LaTeX 的快速入門

蔡炎龍
政治大學應用數學系

2014 年版

目 錄

第 1 章

前言

1.1 這份文件的目的

這份文件是希望提供有心想學 LaTeX 的人, 一份快速入門的文件。我心目中的主要讀者是研究生, 所以我們會以最快的速度去討論怎麼樣把一篇論文完成, 包括 BibTeX 的論文管理。但另一方面來說, 我又希望可以更廣泛的讓 LaTeX 帶入一般文件處理, 而不只是在論文上面, 所以我會將中文 LaTeX 一併帶入。好在現在要用 LaTeX 編輯中文文件已比以前簡單太多, 我們其實不用特別多安裝什麼, 就可以使用!

中文方面我們採用 XeLaTeX, 這套系統的好處是可以用自己電腦裡的字型! 壞處是要特別用 XeLaTeX 編譯, 而且如果有共同合作者就要小心系統中是否有相同的字型。我們建議使用 cwTeX Q 五套字型, 不過當然個人使用你大可自由使用你想要的字型。

本文件不介紹安裝的問題, 安裝請參考我另一份文件《中英文 LaTeX 安裝與應用》, 本篇假設大家是安裝完成了。如果是在政大, 可以到應用數學系電腦室, 我們已經設好我們這篇文章討論應有的 LaTeX 環境。

另外, 爲了順利的使用 LaTeX, 你應該要有個順手的純文字編輯器。我個人推薦的編輯器如下:

- TeXWorks (所有平台都有)
- Vim (Unix-like 系統, 如果自認 Geek 級使用者)
- TeXShop (Mac OS X, 事實上我最偏好這一個)

我們只準備使用 PDFLaTeX 和 XeLaTeX, 這樣我們所有的 LaTeX 檔, 都直接產生 PDF 文件。中文編碼我們只準備使用 UTF-8, 這除了是個潮流, 也讓英文和中文基本上用的流程是完全一樣。

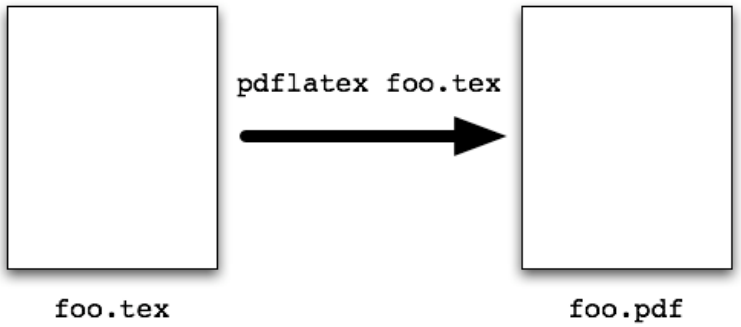
1.2 版本資訊

這份文件是在 2007 年 7 月 16 日完成第一版初稿, 7 月 18 日改用 XeLaTeX 進行改版。2013 年再做修正, 2014 年在 GitHub 上公開所有檔案。

第 2 章

LaTeX 極速入門

很多人說 LaTeX 很難, 其實 LaTeX 實在沒什麼難的。我們只不過是做一個純文字檔, 存成 `.tex` 這樣的檔案, 然後使用 `pdflatex` 這個指令, 馬上就產生一篇高品質的 PDF 文件。



我們這裡很快的來看一下這個 `.tex` 的純文字檔應該長什麼樣子。

2.1 最簡單的 LaTeX 文件

最簡單的 LaTeX 檔案是長這個樣子。

```
\documentclass{article}
\begin{document}

% 內文, 文章的內容

\end{document}
```

可以試打一些內容進去看看, 存成 `.tex` 檔, 再用 `pdflatex` 編譯。要注意目前還不能用中文。

2.2 完整的 LaTeX 格式

一份完整的 LaTeX 文件的架構大概如下。

```
\documentclass{article}

% 設定區, 我們還不會

\title{文章的標題}
\author{作者}

\begin{document}
\maketitle

% 內文, 文章的內容

\end{document}
```

框起來的部份就是我們需要打字進去的地方。你可以試打一些東西進去, 然後 LaTeX 會自動幫你印出標題、作者、有分節的文件。是不是非常容易? LaTeX 的一個特性就是, 你可

