LaTeX Tutorial

LaTeX 介紹

CHUN-JEN SHIH

revised by Jian-Jiun Ding

施淳仁 著　　丁建均老師 修訂

Graduate Institute of Communication Engineering

National Taiwan University

國立台灣大學電信工程學研究所

Contents

0.1 Editing Environment

LaTeX is a powerful typesetting system. Using LaTeX, we can produce professional academic papers, delicate presentation slides and even various styles of resume. To produce a documentation using LaTeX, one needs an editing and compiling environment. There are offline softwares and online websites that can serve as such useful environment. We take TeXstudio as an example for offline softwares and Overleaf as an example for online websites.

0.1.1 TeXstudio

To use TeXstudio as an editing environment, one needs to download the software execution application from its website TeXstudio Website. One also needs to download LaTeX operating system, e.g., MikTeX. After these basic setups, one can start to edit LaTeX documents in TeXstudio. The following is a screenshot of the editing interface of TeXstudio. As we can see, on the boundaries of TeXstudio are sidebars with many buttons, which enables us to perform

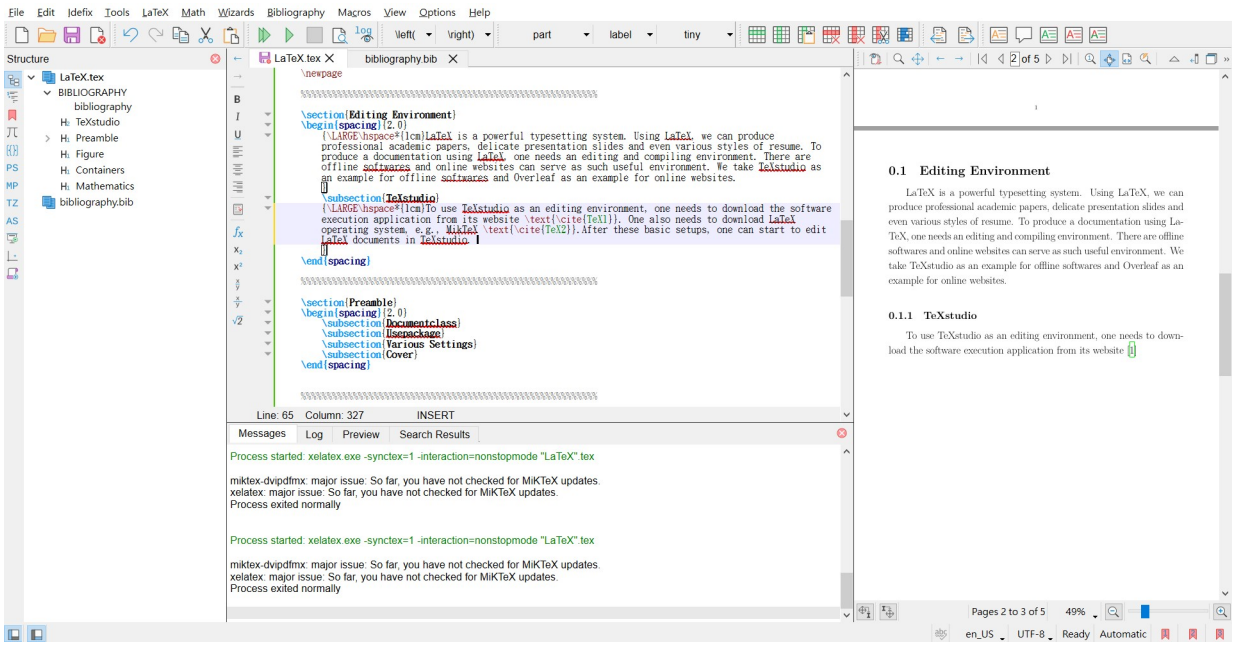

Figure 1: TeXstudio Screenshot

various settings and gives us useful shortcuts to LaTeX commands. The readers can resort to TeXstudio documentation for a detailed and comprehensive tutorial for TeXstudio.

0.1.2 Overleaf

Overleaf is an online website that supports us to write LaTeX documents on it. One needs to go to its website and registers an account. Then, one can starts a project. The following figure is the screenshot of the Overleaf editing interface. After finishing editing, one can down-

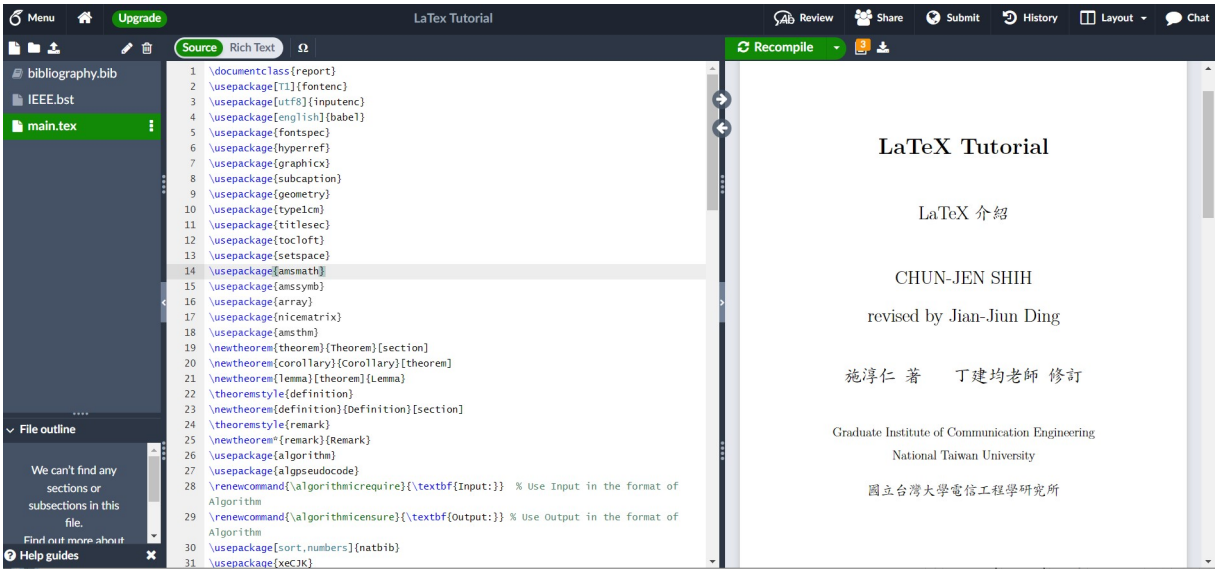

Figure 2: Overleaf Screenshot

load the pdf document. Furthermore, Overleaf enables us to cooperate with other people on the same project. This is a quite useful function.

Finally, we make a remind that if one wants to write a document that contains Chinese characters he needs to set the compiler as XeLaTeX.

