cwTEX

排版系統

"[T]he TEX research project that I embarked on was driven by two major goals. The first goal was *quality*: we wanted to produce documents that were not just nice, but actually the best."

"I never intended to have a system that would be universal and used by everybody. I always wanted to write a system that would be used for just the finest books."

"The current version number for TEX is 3.1, and for METAFONT it is 2.7. If corrections are necessary, the next versions of TEX will be 3.14, 3.141, then 3.14159, ..., converging to the ratio of a circle's circumference to its diameter; for METAFONT the sequence will be 2.71, 2.718, ..., converging to the base of natural logarithms.

Donald E. Knuth
Digital Typography (1999)

cwTEX
排 版 系 統

吳 聰 敏·吳 聰 慧

第 2 版

編目: ISBN 957-97454-0-4
國立中央圖書館
分類碼: 312.949T46
CIP 流水號: 86011499

cwTEX 網址:
http://ceiba.cc.ntu.edu.tw/tmwu
中文 TEX 之 BBS 討論站:
telnet://140.112.18.32

吳聰敏 · 吳聰慧

封面設計: 吳聰敏
內文設計/排版: 吳聰敏

本書使用字體包括:
cwTEX 中文字體
Adobe Minion condensed
Computer Modern typewriter
Y&Y Mathtime plus

總經銷:
翰蘆圖書出版有限公司
台北市懷寧街 92 號 5 樓
電話: (02)2382-2333
傳眞: (02)2388-6655
郵撥: 15718419
Email: hanlu@hanlu.com.tw
http://www.hanlu.com.tw

經銷處
碁珠資訊股份有限公司
電話: (02)2788-2408
http://www.gotop.com.tw

1997 年 10 月初版
2000 年 3 月 2 版

定價: 新台幣 600 元

目錄

序

cwTEX 系統從開始發展迄今, 已超過十年。開始發展此一系統的動機很簡單: 因爲在學術界的工作需要一套排版學術論文的軟體。幾年之後, 我們才發現 Knuth 教授當初也是因爲同樣的理由而創造 TEX 系統。

cwTEX 系統是以 TEX 爲骨幹。發展軟體時, 我們希望它的能力、品質與英文 TEX 系統相當。但是, 中文書寫系統與英文畢竟差異很大, 在某些地方我們實在是找不出好的解決之道。不過, 我們認爲 cwTEX 的能力與排版品質, 相較於目前任何中文排版系統都毫不遜色。這當然是要歸功於 TEX 幾近完美的設計。

本書初版於 1997 年出版以來, LATEX 排版系統有許多新的發展, 其中最重要的可能是網路出版; 因此, 以 LATEX 作爲網路出版的工具程式也陸續出現。利用這些軟體工具, LATEX 的排版結果很容易可以送上網站。因應此一發展, 本版對於網路出版另闢一章介紹。除了網路出版之外, 第2版在許多地方與初版差異甚大, 簡單說明於下:

- cwTEX 中文字體指令之語法改爲與 LATEX 一致。譬如, 中文字體指令有效範圍僅限於指令環境或大括號內; 另外, 中文字體指令之後可續鍵入文字, 不須另換新行。
- cwTEX 提供簡單的中文字體巨集指令功能, 用於設定文稿章節標題與註解之中文字體。
- 每一中文字體可以作斜體與水平放大/縮小之變形。
- 新版 DVIPS 能正確處理中文 PostScript 字型檔, 因此文稿排版結果可直接以描邊字型列印。
- 介紹一些新的巨集套件, 使文稿編排設計更爲容易。譬如, titlesec 巨集套件使章節標題設計變得容易, 且富有彈性。又如, fancyvrb 巨集套件之 Verbatim 指令環境可以照列中文與排版指令。

本書初版發行時, Win95 作業系統雖已出現, 但專以此作業系統爲對象之 TEX 系統尙不十分成熟。兩年來, 專門在 Win95/98 系統上使用之 TEX 系統包括 MiKTEX 與 fpTEX 兩套。目前, cwTEX 可以在這兩套系統上使用。除了 Win95/98 之外, cwTEX 也可以在 Windows NT 或 Windows 2000 上執行。事實上, 除了 Windows 平台之外, 一些熱心人士正幫忙將 cwTEX 移植到 Linux, FreeBSD 平台上。如果有新的進展, 我們將在 cwTEX 網站上公佈。

使用 cwTEX 給我們帶來相當的便利, 甚至是樂趣; 我們希望你也有同樣的感受。發展軟體的過程中, 張淸溪、林向愷、鍾經樊、古慧雯與宋玉生教授提供許多寶貴意見。歷年來台大經研所的研究生不少人以此系統排版論文, 等於是幫忙測試軟體。其中, 盧敬植、曲祉寧、陳宜廷等特別熱心幫忙。台大經濟系、淡大數學系及嘉南藥理學院提供硬體設備與軟體; 陳兪成先生細心校閱初稿; 吳慕凡小姐繪製插圖; 我們很感激他們的幫助。台大經濟系的研究助理陶曉昀、陳香如、徐中貴、楊文琦、湯雅慧等, 特別是張素惠小姐, 對於發展此一系統貢獻良多, 我們在此一併致謝。

準備第2版過程中得到許多人的協助。首先, 許多的熱心使用者提出各種建議, 但限於時間與能力, 部分建議無法完成, 謹在此表示感激與歉意。朱雅珍從美國寄來參考書籍, 陳兪成再度細心校閱初稿, 翁明德積極推薦 fpTEX, 並提供實質幫助; 他們的協助使得軟體與手册的品質得以大幅提升。最後, 沒有林佳蓉與曲祉寧與辛苦工作, 我們的改版工作是難以完成的。我們也藉此機會特別表示感謝。

幾十年來, 父母樸素、平實的生活態度一直是我們的榜樣。他們給與子女們的持續關懷, 對於我們的工作, 包括發展 cwTEX 系統在內, 有積極的鼓勵作用。我們也在此表達感謝之意。

本版初稿完成之際恰逢台灣遭受百年來最大的集集大地震 (1999 年 9 月 21 日凌晨 1 時 47 分)。謹在此誌念地震中不幸傷亡的人們。

吳聰敏 · 吳聰慧

1 前言

本書介紹 cwTEX 排版系統。cwTEX 包含TEX 排版系統, 中文字轉換程式與各式中文字體。TEX 系統是美國 Stanford 大學 Donald E. Knuth (高德納) 教授所發展, 目前是美國數學學會 (American Mathematical Society) 出版學術期刊與書籍的排版系統。一般認爲 TEX 的特點是排版數學文稿能力特別強。不過, 根據 Knuth 的夫子自道:[1]

> "[T]he TEX research project that I embarked on was driven by two major goals. The first goal was *quality*: we wanted to produce documents that were not just nice, but actually the best."
>
> "I never intended to have a system that would be universal and used by everybody. I always wanted to write a system that would be used for just the finest books."

