

國立台灣大學機械工程學系
系統最佳化實驗室

Latex Equation and Debug

By Tzu-Chieh Hung

03/20/2014

第 1 章 數學排版

在 L^AT_EX 的數學模式中，所有的文字、符號預設都為斜體字，如果要在數學模式中插入一般的正常文字，需要使用 `\mbox{}` 或 `\textmr{}` 將文字包圍起來。除了字型會被自動調整之外，所有的空白也都會被忽略。如果需要插入空白，可以使用調整橫向空間的相關指令，如：`\hspace{}`、`\quad`、`\thinspace` 等指令進行調整。

L^AT_EX 中的數學模式可分為隨文模式（math in line mode）與展式模式（math display mode）兩種。隨文模式用在文字與數學式混合在一起的排版；展式模式則會將數學式單獨形成一行，且會與正常文字有一定的空間區隔。

1.1 隨文模式（math in line mode）

以下三種方法可以直接進入隨文數學模式：

1. `$` 數學式 `$`：這個方法通常用來插入簡單的數學式 $f(x)$ 或符號 $\mu + \sigma$ 。

這個方法通常用來插入簡單的數學式 `$f(x)$` 或符號 `$\mu + \sigma$`。

2. `\begin{math}` 數學式 `\end{math}`：如果遇到比較長的數學式，則可以改用環境的方式來輸入，不過要注意的是在這個環境的上下不能留空白行，否則它會另起段落，這樣就失去隨文模式的意義了。 $f(x) = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + \dots$ 。各位可以看到在隨文數學式中，L^AT_EX 會自動換行。

如果遇到比較長的數學式，則可以改用環境的方式來輸入，不過要注意的是在這個環境的上下不能留空白行，否則它會另起段落，這樣就失去隨文模式的意義了。

```
\begin{math}
f(\mathrm{x}) = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9
+ \cdots
\end{math}
```

。各位可以發現在隨文數學式中，`\LaTeX` 會自動換行。

3. `\(` 數學式 `\)`：這是 `\begin{math}` 數學式 `\end{math}` 的省略寫法。在隨文數學式中，數學式的整體高度會被限制為字體高度，如果要取消限制，可以使用 `\displaystyle` 強制顯示為展式模式。 $f(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$ 。

```
\(
f(x,\mu,\sigma) = \frac{1}{\sigma \sqrt{2\pi}}
\displaystyle\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]
\)
```

1.2 展式模式 (math display mode)

展式數學式通常用來描述獨立與比較複雜的數學式，它會讓數學式獨立成爲一行，且會使用較大的數學符號與字體，有需要的話，也可以加入編號。

以下三種方法可以進入一般的展式模式：

1. `\begin{displaymath}` 數學式 `\end{displaymath}`

$$f(x, \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2 \right]$$

```
\begin{displaymath}
f(x,\mu,\sigma) = \frac{1}{\sigma \sqrt{2\pi}}
\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]
\end{displaymath}
```

2. `\[` 數學式 `\]` 這是 `\begin{displaymath}` 數學式 `\end{displaymath}` 的省略寫法。這兩種展式數學式都不會有方程式編號。
3. `\begin{equation}` 數學式 `\end{equation}`：這是有方程式編號的數學式。如果要取消方程式編號，則可以使用 `equation*`。

$$f(x, \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp \left[-\frac{1}{2} \left(\frac{x - \mu}{\sigma} \right)^2 \right] \quad (1.1)$$

```
\begin{equation}
f(x,\mu,\sigma) = \frac{1}{\sigma \sqrt{2\pi}}
\exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]
\end{equation}
```

1.3 矩陣與多行方程式

除了最簡單的展式模式之外，我們還常會用到矩陣與多行的方程式，在此將逐一介紹。

1.3.1 矩陣

沒有外括號的矩陣可以透過 `\begin{array}` 矩陣內容 `\end{array}` 來達成，矩陣內部的寫法與表格寫法類似，都是使用 `&` 將各個欄位分開。

$$\begin{array}{ccc} a & b & c \\ d & e & f \\ g & h & i \end{array} \quad (1.2)$$

```
\begin{equation}
\begin{array}{ccc}
a & b & c \\
d & e & f \\
g & h & i
\end{array}
\end{equation}
```

在(1.2)中，如果需要加上左右括號，可以使用 `\left[\right]` 將 `array` 環境包圍起來，如(1.3)。值得注意的是，由於大括號為 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 內建指令，所以如果要使用大括號，則需要在前面加上一個 `\`；而如果只需要單邊括號，另一邊則以英文句點取代。

$$\left[\begin{array}{ccc} a & b & c \\ d & e & f \\ g & h & i \end{array} \right] \cdot \left(\begin{array}{ccc} a & b & c \\ d & e & f \\ g & h & i \end{array} \right. \quad (1.3)$$

```
\begin{equation}
\left[
\begin{array}{ccc}
a & b & c \\
d & e & f \\
g & h & i
\end{array}
\right]
\cdot
\left(
\begin{array}{ccc}
a & b & c \\
d & e & f \\
g & h & i
\end{array}
\right.
\end{equation}
```

使用上述方法，可以簡單的寫出分段方程式：

$$f(x) = \begin{cases} ax + b & , \quad x \geq 0 \\ cx + d & , \quad x < 0 \end{cases}$$

```
\[
f(x) =
\left\{
\begin{array}{ccc}
ax+b & , & x\geq 0 \\
cx+d & , & x< 0
\end{array}
\right.
\]
```

