LaTeX notebook

Wentao Lu

February 12, 2020

Contents

1 Section 1: a sample hierarchical structure

Sections are numbered and will appear in the table of contents.

1.1 Subsection 1

Subsections are also numbered and will appear in the table of contents.

1.1.1 Subsubsection 1

And so forth...

Paragraph 1 Paragraphs aren not numbered and won't show in the table of contents. Here this paragraph is given a name.

By contrast, this paragraph is **not** given a name, it's an *anonymous* paragraph. In practice, paragraphs do not need names unless we are writing math papers or technical documentation. It's also somewhat clumsy to start a new paragraph with \paragraph{} unless our sections and paragraphs are very deeply nested. More often, we can simply place a newline command \newline or \\\\ at the end of the previous paragraph.

LaTeX will automatically indent the first line of each paragraph that doesn't immediately follow a section heading. Indeed, this format looks better.

If you'd like to get rid of an indent anyway, use the \noindent command to start a paragraph, but this format does not look pretty and is not recommended since it breaks the default class structure.

2 Math

There are two major modes of typesetting math in LaTeX one is embedding the math directly into your text by encapsulating your formula in dollar signs and the other is using a predefined math environment.

2.1 Using inline math - embed formulas in your text

If you need to typeset a single math symbol or formula, surround it with dollar signs: this formula $f(x) = x^2$ is an example.

2.2 The *equation* and *align* environment

The most useful math environments are the *equation* environment for typesetting single equations and the *align* environment for multiple equations and automatic alignment. These two environments also take care of equation numbers for us.

The automatic numbering is a useful feature, but sometimes it's necessary to remove them for auxiliary calculations. LaTeX doesn't allow this by default, but we can include a package called *amsmath* and add a * after the environment name. The asterisk (e.g. equation*, align*) only indicates, that I don't want the equations to be numbered. Note that it is not possible to enter two equations in a single *equation* environment, it will result in a compilation error.

$$f(x) = x^2 \tag{1}$$

$$f(x) = x^3 + 3x + 13$$

The *align* environment will align the equations at the ampersand &. While it is possible to enter many equations in a single *align* environment, different equations have to be separated by a linebreak \\\\. There is no alignment when using the simple *equation* environment.

$$\begin{aligned} 1 + 2 &= 3 \\ 1 &= 3 - 2 \end{aligned}$$

2.3 Mathematical notation examples

All mathematical expressions have a unique command with unique syntax.

$$\begin{aligned} g(x) &= \frac{1}{x} \\ a(x) &= \int_b^a \frac{1}{3} x^3 \\ b(x) &= \frac{1}{\sqrt{x}} \end{aligned}$$

Use the *matrix* environment to typeset matrices. $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Scale parentheses with \left(\right) automatically. $\left(\frac{1}{\sqrt{x}}\right)$

3 Images and figures

3.1 Captioned images and figures

The *figure* environment takes care of the numbering and positioning of the image within the document. All figures will be indexed automatically and tagged with successive numbers when using the *figure* environment and the *graphicx* package.

At some point, you will notice that the figure doesn't necessarily show up in the exact place as you put your code in the .tex file. If your document contains a lot of text, it's possible that LaTeX will put the figure on the next page, or any other page where it finds sufficient space. To prevent this behavior, it's necessary to set the *float* value for the *figure* environment. Possible *float* values are:

- h (here) - same location
- t (top) - top of page
- b (bottom) - bottom of page
- p (page) - on an extra page
- ! (override) - will force the specified location