顯然, 品質才是 TEX 系統的眞正精神。

cwTEX 以 TEX 爲基礎, 但加強其功能, 使它可以排版中文。本書說明如何使用 cwTEX 排版中英文書籍與文稿。TEX 的排版指令多而且複雜。對於專業排版者來說, 這有「如魚得水」的感覺。但對於一般的使用者, 可能望之而卻步。爲了因應一般使用者的需求, 許許多多的專家寫了一套套的巨集指令 (macros)。利用這些現成的巨集指令, 以 TEX 排版書籍稿件並不困難。各巨集指令中, 最有名也最廣爲使用的是 LATEX, 原始作者爲 Leslie Lamport。本書主要介紹 LATEX 指令。

Knuth 教授原先是在大型電腦上發展 TEX 系統。完成之後, 他把原始程式全部公開。因此有人把 TEX 移殖 (porting) 到各種作業系統上。目前, 大至超級電腦, 小至個人電腦上, 都有 TEX 系統。其中, 大部分都是免費使用。本書初版出版於 1997 年, 當時雖然許多個人電腦所使用的作業系統已逐漸由 MSDOS 與 Win31 轉爲 Win95, 但是專屬於 Win95 之 TEX 系統尙未成

[1] 見 Knuth (1999), 頁 559, 616。

熟。因此, 本書初版所介紹的是使用於 MSDOS 作業系統上之 emTEX。兩年來, 專以 Windows 系統爲對象的 TEX 已有好幾套, 其中可供免費下載的包括 MiKTEX 與 fpTEX 兩套。目前, cwTEX 主要即結合這兩套系統發展。

以上所介紹的與 TEX 有關的軟體可簡單歸納如下:

- TEX: Knuth 所發展之排版系統;
- LATEX: 方便使用之巨集套件組合;
- MiKTEX/fpTEX: 可於 Win95/98/NT 平台上執行之 TEX 系統;
- cwTEX: 讓 TEX 可以排版中文稿件之工具程式與字型檔案。

要讓 TEX 處理中文, 基本上有兩種方法。第一是直接修改 TEX 原始程式碼。我們採取的是較間接的第二種方法: 寫一個中文字碼轉換程式 (亦稱爲前階處理程式 preprocessor), cwtex。此程式的主要功能是將中文字碼轉換成 TEX 的格式。我們另外造出符合 TEX 規格的各式中文字型。如此一來, TEX 即可以編排中文字。此種方法較簡單, 而且程式出現錯誤的機率較低。

我們採取前階處理程式來處理中文還有另外一個原因。完整的排版過程包括輸入文字及指令, 編排 (text formatting), 預視及列印。要在顯示器上預視排版結果需要預覽軟體; 在不同印表機上列印也需要特別的列印程式。自從 TEX 出現之後, 很多人寫了預覽/列印軟體及相關的工具程式 (utilities), 絕大部分是免費提供使用。cwTEX 前階處理程式轉換中文時, 完全依照 TEX 之規格。因此, 幾乎所有的預覽/列印軟體及工具程式都可用於中文稿件上, 不須修改。

LATEX 是由一大群巨集指令組合而成。從開始發展以來, 許多使用者寫了更多的巨集指令擴充其功能。目前有一群專家正在更新 LATEX 巨集指令。此一工作完成之後, 將稱爲 LATEX 第3版, 目前流通之版本稱爲 LATEX2ϵ; 舊有版本則稱爲 LATEX 2.09。本書所介紹的是 LATEX2ϵ 之指令。

除了 LATEX 及 cwTEX 中文指令外, 本書也扼要介紹一些排版的觀念。文稿排版品質的好壞, 除了排版系統的能力之外, 更重要的是排版者的鑑賞能力。LATEX 的排版能力一流。但是, 頂級的咖啡機器 (如 Rancilio Audrey) 若使用不當, 也可能燒出如 Agatha Christie 所說的「名爲咖啡的可疑液體」。我們從排版文獻中整理出一些應注意的事項。了解這些基本觀念, 排版時可以避免一些常見的錯誤。

本書是 cwTEX 排版系統之使用手册。學習電腦排版, 必須實際動手, 不能光說不練。要學習使用 cwTEX 系統, 請先安裝程式。第4章說明如何安裝 cwTEX 系統。初學習排版, 請先閱讀2-6章。第2章提供6個排版例子, 第3章說明完整的排版過程; 你可以依樣輸入文稿、排版、預視與列印。第4章除了說明安裝軟體的方法之外, 也介紹如何透過 WinEdt 軟體介面在 Win95/98 下排版。第5章說明排版指令的概念及輸入文稿的原則。第6章說明選用中英文字體的方法。

第7、8兩章說明章節與段落設計指令。若是排版書籍, 我們首先須設計好版面樣式, 再以這兩章的指令編排。LATEX 提供一些現成的版面格式, 可供直接使用。第9章說明數學式之排版, 對某些人來說, 這可能是 TEX 最引人入勝之處。第10章爲表格編排。表格排版指令較複雜, 因此我們提供較多的例子說明。

新版 LATEX 一個重要的發展方向是與 PostScript 繪圖語言結合。此一結合讓我們可以使用各種專業品質之英數字體、引用外製圖形、或者在文稿中直接繪製圖形。第11-13章即介紹相關之指令與巨集套件。

LATEX 事實上是以 TEX 爲骨幹的一套巨集指令, 第14章說明巨集指令之概念及使用方法。第15章介紹一些有用的工具, 包括: 排版書信、編製索引、排版投影片等等。第16章說明網路出版的相關工具與軟體。cwTEX 系統所提供的字型檔大部分都含有13,503個中文字。即使如此, 我們偶而還是會碰到一些字型檔內所沒有的中文字。遇有此種狀況, 必須自行造字。第17章說明造新字的方法。

cwTEX 爲幕後排版系統, 下指令或輸入文字時, 難免出現錯誤。第18章說明如何偵測錯誤。最後, 第19章說明如何下載並使用新出版之巨集套件。

* * *

本書的排版幾乎使用了本書所介紹的各種巨集套件與程式。一開始, 我們是分章編排。到了最後階段, 我們創造一主檔案, 取名爲 cxbook.ctx, 其內引入全書各章。排版過程中, 我們測試 MiKTEX 與 fpTEX, 也測試 emTEX 之 htex386。因爲全書內容複雜, 我們發現 htex386 與 fpTEX 可以毫無錯誤地正確編排。如果使用 MiKTEX, 則須以 lambda 編排。

使用 DVIPS 程式將整本書轉換爲一個 PostScript 檔案, 執行時會出現錯誤訊息; 問題似乎和作業系統管理記憶體之效率有關。如果僅轉換全書的

一部分, 如封面至第100頁, 或第101頁至300頁, 則不會有問題。我們最後是以 omega 程式集之 odvips 程式轉換全書, 結果正確無誤。以下是 LaTeX 排版之綜合訊息:

```
Here is how much of TeX's memory you used:
 6115 strings out of 10867
 71012 string characters out of 171683
 156152 words of memory out of 263001
 8540 multiletter control sequences out of 10000+0
 214756 words of font info for 396 fonts,
    out of 400000 for 1000
 14 hyphenation exceptions out of 1000
 35i,23n,46p,1651b,701s stack positions
    out of 300i,100n,500p,50000b,4000s
```

2 例子

學習排版的捷徑是參考現成的例子。本章列出6個排版例子, 並加上簡單的說明。這幾個例子之檔案皆置於 c:\texmf\cwtex\examples 檔案夾內。若自行排版這幾個例子, 得到的結果與本章所展示者會稍有差異, 原因是本書排版使用了一套特別的英文字體。

除了這6個例子外, 上述檔案夾內尚含有其他例子。其中, paper1.zip, paper2.zip 等是短篇論文的例子; thesis1.zip 是博碩士論文的例子。博碩士論文採用一主檔案, 其內再引入各章之子檔案。LaTeX 可用以排版信函, letter1.ctx, letter2.ctx 等檔案是信函之例子。此外, NSCproj.ctx 是國科會研究補助之申請表格; NSCvita.ctx 是個人資料表。不過, 國科會表格經常變動, 以上檔案的排版結果與國科會最新格式可能稍有差異。

國民所得兩萬美元

吳聰敏

1999.9.9

台灣的生活水準在提升嗎？若從大街小巷充斥著賣名牌的商店來看，答案是肯定的。但如果從生活環境品質日益惡化來看，答案則是否定的。

三十年前，台灣的所得尚低。一般民衆最關心如何提升所得；因此政府施政也以提升所得爲主要目標，這不難理解。但是，所得逐漸上升之後，民衆的偏好、需求會逐漸改變。每一個國家都是如此，台灣也不例外。最簡單一個例子，幾年來出國旅遊風氣日盛，這表示民衆對休閒的需求日增。因爲台灣本島可供休閒旅遊的場所太少，大家只好往國外跑。(但是，並不是每一個人都有能力到國外渡假。)