0.2 Preamble

Before writing the main contents of a LaTeX document, one needs to define the basic information of this document. For example, the

document class should be defined, various packages should be imported and miscellaneous settings should be set. These constitute the preamble of a LaTeX document. Note that the main contents are sandwiched within "\begin{document}" and "\end{document}" and the preamble commands are before "\begin{document}".

0.2.1 Documentclass

When the LaTeX compiler deals with a LaTeX document, it should need to know the type of this document. This can be done by using the command

$$\backslash documentclass[options]\{class\} \tag{1}$$

"class" is the desired type and "options" determines some further properties. We make some examples.

1.
$$\backslash documentclass[a4paper]\{article\} \tag{2}$$
   - If our pdf document is desired to be journal papers, conference papers or short letters, etc., we use "article"
   - "a4paper" is the size of the resulting pdf document.

2.
$$\backslash documentclass\{report\} \tag{3}$$
   - "report" is used for long papers or short books with a few sections or chapters.

3.
$$\backslash documentclass\{book\} \tag{4}$$
   - "book" can support ordinary long books.

4.
$$\backslash documentclass\{beamer\} \tag{5}$$
   - "beamer" is a powerful class that we can use to design beautiful and professional slides.

0.2.2 Usepackage

In order for the LaTeX operating system to support various functions or settings to generate a desired pdf document, we need to import miscellaneous packages. The command is

$$\backslash usepackage[options]\{package\} \tag{6}$$

In this section, we directly introduce various packages that are important to generate pdf documents.

1.

$$\backslash usepackage[utf8]\{inputenc\} \tag{7}$$

- The inputenc package allows one to specify the input encoding of his LaTeX document. In this way, the LaTeX operating system knows how to interpret LaTeX files.
- utf8 is a popular and default input encoding.

2. Packages related to fonts and words

(a)

$$\backslash usepackage[T1]\{fontenc\} \tag{8}$$

- The T1 font encoding is a 8-bit encoding. If we do not specify it, then some special characters like those accented cannot be supported.

(b)

$$\backslash usepackage\{fontspec\} \tag{9}$$

- To specify specific English and Chinese fonts, one should use this package.

(c)

$$\backslash usepackage\{type1cm\} \tag{10}$$

- If one wants to set the size of fonts, he should use this package.

- The command to set the font size is

$$\backslash fontsize\{arg1\}\{arg2\}\{arg3\} \tag{11}$$

arg1 controls the size of fonts and arg2 controls the line spacing. Both use "pt" as the unit. arg3 is the texts we want to operate on.

(d)

$$\backslash usepackage\{xeCJK\} \tag{12}$$

- If one wants to write Chinese characters, he should use this package.
- One can use the command

$$\backslash setCJKmainfont\{cwTeXKai\} \tag{13}$$

to set a specific Chinese font. cwTeXKai is one specific font and there are many more choices.

- It is preferable that the following two commands are also added.

$$\backslash XeTeXlinebreaklocale\ "zh" \tag{14}$$

$$\backslash XeTeXlinebreakskip\ =\ 0pt\ plus\ 1pt \tag{15}$$

The first one tells the XeLaTeX compiler to break a line in the "Chinese" way. The second one asks the XeLaTeX compiler to add 0pt to 1pt flexible spacing between characters so that the typesetting of lines can be neat.

(e)

$$\backslash usepackage[german, french, english]\{babel\} \tag{16}$$

- To specify languages we may use in our pdf document, we need to import the babel package. The arguments in the brackets are those languages.
- The following command can be applied to select a specific language to use.

$$\backslash selectlanguage\{german\} \tag{17}$$

(f)

$$\backslash usepackage\{xcolor\} \tag{18}$$

- To specify the colors of words, one needs to import the xcolor package.
- One can use the command

$$\backslash\{color\{red\}text\} \tag{19}$$

"red" can be changed to other desired color. one can also use the equivalent command

$$\backslash textcolor\{red\}\{text\} \tag{20}$$

3. Packages related to setting of space

(a)

$$\backslash usepackage\{setspace\} \tag{21}$$

- This package can support us to control spacing between lines.
- There are several commands to use as follows.

$$\backslash singlespacing \tag{22}$$
$$\backslash onehalfspacing \tag{23}$$
$$\backslash doublespacing \tag{24}$$

These commands can set the line spacing to be one, one half and two units, respectively. If one wants to set a specific unit of line spacing, he can use the following two commands.

$$\begin{array}{l}\backslash begin\{spacing\}\{3.0\}\\ \backslash end\{spacing\}\end{array} \tag{25}$$

or

$$\backslash setstretch\{3.0\} \tag{26}$$

3.0 can be changed to any desired unit of line spacing.

(b)

$$\backslash usepackage\{geometry\} \tag{27}$$

- This package enables us to modify the size and margins of

pages.

- One can use the following command.

$$\backslash geometry\{left = 2cm, right = 2cm, top = 3cm, bottom = 1cm\} \tag{28}$$

This command assigns 2cm, 2cm, 3cm and 1cm to the left, right, top and bottom margins respectively.

4. Packages related to figures

(a)

$$\backslash usepackage\{graphicx\} \tag{29}$$

- To insert figures, the graphicx package should be imported.

(b)

$$\backslash usepackage\{subcaption\} \tag{30}$$

- To arrange multiple sub-figures in one figure, the subcaption package should be used.

(c)

$$\backslash usepackage\{wrapfig\} \tag{31}$$

- To wrap the text around a figure, one needs to import this package.

5. Packages related to tables

(a)

$$\backslash usepackage\{array\} \tag{32}$$

- This package enables us to control the widths of tables.

(b)

$$\backslash usepackage\{multirow\} \tag{33}$$

- In order to make a row or a column span several cells, this package should be used.

6. Packages related to Mathematics

(a)

$$\usepackage\{amsmath, amsthm, amsfonts, amssymb\} \tag{34}$$

- These are four common mathematics packages.

(b)

$$\usepackage\{mathrsfs\} \tag{35}$$

- This package supports the Raph Smith's Formal Script font in mathematics. One can use the command

$$\mathscr\{\} \tag{36}$$

(c)

$$\usepackage\{bm\} \tag{37}$$

- If one wants to make texts in the math mode be bold, he should use the following command.

$$\bm\{text\} \tag{38}$$

(d)

$$\usepackage\{algorithm\} \tag{39}$$
$$\usepackage\{algpseudocode\} \tag{40}$$

- To typeset algorithms, these two packages should be imported.

(e)

$$\usepackage\{nicematrix\} \tag{41}$$

- To further control matrices or add extra properties to them, one can use the bNiceMatrix, pNiceMatrix, etc, which are supported by the "nicematrix" package.

7. Packages related to References

(a)

$$\usepackage[square, numbers]\{natbib\} \tag{42}$$

- "natbib" is a package to manage the bibliography.

