

Assignment 4

Aim:

Introduction to Latex

Objective:

- Understand basic document classes of Latex
- documents containing sectioning, cross-references, tables and figures
- Type setting of Basic mathematical formulae
- Automatic generation of table of contents, bibliographies and indexes

Theory:

LATEX is a document preparation system for high-quality typesetting. It is most often used for medium-to-large technical or scientific documents but it can be used for almost any form of publishing. \LaTeX is not a word processor! Instead, \LaTeX encourages authors not to worry too much about the appearance of their documents but to concentrate on getting the right content.

1 Document Basics

Your file should be saved as a file ending with ".tex". The first line of your ".tex" file should be `\documentclass{document_class}`. You need to use document class report for your seminar report; to do this, Put `\documentclass{article}` at the top of file. Other document classes include book, article and letter. Following `\documentclass` is usually a series of package inclusion commands, macro definitions, document specifications, etc. This section of your document is called your documents preamble. Later on, we will discuss the preamble in more detail. After the preamble, we begin your document content itself using the command `\begin{document}`. At the very end of your ".tex" file, put the command `\end{document}`. Your documents contents goes in between these two commands. Throughout your file, you can add comments that will not be compiled by LATEX by putting a % symbol at the beginning of the comment lines.

2 Text Mode Basics

There are two modes in \LaTeX : text mode and math mode. By default when you start writing in your document, you are in text mode. We here discuss some commands to use in text mode; we will then talk about how to enter, code, and exit math mode. (Note: You can write integers in text mode. Most other mathematical symbols need to be written in math mode. Integers can of course also be written in math mode).

- If you press return twice in text mode, you get a new paragraph. \LaTeX indents each new paragraph. If you don't want your paragraph to be indented, type `\noindent`.
- You can skip vertical space by typing `\bigskip`, `\medskip`, or `\smallskip`. You can alternatively specify exactly the amount of vertical or horizontal space that you want to skip using the commands `\vspace`, `\vspace*`, `\hspace`, and `\hspace*`; we later discuss these commands.
- You can break a line by typing `\\`. The command `\linebreak` will break a line like `\\` does, but will force the text to fill the line completely. You can break a page by typing `\pagebreak`.
- You can center text by typing `\begin{center} text \end{center}`; underline it by typing `\underline{text}`; make it boldface or italic by typing `{\bf text}` or `{\it text}` respectively; or make it different sizes by typing `{\Huge text}`, `{\huge text}`, `{\LARGE text}`, `{\Large text}`, `{\normalsizetext}`, `{\small text}`, `{\footnotesizetext}`, `{\scriptsizetext}`, or `{\tinytext}`.
- You can make lists of items by using the following commands:
`\begin{enumerate}`
`\item[first_item_label] first item`
`\item[second_item_label] second item`
`\end{enumerate}`.
An item label can be a bullet (`\bullet`), a number, a letter, etc.
- To create some special characters, like `$`, `%`, `{}`, `&` and `#`, you must precede them with backslashes. (These characters are used to indicate commands within the \LaTeX language, so you need to tell \LaTeX you are not giving it a command.)
 - To create a backslash (that is, `\`), type `\backslash` in math mode (see next section for details on math mode).
 - To make left quotation marks, type `\lq \lq`.
 - To underline text, type `\underline{your text}`.
 - To insert an extra space between characters (for instance, when two consecutive characters are too printed close together for your liking), type `\`, (that is, a backslash followed by a comma) for a small extra space, and type `\` for a larger one. This also works in math mode.
 - To make the word \LaTeX appear in its trademark style, type `\LaTeX`.

3 Math Mode Basics

- You put work in math mode by surrounding it by `$` or `$$`. `$` puts your work in inline math mode, which, well, displays your work inline. You can make inline work larger by typing `$$\displaystyle{your math}$$`. `$$` puts your work in display math mode, making it large and centering it on the page on a new line.
- Making Greek letters in math mode is easy: for example, you type `\Gamma` for a capital gamma, and `\rho` for a lowercase rho.
- You type `x^y` to raise x to the power of y , and `x_y` to get x with subscript y .
- You can use operators like `\sum`, `\prod`, `\int`, `\cap`, `\cup`, `\vee`, and `\wedge`, which create symbols, and operators like `\gcd` and `\cos` which create words (in roman font, the font used by default in text mode). You can currently find a nice list of symbols at <http://omega.albany.edu:8008/Symbols.html>.
- You get fractions by typing `\frac{num}{denom}`, and get, say, the n th root of x by typing `\sqrt[n]{x}`.
- To make delimiters (such as parentheses) the correct size for the expressions they are containing, type, for instance, `\left(expression\right)`. To suppress the appearance of a delimiter, use a period: for instance, `\left. expression \right]` will create a righthand bracket of the right size, but create no lefthand delimiter.
- To un-italicize (make roman) an expression in math mode, type `\mathrm{expression}`. Alternatively, you could use the command `\mbox{expression}`.
- The environments `eqnarray` and `eqnarray*` make equations line up nicely on the page. The entries in equation arrays are automatically in math mode; however, the equation array itself must occur in TEXT MODE. A typical equation array might be coded as follows (the ampersands create column jumps, and the `\\s` create row jumps).

