Miniguide to LaTeX

1 Introduction

LaTeX is a document preparation system designed to produce high-quality technical and scientific texts, functioning as a markup language where the text structure is described by the user and the formatting is done by the system.

In contrast to traditional word processors, LaTeX offers a distinct workflow: instead of formatting text visually, the user writes commands that define document structure. This ensures automatic formatting, especially in elements like mathematical equations, and also guarantees file stability as they are stored as plain text which prevents data corruption. Additionally, portability is maximized, as these files work on any operating system and the user has more precise control over all aspects of the document.

In this guide, the usage of LaTeX through the *Overleaf* online platform is encouraged for convenience, therefore I won't discuss how to do locally install it on macOS or Windows machines.

2 Basic document structure

Every LaTeX document follows this fundamental structure:

```
% Document type and options
\documentclass[12pt,a4paper]{article}

% Preamble - Packages and settings
\usepackage[utf8]{inputenc}   % Accent support
\usepackage[T1]{fontenc}      % Font encoding
\usepackage{amsmath}          % Mathematical support
\usepackage[english]{babel}   % English language

% Document start
\begin{document}

% Arbitrary content

\end{document}
```

In terms of customization, *packages* are used that bring various functionalities, some of these essential packages are `inputenc` for special character support, `fontenc` for font encoding, `babel` for different languages, `amsmath` for mathematical formulas, `graphicx` for image insertion, and `geometry` for configuring page margins.

LaTeX offers various document classes: scientific articles (class `article`), reports and dissertations (class `report`), books (class `book`), presentations (class `beamer`), and formal letters (class `letter`).

It is recommended to compile the file frequently to verify that everything is working correctly and, if help is needed, consult the official LaTeX documentation as it contains detailed information about all available commands and packages.

There is a clear and hierarchical structure to organize LaTeX documents. In order to define the title, author, date, and create sections, subsections, and sub-subsections, use the following commands:

```
\title{Document Title}
\author{Your name}
\date{\today} % current date
\maketitle % composes the title

\section{Main Section}
\subsection{Subsection}
\subsubsection{Sub-subsection}
```

Through LaTeX, the user can use a variety of commands to format text precisely and consistently. To highlight words or phrases, there's the `\textbf{}` command for bold, `\textit{}` for italic, `\underline{}` for underline, `\texttt{}` for font (common when showcasing code), and `\emph{}` for emphasis. Adjust font size with commands `\tiny`, `\small`, `\normalsize`, `\large`, and `\huge`.

To control text organization, there are specific commands such as a blank line or the `\par` command starting a new paragraph. To break a line within a paragraph, use `\\` or `\newline`. The `\noindent` command removes the default indentation at the beginning of a paragraph. Additionally, control of the horizontal and vertical spacing can be done via the `\hspace{}` and `\vspace{}` commands, respectively.

Let's recap some of this information below:

```
\textbf{bold text}
\textit{italic text}
\underline{underlined text}
\texttt{monospaced text}
\emph{emphasized text}

{\tiny tiny text}
{\small small text}
{\normalsize normal text}
{\large large text}
{\huge huge text}
```

Lists are commonly created with bullets, numbered, or descriptive. Bullet lists are ideal for listing items without a specific order, while numbered lists are used when item order is relevant. Descriptive lists, however, associate a label (term) with a more detailed description. The syntax for creating these lists is relatively simple and involves using the `itemize`, `enumerate`, and `description` environments, respectively. Within these environments, each list item starts with the `\item` command followed by the item text.

For example, when creating a bullet list we'd start off by writing an `itemize` environment and inside it list the items using the `\item` command. To customize any list appearance, LaTeX offers options to modify the bullet type, numbering, and spacing between items.