以專注在文章的內容上，要美化什麼的可以最後慢慢調。

第 3 章

LaTeX 數學式子基礎

3.1 數學模式

很多人聽說 LaTeX, 都是聽說它對數學符號處理功力很強。我們來看看要怎麼打入數學符號。LaTeX 有兩種數學模式, 分別是:

- 隨文模式 (`inline mode`)
- 展示模式 (`display mode`)

我們來看看怎麼樣使用。

隨文模式

所謂隨文模式就是數學式子要插在文中, 使用的方式是把數學式子放入兩個 $ 的符號中。比方說下面這個例子:

The formula $f(x)=x^3-2x+6$ is important in this case.

```
The formula $f(x)=x^3 - 2x +6$ is
important in this case.
```

3.2 展示模式

所謂展示模式的數學式子, 是把數學式**獨立**、**置中**表示。展示模式有很多下指令的方式, 我們可以把數學式子用 "$$ ...$$", "\[...\]", 或 "\begin{equation} ...\end{equation}" 等方式表示, 比方說

The formula

$$f(x)=x^3-2x+6$$

is important in this case.

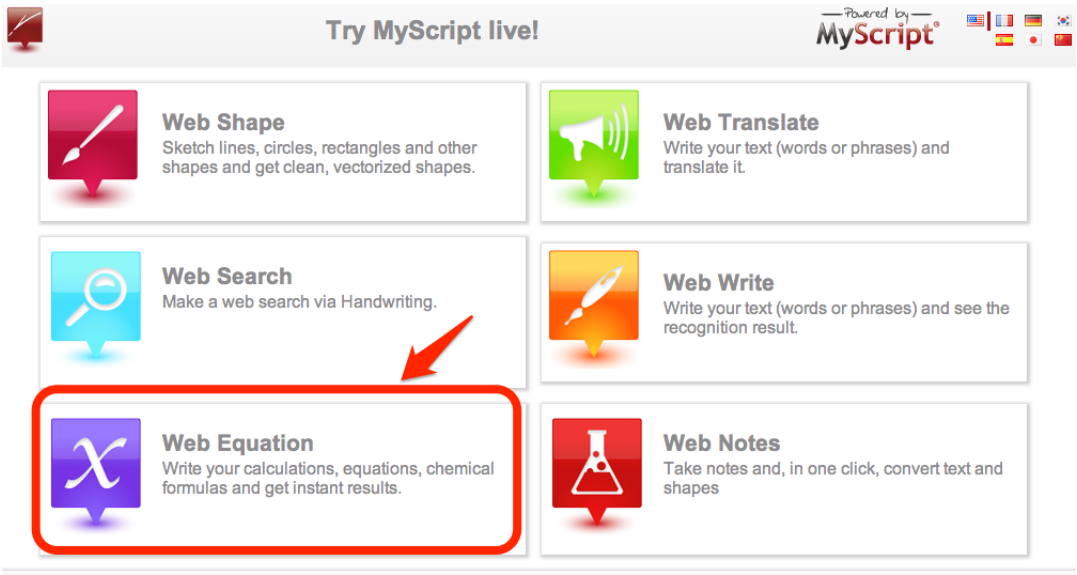
```
The formula
\[
f(x)=x^3 - 2x +6
\]
is important in this case.
```