- The options "[square,numbers]" enable squared brackets and numeric citations respectively.

(b)

$$\backslash usepackage[argument]\{biblatex\} \tag{43}$$

- "biblatex" is another package to manage the bibliography.
- one can use some arguments to set the style of the bibliography section. For example, "style=numeric" defines the bibliography style and citation style to be "numeric".

8.

$$\backslash usepackage\{hyperref\} \tag{44}$$

- We can create hyperlinks in the pdf document by using this package and the following commands.

$$\backslash url\{http://www.overleaf.com\} \tag{45}$$
$$\backslash href\{http://www.overleaf.com\}\{link\} \tag{46}$$

If 45 is used, the text "http://www.overleaf.com" will be a hyperlink linked to this website. If 46 is used, the text "link" will be a hyperlink linked to the website "http://www.overleaf.com".

9.

$$\backslash usepackage\{titlesec\} \tag{47}$$

- To customize the styles of titles, this package should be used.
- Titles include "part", "chapter", "section", "subsection", "subsubsection", "paragraph", "subparagraph". One can use commands like

$$\backslash section\{titlebody\} \tag{48}$$

to generate a title. "section" can be replaced with other possible titles. If one wants the title to be unnumbered, he can add an extra asterisk sign like

$$\backslash section^*\{titlebody\} \tag{49}$$

- The following command can be used to set the styles of titles.

$$\begin{aligned}\backslash titleformat\{<title>\}\{<format>\}\\ \{<label>\}\{<sep>\}\end{aligned} \tag{50}$$

  – "<title>" is the title we want to operate on. For example, \part, \chapter, etc.
  – "<format>" is the format we want to apply on the title label and the title body. For example, \Huge\bfseries.
  – "<label>" specifies the title label. For example, 第,\thechapter, 章. In this way, the title label will be something like " 第 2 章".
  – "<sep>" is the horizontal spacing between the title label and the title body and it must be a length and not be empty. For example, 20pt.

10.

$$\backslash usepackage\{tocloft\} \tag{51}$$

- This package can control how we typeset the table of contents (ToC), the list of figures (LoF) and the list of tables (LoT).
- One can use the following commands to create the ToC, the LoF and the LoT, respectively.

$$\backslash tableofcontents \tag{52}$$
$$\backslash listoffigures \tag{53}$$
$$\backslash listoftables \tag{54}$$

- The title texts of the TOC, the LoF and the LoT are determined by the values of the following identities.

$$\backslash contentsname \tag{55}$$
$$\backslash listfigurename \tag{56}$$
$$\backslash listtablename \tag{57}$$

  One can change the values by the command "renewcommand"

(which will be introduced in section 0.8). For example,

$$\renewcommand\contentsname{Summary}$$

- The fonts of the section and subsection titles are determined by the following identities.

$$\cftsecfont \tag{58}$$
$$\cftsubsecfont \tag{59}$$

One can also change them by the command "renewcommand". For example,

$$\renewcommand\cftsecfont{\fontsize{20pt}{20pt}\selectfont}$$

- If one wants to manually add contents into the ToC, the following command should be used.

$$\addcontentsline{toc}{< title >}{titletext} \tag{60}$$

For example,

$$\addcontentsline{toc}{\chapter}{References}$$

11.

$$\usepackage{varwidth} \tag{61}$$

- If one creates a box using the "fbox" command (which will be introduced in section 0.8) and wants to insert multiple lines inside the box, he should import this package.

12.

$$\usepackage{mhchem} \tag{62}$$

- This package provides commands for typesetting chemical notations, molecular formulae and equations.
- We give an example. We can express $^{95}_{36}\mathrm{Kr}$ using the command

$$\ce{^{95}_{36}Kr}$$

0.2.3 Cover

One can use the following commands to typeset the cover of a pdf document.

$$\backslash title\{\} \tag{63}$$
$$\backslash author\{\} \tag{64}$$
$$\backslash date\{\} \tag{65}$$

It is clear that 63 determines the title of this document, 64 tells the author of this document and 65 shows the date when this document is done. After setting these preamble commands, we should add the following command

$$\backslash maketitle \tag{66}$$

after "\begin{document}" to make a cover that follows these settings.

0.3 Figure

To include figures in the document, one needs to use the following command.

$$\begin{aligned} &\backslash begin\{figure\}[htbp] \\ &\backslash end\{figure\} \end{aligned} \tag{67}$$

There are four optional arguments, which are "h", "t", "b", "p", to control where the figure is placed. "h" stands for "here", which tells the compiler to put the figure here. "t" stands for "top", which tells the compiler to put the figure at the top of the current page. "b" stands for "bottom", which tells the compiler to put the figure at the bottom of the current page. "p" stands for "page", which tells the compiler to put the figure at the top of the next page. The former the argument, the greater priority it has. For example, "htbp" means that the compiler will first try to put the figure here. If this cannot be done, it proceeds to try to put it "t", "b" and "p". Furthermore, one can use

the command

$$\backslash centering \tag{68}$$

to make the figure lie horizontally center at the page.

The command

$$\backslash includegraphics[options]\{figure\} \tag{69}$$

is what we need to use to include a figure. "figure" is the path of inserted figure. As for options, we list some useful choices below.

- scale=0.8 : this argument will scale the image 0.8 of its real size.
- width=4cm, height=6cm : we can directly assign the width and height of the image. If only one (width or height) is assigned, the other one (height or width) will be scaled to keep the aspect ratio.
- width=\textwidth : we can also directly set the width of the image to be the same with that of the text.
- angle=45 : we can also rotate the figure

After including the image, we can give a caption to it so that we can describe what this image is about. The command is

$$\backslash caption\{\} \tag{70}$$

Besides simply include a single figure, we also have the need to include multiple subfigures in one single figure. We can achieve this using the "subfigure" environment within a "figure" environment. Concretely speaking, the command related to "subfigure" is

$$\begin{aligned} &\backslash begin\{subfigure\}\{width\} \\ &\backslash end\{subfigure\} \end{aligned} \tag{71}$$

"width" controls the width of this subfigure. We often using something like "0.3\textwidth", which means the width is 0.3 the textwidth. Then, within 71, we also use the "includegraphics" command with the

option set as "width=\textwidth". We can give an example as follows.

$$
\begin{aligned}
&\backslash begin\{figure\}[h]\\
&\quad \backslash centering\\
&\quad\quad \backslash begin\{subfigure\}\{0.5\backslash textwidth\}\\
&\quad\quad\quad \backslash includegraphics[width = \backslash textwidth]\{subfigure1\}\\
&\quad\quad\quad \backslash caption\{caption1\}\\
&\quad\quad \backslash end\{subfigure\}\\
&\quad \backslash hspace^*\{0.2cm\}\\
&\quad\quad \backslash begin\{subfigure\}\{0.5\backslash textwidth\}\\
&\quad\quad\quad \backslash includegraphics[width = \backslash textwidth]\{subfigure2\}\\
&\quad\quad\quad \backslash caption\{caption2\}\\
&\quad\quad \backslash end\{subfigure\}\\
&\backslash end\{figure\}
\end{aligned}
$$

Between subfigures, we can add some spacing to separate them. We can use a command like "\hspace*{0.2cm}" (which will be introduced in section 0.8).