4 Preamble Basics

One of the things to do in the preamble is specify any packages you wish to use when compiling the document. Some useful packages are `amsfonts` (which allows you to make the traditional symbols for the sets of the integers, the reals, etc.); `amsmath` (which provides you with the math operation `\mod`, and with easy ways of making matrices); and `verbatim`, which allows you to use a comment environment (this environment begins with the command `\begin{comment}` and ends with the command `\end{comment}`). In particular, using `amsfonts` you can create the traditional symbol for the set of integers by typing `\mathbb{Z}`; for the set of reals by typing `\mathbb{R}`; etc. To use a particular package, put the command `\usepackage{package_name}` in your preamble. E.g., if you want to use commands from the package `amsfonts` you need to put the command `\usepackage{amsfonts}` in preamble. \LaTeX will automatically put page numbers centered at the bottom of pages, and will number the first page (at least, in article mode).

If you want your page numbers to instead appear in the upper right-hand corner of pages, put the command `\pagestyle{myheadings}` in your preamble. If you wish to suppress the page numbers appearance on your first numbered page, put the command `\thispagestyle{empty}` in your preamble. This will work UNLESS you are using the `\maketitle` command or dividing your book or report into chapters. (We later discuss how to suppress its appearance in those particular cases.) You can write your own definitions in the preamble. For example, to define a symbol, `\gzero`, as a capital gamma with a subscript of 0, you would put in the preamble the command:

```
\def\gzero{\Gamma_0}
```

You can include variables in your definitions. For example, to define a symbol, `\g`, as a capital gamma with a variable subscript, you could put in the preamble the command:

```
\def\g#1{\Gamma_{#1}}
```

Then to get in your document, say, a capital gamma with a subscript of x , you would type (in math mode) `\g{x}`. Of course, you could also just directly type the command `\Gamma_{x}` in your document. But when relatively long commands come up frequently in your document, it can save you a lot of time and typing if you abbreviate the commands with definitions in your preamble. For example, suppose the 3×3 identity matrix, I_3 , over the reals comes up a lot in my document. We will see in a later section that perhaps the easiest way to make this matrix is a command such as the following (using the package `amsmath`):

```
\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}
```

Then any time you want to refer to this matrix in the document, you need only type `\i3`. You can create commands with more than one variable. For instance, you could create a symbol, `\gg`, that is a capital gamma with both a variable subscript and a variable superscript:

```
\def\gg#1#2{\Gamma_{#1}^{#2}}
```

You could then make, say, a capital gamma with a subscript of 17 and a superscript of 4 by typing:

```
\gg{17}{4}
```

Here is a good way of making words appear in roman font in math mode (the way, for example, the commands `\gcd` and `\cos` appear in math mode): use the `\mathop` command. As an example, to define the word `Im` (for image or imaginary) in this manner, we could make the definition-

```
\def\Im{\mathop{\rm Im}}
```

5 Differences Between MS Word & L^AT_EX

- The strength of Word is in writing short, relatively simple documents, since you immediately see how what you wrote looks like (WYSIWYG); whereas writing raw L^AT_EX is somewhat slower, because you first write the contents only and then you generate an output file with layout in a separate step.
- Basic Word features are very easy to use and everybody can produce a simple document with reasonable layout. Using advanced features like automatic numbering, links, and citations require a comparable learning time to L^AT_EX. You will have to initially invest some hours to learn L^AT_EX without being able to immediately produce text. Ultimately, you will be faster typing than clicking through Word icons and menus.
- Word type setting quality is ok for most purposes but its far from professional. Plus, you will only be able to reach good quality if you know Word well, which is not the case for the average users. L^AT_EX, on the other hand, provides you with professional layout out of the box. If you want to get into the nitty gritty, you can adapt the default to your own taste but the standard is already very high without any customisations.
- Here the main shortcoming is that Word does not include a citation utility. You will need to purchase additional software to fill the gap. Bibliography programmes come with their own price, learning curve, and disadvantages. Since L^AT_EX is open source many generations of scientists have written the extensions they needed. You will be able to find all commonly and most rarely used features. References are handled by BibTeX and a database front end like BiBDesk but unlike in Word integration is seamless and the extensions are free.
- MS Word is a proprietary software whereas L^AT_EX is an open-source software.

Word

sequence (in any order). Formally, we say that a rule $I_a \Rightarrow I_b$ occurs in a sequence $s = \langle I_1, I_2, \dots, I_n \rangle$ if and only if there exists an integer k such that $1 \leq k < n$, $I_a \subseteq \bigcup_{i=1}^k I_i$ and $I_b \subseteq \bigcup_{i=k+1}^n I_i$.

Latex

same sequence (in any order). Formally, we say that a rule $I_a \Rightarrow I_b$ occurs in a sequence $s = \langle I_1, I_2, \dots, I_n \rangle$ if and only if there exists an integer k such that $1 \leq k < n$, $I_a \subseteq \bigcup_{i=1}^k I_i$ and $I_b \subseteq \bigcup_{i=k+1}^n I_i$.

6 Advantages of L^AT_EX

Superficially, one of the advantages of L^AT_EX over other more traditional systems (e.g. Word or OpenOffice) is the high typographical quality of the documents that you'll be able to produce. This is particularly true for documents that are heavy on mathematics, but documents for any other area could also take advantage of these qualities.

A less obvious advantage, but much more important, is that L^AT_EX allows you to clearly separate the content from the format of your document. As a writer (scientist, researcher or not), this gives you the opportunity to focus on the what, the creative part of your work, rather than the how is it going to look printed out in paper (that is the work of L^AT_EX document class designers).

Conclusion:

We have successfully been introduced to and understood the basics of L^AT_EX.