3.3 好用的 LaTeX 工具

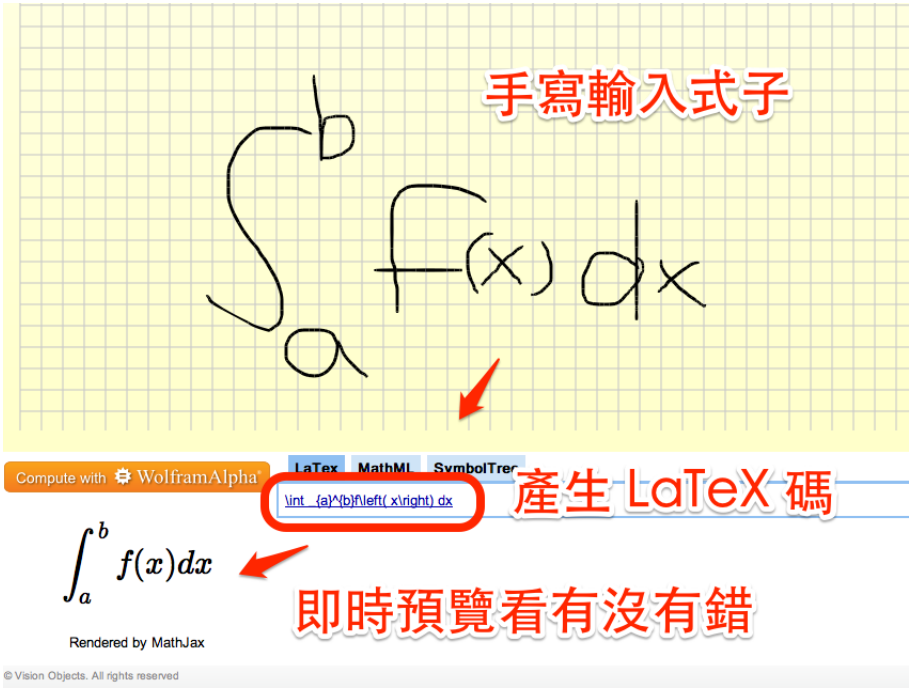
3.3.1 MyScript 的 Web Equation

MyScript 是一個「手寫辨識」的程式庫, 他們做了網路示範版本。你可以到他們示範網頁:

`http://webdemo.visionobjects.com/portal.html?locale=en`



我們要用的是 Web Equation。然後你就用手寫出一個數學式子, Web Equation 辨識輸出 LaTeX 碼, 而且即時預覽, 讓你知道辨識結果有沒有問題!

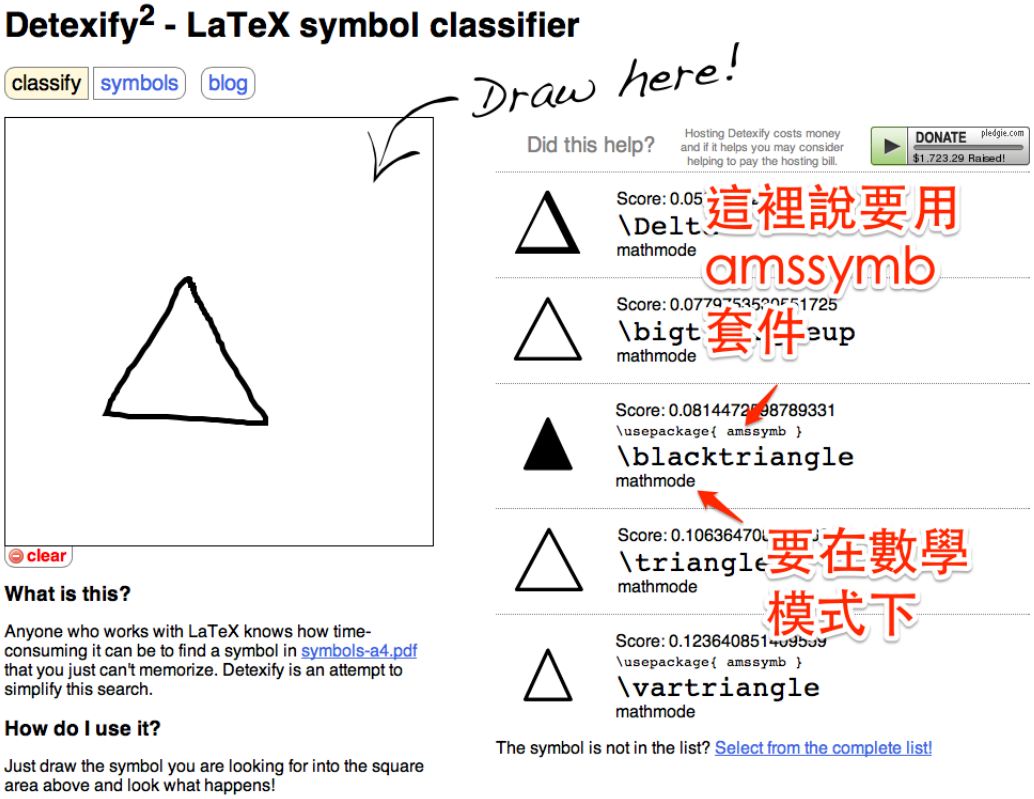


3.3.2 Detexify 手寫符號

接著我們介紹和 Web Equation 有點像, 但這次著重在符號上。我們常常會碰到某個 LaTeX 符號不知怎麼打, 最容易的方式當然是寫出我們要的符號, 電腦就告訴我們怎麼下指令! Detexify 就是做這樣的工作:

`http://detexify.kirelabs.org`

你手寫一個符號, Detexify 會列出它覺得長得像的, 還告訴你需不需要特別套件、要在文字模式還是數學模式下使用。



3.3.3 LaTeX 符號大全集

如果你有什麼符號, 實在不知怎麼打出來, 用了我們介紹的方式也找不到。現在介紹一個殺手 – Scott Pakin 的 “The Comprehensive LaTeX Symbol List”。你可以在這裡下載:

`http://www.ctan.org/tex-archive/info/symbols/comprehensive/`

這有完整到什麼程度呢? 完整到你沒有想過 LaTeX 可以打出的「符號」都可以找到!

第 4 章

使用 XeLaTeX 來打中文

4.1 XeLaTeX 的快速入門

前一章介紹的 CJK-LaTeX 麻煩的地方是要安裝特別的字型, 我們可不可以直接使用系統上的字型呢? 答案是可以的! 只要使用 XeLaTeX, 你可以使用自己電腦裡的字型。

XeLaTeX 是由 Jonathan Kew 開發的, 中國南開大學孫文昌教授爲 XeLaTeX 寫了對中文使用者很方便的 xeCJK 套件, 省了很多設定上的麻煩。所以我們主要介紹 xeCJK 的用法。在編譯時還是要用 XeLaTeX 編譯。

我們首先要選擇系統裡的一個字型, 找到它的名稱, 接著就照以下範例打, 就可以完全不用裝字型 (或依《中英文 LaTeX 安裝與應用》安裝 cwTeX-Q 字型), 馬上使用中文 LaTeX! 現在假設我們想用 cwTeX-Q 的明體字 (`cwTeX Q Ming Medium`), 我們可以這樣做:

```
\documentclass{article}

\usepackage{xeCJK}
\setCJKmainfont{cwTeX Q Ming Medium}
```

```
\begin{document}

文章內容如一般 \LaTeX, 還可打中文!

\end{document}
```

出來的結果如下:

文章內容如一般 LaTeX, 還可打中文!

所以基本上就是一般的 LaTeX, 只有先引用套件

```
\usepackage{xeCJK}
```

然後再設定我們要用的字型

```
\setCJKmainfont{cwTeX Q Ming Medium}
```

就可以了!

唯一要小心是在編譯時, 我們要選用 XeLaTeX 編譯。比方說, 我們的 LaTeX 檔叫 `foo.tex`, 那我們就可以這樣編譯:

```
xelatex foo.tex
```

4.2 XeLaTeX 中文字型設定

我們打文章, 很可能正常的字體用明體, 粗體字用黑體, 這樣 XeLaTeX 下要怎麼做呢? 答案是我們在設定中文字型時可以加一些參數。比方說

```
\setCJKmainfont[BoldFont={cwTeX Q Hei Bold}]
{cwTeX Q Ming Medium}
```

這樣主要的字體仍然是使用 `cwTeX Q Ming Medium` (明體), 而粗體字會用 `cwTeX Q Hei Bold`。

我們看以下的例子。

這裡就是最**重要**的地方。

```
這裡就是最{\bf{重要}}的地方。
```

我們更可以再設用楷書爲「斜體」字, 如

```
\setCJKmainfont[BoldFont={cwTeX Q Hei Bold},
ItalicFont={cwTeX Q Kai Medium}]{cwTeX Q Ming Medium}
```

4.3 XeLaTeX 英文字型設定

XeLaTeX 當然不只可以用系統的中文字型, 英文字型也可以任意選用! 現在假設我們中文字型設好了, 可不可以顯示英文時用英文字型呢? 當然可以! 我們只要設

```
\setromanfont{Times}
```

就可以把英文字型設成 “Times” 字型。但這樣設我們用一般英文 LaTeX 打標點等方式會看來怪怪的。這時我們可以設

```
\setromanfont[Mapping=tex-text]{Times}
```

就好了。我們還可以分別設無描邊字型、等寬字型等等, 現舉例如下:

```
\setromanfont[Mapping=tex-text]{Times}
\setsansfont[Scale=MatchLowercase, Mapping=tex-text]{Arial}
\setmonofont[Scale=MatchLowercase]{Courier}
```