Let's understand this better:

```
% Bullet list:
\begin{itemize}
    \item First item
    \item Second item
    \item[-] Item with hyphen
\end{itemize}

% Numbered list:
\begin{enumerate}
    \item First item
    \item Second item
    \item[*] Item with asterisk
\end{enumerate}

% Description list:
\begin{description}
    \item[Term] Description
    \item[Another] Another description
\end{description}
```

You probably already noticed this, but comments can be made with the % character. Another example:

```
% This is a line comment
%
% This is a block
% of comments
```

Besides basic formatting commands, LaTeX offers a variety of environments to format text in specific ways: for example, to highlight quotes, you can use the `quote` and `quotation` environments, both of which center and format the text to indicate it's a quote. The `quote` environment is used for short quotes, while `quotation` is more suitable for longer quotes, such as those with multiple paragraphs.

When preserving the exact formatting of a given text, such as with programming code or command output examples, the `verbatim` environment is ideal. Within this environment, all characters, including spaces and line breaks, are displayed exactly as they were typed, without any interpretation by the LaTeX compiler:

```
% Quotes
\begin{quote}
    Centered short quote.
\end{quote}

\begin{quotation}
    Long quote with paragraphs.
\end{quotation}

% Verbatim text (preserves spaces and breaks)
\begin{verbatim}
    Text exactly
    as typed
```

To control paragraph indentation, the `\parindent` value can be adjusted to any arbitrary number. Line spacing is defined with the `\linespread` command, and text alignment can be centered, left-aligned, or

right-aligned using the `center`, `flushleft`, and `flushright` environments.

To organize text in columns, the `multicol` package is useful. When using LaTeX, it's important to avoid some common errors, such as using `\\` to separate paragraphs (the correct way is to leave a blank line), mixing different formatting commands without grouping them properly, forgetting to close environments, and using multiple white spaces (which will be ignored).

Suggestion for practicing
Use the knowledge acquired so far to apply the elements of structure, lists, formatting, and text alignment to a blank document.

3 Advanced Elements

3.1 Floating Elements

In LaTeX, tables are constructed using the `tabular` environment, which allows defining the number of columns and text alignment in each column. Figures, in turn, are inserted into the document using the `figure` environment and the `includegraphics` command.

Both elements, tables and figures, are considered *floating*, meaning LaTeXcan position them automatically on the page, seeking the best fit. To control the position of a floating element, the options `[h]` (here), `[t]` (top), `[b]` (bottom), and `[p]` (separate page) should be used within brackets after the environment name. Additionally, you can add captions to tables and figures using the `caption` and `label` commands, respectively, making it easy to reference these elements in other parts of the document:

```
    \begin{table}[htbp]
    \centering
    \caption{Table Title}
    \begin{tabular}{|l|c|r|}  % | for borders, l-left, c-center, r-right
        \hline
        Column 1 & Column 2 & Column 3 \\
        \hline
        Item 1 & 123 & A \\
        Item 2 & 456 & B \\
        \hline
    \end{tabular}
    \label{tab:example}
\end{table}
```

It's interesting to use packages like `booktabs` for more elegant tables, `tabularx` for adjusting table width, and `multirow` for merging cells.

```
\usepackage{graphicx}

\begin{figure}[htbp]
    \centering
    \includegraphics[width=0.8\textwidth]{image.png}
    \caption{Figure description}
    \label{fig:example}
\end{figure}
```

When inserting images, place `label` after `caption` and use formats like PNG.

To add color to text, the `xcolor` package is essential. This way custom colors can be defined and applied to different text elements, such as words or entire blocks.

Additionally, it's possible to create highlight effects, such as colored boxes with or without borders. Cross-references are another powerful LaTeX tool, allowing for the connection of different parts of a document, such as sections, figures, and tables. To cross-reference, use the `ref`, `pageref`, and `nameref` commands, which insert the number, page number or name of an element, respectively. To manage a list of bibliographic references, use the `cite` and `bibliography` commands. Don't forget to choose an appropriate citation style, such as `plain`.