If one wants to wrap the text around a figure, he can use the following command.

$$
\begin{aligned}
&\backslash begin\{wrapfigure\}\{option\}\{width\}\\
&\backslash end\{wrapfigure\}
\end{aligned}
\tag{72}
$$

"option" determines the relative position between the text and the figure. If "r" is used, then the figure will appear to the right of the text while it appears to the left when "l" is used. If the format of the document is a book, then use "o" and "i" instead, where the former denotes the outer edge and the latter denotes the inner edge of the page. In the following, we give an example that involves the "wrapfigure" and

"subfigure" together.

$$
\begin{aligned}
&\backslash begin\{wrapfigure\}\{r\}\{6cm\}\\
&\quad \backslash centering\\
&\quad\quad \backslash begin\{subfigure\}\{1.15\backslash textwidth\}\\
&\quad\quad\quad \backslash includegraphics[width = \backslash textwidth]\{subfigure1\}\\
&\quad\quad\quad \backslash caption\{caption1\}\\
&\quad\quad \backslash end\{subfigure\}\\
&\quad\quad \backslash begin\{subfigure\}\{1.2\backslash textwidth\}\\
&\quad\quad\quad \backslash includegraphics[width = \backslash textwidth]\{subfigure2\}\\
&\quad\quad\quad \backslash caption\{caption2\}\\
&\quad\quad \backslash end\{subfigure\}\\
&\backslash end\{wrapfigure\}
\end{aligned}
$$

0.4 Containers

In this section, we will introduce two kinds of containers that provide an environment where one can effectively show their information. These are lists and tables respectively.

1. Lists

   - One can use the "itemize" environment for unordered lists. The command is

   $$
   \begin{aligned}
   &\backslash begin\{itemize\}\\
   &\backslash end\{itemize\}
   \end{aligned}
   \tag{73}
   $$

   - One can use the "enumerate" environment for ordered lists. The command is

   $$
   \begin{aligned}
   &\backslash begin\{enumerate\}\\
   &\backslash end\{enumerate\}
   \end{aligned}
   \tag{74}
   $$

   - In both environments, one uses the following command to list

an item.

$$\backslash item[label\ text]\ \ text\ of\ entries \tag{75}$$

"label text" is an optional argument that can change the default label of individual entries. We give an example as follows.

$$\begin{aligned}&\backslash begin\{itemize\}\\&\quad\backslash item[NOTE]\ \ an\ example\\&\backslash end\{itemize\}\end{aligned}$$

- We can use the following commands to change the counter of an "enumerate" list.

$$\backslash setcounter\{counter\}\{number\} \tag{76}$$
$$\backslash addtocounter\{counter\}\{number\} \tag{77}$$

The command 76 sets the "counter" to "number" and the command 77 adds "number" to the current value of "counter". For example,

$$\begin{aligned}&\backslash begin\{enumerate\}\\&\quad\backslash setcounter\{enumi\}\{2\}\\&\quad\backslash item\ \ third\ item\\&\backslash end\{enumerate\}\end{aligned}$$

"enumi" is the name of the counter, which belongs to the first level of "enumerate". One can use "enumii", "enumiii" and " enumiv" for deeper nested levels of "enumerate".

2. Tables

- One can create a table through a combination of the "table" and "tabular" environment.
- The "table" environment enables us to use "caption" and "label" (which will be introduced in section 0.8) command. It also determines where we place the table just like how we control a figure. That is, we also use the arguments "h", "t", "b", "p" (see section 0.3).
- The "tabular" environment defines the appearance of the table

itself. It uses ampersands & to differentiate each column and newline symbols \\ to differentiate each row. If we do not want to start a new row but just want to start a new line within a cell, we need to use

$$\backslash newline \tag{78}$$

To add vertical lines to separate columns, a symbol | should be used in the argument passed to the tabular environment (we will explain more clearly in the following). To add horizontal lines to separate rows, one can use the command

$$\backslash hline \tag{79}$$

If we do not want a complete horizontal line that spans all columns but a partial horizontal line that only spans some consecutive columns, we should use the following command.

$$\backslash cline\{i-j\} \tag{80}$$

That means a partial horizontal line that starts in column i and ends in column j.

- We give an example of a combination of commands to create a table.

$$\begin{aligned}
&\backslash begin\{table\}[htbp]\\
&\quad \backslash centering\\
&\quad \backslash begin\{tabular\}\{argument\}\\
&\qquad \backslash hline\\
&\qquad cell1 \ \& \ cell2\backslash\backslash\\
&\qquad cell3 \ \& \ cell4\backslash\backslash\\
&\qquad \backslash hline\\
&\quad \backslash end\{tabular\}\\
&\backslash end\{table\}
\end{aligned}$$

"argument" determines the format of columns. We give some examples below.

$-\{c\ c\ c\}$: this tells the compiler to create a table with three

columns and the text of each column should be centered.

- $\{|c|c|c|\}$: just as we have mentioned, the symbol | denotes a vertical line. Hence, this argument means there are additional vertical lines between each columns.
- $\{||c\ c\ c||\}$: similar to the second one, we can also make a variation on how we place the vertical lines.
- $\{l|c|r\}$: besides "c", one can also use "l" to denote left alignment and "r" to denote right alignment.
- $\{m\{5em\}|m\{1em\}|m\{2em\}\}$: this argument requires the importation of the "array" package. The parameter "m$\{1em\}$" sets a fixed width of 1em to the second column. Note that we can also use "p$\{1em\}$" and "b$\{1em\}$". The differences of them lie in how the text is aligned vertically. For "m$\{1em\}$", the text is aligned vertically in the middle. For "p$\{1em\}$", the text is aligned vertically at the top. For "b$\{1em\}$", the text is aligned vertically at the bottom.
- $\{*\{3\}\{|c\}\}$: this is equivalent to $\{|c|c|c\}$. That is, it is the same as repeating "|c" for three times.
- $\{|p\{2.5cm\} < \{\backslash centering\}|p\{12.5cm\} < \{\backslash centering\}|\}$: we use

$$< \{\backslash cmd\} \tag{81}$$

to execute a command right after each column element. Note that this example is equivalent to "$\{| > \{\backslash centering\}p\{2.5cm\}| > \{\backslash centering\}p\{2.5cm\}\}$" since

$$> \{\backslash cmd\} \tag{82}$$

executes a command right before each column element.