新用的 `Scale` 參數設 “MatchLowercase” 是保證各字型小寫的 “x” 是一樣大小的。

第 5 章

使用 AMS-LaTeX

5.1 引入 AMS-LaTeX

AMS 美國數學學會的 LaTeX 套件已然成爲一種標準。通常會用到的有三個套件:

- `amssymb`: 提供一些原本 LaTeX 沒有的符號, 比方說 $\mathbb{R}$, $\mathbb{C}$, 等等。
- `amsmath`: 提供一些好用的環境, 比方說 `align` 環境等等。
- `amsthm`: 提供比較好的使用定理、定義等的環境。

如果使用一般的 `article class`, 建議每次都把三個套件讀進來:

```
\usepackage{amssymb, amsmath, amsthm}
```

5.2 使用 AMS Article Class

使一個使用 AMS-LaTeX 的方式是使用 AMS 提供的個文章類型, 叫 AMS Article。要使用就是設定使用 `amsart`:

```
\documentclass{amsart}
```

它會自動讀入 `amsmath`, `amsthm` 兩個套件, 和部份 `amssymb` 套件 (比方說有 $\mathbb{R}$)。如果需要全套的 `amssymb`, 還是要自行讀入:

```
\usepackage{amssymb}
```

第 6 章

定理環境的使用

我們寫數學文章, 總會出現定義, 定理, 證明等等。我們在 LaTeX 要處理這些東西是很容易的。

6.1 基本定理環境

在設定區設定:

```
\newtheorem{thm}{Theorem}
```

意思是我們要先建一個新的定理環境, 叫做 thm, 顯示時標示爲 “Theorem”。比方說:

Theorem 1. *The statements of the theorem.*

```
\begin{thm}
The statements of the theorem.
\end{thm}
```

6.2 定理的編號

你可以發現定理的編號會自動從 1, 2, 3, 等等編下去。但是有時我們要依節次來標, 比方說第一節的第一個定理的編號是 1.1, 然後 1.2, 1.3, 這樣下去, 要怎麼做呢? 很容易, 加個 `section` 參數就好。比如說在設定時我們設:

```
\newtheorem{thm}{Theorem}[section]
```

那麼在本節 (第 6 節) 使用定理環境會變成下面這個樣子。

Theorem 6.2.1. *The statements of the theorem.*

```
\begin{thm}
The statements of the theorem.
\end{thm}
```

我們如果定了兩個定理環境, 他們原本是互不相干的, 所以會各自編號。比如說如果我們有設了 `thm`, `lem` 兩個定理環境:

```
\newtheorem{thm}{Theorem}
\newtheorem{lem}{Lemma}
```

引用起來會是像這樣子:

Lemma 1. *The statements of the lemma.*

Theorem 1. *The statements of the theorem.*

```
\begin{lem}
The statements of the lemma.
\end{lem}
\begin{thm}
The statements of the theorem.
\end{thm}
```

這樣子的編號方式, 我們無法知道 Lemma 7 和 Theorem 3 倒底是哪一個先出現, 哪一個後出現。要用統一的編號。比方說引理 1 之後是定理 2 等等, 就要用下面的方式宣告定理環境。

```
\newtheorem{thm}{Theorem}
\newtheorem{lem}[thm]{Lemma}
```

請比較和以前有什麼不一樣。

Lemma 1. *The statements of the lemma.*

Theorem 2. *The statements of the theorem.*

```
\begin{lem}
The statements of the lemma.
\end{lem}
\begin{thm}
The statements of the theorem.
\end{thm}
```

最後, 如果需要完全沒有編號的定理環境, 就要像下面這樣加上星號。

```
\newtheorem*{mainthm}{Main Theorem}
```

Main Theorem. *The statements of the theorem.*

```
\begin{mainthm}
The statements of the theorem.
\end{mainthm}
```

6.3 不同定理風格 (amsthm)

使用 `amsthm`, 可以指定三種不同的定理風格: `plain`, `definition`, 和 `remark`。使用方式是在定義定理環境之前, 先下達:

```
\theoremstyle{[定理風格]}
```

舉例來說, 假設我們定義下列的定理環境:

```
\theoremstyle{plain}
\newtheorem{thm}{Theorem}

\theoremstyle{definition}
\newtheorem{defn}{Definition}

\theoremstyle{remark}
\newtheorem{rmk}{Remark}
```

會產生如下的效果。

Theorem 1. *The statements of the theorem*.

Definition 1. The statements of the definition.