3.2 Basic customization

To customize page geometry, the `geometry` package allows for defining paper size, margins, and internal spacing. The `fancyhdr` package offers advanced features for creating custom headers and footers, allowing the user to define different content for left, right, and center margins, both in the header and footer.

Not only that, but LaTeX allows adjusting line spacing using the `linespread` command and paragraph spacing with `parskip`. To control list margins, use the `leftmargin` command. Check it out:

```
% Page configuration
\usepackage{geometry}
\geometry{
    a4paper,
    margin=2.5cm,
    top=3cm,
    bottom=3cm
}

% Header and Footer
\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyhead[L]{Left Header}
\fancyhead[R]{Right Header}
\fancyfoot[C]{Page \thepage}

% Spacing
% Between lines
\linespread{1.5}

% Between paragraphs
\setlength{\parskip}{1em}

% List margins
\setlength{\leftmargini}{2em}
```

Suggestion for practicing
Use the information learned in this section to apply in the earlier document graphic elements and color customization; include a custom title and make a footer with page numbering.

3.3 Mathematics

To insert mathematical formulas in text, use the `inline` mode with $a + b$ and to highlight equations and the `display` mode with the `equation` environment. Operators like $+, -, *, /$ are used to perform basic operations. Mathematical symbols like $\alpha, \beta, \sum, \int$ and many others are available.

LaTeX offers environments to create matrices, fractions, and more complex equations. Packages like `amsmath` and `amssymb` allow writing more sophisticated expressions. Add them to the preamble:

```
\usepackage{amsmath}    % main package for mathematics
\usepackage{amssymb}    % additional mathematical symbols
```

Let's see how this works in practice:

```
% Inline mode (in the text line)
This is an equation $x + y = z$ in the middle of text

% Display mode (centered equation)
\[
    x + y = z
\]

% Environment for equations (with numbering)
\begin{equation}
    x + y = z
\end{equation}
```

Equations can be aligned:

```
\begin{align}
x + y &= z \\
2x + 3y &= 5z
\end{align}
```

Basic operations are represented as follows:

```
% Arithmetic operators
$a + b$         % addition
$a - b$         % subtraction
$a \times b$    % multiplication
$a \div b$      % division
$\frac{a}{b}$   % fraction

% Powers and indices
$x^2$           % squared
$x_i$           % subscript
$x^2_i$         % combined
```

Mathematical symbols are fully supported:

```
% Greek symbols

$\alpha, \beta, \gamma$
$\pi, \theta, \phi$

% Sets

$\in, \notin$           % in a given set, not a given set
$\subset, \supset$      % subset, superset
$\cup, \cap$           % union, intersection

% Logic

$\forall, \exists$     % para todo, existe
$\implies, \iff$       % implica, se e somente se
```

Matrices can be written in LaTeX these ways:

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

% Com parênteses
\begin{pmatrix}
a & b \\
c & d
\end{pmatrix}

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

Other relevant symbols:

```
% Sum
\sum_{i=1}^{n} x_i

% Integral
\int_{a}^{b} f(x) \, dx

% Limits
\lim_{x \to \infty} f(x)

% Product
\prod_{i=1}^{n} x_i
```

4 Conclusion

Although LaTeX can seem intimidating at first, I sincerely hope this short guide helped you to have a basis on what this powerful tool can do. The topics addressed were only the fundaments necessary to

write a basic document, from basic structure to slightly more advanced format elements. With practice and experimentation, LaTeXwill reveal itself to be the ideal tool for producing technical and academical documents with professional excellency.

4.1 Additional resources

I believe to be important to mention some interesting environments to get to know better LaTeXand, if necessary, help. This can be done through whether the official **TeX** structure, the subreddit **r/latex** or **Tex Stack Exchange**.
Overleaf allows for sharing diverse *templates* of documents on their platform, free to use. You can visualize them at `https://www.overleaf.com/latex/templates`.

Good luck!