- $\{r@\{.\}l\}$: the special expression

$$@\{\} \tag{83}$$

will remove the "\tabcolsep" (see the setlength command in the section 0.8) that exists between columns and replace it with whatever is inside the curly brackets. Hence this exam-

ple illustrates a table with two columns, the first one embraces text aligned horizontally to the right and the second one embraces text aligned horizontally to the left and appended "." at the beginning.

- To design a more complex table, one may import the "multirow" package and use the "multirow" and "multicolumn" command. The "multirow" command is

$$\backslash multirow\{number\ of\ rows\}\{*\}\{content\} \tag{84}$$

This command will combine "number of rows" rows into one cell and contains "content". Note that it is necessary to omit the content of the same row in the following lines. The "multicolumn" command is

$$\backslash multicolumn\{number\ of\ columns\}\{alignment\}\{content\} \tag{85}$$

This command will combine "number of columns" columns into one cell formated by "alignment" and contain "content". We give an example of how these commands can be used.

```
\begin{table}[htbp]
  \LARGE
  \centering
  \begin{tabular}{*{3}{|c|}}
    \hline
    value1 &value2 & value3\\
    \hline
    \multicolumn{2}{|l|}{\multirow{2}{*}{45}} & 78\\
    \multicolumn{2}{|l|}{} & 23\\
    \hline
  \end{tabular}
\end{table}
```

This is the source code of the following table.

value1	value2	value3
45		78 23

0.5 Mathematics

In this section, we introduce some important environments and commands regarding mathematics.

1. Math Expressions

   - To write equations or construct various math components, e.g., cases and matrices, we need to learn how to write math expressions. Those math expressions are the basic ingredients of different math environments.
   - One can refer to the following three sources for syntax and commands of mathematical symbols.
     – List of mathematical symbols by subject
     – MathJax basic tutorial and quick reference
     – LATEX Mathematical Symbols

     One can almost find all he needs to write miscellaneous math expressions in those three sources.

2. Cases

   - The ”cases” environment renders a large curly-brace to the left of multiple lines, which are expressions based on different conditions.

- We give an example of how we can use it.

$$\begin{aligned} &\backslash begin\{cases\} \\ &\quad expression1 \ \& \ condition1\backslash\backslash \\ &\quad expression2 \ \& \ condition2\backslash\backslash \\ &\backslash end\{cases\} \end{aligned} \tag{86}$$

  Similar to what we have introduced in the previous section, the ampersands & is used to separate items and the newline symbol \\ is used to start a new line.

3. Equations

- There are several ways to write an equation. In the following, we will give some examples to illustrate how we can use different commands to meet our end.
  - We can use

$$\$math\ expression\$ \tag{87}$$

    to write an inline-mode math equation.
  - One can use the following command to create a display-mode math equation.

$$\begin{aligned} &\backslash begin\{equation\} \\ &\quad math\ expression \\ &\backslash end\{equation\} \end{aligned} \tag{88}$$

    Note that this kind of math equation is numbered. If we want to create a unnumbered one, we need to use

$$\begin{aligned} &\backslash begin\{equation^*\} \\ &\quad math\ expression \\ &\backslash end\{equation^*\} \end{aligned} \tag{89}$$

    instead. If the math expression spans multiple lines, we may want to align each line neatly. If this is the case, we need to add an extra "aligned" environment and use the ampersands & to indicate where to align. We directly give an example as

follows.

$$\begin{aligned}
&\backslash begin\{equation^*\}\\
&\quad \backslash begin\{aligned\}\\
&\qquad \&\backslash int\hat{}3_0x\hat{}2dx\\
&\qquad = \&\backslash, \backslash, 9\\
&\quad end\{aligned\}\\
&\backslash end\{equation^*\}
\end{aligned}$$

This is the source code of the following equation

$$\begin{aligned}
&\int_0^3 x^2 dx\\
&= 9
\end{aligned}$$

– One can also use the "align" environment that combines " equation" and "aligned" together. For the above example, we can also rewrite it using "align". The source code is as follows.

$$\begin{aligned}
&\backslash begin\{align^*\}\\
&\quad \&\backslash int\hat{}3_0x\hat{}2dx\\
&\quad = \&\backslash, \backslash, 9\\
&\backslash end\{align^*\}
\end{aligned}$$

However, note that if we use a numbered equation system; that is, "align" without the asterisk symbol, then each line of the math expression will be assigned an equation number.

4. Matrices

- To write matrices, one needs to import the "amsmath" package.
- There are six matrix environments we can use to create matrices with different appearances. These are "matrix", "pmatrix", " bmatrix", "vmatrix", "Bmatrix", and "Vmatrix" respectively. " p" stands for parentheses, "b" stands for brackets, "v" stands for verts, "B" stands for braces and "V" stands for double verts.

- We give an example below to show how to construct a matrix.

$$\begin{aligned}&\backslash begin\{Bmatrix\}\\&\quad value1\ \&\ value2\backslash\backslash\\&\quad value3\ \&\ value4\backslash\backslash\\&\backslash end\{Bmatrix\}\end{aligned} \tag{90}$$

  This is the source code of the following matrix

$$\begin{Bmatrix} value1 & value2 \\ value3 & value4 \end{Bmatrix}$$

  Similar to what we have introduced in the previous section, the ampersands & is used to separate items and the newline symbol \\ is used to start a new line. One can try himself other possible types of matrices.

- To create more complex and advanced matrices, one needs to import the "nicematrix" package and use environments like "bNiceMatrix" and "pNiceMatrix". The readers can refer to this tutorial for detailed functions, properties and commands of "nicematrix". In the following, we just give a simple example.

$$\begin{aligned}&\backslash begin\{bNiceMatrix\}[last-row, last-col]\\&\quad value1\ \&\ value2\ \&\ rowdim1\backslash\backslash\\&\quad value3\ \&\ value4\ \&\ rowdim2\backslash\backslash\\&\quad coldim1\ \&\ coldim2\ \&\ \backslash\backslash\\&\backslash end\{bNiceMatrix\}\end{aligned}$$

  This is the source code of the following matrix.

$$\begin{array}{cc} \left[\begin{array}{cc} value1 & value2 \\ value3 & value4 \end{array}\right] & rowdim1 \\ rowdim2 & \\ coldim1 \quad coldim2 & \end{array}$$

  As we can see, by setting the arguments as "last-row" and "last-col", we can add additional notations on the right side and bottom of this "bmatrix". "first-row" and "first-col" are also

similar arguments one can use.