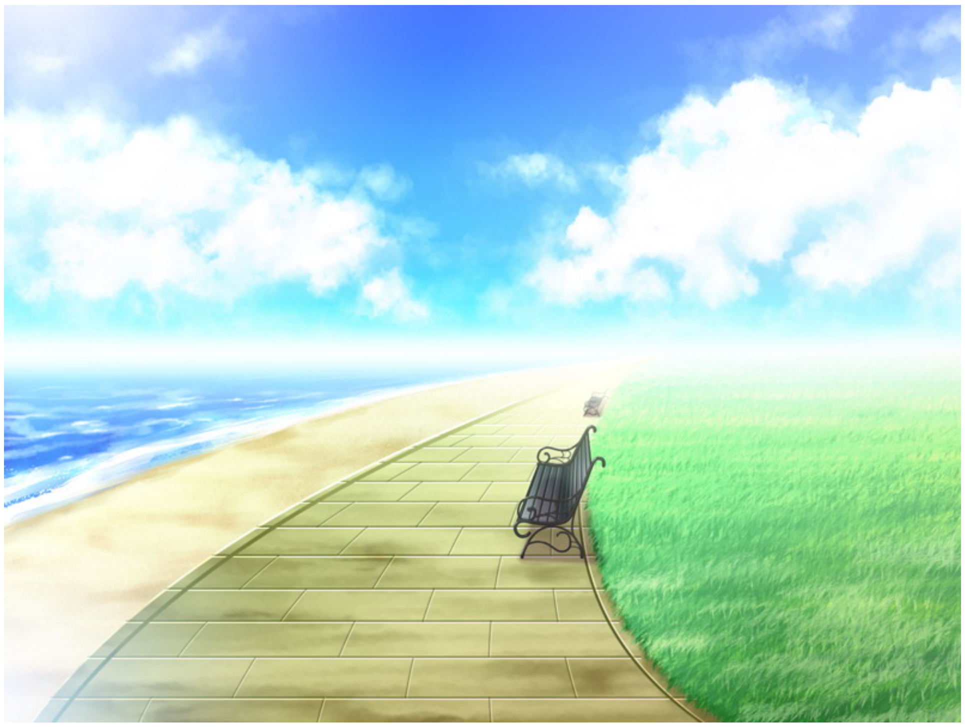

Figure 1: This is a sample image.

3.2 Multiple images and subfigures

The *subfigure* environment allows you to place multiple images at a certain location next to each other and the usage is pretty straightforward.

First, we need to include the *subcaption* package to the preamble. Next, we need to add multiple *subfigure* environments within a *figure* environment. Note that we must manually set the width of the image. If there are two images aligned next to each other, their widths should both be set to 0.4, but they still fill up the whole space. In such cases, we should always set the width to .1 less than we expect.

Likewise, if there are three images aligned next to each other, we should consecutively add three subfigures, each with a 0.2 \linewidth (1/3 - 0.1 = 0.2).

(a) Coffee. (b) More coffee.

Figure 2: Two cups of coffee next to each other.

(a) Coffee. (b) More coffee. (c) Tasty coffee.

(d) Too much coffee.

Figure 3: The same cup of coffee. Multiple times.

4 Bibliography

For simplicity, better compatibility and maintainability, it is recommended that we use *bibtex* and *biblatex* to create the bibliography. Briefly speaking, *bibtex* is an external program that processes bibliography information in your .bib file, while *biblatex* is a LaTeX package that formats citations and bibliographies. In other words, *bibtex* reads input from our database (i.e. the .bib file) and transforms that .bib file into a .tex understandable one, and then, the LaTeX package *biblatex* gives us access to use citing commands and display our bibliography in the final output .pdf file, this is how the pipeline works in general, despite that the real workflow is much more complicated.

Alternatively, we also have some other choices such as *biber* and *natlib*, which are not in the scope of this notebook. For those who are interested, this post contains a detailed introduction and comparison between all these tools.

4.1 Creating a .bib file

First we have to create a .bib file (bibliography database), which contains our bibliographic information. Here we have some examples.

```
@article{ahu61,
   author  = {Arrow, Kenneth J. and Leonid Hurwicz and Hirofumi
       Uzawa},
   title   = {Constraint qualifications in maximization problems},
   journal = {Naval Research Logistics Quarterly},
   volume  = {8},
   year    = 1961,
   pages   = {175-191}
}

@book{ab94,
   author    = {Charalambos D. Aliprantis and Kim C. Border},
   year      = {1994},
   title     = {Infinite Dimensional Analysis},
   publisher = {Springer},
   address   = {Berlin}
}

@incollection{m85,
   author    = {Maskin, Eric S.},
   year      = {1985},
   title     = {The theory of implementation in {N}ash equilibrium
       : a survey},
   booktitle = {Social Goals and Social Organization},
   editor    = {Leonid Hurwicz and David Schmeidler and Hugo
       Sonnenschein},
   pages     = {173-204},
   publisher = {Cambridge University Press},
   address   = {Cambridge}
}

@inproceedings{ah2006,
   author    = {Aggarwal, Gagan and Hartline, Jason D.},
   year      = {2006},
   title     = {Knapsack auctions},
   booktitle = {Proceedings of the 17th Annual ACM-SIAM Symposium
```

```
        on Discrete Algorithms},
    pages     = {1083-1092},
    publisher = {Association for Computing Machinery},
    address   = {New York}
}

@techreport{arrow48,
    author      = {Arrow, Kenneth J.},
    title       = {The possibility of a universal social welfare
        function},
    institution = {RAND Corporation},
    year        = {1948},
    number      = {P-41},
    type        = {Report}
}

@unpublished{fk1988,
    title  = {A theory of learning, experimentation, and
        equilibrium in games},
    author = {Fudenberg, Drew and Kreps, David M.},
    year   = {1988},
    note   = {Unpublished paper}
}
```

latex.bib

Now let's closely look at the reference blocks in our .bib file and explain each field. There are three mandatory fields for all document classes. Be it an article, a book or an incollection, we must have the **author, year, title** fields. The string on the first line after { is a label that we use to refer to the item when we cite it. Inside the **author =** {} braces is the list of authors connected by ”and”, not by commas. Here commas are used to separate different parts of an author's name, where we should write in the form of ”last name, first name”. Inside the **title =** {} braces is the title of the reference, without quotation marks. For the **year** field, it must consist of only digits, and the braces around it can be omitted. All other fields are somewhat self-explanatory so we won't bother to cover them.