公元兩千年時，台灣每人平均國民所得高達兩萬美元。上下班時，每人身著名牌，在公車或自用車子裡動彈不得，空氣品質及水質比今天更惡化；遇到「假日」除非有錢出國，否則只好呆在客廳裡吹冷氣、看電視。這樣子的兩萬美元所得的意義在那裡呢？

```
% file: examp1.ctx
\documentclass[12pt]{article}
\title{\bb17 國民所得兩萬美元}
\author{\k14 吳聰敏}
\date{1999.9.9}
\begin{document}
\maketitle
\fontsize{12}{20pt}\selectfont
\m12
台灣的生活水準在提升嗎?
若從大街小巷充斥著賣名牌的商店來看,
答案是肯定的。
但如果從生活環境品質日益惡化來看,
答案則是否定的。

三十年前,台灣的所得尚低。
一般民衆最關心如何提升所得;
因此政府施政也以提升所得為主要目標,這不難理解。
但是,所得逐漸上升之後,民衆的偏好 ...
每一個國家都是如此,台灣也不例外。
最簡單一個例子,幾年來出國旅遊風氣日盛,
這表示民衆對休閒的需求日增。
因為台灣本島可供休閒旅遊的場所太少,
大家只好往國外跑。
(但是,並不是每一個人都有能力到國外渡假。)

公元兩千年時,台灣每人平均國民所得高達兩萬美元。
上下班時,每人身著名牌,
在公車或自用車子裡動彈不得,
空氣品質及水質比今天更惡化;
遇到「假日」除非有錢出國,
否則只好呆在客廳裡吹冷氣、看電視。
這樣子的兩萬美元所得的意義在那裡呢?
\end{document}
```

- LaTeX 的第一道指令通常是 \documentclass, 文稿內容則置於 document 指令環境內: 以 \begin{document} 開始, \end{document} 結束。
- 排版指令皆以反斜線 \ 起頭, 但註銷指令是 %。
- \bb17, \k14, \m12 爲中文字體指令, 分別表示選用 17 點粗黑字體, 14 點楷體, 與 12 點明體中文字。
- 空一行表示本段結束。輸入文稿時, 請盡量在標點符號後換行。
- \title 指令用於排版標題, \author 用於排版作者名字, \date 用於排版日期。
  \maketitle 指令指示將該行以上之文字全部排版於標題頁之內。
- \fontsize 指令選項選定 12 點英數字體, 行距爲 20 點。

台灣長期總產出之變動

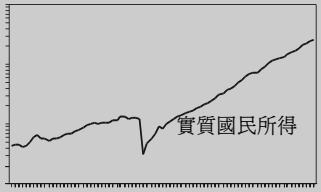
吳聰敏*

1994.12

台灣長期總體經濟表現, 可以簡單分清國統治時期、日治時期及戰後時期來討論。其中, 清末日初是一個重要的轉捩點。

1 長期國內生產毛額之變動

百年來台灣的國民所得有很大的變化。下圖畫出台灣長期國民所得之變動。[1] 由此圖可以看出, 日治時期國民所得成長率低於戰後時期。



2 國內生產毛額的變動

日治初期總督府推動**資本主義化**的政策, 其目的是要改善台灣的投資環境, 吸引日本企業家前來投資。日本人在台灣進行土地與戶口調查, 開闢道路, 統一貨幣與度量衡。

*作者任教於台大經濟系。

[1]日本人曾簡單估算 1938-44 年之國民所得, 刊於《台灣金融經濟月報》1944 年 8 月號。

```
% file: examp2.ctx
\documentclass[11pt]{article}
\usepackage{graphicx,psfrag}
\title{\m17 台灣長期總產出之變動}
\author{\m11 吳聰敏\thanks{\m10
作者任教於台大經濟系。}}
\date{\small 1994.12}
\begin{document}
\maketitle
\fontsize{10.95}{18pt}\selectfont
\m11
台灣長期總體經濟表現，
可以簡單分清國統治時期、
日治時期及戰後時期來討論。
其中，清末日初是一個重要的轉捩點。

\section{\m14 長期國內生產毛額之變動}
百年來台灣的國民所得有很大的變化。
下圖畫出台灣長期國民所得之變動。\footnote{\m10
日本人曾簡單估算1938-44年之國民所得，
刊於《台灣金融經濟月報》1944年8月號。}
由此圖可以看出，
日治時期國民所得成長率低於戰後時期。

\begin{center}
\psfrag{GDP}{實質國民所得}
 \includegraphics[width=7cm]{examp.eps}
\end{center}

\section{\m14 國內生產毛額的變動}
日治初期總督府推動{\bb11 資本主義化}的政策，
其目的是要改善台灣的投資環境，
吸引日本企業家前來投資。
日本人在台灣進行土地與戶口調查，開闢道路，
統一貨幣與度量衡。
\end{document}
```

- \usepackage 指令用於引入其他巨集套件。本例子引入 graphicx 與 psfrag 巨集套件, 其功能是引用外製圖形。
- \thanks 指令用於排版致謝詞。
- \date 指令內之 \small 指令指示選用較小之英數字體。
- 排版節標題可使用 \section 指令。
- 排版註解指令爲 \footnote, LaTeX 會自動替章節標題與註解加上數字編號。

戰後初期, 國民政府在台灣實施全面性的經濟管制, 其手段包括: 民營企業公營化、獨占與聯合壟斷、價格控制等。

1 文獻檢討

有關於台灣戰後初期經濟的文獻很少, 連溝口敏行的專著,《台灣・朝鮮經濟の成長》, 也未討論此一時期的經濟狀況。日本大藏省所出版的《日本人の海外活動に關する歷史的調查》, 則收集不少此時期的統計資料。

2 貿易管制

1946-48年間, 台灣的砂糖絕大部分銷往上海。國民政府極力壓低砂糖的銷售價格。價格一受到管制, 台糖的銷售收入也大受影響。表1列出台灣出口至大陸的糖、煤及總出口之物價指數。

表 1: 台灣對大陸出口金額

	1946	1947	1948
糖	100	392	436
煤	100	455	4,736
物價指數	100	463	2,871

說明: 金額單位爲台幣百萬元。

根據台糖公司本身的估計, 干預政策使台糖在1947-49年間,「損失達新臺幣一億四千餘萬元」。事實上, 國民政府的管制政策是戰後初期台灣惡性物價膨脹的主要因素之一。

```
\documentclass[12pt]{article}
\usepackage{booktabs}
\input{mymacro}
\renewcommand{\tablename}{\m12 表}
\belowcaptionskip=6pt
\begin{document}
\sz12\m12
戰後初期，國民政府在台灣實施全面性的經濟管制，
...

\section{\r14 文獻檢討}
有關於 ...《台灣 $\cdot$ 朝鮮經濟の成長》，...
日本大藏省 ...《日本人の海外活動に關する歷史 ...

\section{\r14 貿易管制}
1946-48年間，台灣的砂糖 ...
表\Z\ref{export}\Z 列出台灣出口至大陸的糖、...

\begin{table}[h]
\begin{center}
\caption{台灣對大陸出口金額} \label{export}
\begin{tabular}{rrrr}
\toprule
        &  1946  & 1947 &  1948\\
\midrule
    糖  &   100  &  392 &   436\\
    煤  &   100  &  455 & 4,736\\
物價指數 &  100  &  463 & 2,871\\
\bottomrule
\end{tabular}\par
\parbox{5.4cm}{說明: 金額單位為台幣百萬元。}
\end{center}
\end{table}

根據台糖公司本身的估計，... 因素之一。

\par\vfill\noindent \jobname.ctx (\today)
\end{document}
```

- 排版表格可以使用 tabular 指令環境。\usepackage{booktabs} 指令用以引入 booktabs 巨集套件, 使表格排版更精美。
- 第 7 行之 \sz12 爲一改變字體大小及行距之巨集指令; 巨集指令檔名爲 mymacro.tex; 於文稿前端以 \input 指令引入。
- cwTEX 系統可以排版日文假名。
- \section 指令內之 \r14 選用圓體 14 點中文字體。
- \Z 用於稍加大中文與阿拉伯數字之間距。
- 表格置於 table (浮動版面) 指令環境內, LATEX 會自動選擇適當的置放點。\caption 用於排版圖表標題。我們尙可進一步使用 \label 與 \ref 配對指令以引述圖表之編號。
- 文稿末端之 \jobname 指令用於排版檔名, \today 用於排版日期。