Remark 1. The statements of the remark.

```
\begin{thm}
The statements of the theorem.
\end{thm}
\begin{defn}
The statements of the definition.
\end{defn}
\begin{rmk}
The statements of the remark.
\end{rmk}
```

6.4 定理的引用

我們會引用到的定理, 就用 \label{引用代碼} 來標記。比方說

Theorem 2. *The statements of the theorem.*

```
\begin{thm}\label{T:major}
The statements of the theorem.
\end{thm}
```

要引用的時候, 就是用 ~ \ref{引用代碼}:

Please refer to Theorem 2.

```
Please refer to
Theorem~\ref{T:major}.
```

第 7 章

插入圖片

7.1 插入圖片的基本方法

這裡建議使用 graphicx 套件:

```
\usepackage{graphicx}
```

假設我們要插入 pic.png 這個圖檔, 使用

```
\includegraphics[width=5cm]{pic.png}
```

即可。自然, width 是可依你需要設定的。建議使用的圖檔格式爲:

```
.png, .pdf, .jpg
```

7.2 圖片置中

要把圖片置中, 使用 center 環境即可:

```
\begin{center}
\includegraphics[width=5cm]{pic.png}
\end{center}
```

7.3 figure 的使用方法

上面的圖形基本上是在那插入, 就會放在那裡。但正式排版中, 常會依版面情況調整位置, 且會有提示文字。這時要使用 figure 環境。一般要置中, 又有說明的圖會這樣引用:

```
\begin{figure}
\begin{center}
\includegraphics[width= 圖形寬度]{檔案名稱}
\end{center}
\caption{圖形的文字說明}
\end{figure}
```

LaTeX 會幫你把圖形放在它認爲合適的地方。如果你對放的位置很有意見, 可以加入 h, t, b, 或 p 等參數改變。比方說使用

```
\begin{figure}[h] ...
```

這些參數代表你希望放置的位置分別是:

- h: 放在此處
- t: 放在頂端
- b: 放在底端
- p: 在本頁

事實上你也可以同時用 [htbp], 這是告訴 LaTeX 你希望放在這一頁, 但到底怎麼放讓 LaTeX 自己「看著辦」。

7.4 圖形的引用

圖形的引用其實和定理引用一樣。你只要在想引用的圖提示文字加上 \label, 比方說:

```
\caption{圖形的提示文字}\label{引用代碼}
```

要引用時則如下範例:

```
參考圖 ~\ref{引用代碼}...
```

就可以了。

第 8 章

列表

我們這裡介紹怎麼樣在 LaTeX 使用文書處理常用的列表。

8.1 基本列表

要分點列表的基本方式如下:

- 微積分
- 高等微積分
- 線性代數

```
\begin{itemize}
\item 微積分
\item 高等微積分
\item 線性代數
\end{itemize}
```

8.2 數字列表

我們再看要以 1, 2, 3 等標示的列表怎麼做。

1. 微積分
2. 高等微積分
3. 線性代數

```
\begin{enumerate}
\item 微積分
\item 高等微積分
\item 線性代數
\end{enumerate}
```

8.3 定義型列表

第三種定義型列表使用方式如下。

微積分 大一必修

高等微積分 大二必修

線性代數 大二必修

```
\begin{description}
\item [微積分] 大一必修
\item [高等微積分] 大二必修
\item [線性代數] 大二必修
\end{description}
```

第 9 章

陣列和表格

9.1 陣列的使用

陣列就是如同矩陣型的排列。我們可以看一下一個例子。

$$
\begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{array}
$$

```
\[
\begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{array}
\]
```

這裡要說明一下。

```
\begin{array}{ccc}
```

是表示要用陣列, 這個陣列有三行, 每一行都要對齊中間 (c)。對齊的方式有三種選擇:

- c: 對齊中間
- l: 對齊左邊
- r: 對齊右邊

我們要一列一列輸入, 要換行時用 "\\" 換行, 每一欄用 "&" 隔開。

9.2 表格的使用方式

表格的使用方式非常接近陣列的使用。

item 1	item 2	item 3
1	2	3
4	5	6

```
\begin{tabular}{ccc}
item 1 & item 2 & item 3 \\
1 & 2 & 3 \\
4 & 5 & 6
\end{tabular}
```