For an article, we must also specify the **journal** field. For a book, the **publisher** field is also mandatory. Besides, the **author** field can also be replaced by an **editor** field, either of the two must be specified. For an incollection document class, fields **booktitle** and **publisher** are both mandatory. Likewise, **booktitle** is mandatory for inproceedings, **institution** is mandatory for techreport, and **note** is mandatory for unpublished document class.

Inside each reference block structure, the order of the fields is unimportant. Our .bib file can contain references we don't cite. *bibtex* will put in the list of references at the end of our paper only the ones that we cite.

4.2 Reading database and formatting references

Next, in the preamble we have to include the *biblatex* package, specify *bibtex* as the backend engine, choose a bibliographic formatting style, and then use the command \bibliography to tell LaTeX the location of our .bib file. Anywhere within the document, we can cite references using \cite and \autocite.

For example, this is a citation[1] which is numbered and appears in the footnotes, and this is a citation whose full text description is directly embedded in the paragraph but does not appear in the footnotes: Kenneth J. Arrow, Leonid Hurwicz, and Hirofumi Uzawa: *Constraint qualifications in maximization problems*. In: *Naval Research Logistics Quarterly* 8 (1961), pp. 175–191.

Here's another citation[2] for which we have also specified the page numbers.

Then, at the end of the document, we can use \printbibliography to create a separate reference page which will list all citations that have been cited.

Finally, it's worth mentioning that there are many bibliography styles that we can use when we include the package *biblatex*. This post has listed some of the most commonly used styles, with sample snapshots of how they look like. It is noteworthy that for different styles, the behaviors of the citing commands we mentioned above are also different. For some styles, the reference page could be empty, or the citation full text cannot be embedded. This is just a tutorial notebook, but when we write a formal paper, we should always stick to the style that is required by the university department.

[1] Charalambos D. Aliprantis and Kim C. Border: *Infinite Dimensional Analysis*. Berlin: Springer, 1994.

[2] Drew Fudenberg and David M. Kreps: "A theory of learning, experimentation, and equilibrium in games". Unpublished paper. 1988, pp. 32-35.

5 Footnotes

Footnotes are numbered automatically. For example, we can create a footnote of what coursera[3] is, and later refer to it as footnote 3.

6 Tables

Tables in LaTeX can be created through a combination of the *table* environment and the *tabular* environment. The *table* environment part contains the caption and defines the float for our table, i.e. where in our document the table should be positioned and whether we want it to be displayed centered. The \caption and \label commands can be used in the same way as for pictures. The actual content of the table is contained within the *tabular* environment.

The *tabular* environment uses ampersands & to separate columns and newline symbol \\\\ to separate rows. A horizontal line can be added with the \hline command. Within a column, if we want to have numbers aligned at the decimal point, we can include the *siunitx* package for this purpose, and setup the alignment using \sisetup.

6.1 A simple table

This is a simple table that works but is not very readable.

Table 1: A simple table.

Value 1	**Value 2**	**Value 3**
α	β	γ
1	110.1	a
2	10.1	b
3	23.113231	c

[3]An American online learning platform founded by Stanford professors Andrew Ng and Daphne Koller that offers massive open online courses (MOOC), specializations, and degrees.

Table 2: A simple table - numbers aligned.

Value 1	**Value 2**	**Value 3**
α	β	γ
1	110.10	a
2	10.10	b
3	23.11	c

6.2 Cells spanning multiple rows or multiple columns

Sometimes it's necessary to make a row or column span several cells. For this purpose we can use the *multirow* package. This package allows us to use the *multirow* and *multicolumn* environments, which make it easy to create a cell spanning multiple rows or columns.

Table 3: Multirow table.

Value 1	**Value 2**	**Value 3**
α	β	γ
12	1110.10	a
	10.10	b
3	23.11	c
4	25.11	d

If we want a cell to span multiple columns, we have to use the *multicolumn* command. The usage differs a bit from *multirow* command, since we also have to specify the alignment for our column. For example, in the row where we have a cell that spans two columns, there's only one column separator & (instead of two for all other rows).

Of course it's also possible to combine the two features, to make a cell spanning multiple rows and columns. To do this, we simply nest the *multirow* command inside the *multicolumn* command. The trick here is that, we have to add another *multicolumn* statement for as many rows as we're combining.

Table 4: Multicolumn table.

