manipulate these dimensions.

[1] Footnotes are a classy way to do that. Making footnotes is fairly simple in LaTeX .
† using BibTeX which automatically takes care of the bibliography formatting

3 Mathematics

Make sure you have imported the `amsmath`, `amsthm`, `amssymb`, and `esint` packages, in that order. Have a look at the tutorial to understand how they work, and what theydo. In particular the `amsthm` package makes defining ordered theorem-like environments very convinient.

Theorem 1 (Markov's Inequality)**.** *If X is a non-negative random variable and $a > 0$ then*

$$P(X \geq a) \leq \frac{E(X)}{a}$$

Corollary 2 (Chebyshev's Inequality)**.** *Let X be an integrable random variable with finite expected value μ and finite variance $\sigma^2 \neq 0$. Then for any $k \in \mathbb{R},\ k > 0$,*

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$$

Proof. Let the random variable $Y := (X - \mu)^2$. Observe that Y is a non-negative random variable; and from the definition of variance, $E(Y) = \sigma^2$, which is positive from the premise. Hence, we can appeal to Theorem 1

$$P(Y \geq k^2\sigma^2) \leq \frac{\sigma^2}{k^2\sigma^2} = \frac{1}{k^2}$$

However, $P(Y \geq k^2\sigma^2) = P(|X - \mu| \geq k\sigma)$ and we are done. □

Consider a plane $z = ax + by + c$. Usually, three points on the plane are sufficient to determine the parameters a, b, c; however, we're given a data set of several points: x and y coordinates are known perfectly, but the z coordinates are corrupted by a Gaussian noise $\mathcal{N}(0, 1)$. Here's how we solve for the maximum likelihood estimates of the parameters:

$$\begin{bmatrix} \sum_i x_i^2 & \sum_i x_i i y_i & \sum_i x_i \\ \sum_i x_i y_i & \sum_i y_i^2 & \sum_i y_i \\ \sum_i x_i & \sum_i y_i & \sum_i 1 \end{bmatrix} \begin{bmatrix} \hat{a} \\ \hat{b} \\ \hat{c} \end{bmatrix} = \begin{bmatrix} \sum_i x_i z_i \\ \sum_i y_i z_i \\ \sum_i z_i \end{bmatrix} \tag{1}$$

The above equaion (we can also refer to it as equation 1 as we created a label) is obtained by setting the partial derivatives of the log-liklihood to 0, e.g. $\partial\mathcal{L}/\partial a = 0$ etc.

Observe that we also cross referenced Theorem 1 in the proof of Corollary 2. And here we just did it again. This is how we use labels in tandem with the `hyperref` package.

Theorem 3 (De Morgan's Law)**.** $\neg(a \vee b) \iff \neg a \ \wedge\ \neg b$ *Alternately, in the context of sets, we have* $\overline{\mathrm{A} \cup \mathrm{B}} = \overline{\mathrm{A}} \cap \overline{\mathrm{B}}$.

Remark. This is closely related to the following rule in the first order propositional logic:
$\neg(\exists x.p(x)) \iff \forall x.\neg(p(x))$

We almost always want to site sources for the claims we make, because when you publish, it's pointless to keep reinventing the wheel. For instance, the Kronecker's Theorem on simultaneous Diophantine approximations is a standard but powerful result, it makes sense to refer the reader to [2, Chap. 7, Sec. 1.3, Prop. 7] for a statement and proof. This source is a book. Make sure you enter the relevant information in the appropriate fields in the BibTeX entry.

In Computer Science, most citations are journal or conference papers. A discussion on algebraic numbers is incomplete without citing the Mignotte separation bounds, given in a modestly titled conference paper [3]. We can even cite multiple sources with a single command [1, 4]. These are journal articles. If possible, it's best to also specify the volume and page numbers. Your aim shouls be to point the reader to what you're quoting as unambiguously as possible.

4 Computer Science

4.1 Algorithms

We have used the `algorithm2e` package.

Algorithm 1: Dijkstra's Algorithm

Data: Graph, source
Result: Distances of each vertex from the source and starting edges of the shortest paths

```
create priority queue Q of vertices of Graph //min-heap
foreach vertex v in Graph do
    dist[v] ← ∞
    prev[v] ← UNDEFINED
    add v to Q
end
dist[source] ← 0
while Q is not empty do
    u ← Q.extract()
    foreach neighbour v ∈ Q of u do
        alt ← dist[u] + length(u, v)
        if alt < dist[v] then
            dist[v] ← alt
            prev[v] ← u
        end
    end
end
return dist[], prev[]
```

4.2 Environments & Code

Listing 1: [LaTeX]TeX
The Listings Package

```
\begin{lstlisting}
%Your code goes here.

%This is usually how to present code in \LaTeX, without worrying about
    accidentally invoking a macro. Yes, the text is displayed verbatim.
\end{lstlisting}
\label{trickq}
```

For our document, however, we noticed that naïve methods dont work, and defined a custom environment with `\lstnewenvironment`. This takes care of title formatting and vertical spacing too! Our environment has a counter associated with it, and takes the language and title as arguments. Other code formatting options are specified with `\lstset{...}`. You might want to have a look at the tutorial!