5. Algorithms

- One can create an algorithm through a combination of the ” algorithm” and ”algorithmic” environment.
- The ”algorithm” environment enables us to use ”caption” and ”label” command. Similar to tables, we can also use the arguments ”h”, ”t”, ”b” and ”p” to determine where we place an algorithm.
- The ”algorithmic” environment enables us to typeset the bodies of the algorithm. We directly give an example of how we can construct an algorithm.

```
\begin{algorithm}[htbp]
  \caption{algorithm1}
  \begin{algorithmic}
    \Require\\\hspace*{-0.5cm}
    Parameters\\\hspace*{-0.5cm}
    \textbf{Initialization :}\\\hspace*{-0.5cm}
    InitializationText\\\hspace*{-0.5cm}
    \textbf{Iteration :}\\\hspace*{-0.5cm}
    IterationText\\\hspace*{-0.5cm}
    \textbf{Output :}\\\hspace*{-0.5cm}
    OutputText
  \end{algorithmic}
\end{algorithm}
```

This is the source code of the following algorithm.

Algorithm 1 algorithm1

Require:
Parameters
Initialization:
InitializationText
Iteration:
IterationText
Output:
OutputText

6. Theorems, Lemmas, Corollaries, Definitions, Proofs and Remarks, etc.

   - In a mathematical document, there are theorems, lemmas, corollaries, definitions, proofs, remarks, etc. We can use the "newtheorem" command to define mathematics environments. Note that we should import the "amsthm" package.
   - We can use the following two kinds of commands.

$$\backslash newtheorem\{name\}\{PrintedTitle\}[numberby] \quad (91)$$
$$\backslash newtheorem\{name\}[counter]\{PrintedTitle\} \quad (92)$$

   "name" is the user-defined math environment. "Printed Title" is the title of this environment. "numberby" is the name of the section level (section, theorem, etc.) where the numbering takes place. "counter" is the name of the counter by which the numbering of this environment follows. We give two examples to show how to use them. The first example is

$$\backslash newtheorem\{theo\}\{Theorem\}[section]$$

   In this example, we define a math environment called "theo". When we use it by the following command

$$\backslash begin\{theo\}[name]$$
$$\backslash end\{theo\}$$

   we construct a "theo" environment. Since we set the "numberby" as "section", if the section is ,say 1.2, the title will be something like "Theorem 1.2.2 (name)". That is, this theorem is the 2rd theorem in section 1.2 and has a name "name". The

second example is

$$\backslash newtheorem\{lemma\}[theo]\{Lemma\}$$

In this example, we define a math environment called "lemma". When we use it by the following command

$$\begin{array}{l}\backslash begin\{lemma\}\\ \backslash end\{lemma\}\end{array}$$

we construct a "lemma" environment. Since we set the "counter" as "theo", if the numbering of the latest "theo" environment is, say 1.2.2, the title will be "Lemma 1.2.2", i.e., the "lemma" environment follows the same numbering as the "theo" environment.

- To differentiate different math environments, besides the title of them, we can also use different styles. There are three kinds of styles, which are the "definition", the "plain" and the "remark" styles respectively. One can refer to this tutorial for details of these styles. One can use the following command

$$\backslash theoremstyle\{plain\} \tag{93}$$

to set a specific style for some math environments. This command should be in front of the "newtheorem" command. For example,

$$\begin{array}{l}\backslash theoremstyle\{definition\}\\ \backslash newtheorem\{theo\}\{Theorem\}[section]\\ \backslash newtheorem\{lemma\}[theo]\{Lemma\}\end{array}$$

In this way, the "theo" and the "lemma" environments will follow the "definition" style.

- The "amsmath" package provides the "proof" environment. We can use

$$\begin{array}{l}\backslash begin\{proof\}\\ \backslash end\{proof\}\end{array} \tag{94}$$

to invoke this environment. The default QED symbol is □. If we want to change it, we should use the following command.

$$\backslash renewcommand\backslash qedsymbol\{Q.E.D.\} \tag{95}$$

The "renewcommand" will be introduced in section 0.8 and the "\qedsymbol" is the value of the QED symbol. In this example, we change □ to Q.E.D..

- Note that the commands 91, 92, 93 should be put in the preamble of the LaTeX document.

0.6 Reference

If one imports the "natbib" package 42, he can use the following commands to create a bibliography.

$$\backslash bibliographystyle\{ieeetr\} \tag{96}$$
$$\backslash bibliography\{bib\ document\} \tag{97}$$

Command 96 is used to incorporate a specific bibliography style to create one. One can replace "ieeetr" with other styles. As for 97, it is used to import a bibliography document (with an filename extension ".bib"). The contents of a bibliography document are standard bibtext references. We give a snapshot of what a bibliography document might be. If one wants to refer a specific paper, he can search it by the Google Scholar. As figure 4 shows, one first clips the button inside the red circle, then clips the button inside the green circle, and he can copy the bibtex reference of it. After including the bibtex reference of a specific source in the bibliography document, one can cite this source in the article. For example, in figure 3, the bibtex reference of the paper "CoSaMP:Iterative signal recovery from incomplete and inaccurate samples" has been included. One can use the following command to cite this in the article.

$$\backslash cite\{cosamp\} \tag{98}$$

```
@article{cosamp,
  title={CoSaMP: Iterative signal recovery from incomplete and inaccurate samples},
  author={Needell, Deanna and Tropp, Joel A},
  journal={Applied and computational harmonic analysis},
  volume={26},
  number={3},
  pages={301--321},
  year={2009},
  publisher={Elsevier}
}

@article{romp1,
  title={Signal recovery from incomplete and inaccurate measurements via regularized
  orthogonal matching pursuit},
  author={Needell, Deanna and Vershynin, Roman},
  journal={IEEE Journal of selected topics in signal processing},
  volume={4},
  number={2},
  pages={310--316},
  year={2010},
  publisher={IEEE}
}
```