Value 1	**Value 2**	**Value 3**
α	β	γ
12		a
2	10.10	b
3	23.11	c
4	25.11	d

Table 5: Multirow and -column table.

Value 1	**Value 2**	**Value 3**
α	β	γ
1234		a
		b
3	23.11	c
4	25.11	d

6.3 Prettier tables with booktabs

The *booktabs* package provides much prettier horizontal separators. We can replace *\hline* with *\toprule*, *\midrule* and *\bottomrule*.

Table 6: Table using booktabs.

Value 1	**Value 2**	**Value 3**
α	β	γ
1	110.10	a
2	10.10	b
3	23.11	c

6.4 Multipage tables

If you have a lot of rows in your table, you will notice that by default, the table will be cropped at the bottom of the page, which is certainly not what you want. The package *longtable* provides a convenient way, to make tables span multiple pages. It's actually easier to use this package since the *longtable* environment combines the previous *table* and *tabular* environment into a single environment. Using this environment, we create a table, that is automatically split between pages, if it has too many rows.

Table 7: Multipage table.

Value 1	**Value 2**	**Value 3**	**Value 4**	**Value 5**
α	β	γ	ϵ	η
10	110.10	a63	apple	orange
29	10.10	b73	apple	orange
38	23.11	c65	apple	orange
47	523.10	a98	apple	orange
56	120.12	b15	apple	orange
65	3.72	c35	apple	orange
74	46.18	a90	apple	orange
83	14.00	b34	apple	orange
92	9.14	c72	apple	orange
01	532.24	a84	apple	orange
10	110.10	a56	apple	orange
29	10.10	b38	apple	orange
38	23.11	c93	apple	orange
47	523.10	a54	apple	orange
56	120.12	b66	apple	orange
65	3.72	c86	apple	orange
74	46.18	a33	apple	orange
83	14.00	b26	apple	orange
92	9.14	c18	apple	orange
01	532.24	a27	apple	orange
10	110.10	a58	apple	orange
29	10.10	b88	apple	orange
38	23.11	c94	apple	orange
47	523.10	a56	apple	orange
56	120.12	b00	apple	orange

6.5 Generating tables from .csv files

When writing papers, it is sometimes necessary to present a large amount of data in tables. Writing such tables by hand is very time-consuming and error-prone. To avoid this, we can simply import the data directly from

.csv (comma-separated value) files. For this purpose, we can use the *pgfplotstable* package. Similarly, there's also a package called *pgfplots* for automatic plot generation from .csv files. However, this notebook will not cover them here because such cases are rare. It's not useful to learn about the redundant features since we already have better tools such as Python for tackling with these tasks. The point is, for whatever task we have, wisely choose the most appropriate toolbox.

7 Source code

The required packages and settings have already been setup in the preamble part, and the comments should be self-explanatory. To actually insert source code in the document, there are mainly two ways. One way is to directly copy and paste source code into the *lstlisting* environment, but indentation could be troublesome since it will not ignore the indentation and spaces in the .tex file. As an alternative, it's preferable to use the \lstinputlisting command, which is able to read in a source code script file and outputs it to the .pdf, without messing up with our .tex file.

```
        # indentation from the .tex file not ignored
        # when we directly copy and paste code
        @functools.singledispatch
        def show(obj):
            print(obj, type(obj), "obj")
```

```
#include <iostream>
using namespace std;
int main() {  // no redundant indentation if we read from a script
    cout << "Hello, World!" << endl;
    return 0;
}  // the script path and filename is listed below the code block
```

support/HelloWorld.cpp

8 Hyperlinks

We can use the *hyperref* package which allows us to set links with a description as well as add bare urls to the document. After the hyperlink is inserted, if there's a colored box shown around the word. Don't worry, this box is not going to show up in our printed document, but only if we view it on the computer.

This is my homepage: neo-mashiro.

A bare URL without an description: `https://github.com/neo-mashiro`.

My email address is: neo-mashiro@hotmail.com.

9 Lists

Unordered lists use the *itemize* environment.

- One
- Two
- Three

Ordered lists use the *enumerate* environment.

1. One
2. Two
3. Three

Nested lists are the products of nested list environments.

1. One
    (a) Two
    (b) Three
2. Four

It's easy to change the numbering scheme or the bullets of a list.

– Dash

– Dash

∗ Asterisk

List of Figures

List of Tables

References

Aliprantis, Charalambos D. and Kim C. Border: *Infinite Dimensional Analysis*. Berlin: Springer, 1994.

Arrow, Kenneth J., Leonid Hurwicz, and Hirofumi Uzawa: *Constraint qualifications in maximization problems*. In: *Naval Research Logistics Quarterly* 8 (1961), pp. 175–191.

Fudenberg, Drew and David M. Kreps: “A theory of learning, experimentation, and equilibrium in games”. Unpublished paper. 1988.