論「米糖相剋」

古慧雯 · 吳聰敏*

1995.12

稻米和蔗糖是日治時期最重要的產業, 文獻上所謂的「米糖相剋」問題, 是在探討甘蔗而和稻作競地的情形。

1 模型

影響作物選擇的因素很多, 包括各作物的單位價格及其生產力。根據以上的分析, 農夫選種甘蔗的條件是:

$$\frac{q_C^k(t)}{q_{R1}^k(t)} \geq \frac{P_{R1}(t)}{P_C(t+3)} \frac{1+\delta+\delta^2}{\delta^2} \tag{1}$$

很明顯地, t 時種蔗的總人口或總面積, 繫乎其時各農夫蔗稻的相對生產力, 以 $F_t(\cdot;\alpha_t)$ 表示 $q_C^k(t)/q_{R1}^k(t)$ 之累積分配函數, 其中參數 α_t 反映農業政策對蔗稻相對生產力的衝擊。α_t 值較大時, 蔗稻相對生產力的分配亦較具「隨機優勢」, 亦即新政策有利於蔗作勝過稻作: $\partial F_t/\partial \alpha_t < 0$。

由按式 (1), t 時甘蔗植付面積之比率爲:

$$1 - F_t\left(\frac{P_{R1}(t)}{P_C(t+3)} \frac{1+\delta+\delta^2}{\delta}; \alpha_t\right)$$

*感謝楊文琦、徐中貴、賴香吟小姐協助搜集資料。

```
% file: examp4.ctx
\documentclass[11pt]{article}
\input{mymacro}
\usepackage{mathptmx}

\title{\m17 論「米糖相剋」}
\author{\m12 古慧雯 $\cdot$ 吳聰敏\thanks{\m10
感謝楊文琦、徐中貴、賴香吟小姐協助搜集資料。}}
\date{\small 1995.12}
\begin{document}
\maketitle
\sz11\m11
稻米和蔗糖是日治時期最重要的產業，...

\section{\m14 模型}
影響作物選擇的因素很多，...
根據以上的分析，農夫選種甘蔗的條件是:
\begin{equation}
 \frac{q_C^k(t)}{q_{R1}^k(t)} \ge
 \frac{P_{R1}(t)}{P_C(t+3)}
 \frac{1+\delta+\delta^2}{\delta^2}
\end{equation}
很明顯地，$t$ 時種蔗的總人口或總面積，
繫乎其時各農夫蔗稻的相對生產力，
以 $F_t(\cdot; \alpha_t)$ 表示
$q_C^k(t)/q_{R1}^k(t)$ 之累積分配函數，
其中參數 $\alpha_t$ 反映 ...
$\alpha_t$ 值較大時，...
新政策有利於蔗作勝過稻作:
$\partial F_t/\partial \alpha_t < 0$。

由按式 (1)，$t$ 時甘蔗植付面積之比率為:
\[
 1 - F_t \left( \frac{P_{R1}(t)}{P_C (t+3)}
 \frac{1+\delta+\delta^2}{\delta};
 \alpha_t\right)
\]
\end{document}
```

- 數學式須於數學模式中排版, 此例中分別使用 equation 指令環境與 \[... \] 編排展示數式; 前者有自動編號之功能。隨文數式或數學符號則以 $...$ 排版。
- 數學符號是以指令排版, 譬如 α_t 產生 α_t。
- 本例使用 mathptmx 巨集套件排版, 選用特別的 PostScript 字體排版英數文字。

從生產面計算國內生產毛額時, 通常把產業區分爲: (1) 農畜業、(2) 製造業與 (3) 服務業。

1 統計資料之性質與來源

農畜業主要產品生產額之原始資料來源主要爲《總督府統計書》與《臺灣農業年報》。

2 農畜業

農畜業分普通作物、特用作物等五大類。

2.1 普通作物

普通作物包括米穀、甘藷等項。蓬萊米自 1920 年中期開發成功之後, 其產量在短時間內就凌駕在來米之上。但是, 1946-48 之間在來米產量反而高於蓬萊米。戰後初期, 台灣化學肥料極爲缺乏; 影響所及農民只好暫時改種在來米。[1]

2.2 特種作物

特種作物主要爲甘蔗與粗製茶。1902 年之甘蔗生產額仍低於粗製茶; 但翌年開始, 甘蔗生產額即超過粗製茶, 1910 年代開始, 糖業之繁榮使特種作物生產額比率大幅上升, 最高約達 80%。

[1] 請見于景讓 (1949), 頁 14-16。

```
% file: examp5.ctx
\documentclass[11pt]{article}
\input{mymacro}

\ctxfdef{\section}{\bb14}
\ctxfdef{\subsection}{\bb12}
\ctxfdef{\footnote}{\m10}

\begin{document}
\sz11\m11
從生產面計算國內生產毛額時，
通常把產業區分為：(1) 農畜業、(2) 製造業與 ...

\section{統計資料之性質與來源}
農畜業主要產品生產額之原始資料來源主要為
《總督府統計書》與《臺灣農業年報》。

\section{農畜業}
農畜業分普通作物、特用作物等五大類。

\subsection{普通作物}
普通作物包括米穀、甘藷等項。
蓬萊米自1920年中期開發成功之後，
其產量在短時間內就凌駕在來米之上。
但是，1946-48之間在來米產量反而高於蓬萊米。
戰後初期，台灣化學肥料極為缺乏；
影響所及農民只好暫時改種在來米。\footnote{
請見于景讓 (1949)，頁14-16。}

\subsection{特種作物}
特種作物主要為甘蔗與粗製茶。
1902年之甘蔗生產額仍低於粗製茶；
但翌年開始，甘蔗生產額即超過粗製茶，
1910年代開始，
糖業之繁榮使特種作物生產額比率大幅上升，
最高約達80\%。
\end{document}
```

- cwTEX 提供簡單的中文標題巨集指令 \ctxfdef 以方便選用章節標題之中文字體與註解之中文字體。
- 本例設定節標題使用 \bb14, 小節標題使用 \bb12 中文字體; 註解則使用 \m10 字體。

國民所得兩萬美元

台灣的生活水準在提升嗎？若從大街小巷充斥著賣名牌的商店來看，答案是肯定的。但如果從生活環境品質日益惡化來看，答案則是否定的。

三十年前，台灣的所得（income）尚低。一般民衆最關心如何提升所得；因此政府施政也以提升所得爲主要目標，這不難理解。但是，所得逐漸上升之後，民衆的偏好、需求（demand）會逐漸改變。每一個國家都是如此，台灣也不例外。最簡單一個例子，幾年來出國旅遊風氣日盛，這表示民衆對休閒的需求日增。因爲台灣本島可供休閒旅遊的場所太少，大家只好往國外跑。（但是，並不是每一個人都有能力到國外渡假。）

公元兩千年時，台灣每人平均國民所得高達兩萬美元。上下班時，每人身著名牌，在公車或自用車子裡動彈不得，空氣品質及水質比今天更惡化；遇到「假日」除非有錢出國，否則只好呆在客廳裡吹冷氣、看電視。這樣子的兩萬美元所得的意義在那裡呢？

```
% file: examp6.ctx
\documentclass[12pt]{article}
\usepackage{multicol,lscape}
\raggedcolumns
\columnsep=1cm
\pagestyle{empty}
\begin{document}
\begin{landscape}
\vk17
\hspace*{1cm}國民所得兩萬美元

\bigskip
\fontsize{12}{22pt}\selectfont\vm12
\begin{multicols}{2}
\noindent
台灣的生活水準在提升嗎?
若從大街小巷充斥著賣名牌的商店來看,
答案是肯定的。
但如果從生活環境品質日益惡化來看,
答案則是否定的。

三十年前,台灣的所得 (income) 尚低。
一般民衆最關心如何提升所得;
因此政府施政也以提升所得為主要目標,
這不難理解。
但是,所得逐漸上升之後,民衆的偏好 ...
...
大家只好往國外跑。
(但是,並不是每一個人都有能力到國外渡假。)

公元兩千年時,台灣每人平均國民所得高達兩萬美元。
上下班時,每人身著名牌,
在公車或自用車子裡動彈不得,
空氣品質及水質比今天更惡化;
遇到「假日」除非有錢出國,
否則只好呆在客廳裡吹冷氣、看電視。
這樣子的兩萬美元所得的意義在那裡呢?
\end{multicols}
\end{landscape}
\end{document}
```