除了 array 環境之外，matrix 環境也提供了許多有分界符號的矩陣。

$$\begin{matrix} a & b \\ c & d \end{matrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \left\{ \begin{matrix} a & b \\ c & d \end{matrix} \right\} \left| \begin{matrix} a & b \\ c & d \end{matrix} \right| \left\| \begin{matrix} a & b \\ c & d \end{matrix} \right\|$$

```
\[
\begin{matrix}
a & b \\
c & d
\end{matrix}
\begin{pmatrix}
a & b \\
c & d
\end{pmatrix}
\begin{bmatrix}
a & b \\
c & d
\end{bmatrix}
\left\{
\begin{matrix}
a & b \\
c & d
\end{matrix}
\right\}
\left|
\begin{matrix}
a & b \\
c & d
\end{matrix}
\right|
\left\|
\begin{matrix}
a & b \\
c & d
\end{matrix}
\right\|
\]
```

```
c & d
\end{Vmatrix}
\]
```

1.3.2 多行方程式

`align` 環境可以用來輸入多行方程式，在 `align` 環境中，可以使用 `&` 將特定位置對齊，以我們常用的最佳化方程式來說：

$$\min_x f(x) \tag{1.4}$$

$$\text{s.t. } g(x) \leq 0 \tag{1.5}$$

$$h(x) = 0 \tag{1.6}$$

```
\begin{align}
\min_x \quad & f(x) \\
\text{s.t.} \quad & g(x) \leq 0 \\
& h(x) = 0
\end{align}
```

在上述方程組中， $f(x)$ 、 $g(x)$ 與 $h(x)$ 透過前面的 `&` 符號而對齊。在此方程組中，我們通常會希望方程式標號是一組的，而不是分開為三個標號，這時候就可以使用 `subequations` 環境將整個 `align` 環境包起來，並透過不同的 `label` 來呼叫方程式。如(1.7)表示的就是整個方程組，而(1.7a)就是指目標函數。此外，若不要某行方程式的標號，則只要在該行方程式後面加上 `\notag` 指令即可。

$$\min_x f(x) \tag{1.7a}$$

$$\begin{aligned} \text{s.t. } g(x) &\leq 0 \\ h(x) &= 0 \end{aligned} \tag{1.7b}$$

```
(\ref{eq:opt})表示的就是整個方程組，而(\ref{eq:obj})就是指目標函數。
\begin{subequations}
\begin{align}
\min_x \quad & f(x) \label{eq:obj} \\
\text{s.t.} \quad & g(x) \leq 0 \quad \notag \\
& h(x) = 0 \quad \notag
\end{align}
\label{eq:opt}
\end{subequations}
```

另外一種情況是我們並不需要方程式的子標號，這時候只要在 `align` 環境中再使用 `split` 環境包起來即可。

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq 0 \\ & h(x) = 0 \end{aligned} \tag{1.8}$$

```
\\begin{align}
\\begin{split}
\\min_x \\quad & f(x) \\
\\mbox{s.t.} \\quad & g(x) \\leq 0\\
& h(x) = 0
\\end{split}
\\end{align}
```


第 2 章 編譯與 Debug

在一般的文件中， \LaTeX 僅需要編譯 `.tex` 檔所以並不會有太大的問題。但在論文寫作的過程中，由於加入了參考文獻的 `.bib` 檔，所以常會出現編譯錯誤或是參考文獻與 `.bib` 檔案內容不合的情況，因此在這裡簡述整個 \LaTeX 的編譯過程，若各位在編譯過程中遇到問題，請依循以下步驟進行編譯：

1. pdfLaTeX: 在第一次的編譯過程中， \LaTeX 會截取 `.tex` 檔內的各種 `\cite{}` 與 `\label{}`，因此在經過第一次編譯之後，所有文章中呼叫的方程式、圖片與表格都會以`??`呈現，而參考文獻則是以`[?]`呈現。
2. BibTeX: 如果有使用到參考文獻的 `.bib` 檔，則需要進行這一個步驟，讓程式在 `.bib` 檔內抓取相關的資料。
3. pdfLaTeX: 在第二次編譯過程中， \LaTeX 已經有參考文獻的相關資料，但是還不確定其順序，所以仍是以`[?]`呈現；而在方程式、圖、表的部分，因為在前一次的編譯中已經有圖表的出現順序，所以會完整呈現圖表的編號。
4. pdfLaTeX: 最後一次編譯會將原本呈現`[?]`的參考文獻顯示為正確的編號，在文章最後面也會依序條列出參考文獻。

2.1 常見問題與處理方式

在 \LaTeX 中，較常見且通常看不出原因的問題有兩種：

1. 編譯錯誤一次後，怎麼修改甚至改回之前編譯正確的檔案， \LaTeX 仍然無法編譯：

針對這個問題，主要是發生在 `.aux` 檔案因為編譯失敗中止而沒有回復到正常的狀態，所以只要在 File 選單中選擇 Remove Aux Files 或是直接進資料夾刪除 \LaTeX 自動產生的檔案後，再重新編譯即可恢復正常。

2. 修改 `.bib` 檔案之後， \LaTeX 可以正常編譯，但是輸出的參考文獻與 `.bib` 檔案內的參考文獻資料不同：

這個問題通常是發生在新增或更新 `.bib` 檔內的資料之後，發現無論怎麼做，編譯出來的 `.pdf` 都還是原本的資料。這是因為 \LaTeX 在執行的過程中由 `.bib` 檔產生了一個 `.bbl` 檔，而在編譯的時候，參考文獻其實是從 `.bbl` 檔抓取的，所以只要去資料夾內刪除 `.bbl` 檔，然後再依照上述的編譯步驟編譯一次即可。

萬用解：刪除資料夾內，系統自動產生的所有檔案，然後依照上述過程重新編譯。如果還是無法編譯，那就是你寫錯了！