如果要加橫線, 加入 \hline:

item 1	item 2	item 3
1	2	3
4	5	6

```
\begin{tabular}{ccc} \hline
item 1 & item 2 & item 3 \\ \hline
1 & 2 & 3 \\ \hline
4 & 5 & 6 \\ \hline
\end{tabular}
```

加直線更方便, 在對齊設定那加就可以了:

item 1	item 2	item 3
1	2	3
4	5	6

```
\begin{tabular}{|c|c|c|} \hline
item 1 & item 2 & item 3 \\ \hline
1 & 2 & 3 \\ \hline
4 & 5 & 6 \\ \hline
\end{tabular}
```

9.3 一般的括號和會變大的括號

在 LaTeX 裡, 要打出小括號到大括號方法如下:

- 小括號: ( )
- 中括號: [ ]
- 大括號: \{ \}

問題是如果你想打一個矩陣, 配合 `array` 使用, 會出現一個好笑的結果:

$$
(\begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{array})
$$

```
\[
(\begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{array})
\]
```

要改正這個缺點, 我們要用「可自調大小的括號」。方式很簡單, 在左邊的括號前加 `\left`, 右邊加 `\right` 就可以。比方說:

$$\left(\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}\right)$$

```
\[
\left(\begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9
\end{array}\right)
\]
```

9.4 陣列和括號的應用

注意前面「會變大小的括號」是一定要成對出現的。如果我們已經用了 \left, 一定要有 \right。不過, 我們其實可以只要一邊, 比方說左邊的括號, 而右邊可用 \right. 表示不要顯示任何括號。我們來看一個應用:

$$|x| = \left\{ \begin{array}{rr} x, & \text{if } x \geq 0 \\ -x, & \text{if } x < 0 \end{array} \right.$$

```
\[
|x| = \left\{
\begin{array}{rr}
 x, & \mbox{if $x \geq 0$} \\
-x, & \mbox{if $x < 0$}
\end{array} \right.
\]
```

這裡我們新學一件事, 那就是**如果數學式中我們要打入一些純文字, 可以用 \mbox 指令**。而在 \mbox 裡面的是純文字, 所以再打數學符號就要加上錢的符號了。

第 10 章

BibTEX 入門

10.1 為什麼要用 BibTEX?

使用 BibTEX 並不是在 LATEX 下引用論文唯一方式。開始的時候, 用 BibTEX 可能還會覺得比較麻煩, 因爲你爲了論文, 還要建立一個 `.bib` 純文字檔, 內容是你要引用的論文資訊。我們爲什麼要這麼麻煩呢? 把要引用的論文全寫在本文那個 LATEX 檔不是很好嗎? 這其實不是那麼好。

我們常會碰到這個情況: 你找了一堆論文, 其實你也不知道哪篇對你有用, 所以暫時沒有放入你文章後面的參考資料清單中。有一天, 你發現某某篇有用, 結果一時之間找不到那篇在哪裡! 就算你眞的把論文弄好, 每次還要手動排序! 萬一有一天你發現你論文格式用的不合教授/期刊的要求, 你還得重新修正!

如果用 BibTEX, 你完全不用擔心這件事! 你覺得有可能參考的, 你就把它建檔進去, **BibTEX 只會列出你真的有引用的文章**, 而且**幫你排序**, 你也可以**隨時指定, 更換論文排列和引用樣式**。更方便的是, 如果你下一篇文章也是同領域, 使用 BibTEX 可以再用完全一樣的 `.bib` 檔。你也可以爲了新的文章加入新的論文, 但是不會影響原來文章的編譯。更棒的事可能是, 你可以和大約同領域的教授, 同學共同分享 `.bib` 檔, 這樣大家都可

以省些力氣!

10.2 BibTEX 的檔案內容

BibTEX 的檔案是一個純文字, 以 .bib 爲副檔名的檔案, 內容就是紀錄每一篇你有興趣文章的資訊。我們看一個例子會更加明白:

```
@article{tx07,
Author = {Tsai, Yen-lung and Xia, Eugene Z.},
Journal = {Proc. Amer. Math. Soc.},
Volume = {135},
Number = {8},
Pages = {2365-2367}
Title = {Non-abelian local invariant cycles},
Year = {2007}}
```

這裡最前面 tx07 是我們自己設的文章引用代碼, 後面的 Author 自然是作者, Journal 是期刊名稱等等。其實因爲 LATEX 和 BibTEX 可以說是一個標準, 有很多方式可以「生成」這些編碼, 我們不一定要這樣自己慢慢照這樣打入, 所以這裡就不詳細說明了。