- cwTEX 提供垂直中文字體。但實際上之應用尚待進一步測試。
- 目前, 中文垂直字體僅提供 5 種: 明體、粗黑體、圓體、楷體、與仿宋體, 而且只有常用字。直排字體指令是在橫排字體指令之前加上 v, 例如, 明體12點指令爲: \vm12; 揩體17點指令爲: \vk17。

3 排版方法

個人電腦自 1980 年代初期逐漸普及以來, 使用最廣的應用軟體可能是文書排版軟體。文書排版軟體可分爲兩大類: 幕前排版 (What-You-See-Is-What-You-Get) 與幕後排版。絕大部分的英文文書處理軟體, 如 Word, Ami Pro, 或 Word Perfect 等, 都是幕前排版軟體, TEX 則是幕後排版軟體。

使用幕前排版軟體排版很方便。當我們輸入文字時, 軟體立即編排版面。最後的列印結果和顯示器所見到的幾乎完全相同。相反的, 若使用幕後排版軟體, 我們必須先以文字編輯軟體 (editor) 輸入文稿與排版指令, 以排版程式編排 (typeset); 再利用預覽軟體 (previewer) 觀看編排結果。若一切無誤, 最後由列印機印出。幕後排版之過程間接, 但如果你追求的是品質, 幕後排版軟體有時是唯一的選擇。在專業排版領域內, TEX 是一套有名的高品質幕後排版系統。

TEX 排版軟體可以在不同的作業系統上使用, 爲了易於辨識, 不同作業系統上之TEX 系統各有其稱呼。譬如, 1990 年代初期的 MSDOS 與 OS/2 作業系統使用者大都使用 emTEX。本書主要介紹 MiKTEX 與 fpTEX 兩套系統, 這是針對 Windows 作業系統所發展出來的。

欲使用 TEX 排版, 電腦中須安裝 MiKTEX 或 fpTEX 軟體。不過, TEX 是幕後排版, 除了安裝 TEX 之外, 我們還需要預覽/列印軟體。Windows 系統下有好幾套預覽/列印軟體, 本章將分別介紹 YAP/Windvi, GSview 與 Acrobat Reader 三種。這三種軟體各有其特點, 可滿足不同使用者之需求。除了以上軟體之外, 我們還需要一套文字編輯軟體以輸入文稿。本章主要介紹 WinEdt, 此一軟體與 TEX 結合得很好; 更重要的是, 它能辨識中文, 輸入中文字時不致出現錯誤。

功能較強的文字編輯軟體都可以讓使用者定義功能鍵。譬如, 大部分應用軟體都將 [F1] 功能鍵定義爲「功能說明」。使用軟體時若遇到困難, 按下 [F1] 鍵, 視窗內即出現相關問題之說明。如果事先設定好功能鍵, 輸入排版文稿之後, 按下功能鍵即可執行排版與預覽/列印動作, 不須離開文字編

輯軟體視窗。如果你依下一章所介紹的方法成功安裝軟體, 安裝程式將在 WinEdt 中設定幾個功能鍵, 讓排版工作變得輕而易舉。以下說明如何透過 WinEdt 之功能鍵或功能圖像 (icon) 進行排版。

3.1 排版步驟

cwTEX 安裝完成之後, 硬碟中將新增 \texmf 與 \localtexmf 兩個檔案夾。在 \texmf\cwtex\examples 檔案夾下有幾個測試檔, 其中之一爲 test.ctx。以 WinEdt 開啓此檔案, 其內容如下:

```
\documentclass[12pt]{article}
\begin{document}
\fontsize{12}{18pt}\selectfont
\m12
以 \TeX{} 排版中文很容易;
數學式之排版, 如 $\sqrt{\beta}$, 尤其精確而且簡單。
\end{document}
```

這是一篇簡單的文稿, 第一道指令設定文件類別爲 article, 並選用 12pt 之英數字體; 第 2 行指令爲 document 指令環境之起頭, 指示下一行開始爲文稿之內容。第 3 行以 \fontsize 指令選用 12pt 之英數字體, 行距重新設定爲 18pt; 第 4 行指令 \m12 指示選用 12 點之明體中文字體。最後一指令指示文稿結束。

文稿輸入完畢之後, 我們如何排版呢? 一般而言, 使用 cwTEX 系統排版文稿須經過三個步驟: 第一步是執行 cwtex 將文稿內之中文字轉換爲 TEX 字體指令; 第二步是執行 latex 進行排版; 最後一步是預覽/列印。這三個步驟可以經由不同途徑完成, 每一途徑各有其特點。使用者可依個人偏好, 選擇最適當途徑。以下之說明以 test.ctx 測試檔爲例。

途徑1: cwTEX ➔ LATEX ➔ DVIPS ➔ GSview

途徑 1 是最典型的排版方法。在 WinEdt 視窗內輸入文稿或開啓原已輸入完成之檔案, 首先按 [F9] 功能鍵將中文字轉換爲 TEX 字體指令; 接著按下 [F10] 進行排版。以 test.ctx 爲例, 在 WinEdt 視窗內按下功能鍵 [F9] 即等於是執行下列指令:

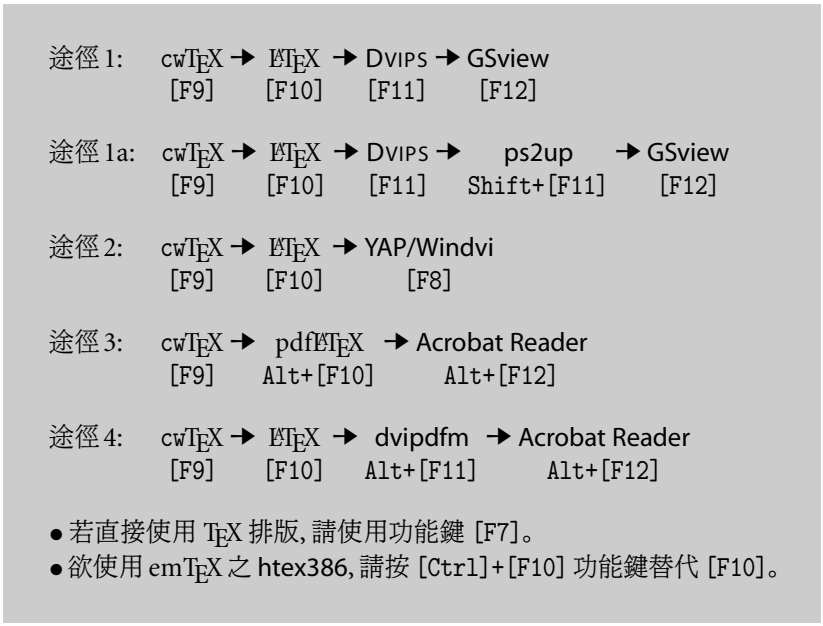


圖 3.1: 排版步驟

```
c:\texmf\cwtex\examples>cwtex -d=c:\xtemp test.ctx
```

執行完畢後, 硬碟內將產生 test.tex 與 cinput.tex 兩檔案。上列指令中之 -d=c:\xtemp 爲 cwtex 程式選項, 其作用是將執行結果所產生之輔助檔案全部移入 c:\xtemp 工作檔案夾內。若不加入此一選項, 排版後所有檔案將儲存於 test.ctx 所在之檔案夾內。將執行結果移入特定位置之工作檔案夾有其方便之處。因爲輔助檔案集中一處, 每隔一段時間清理時, 不須到各檔案夾尋找檔案。