Figure 3: bibliography document

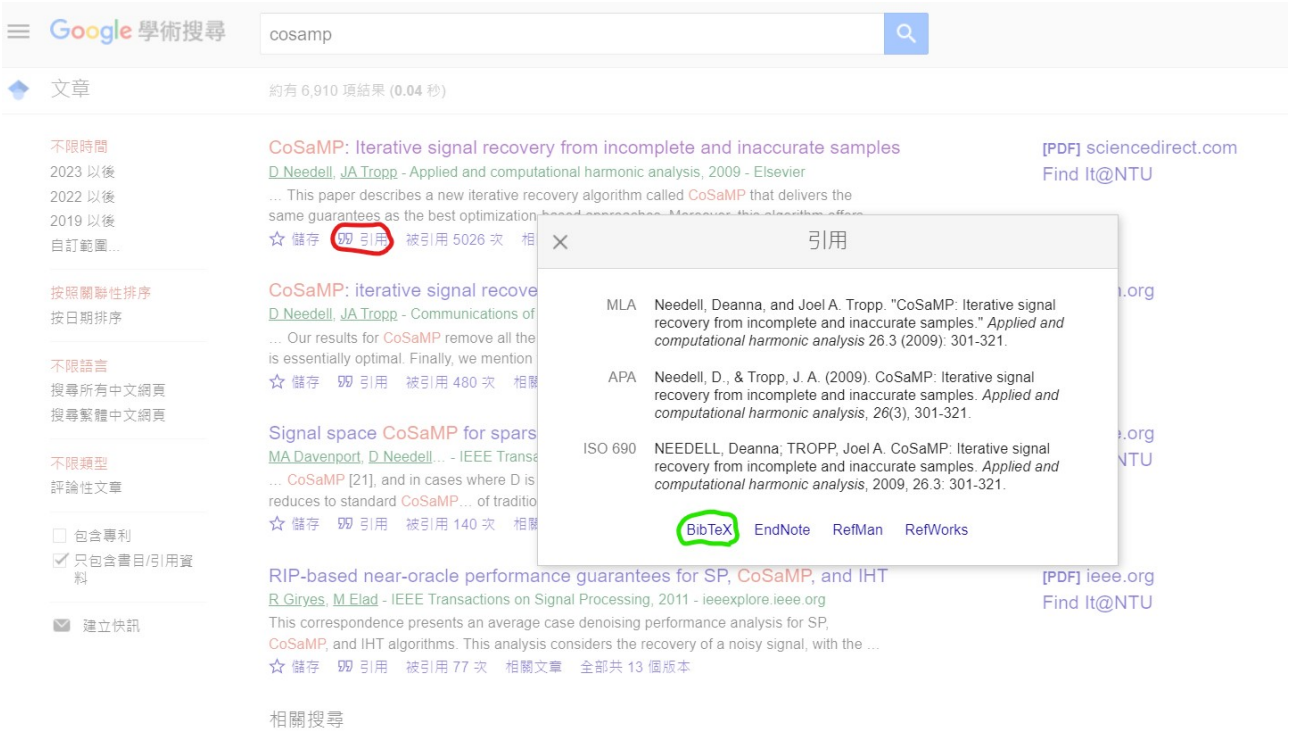

Figure 4: google scholar

Note that in the bibliography section, one can add the following command

$$\backslash nocite\{*\} \tag{99}$$

so that all sources in the bibliography document will be listed in the bibliography section of the article no matter each source is cited or not.

If one imports the "biblatex" package 43, he can use the following commands to create a bibliography.

$$\backslash addbibresource\{bib\ document\} \tag{100}$$
$$\backslash printbibliography[argument] \tag{101}$$

The command 100 has the same function as 97. We can customize the bibliography using 101. We give some examples of how to use it.

- "\printbibliography[title={text}]" : one can add a title of the bibliography using the argument "title".
- "\printbibliography[type=article,title={text}]" : one can only print entries whose type is "article" and let the title of this bibliography section be "text".
- "\printbibliography[type=book,title={text1}]"
  "\printbibliography[keyword={word},title={text2}]" : one can create multiple bibliography subsections. In this example, we create two, the first has entries whose type is "book" and the title of this subsection is "text1", and the second has entries that involve the keyword "word" and the title of this subsection is "text2".

To cite a source in the article, one also uses the same command as 98.

There are still another way to make a reference section. We use the following environment.

$$\begin{aligned}&\backslash begin\{thebibliography\}\{45\}\\&\backslash end\{thebibliography\}\end{aligned} \tag{102}$$

If there are at most nine sources, we should replace 45 with a one-digit number, i.e., 1 to 9. If there are at most ninety-nine sources, we should use a two-digit number, i.e., 10 to 99. To refer each source, we use the following command.

$$\backslash bibitem\{key\}\ \ bibtex\ reference \tag{103}$$

We give an example as follows.

$$
\begin{aligned}
&\backslash begin\{thebibliography\}\{45\}\\
&\quad \backslash bibitem\{cosamp\}\ \ D.Needell\ and\ J.A.\ Tropp,\\
&\quad "Cosamp:\ Iterative\ signal\ recovery\ from\ incomplete\ and\\
&\quad inaccurate\ samples,"\ Applied\ and\ computational\ harmonic\ analysis,\\
&\quad vol.\ 26,\ no.\ 3,\ pp.\ 301–321,\ 2009.\\
&\backslash end\{thebibliography\}
\end{aligned}
$$

0.7 Making Beamer Slides

"beamer" is a quite useful package to make presentation slides. In this section, we will introduce some commands that are specific for beamer.

1. Frame Environment

   - "frame" is the most fundamental environment for beamer. It is used for creating each slide. We can use the following commands to create one.

$$
\begin{aligned}
&\backslash begin\{frame\}\\
&\quad \backslash frametitle\{title\}\\
&\quad Contents\\
&\backslash end\{frame\}
\end{aligned}
\tag{104}
$$

     We can also combine the first and second line into

$$\backslash begin\{frame\}\{title\}$$

   - One can use the following command to make the title center in the frame.

$$\backslash setbeamertemplate\{frametitle\}[default][center] \tag{105}$$

   - If one wants to control the spacing surrounding the frame title,

he can use the following command.

$$\backslash addtobeamertemplate\{frametitle\}\{before\}\{after\} \tag{106}$$

The argument "before" controls the spacing above the frame title. For instance, if it is set as "\vspace*{$-0.1cm$}", the spacing above the frame title will decrease by 0.1cm. The argument "after" controls the spacing below the frame title. For instance, if it is set as "\vspace*{$0.2cm$}", the spacing below the frame title, i.e., the distance between the frame title and the contents, will increase by 0.2cm.

2. The same as what have been introduced in section 0.2.3, we also use the "title", "author" and "date" commands (i.e., 63, 64 and 65) in the preamble to typeset the cover of a beamer document. However, one cannot use 66 to make a cover but use

$$\backslash frame\{titlepage\} \tag{107}$$

instead after "\begin{document}".

3. One can also create the table of contents by the command 52. However, we should enclose it with the "frame" environment as the following shows.

$$\begin{aligned}&\backslash begin\{frame\}\\&\quad\backslash frametitle\{Table\ of\ Contents\}\\&\quad\backslash tableofcontents\\&\backslash end\{frame\}\end{aligned}$$

If one wants to show the ToC at each section, he can add the

following commands in the preamble.

$$\begin{aligned}&\backslash AtBeginSection[]\\&\{\backslash begin\{frame\}\\&\backslash frametitle\{Table\ of\ Contents\}\\&\backslash tableofcontents[currentsection]\\&\backslash end\{frame\}\}\end{aligned} \tag{108}$$

The argument "currentsection" means that only the current section will be highlighted. We can also use "\AtBeginSubsection[]" instead of "\AtBeginSection[]" so that the ToC will appear at the beginning of each subsection.

4. To highlight some texts, one can enclose it with a box. In beamer, there are three defined boxes to use, which are "block", "alertblock" and "examples" respectively. One can use the following command to invoke a box.

$$\begin{aligned}&\backslash begin\{block\}\{title\}\\&\backslash end\{block\}\end{aligned} \tag{109}$$

"block" can be replaced with "alertblock" or "examples", depending on which style of box one wants to use. "title" is the title of the box.