Listing 2: [LaTeX]TeX
Algorithm Boilerplate

```
\documentclass{article}
%...
\usepackage[linesnumbered, ruled, vlined]{algorithm2e
%...
\begin{document}
%...
\begin{algorithm}[h]
\caption{...}
\SetAlgoLined
\DontPrintSemicolon
\KwData{...}
\KwResult{...}
Statement \ ;
%Let magic happen here
$1+1=2$ \ ;
\end{algorithm}
%...
\end{document}
```

Listing 3: Bash
Another Language

```
#some keywords
for echo cat case if esac elif fi awk sed pwd while do
sudo apt get update sleep
```

Listing 4: Python
Keywords, comments, strings

```
from numpy import *
#insanely powerful library
print("Hello Everyone!")
```

The advantage of defining ordered environments like this is that you can refer to this pretty easily if youve left a label properly. If you want to shift them around, the counter and the cross references will do the adjustment automatically. Of the above, Listing 1 is a fiendish trick question, and Listing 2 is rather benevolent.

5 Utilities

5.1 Images

You'll sometimes find the need to include images in your report. Ideally, the image should enhance the document. But yeah, you're probably going to use this technique for a rather dirty life hack. If a prop insists on a LaTeX report, you're just going to dump a photo of your exquisite handwriting to save time, aren't you? *Exploit this loophole at your own risk. The original author of this assignment can neither confirm nor deny his endorsement of the jugaad.*

Figure 2: Aristotle: The first formal logician

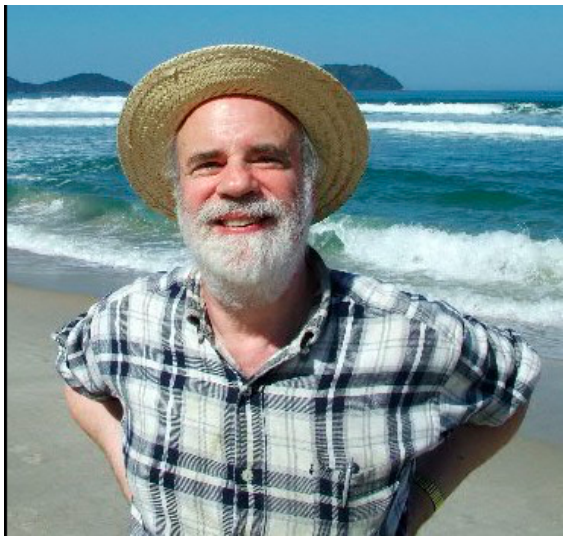

Figure 3: Saul Kripke: we've come a long way since then

Aristotle image source
Saul Kripke image source
Make sure you follow these links, so you know where the hyperlinks lead to when you typeset it yourself.

5.2 Tables

Complexity	**Examples**
$\mathcal{O}(1)$	Computing $(-1)^n$
$\mathcal{O}(log(n))$	Binary Search Insertion & Removal from a min-heap
$\mathcal{O}(nlog(n))$	Merge Sort Fast Fourier Transform
$\mathcal{O}(n^2)$	Bubble Sort First attempt at a problem which has a linear time solution
$2^{\mathcal{O}(log(n))}$	AKS Primality Test
You might want to define a macro to write this notation conveniently	

Table 1: Some common time complexities

This table uses `multirow` as well as `multicolumn`. Replicate it as well as you can.

We have used the fairly popular `plainurl` style.

References

[1] J. P. Bell and S. Gerhold. On the positivity set of a linear recurrence sequence. *Israel Jour. of Math*, 57, 2007.

[2] N. Bourbaki. *General Topology (Part 2): Elements of Mathematics*. Addison-Wesley, 1966.

[3] M Mignotte. Some useful bounds. In *Computer algebra*. Springer, 1982.

[4] James Renegar. On the computational complexity and geometry of the first-order theory of the reals. part i: Introduction. preliminaries. the geometry of semi-algebraic sets. the decision problem for the existential theory of the reals. *J. Symb. Comput.*, 13:255–300, 1992.