按下功能鍵 [F9] 時, 顯示器畫面即開啓 DOS 視窗, 其內顯示 cwtex 之執行訊息。若執行過程未遭遇錯誤, 工作完成之後, DOS 視窗自動關閉。若 test.ctx 檔案內之排版指令有錯誤, 其訊息會凍結在視窗內, 等待進一步處理。功能鍵 [F10] 也是以類似的方式運作, 其作用是執行 latex 程式進行排版。有些排版之錯誤訊息會在 [F10] 階段才出現, 此時須回到 WinEdt 視窗修改文稿檔案。修改完畢, 關掉 latex 視窗, 再按 [F9] 與 [F10] 兩鍵重新排版。

排版之後, 按 [F11] 功能鍵即執行 DVIPS 程式將結果轉換爲 PostScript

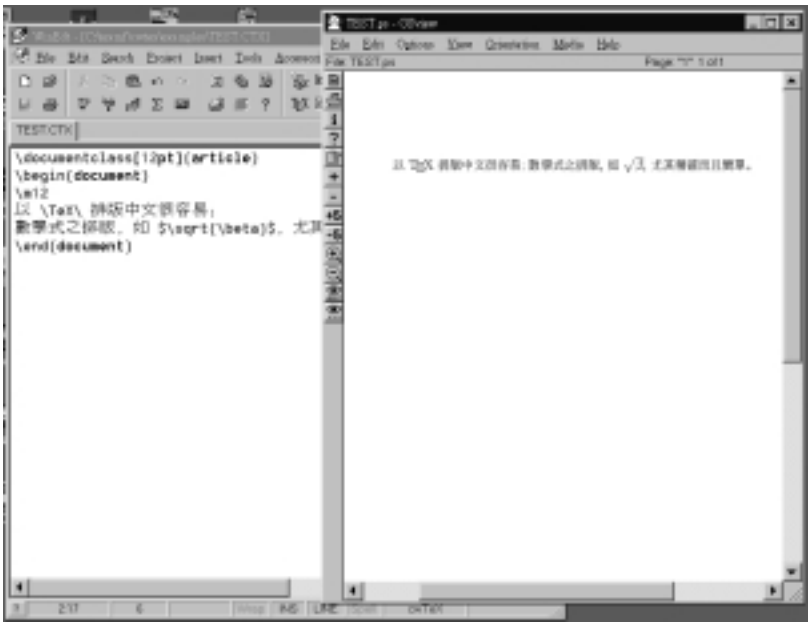


圖 3.2: WinEdt 與 GSview

格式, 檔名爲 test.ps。預覽/列印時, 須按 [F12] 功能鍵啓動 GSview 軟體。圖 3.2 顯示文字編輯視窗與 GSview 預覽結果之畫面。若文稿須修改, 可跳回文字編輯視窗修改原稿, 之後按 [F9], [F10] 與 [F11] 三功能鍵, 再將滑鼠移回 GSview 視窗, 畫面即更新爲修正後的結果。

比起底下介紹的途徑2, 途徑 1 過程多一步驟, 預覽速度稍慢一些, 但它最大的優點是直接使用描邊字型, 因此排版結果可以在高解析度之印表機或輸出機上印出, 品質相當完美。此外, 如果文稿中引用了其他軟體繪製之圖形, 則途徑 1 是最佳選擇。一般的繪圖軟體可將圖形檔儲存爲多種格式, 其中使用彈性最大, 效果最佳之圖形格式是 PostScript。下文介紹之途徑 2 的 YAP 軟體也可預覽/列印 PostScript 圖形, 但有一些限制。有關於引用外製圖形之方法, 請見第 11 章。

途徑1a: cwTEX ➜ LATEX ➜ DVIPS ➜ ps2up ➜ GSview

途徑 1a 與途徑 1 之唯一差別是在預覽/列印之前執行 ps2up 程式對 test.ps 檔案作進一步轉換。轉換之後, 每一頁版面會縮小, 然後兩頁合併爲一頁列

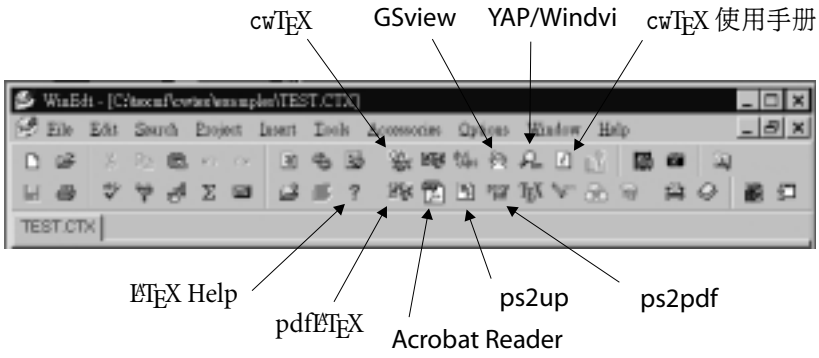


圖 3.3: WinEdt 圖像設定

印。在排版術語中, 這稱爲 2up, 目的是節省用紙。在 WinEdt 視窗內, 我們只須按下 Shift+[F11] 功能鍵, 電腦即自動執行 \texmf\cwtex 檔案夾內之 ps2up 批次檔。此批次檔之內容如下:

```
ren %1.ps %1.psa
call psnup.exe -2 -pletter -s0.8 %1.psa %1.ps
del %1.psa
```

第 2 行指令中之 -s0.8 指示將文稿縮小爲 80%。若有必要, 可自行調整此數字以放大或縮小。另外, -pletter 是設定使用 letter size 紙張, 必要時可改用其他尺寸。

欲使用 ps2up 程式, 文稿前端須加入指令調整版面位置, 否則文字版面可能列印到紙張外面。我們的經驗是在文稿前端加入下列兩行指令:

```
\hoffset=-2cm
\voffset=3cm
```

以上說明使用功能鍵排版的方法。除此之外, 我們也可以點選 WinEdt 視窗內之功能圖像啓動排版程式。如圖 3.3 所示, 視窗上方靠近中央的位置有一 cwTEX 圖像, 以滑鼠點選此圖像, 即等於是按下 [F9] 功能鍵。其右方的三個圖像分別代表 LaTeX, Dvips, 與 GSview, 而 Dvips 圖像下方爲代表 ps2up 程式之圖像。

如果要以 YAP 預覽/列印, 請點選 GSview 右方之圖像。在 MiKTEX 系統下, 此圖像代表 YAP 軟體; 在 fpTEX 系統下, 此圖像代表 Windvi 軟體。回到

cwTEX 圖像, 其下方爲代表 pdfLATEX 之圖像。此一程式可將 TEX 檔案排版爲 PDF 格式, 以下將有進一步說明。pdfLATEX 右方爲 Acrobat Reader, 用於預覽/列印 PDF 檔案。

爲方便查詢, 我們另外設定兩個說明檔圖像。第一個置於 pdfLATEX 左方, 其內容是 LATEX 指令之簡要說明。第二個是本書全部內容, 格式爲 PDF。以滑鼠點選 YAP/Windvi 右方之圖像 (白紙背景上有一個英文 ***i*** 字母), 電腦將啓動 Acrobat Reader 軟體, 並開啓 cwTEX 使用手册以供查詢。

途徑2: cwTEX ➔ LATEX ➔ YAP/Windvi

途徑 2 之排版方式與途徑 1 之差別是預覽/列印之方法不同。以 [F10] 功能鍵排版完成之後, 按 [F8] 使用 YAP 預覽排版結果。若一切滿意, 由 YAP 軟體視窗內可列印結果。YAP 全名爲 Yet Another Previewer, 這是 MiKTEX 所提供的軟體。如果你安裝的是 fpTEX, [F8] 功能鍵將啓動 Windvi 軟體。

