

課程名稱

副標題或授課教師

關於這個版本

說明這份教材的來源、做了哪些整理、哪些地方與原始材料不同。

建議寫清楚三件事：

- **來源**：原始材料是什麼、誰的、什麼授權。
- **處理**：你做了什麼(重排？勘誤？補寫？)，沒做什麼。
- **慣例**：書裡的方塊各代表什麼意思(見下)。

重點方塊：定義、定理、關鍵結論放這裡。

補充說明：解釋為什麼、和別處的關聯、容易踩的坑。

勘誤 L01 · **勘誤**：指出原始材料的錯誤。原稿寫 **錯的**，應為 **對的**。

課堂語錄

原話引用：逐字稿裡值得保留的口述內容。與上面三種分開，是因為它不是在說明數學，而是「當時真的講了什麼」。

Contents

關於這個版本	1
I 第一部分	3
1 這一講的標題	4
1.1 一個小節	4
1.2 四種環境	4
1.3 插圖	5
1.4 表格	6
1.5 交互參照	6

Part I

第一部分

LECTURE 1

這一講的標題

§1.1 小節名稱 · 另一個主題 · 第三個主題

開場一段話，說明這一講要解決什麼問題、和上一講的關係。不要一上來就丟定義——先講「為什麼需要它」。

1.1 一個小節

正文用一般段落寫。行內數學用 `$...$`：例如 $f(x) = x^2$ 、 $\lim_{x \rightarrow c} f(x) = L$ 。獨立數學式用 `\[...\]`：

$$\forall \varepsilon > 0, \exists \delta > 0 \text{ 使得 } \forall x, 0 < |x - c| < \delta \implies |f(x) - L| < \varepsilon.$$

留意 `\eps` 是 `preamble` 定義的縮寫 (ε)，另外還有 `\R \Q \N \Z` ($\mathbb{R} \mathbb{Q} \mathbb{N} \mathbb{Z}$)、`\dx \dt \du` ($dx \, dt \, du$) 這種帶前置細空白的微分符號)、`\ds{y}{x}` ($\frac{dy}{dx}$)。完整清單見 `common/preamble.tex`。

不編號的小節

用 `\subsection*{ }` 開不進目次的小節，適合「例題」「證明」這種重複出現的區塊標題。

1.2 四種環境

定理類

有六個共用同一組編號的環境，語意不同但外觀一致：

定義 1.1 (名稱可省略) · 定義用 `defx`。設 $I \subseteq \mathbb{R}$ 、 $f : I \rightarrow \mathbb{R}$ ，稱 f 在 c 連續，若 $\lim_{x \rightarrow c} f(x) = f(c)$ 。

定理 1.2 (有名字的定理) · 定理用 `thmx`。加 `\label{ }` 之後就能在別處用 `\ref{ }` 引用。

性質 1.3 · 性質用 `propx`——份量比定理輕的結論。

引理 1.4 · 引理用 `lemx`——為了證某個定理而先準備的小結果。

推論 1.5 • 推論用 `corx`——從剛證完的東西直接掉出來的結果。

例 1.6 • 例題用 `exx`。

註 1.7 • 註用 `rmkx`。

證明 • 證明用 `proof`，結尾自動加 $\square$ 。要改標題就寫 `\begin{proof}[反證]` 或 `[解]`。 $\square$

四種方塊

keybox —藍色。放最重要的那句話：定義的核心、定理的敘述、一條要背起來的原則。用得太多就等於沒用，一頁最多一兩個。

classnote —米色。補充說明：為什麼這樣做、和哪一講有關、哪裡容易錯。這是「作者對讀者說話」的地方。

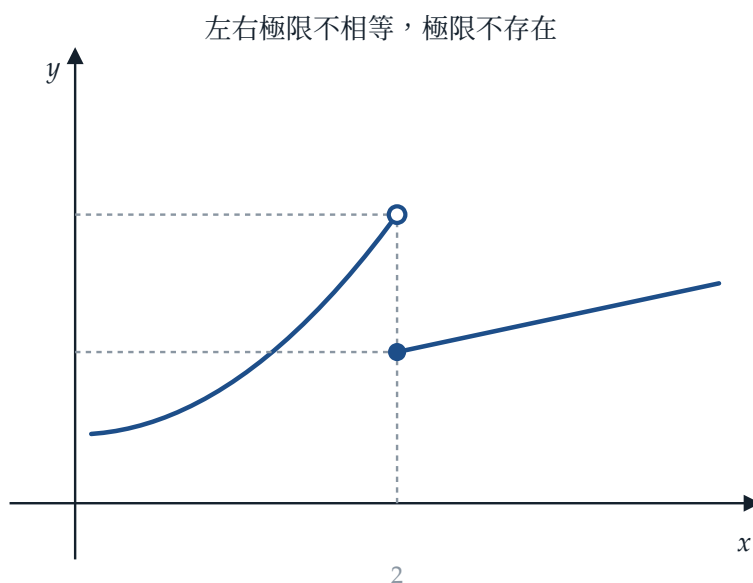
勘誤 L01 • **erratum** —紅色。指出原始材料的錯誤。原稿寫成 $3 + 5 = 7$ ，應為 $3 + 5 = 8$ 。
`\wrong{}` 標紅、`\correct{}` 標藍，兩者在文字與數學模式都能用。

課堂語錄 • 可自訂標題

voice —灰藍。逐字稿的原話引用。

和 `classnote` 的差別：`classnote` 是你在解釋數學，`voice` 是記錄當時說了什麼。分開之後，讀者可以整塊跳過語錄而不影響數學內容，你也不必為了「保留原話」去稀釋正文。

1.3 插圖



用 `\fig{名字}` 引用 `figs/名字.pdf`，預設寬度為 `0.62\linewidth`。要改寬度：`\fig[\linewidth]{名字}` 或 `\fig[0.45\linewidth]{名字}`。

圖是用 `figs/make_figs.py` 畫的，跑 `./build.sh` 會自動重畫。只想重畫一張：`python3 figs/make_figs.py demo-limit`。

為什麼不直接用 TikZ？兩個理由：

1. 畫函數圖形時 `matplotlib` 好寫太多——直接餵 `numpy` 陣列就好，不必自己算取樣點。
2. 出來是向量 PDF，和 TikZ 一樣不失真。

代價是多一個 Python 相依，以及要處理字型一致性（見 `docs/03-figures.md`）。如果你的圖以「示意」為主而非函數曲線，TikZ 可能更合適。

1.4 表格

欄一	欄二	欄三
資料	資料	資料
資料	資料	資料

用 `booktabs` 的 `\toprule` `\midrule` `\bottomrule`，不要畫直線——這是排版慣例，直線會讓表格變吵。

1.5 交互參照

引用上面那個定理：定理 1.2。標籤慣例建議統一，例如 `thm:` `def:` `ex:` `fig:` 開頭，專案大起來之後找得到。

課本習題 • P61(1.2.3.5.11)、P71:39.42.45

用 `\homework{}` 放課本習題指定，會在該處畫一條細線並縮排顯示。