5. We often want to add references in the footnote of each slide. We can use the "biblatex" package 43 and import the bibliography document using 100. However, instead of using 101 to print the references, we use the following command

$$\backslash setbeamertemplate\{bibliography\ item\}\{\backslash insertbiblabel\} \tag{110}$$

and use the following command

$$\backslash footcite\{\} \tag{111}$$

to cite a source.

6. One can use the following command to set the theme of the beamer

slides.

$$\backslash usetheme\{Boadilla\} \tag{112}$$

"Boadilla" is one kind of possible themes. There are many more, e.g., Madrid, Berkeley, to use. The readers can refer to this guide for details of each theme and choose one according to personal preference.

0.8 Miscellaneous Commands

In this section, we will introduce miscellaneous useful commands.

1.

$$\backslash newcommand\{command\}[\#parameter][default]\{function\} \tag{113}$$

- We can use 113 to define a new command. To illustrate how we can do this, we directly give an example as follows.

  $$\backslash newcommand\{\backslash pb\}[3][2]\{(\#2 + \#3)\hat{}\#1\}$$

  "\pb" is the name of the new command. 3 means there are three arguments of this command. 2 is the default value of the first argument, i.e., #1. "(#2 + #3)^#1" is the function of this command. If we use this command as $\backslash pb\{a\}\{b\}$, then it means $(a+b)^2$. If we use this command as $\backslash pb[3]\{x\}\{y\}$, then it means $(x+y)^3$.
- If the command we intend to define has already been defined, an error message will be issued. In this situation, we should use "renewcommand" or "providecommand" instead.
- The usage of

  $$\backslash renewcommand\{command\}[\#parameter][default] \{function\} \tag{114}$$

is just the same as "newcommand". However, this command is used for re-define an already existing command. If this command has not been defined yet, an error message will be issued.

- The usage of

$$\backslash providecommand\{command\}[\#parameter][default]\{function\} \tag{115}$$

is also just the same as "newcommand". If this command really has not been defined yet, then "providecommand" does the same as "newcommand" does, while it does nothing if this command has already been defined.

2.

$$\backslash setlength\{lengthname\}\{value\} \tag{116}$$

- In LaTeX, there are various predefined lengths. We can use this command to set the values of them. In the following, we list some common lengths.
  - \baselineskip : vertical distance between lines in a paragraph
  - \linewidth : the width of the line in the current environment
  - \paperwidth : the width of the page
  - \paperheight : the height of the page
  - \parindent : the indent of a paragraph
  - \parskip : vertical spacing between paragraphs
  - \tabcolsep : the separation between columns in a table (in the tabular environment)
  - \itemindent : the indent of an item (in the item environment)

  For example, we can use

$$\backslash setlength\{\backslash parskip\}\{0.3cm\}$$

  to set the vertical spacing between paragraphs to 0.3cm and use

$$\backslash setlength\{\backslash itemindent\}\{0.5em\}$$

  to set the indent in front of an item to 0.5em.

3. To add horizontal spacing between items (e.g., texts, or figures), one can use the following commands.

$$\backslash quad \tag{117}$$
$$\backslash qquad \tag{118}$$
$$\backslash hspace^{*}\{1cm\} \tag{119}$$

The commands 117, 118 amount to fixed 1em and 2em horizontal spacing respectively. The command 119 provides a flexible setting of the horizontal spacing; that is, one can change 1cm to any other possible lengths (even negative lengths are possible).

4. To add vertical spacing between items, one can use the following commands.

$$\backslash smallskip \tag{120}$$
$$\backslash medskip \tag{121}$$
$$\backslash bigskip \tag{122}$$
$$\backslash vspace^{*}\{1cm\} \tag{123}$$

The commands 120, 121 and 122 amount to fixed 3pt, 6pt and 12pt vertical spacing respectively. The command 123 provides a flexible setting of the vertical spacing; that is, one can change 1cm to any other possible lengths (even negative lengths are possible).

5. One can use the following command to write a footnote.

$$\backslash footnote\{text\} \tag{124}$$

6. One can make a reference of almost everything that is numbered, e.g., sections, equations, figures, algorithms, etc. in the LaTeX document. We can jointly use the following two commands to meet our end.

$$\backslash label\{text\} \tag{125}$$
$$\backslash ref\{text\} \tag{126}$$

First, we use 125 to label a numbered item and then we can refer to

it using 126 anywhere in the LaTeX document. Although the text of the label can be arbitrary, it is better that we follow some rules so that it is more convenient to make a reference. For example, if we want to label an equation, the text can start with an identifier "eq:". If we want to label an algorithm, the text can start with an identifier "alg:".

7. One can start a new page by the following command.

$$\backslash newpage \tag{127}$$

8. One can highlight an item by enclosing it with a box. The following command can be used.

$$\backslash fbox\{text\} \tag{128}$$

For example, we can use the following source code to highlight a math expression : "3+2=5".

$$\backslash fbox\{3 + 2 = 5\}$$

The effect is $\boxed{3+2=5}$. If one wants to insert multiple lines in a box, he can use the "vardwidth" environment. We directly give an example as follows.

$$\begin{array}{l} \backslash fbox\{ \\ \quad \backslash begin\{varwidth\}\{\backslash textwidth\} \\ \quad\quad The\ math\ expression\ is \backslash\backslash \\ \quad\quad 3 + 2 = 5 \\ \quad \backslash end\{varwidth\} \\ \} \end{array}$$

This is the source code of the following box.

The math expression is
3+2=5

As we can see, the argument "\textwidth" sets the width of the box as the length of the longest line.

9. Special Characters :
   - "\backslash" is the command for \
   - "\^{}" is the command for ^
   - "\checkmark" is the command for ✓
10. If one wants to add some items on the top or bottom of another item, he needs to use the command

$$\backslash limits \tag{129}$$

For example, without 129, the following source code

$$\backslash sum\hat{}3\_\{i = 1\}2i$$

will be $\sum_{i=1}^{3} 2i$. However if with 129; that is,

$$\backslash sum\backslash limits\hat{}3\_\{i = 1\}2i$$

then it becomes $\sum\limits_{i=1}^{3} 2i$. In this case, the "\sum" is a math operator. If items in other cases are not math operators, we cannot directly append 129 after that item. Instead, we need to enclose this item with the following command to make it a math operator, and then append 129 after it.

$$\backslash mathop\{\} \tag{130}$$

For example, "\rightarrow" is not a math operator. Hence, we need to enclose it with 130 to make it a math operator. We give an example as follows.

$$\backslash mathop\{\backslash rightarrow\}\backslash limits\_\{test\}$$

This is the source code of the expression : $\mathop{\rightarrow}\limits_{test}$