LATEX 排版後將產出 test.dvi 檔案, 其內並不含有預覽/列印所需之字型檔。以 YAP 預覽/列印時, 軟體會自動自硬碟中取得描點字型 (bitmapped font)。譬如, test.ctx 檔案中使用了 CMR12 之英文字體, 因此 YAP 須取得 CMR12.PK 之描點字型才能預覽/列印。此測試檔檔案中也使用了明體中文字體 M0, 因此軟體也須取得 M0.PK 描點字型。

描邊字型

描點字型

描邊字型可能來自三種來源: 一是 METAFONT 字型檔, 二是 PostScript 字型檔, 第三是 True Type 字型檔。以目前的發展而論, 第二種的 PostScript 字型檔使用最廣, 品質也可能最好。 這三種字型檔都是所謂的描邊字型 (outline font)。描邊字型與描點字型的差別可以由本頁圖例說明。 圖中上方是英文字母 Q 描邊字型的例子, 它是由描繪出字母外框的線條所構成。當 YAP/Windvi 軟體要將排版結果呈現於顯示器上或列印時, 它先使用工具程式由描邊字產生圖下方之描點字型。描點字型是由一個一個的小黑點構成, 這就是我們在顯示器或紙上所看到的。

第一次使用 YAP/Windvi 預覽排版結果時, 軟體視窗內會出現訊息, 說明正在創造描點字型檔。描點字型產生之後即存放於硬碟內, 以供下一次使用。因此, YAP/Windvi 程式使用一段時間之後, 硬碟中已累積了相當數量的描點字型。此時預覽/列印排版文稿即不須再產生描點字型, 排版結果很快出現在顯示器上。

途徑3: cwTEX ➔ pdfLATEX ➔ Acrobat Reader

途徑 2 是將排版結果轉換爲 DVI 格式, 本途徑則是轉換爲 PDF 格式。PDF 格式可以說是 PostScript 檔案格式的特別版本, 主要特性是具有網路出版的功能, 這兩種格式都是 Adobe 公司發展出來的。第 16 章將有進一步說明。文稿 test.ctx 以 [F9] 轉爲 test.tex 之後, 按下 Alt+[F10] 或者點選 pdfLATEX 圖像, 即執行 pdflatex, 排版結果爲 test.pdf。預覽/列印 PDF 檔案, 可使用 Acrobat Reader, 功能鍵爲 Alt+[F12], 或直接點選功能圖像。

除了將 TEX 文稿直接排版爲 PDF 格式之外, pdfLATEX 系統的設計目標之一是要使用 True Type 字體排版文稿, 但這一部份的功能目前仍在發展當中。pdfLATEX 可將文稿直接排版爲 PDF 格式, 功能甚佳。不過, 受限於 PDF 格式之功能, 此一程式對於文稿內引入外製圖形檔有一些限制。譬如, 文稿內無法直接引用 EPS 圖形檔, 不過可以接受 PDF, JPEG 等圖形檔。軟體作者尙提供工具程式可將 EPS 圖形檔案轉換爲 PDF 格式, 請參考其說明檔。

如果文稿內含有外製 EPS 圖形, 而排版結果欲轉成 PDF 格式, 可使用 Ghostscript 6.0 版所提供之工具程式: ps2pdf。請依途徑 1 先將排版結果轉換爲 PostScript 格式, 再點選 ps2pdf 圖像, 工作檔案夾內即產生 PDF 檔案。

途徑4: cwTEX ➔ LATEX ➔ dvipdfm ➔ Acrobat Reader

除了使用 pdfLATEX 之外, 我們也可以使用 dvipdfm 工具程式轉換出 PDF 檔案, 程式作者爲 Mark Wicks。不過, 欲使用此程式, 文稿須先以 LATEX 排版。以 latex 排版之後, 按下 Alt+[F11] 即自動執行 dvipdfm 程式。請注意, dvipdfm 對於外製圖形之格式也有一些限制。譬如, 文稿也不能引用 EPS 圖形檔。

3.2 WinEdt 功能簡介

設定功能鍵的目的是爲了方便排版工作。cwTEX 安裝程式雖然爲 WinEdt 設定一些功能鍵, 但某些情況下使用者可能須再加調整。譬如, 安裝程式假設你是將 GSview 安裝於硬碟 c:。如果 GSview 是安裝於 d:, 功能鍵即無法運作。底下我們將簡單說明如何調整 WinEdt 之功能鍵設定。

表 3.1 列出 WinEdt 軟體之設定。表左欄之功能鍵上一節已略有介紹, 其中 [F7] 是用以啓動 tex 程式。如果文稿是以 TEX 指令排版, 則按 [F9] 之

表 3.1: WinEdt 功能鍵設定

排版與文稿編輯		中文標點符號	
[F9]	啓動 cwtex	Ctrl+Shift+i	、
[F10]	啓動 latex	Ctrl+Shift+o	。
[F11]	啓動 DVIPS	Ctrl+Shift+l	？
[F12]	啓動 GSview	Ctrl+Shift+k	，
Shift+[F11]	啓動 ps2up	Ctrl+Shift+k	；
[F8]	啓動 YAP	Ctrl+Shift+h	「
Alt+[F10]	啓動 pdflatex	Ctrl+Shift+j	」
Alt+[F11]	啓動 dvipdfm	Ctrl+Shift+f	『
Alt+[F12]	啓動 Acrobat Reader	Ctrl+Shift+g	』
[F2]	儲存檔案	Ctrl+Shift+r	《
[F5]	搜尋字串	Ctrl+Shift+t	》
[F3]	再搜尋同一字串	Ctrl+Shift+y	〈
Shift+[F3]	搜尋前一字串	Ctrl+Shift+u	〉
[F6]	搜尋/替換字串		
Ctrl+[F1]	行首加入 % 指令		
Alt+u	取消上一指令（undo）		
Ctrl+g	游標移至特定行（goto）		
Alt+d	刪除本行		
Alt+k	刪除游標至行尾之文字		
Ctrl+[F10]	啓動 htex386		

後, 應按 [F7] 排版。表右欄之功能鍵主要目的是方便輸入中文標點符號。例如, 在 WinEdt 視窗內按 Ctrl+Shift+o 即出現中文句點。Ctrl+Shift+k 按鍵可產生英文逗點, Ctrl+Shift+m 按鍵爲分號, 這是爲了方便注音輸入法者之設定。

表 3.1 左欄下方是一些常用指令之設定。例如, 按 [F2] 鍵即儲存檔案; 功能鍵 [F5] 設定爲搜尋字串, [F6] 則爲搜尋/替換字串。按下 [F5] 功能鍵之後, 視窗內出現一些選項, 我們可以設定僅於選定區域內尋找字串, 或者搜尋整篇文稿。另外, 我們也可設定英文字母是否區分大小寫。功能鍵 Alt+u 設定爲「取消上一指令」, 這是所謂的 undo 功能。譬如, 若你上一個動作是刪除一段文字, 但馬上發現不該刪除, 則按下 Alt+u 即可回復原文字。Alt+u 功能鍵可連續使用, 如果連續按鍵 5 次, 則在此之前 5 個編輯動作將依序取消。

WinEdt 軟體視窗內下方工作列上有幾個選項, 右邊第 3 項爲 [LINE], 此

表示編輯功能是在「文字行」模式。譬如, 如果將游標移於本行之首, 按下 [Shift] 再將游標下移一行, 則本行文字將呈反白。反之, 如果以滑鼠點選 [LINE] 方塊, 編輯功能將進入區域方塊模式 [BLOCK]。在此模式下, 我們可複製或刪除選定之方塊區域。

視窗下方工作列最右邊一項設定所謂的 Document Mode, 原始內定值爲 TeX。如果是中文稿件, 我們建議使用 cwTeX 模式。點選工作列最右邊之方塊, 即出現一小視窗, 由其中之 Document Mode 可選用 cwTeX 模式。WinEdt 文字軟體功能甚多, 自行調整之空間很大。在軟體視窗內按 [F1] 功能鍵即開啓「使用說明」, 有興趣者請自行研究了解。