10.3 文章的引用

最要注意的是上面的例子中一開始的 "tx07" 是我們要引用這篇文章的引用代碼。你在本文中要引用的地方, 請打入

```
~\cite{tx07}
```

就可以了。這裡要討論這個引用代碼的編法。很多人照 LaTeX 之父 Leslie Lamport 的範例, 使用作者: 代碼 做爲引用方式。比方說我們要引用 Yen-lung Tsai 在 2012 年的 “Working with tropical meromorphic functions of one variable” (咳, 我知道沒人會引用, 所以只好自己當範例), 我們可能會用

```
tsai:tropicalmero
```

當做引用代碼。這樣的方式, 在實作時發現相當困擾, 因爲有時很難想到一個好的代碼, 有時好不容易弄了一個很棒的代碼, 要引用時忘了自己原來的代碼, 還要回頭去查。因此, 像不少 LaTeX 使用者的建議一樣, 我會建議直接用 { 作者 + 年份 } 做爲引用代碼。比方說剛剛這篇文章, 我們就用:

```
tsai12
```

這樣引用。你在研究的過程, 你很容易記下來某某人在某某年做了什麼, 所以這樣引用其實更合理方便。兩位或以上的作者, 我們可以用每位作者「姓氏的第一個字母」加再上年分當引用代碼。比如說, Tsai 和 Xia 在 2007 年的文章, 我們就可以照前面用過的 `tx07` 來引用。

當然, 這只是提供參考, 你可以找出自己合適的方式。

10.4 告訴 LaTeX `.bib` 檔在哪裡

這裡還有個問題, 就是你怎麼讓 LaTeX 知道你要使用的 BibTeX 檔在哪裡? 假定我們把我們 BibTeX 檔存成 `reference.bib`, 那麼使用方式就是在 \end{document} 之前加入兩行:

```
\bibliographystyle{plain} % 使用 plain 格式, 可換其他格式
\bibliography{reference} % 使用 reference.bib
```

就可以了。

10.5 BibTEX 的編譯

使用 BibTEX 的 LATEX 文件, 編譯過程有時有點讓人困惑。我們這裡假設以 `foo.tex` 爲我們的 LATEX 檔 (BibTEX 檔叫什麼無妨, 只要我們在文中引用正確的 `.bib` 檔就可以), 我們要產生正確引用的 PDF 檔要用

```
pdflatex foo.tex
bibtex foo.tex
pdflatex foo.tex
pdflatex foo.tex
```

第一次的 `pdflatex`, 你的 LATEX 系統「看到」你要引用的文章, 可是它根本沒有資訊。所以再叫 BibTEX 把需要文章的資訊都抄下來。第二次/第三次 `pdflatex` 就是依得到的文章資訊, 可能排序, 決定出現的篇號, 再填入引用的地方。

有個好消息是, 如果使用 TeXWorks, **你事實上可以用** "`PDFLaTeX+MakeIndex+BibTeX`" **這個預設的編譯法一次搞定**! 萬一沒有搞定再按同樣的方式編譯一次, 不用切來切去。使用 XeLATEX 也有對應的 "`XeLaTeX+MakeIndex+BibTeX`" 。

10.6 如何建立 BibTEX 檔?

其實我們不需要自己打入那些 BibTEX 的資料, 有很多更簡單的方式!

使用 MathSciNet

MathSciNet 是美國數學學會 (AMS) 推出的線上論文查詢系統。幾乎所有重要數學期刊的文章都可以查到, 所以你可以確定某些文章是不是登出來, 在哪登出來的。MathSciNet 可以把找到的文章直接顯示成 BibTEX 格式, 所以你只要 `copy` 和 `paste` 就可以轉貼到你的 `.bib` 檔裡面, 完全不用自己打字!

使用 BibTeX 輔助軟體

BibTEX 輔助軟體可以有比較親切的界面, 讓你方便輸入文章資訊。更重要的是, 如果你的電腦裡有那篇文章, 你可以做一個連結。有一天你想看看這篇文章, 點個兩下就可以打開, 所以你再也不用擔心找不到那篇文章! 我個人推薦兩套在不同平台上免費 BibTEX 輔助軟體:

JabRef (Winodws, Unix-like 或其他可跑 Java 的系統)

BibDesk (Mac OS X)