WinEdt 對大部分常用之編輯指令都設有功能鍵。不過, 每一使用者各有其偏好與習慣。若你不習慣以上之功能鍵設定, 可自行重新設定。請見 4.5 節之說明。

表左欄最下方設定 Ctrl+[F10] 爲啓動 hlatex 程式, 此爲 emTEX 系統之 htex386。依標準 TEX 系統之設計, 排版文稿內不能使用超過256種字型檔。目前版本之 MiKTEX 亦有此限制; 但 fpTEX 則可使用1000種字型檔。若選擇安裝 MiKTEX, 程式會自動加裝 emTEX 之 htex386, 此程式允許使用759種字型檔。實際排版時, 直接以 Ctrl+[F10] 替代 [F10] 功能鍵即可; 其餘步驟與原來相同。

除了 htex386 之外, 我們也可使用 lambda 程式, 此程式也允許文稿內使用較多的字型檔。下一章將有進一步說明。

3.3 排版訊息

執行各排版步驟時, 顯示器上會出現一些訊息。了解這些訊息的意義對於偵測錯誤很有幫助。第一類訊息之內容爲記錄處理過程及結果; 第二類訊息則指出錯誤所在。

執行中文轉換程式時, 顯示器上將出現下列訊息:

```
This is cwTeX preprocessor, version 12.0g.
**TEST.CTX
(TEST.CTX [7])
Output written on c:\xtemp\TEST.tex & cinput.tex.
Transcript written on test.xlg.
```

第 1 行訊息末端的 12.0 爲 cwtex 的版本號碼。第 3 行的 [7] 表示文稿長度爲 7 行。第 4 行訊息則說明中文字轉換之後, 產生 test.tex 與 cinput.tex 兩個檔案。這兩個檔案直接移入 c:\xtemp 檔案夾內。最後一行的意思是說: 中文轉換過程之訊息將記錄於 test.xlg 檔案。因此, 如果來不及從視窗內讀取所有的訊息, 我們仍可事後開啓此一檔案進行了解。

以 latex 編排文稿時, 也會產生許多訊息。latex 處理的並不是原始的 test.ctx, 而是轉換之後的 test.tex 與 cinput.tex。執行指令之後, 一方面顯示器視窗內出現排版訊息; 同時, 所有訊息內容也將記錄於 test.log 檔案內。以測試檔爲例, 排版訊息如下:

```
This is TeX, Version 3.14159 (MiKTeX 1.20) (preloaded ...
**TEST
(TEST.tex
LaTeX2e <1998/12/01>
Babel <v3.6k> and hyphenation patterns for american, ...
ohyphenation, loaded.
(cinput.tex) (d:\texmf\tex\latex\base\article.cls
Document Class: article 1999/01/07 v1.4a Standard LaTeX ...
(d:\texmf\tex\latex\base\size12.clo
...
[1] (TEST.aux) )
Output written on TEST.dvi (1 page, 644 bytes).
```

上述訊息最後一行 ... on TEST.dvi (1 page, 644 bytes). 說明排版結果長度爲 1 頁, 存於 test.dvi 檔案中, 長度爲 644 bytes。此一檔案即可用於預視及列印。

LaTeX 編排之後的檔案, 其附加檔名爲 dvi, 代表 device independent, 其意義是若將此文稿在不同的列表機 (devices) 上印出, 或在不同的顯示器上預視, 版面模樣將完全相同, 差別只是在於精細程度不同。因此, 若我們有一部噴墨印表機, 則文件初稿可以先在此印表機上印出。等到最後的版面確定之後, 再以高密度的雷射印表機或相紙輸出機 (phototypesetter) 印出。

4 安裝與設定

cwTEX 中文處理程式必須附加於現成的 TEX 系統才能作排版工作。目前, 幾乎任何作業系統都可以使用 TEX 排版軟體。中文 cwTEX 程式主要是配合 Windows 平台上之排版系統。但是, 經由熱心人士幫忙, cwTEX 系統已經可以在 Linux 平台上執行。本章介紹如何在 Windows 平台上安裝 cwTEX 系統; Linux 平台上的安裝方法, 請見光碟內所附說明檔。

Windows 平台上有好幾套 TEX 系統, 其中供免費使用的包括 MiKTEX 與 fpTEX 兩套。兩套系統各有長處。MiKTEX 的作者爲 Christian Schenk, 此系統大約在 1998 年秋天漸趨穩定成熟。fpTEX 作者爲 Fabrice Popineau, 此系統是根據 Unix 平台上之 WEB2C 發展出來的。因爲許多專業的 TEX 使用者都是在 Unix 平台上工作, 並發展出許多工具程式出來, 因此 fpTEX 含有較豐富的工具程式。

除了 Windows 平台之外, cwTEX 也可以和 MSDOS 平台的 emTEX 結合使用, 此一系統是由 Eberhard Mattes 所發展。本書第一版即介紹 cwTEX 與 emTEX 之結合系統。但如果你主要是在 Windows 上工作, 使用 MiKTEX 或 fpTEX 較方便。不過, 相較於 MiKTEX 而言, emTEX 有一些優異的功能。譬如, 依原始 TEX 系統之設計, 一篇文章使用之字型檔不得超過256種。如果我們排版內容複雜的中文著作, 字型使用可能超過此一限制。譬如, 本書使用之字型檔合計約 390 種左右, 已超過 TEX 的上限。

在 Windows 平台上, fpTEX 容許使用約1000種字型檔, MiKTEX 目前之版本則僅容許約256種字型檔案。但是, emTEX 系統容許一篇文稿使用759種字體。因此, 如果你選擇安裝 MiKTEX, 則在排版複雜文稿時, 可能無法執行。此時, 可改用 emTEX 排版; 以下將有進一步之說明。

4.1 取得軟體

安裝軟體的第一步驟是取得所有軟體, 如果你購買使用手册, 其中所附光碟

已包含所有軟體。如果沒有光碟片, 你也可以從 cwTEX 網站下載。光碟或網站上之軟體大部分可供個人免費使用, 但原作者仍有版權; 你不可以出售這些軟體牟取利潤。相關的細節, 請參考各軟體之版權聲明。各軟體通常會不定期更新版本。因此, 網站上所存放者, 可能較光碟上之版本爲新; 必要時, 請自行下載更新。

從安裝的角度來看, 全部的排版軟體程式可分爲三部分:

- 工具軟體: 如 WinEdt 文字編輯軟體, GSview 等,
- TEX/LATEX 排版系統: 在 Windows 平台上可選用 MiKTEX 或 fpTEX,
- cwTEX 中文處理系統與中文字型檔。

以上這些軟體如何取得? 先說明前兩類軟體之來源。各國之 TEX 愛好者成立一國際組織, 稱爲 TUG (TEX User's Group), 網址爲:

http://www.tug.org

幾乎所有關於 TEX 系統之資訊都可以自此取得; 而所有免費下載之 TEX 軟體或相關工具程式則儲存於 CTAN (Comprehensive TEX Archive Network)。cwTEX 有一個網站, 你可以經由 cwTEX 網站聯結到 CTAN:

http://ceiba.cc.ntu.edu.tw/tmwu

國內有幾個網站複製了 CTAN 的內容。不過, cwTEX 系統目前並未置於 CTAN 內, 而是直接置於 cwTEX 網站。目前國內至少有三個網站存放 cwTEX 系統; 除了 cwTEX 系統之外, 這三個網站也存放安裝完整排版系統所需之其他軟體。對於一般的使用者而言, 這三個網站所存放之軟體已足供所用, 不須再到 CTAN 下載:

ftp://140.112.150.253/cwTeX（台大經濟系）
ftp://fpt1.sinica.edu.tw/pub2/tex（中研院電算中心）
ftp://192.192.110.1/cwTeX（嘉南藥理學院）

這三個 ftp 網站也可以由 cwTEX 網站連結上去。

以上三個 ftp 網站中, 中研院網站尙複製有 CTAN 網站之軟體與工具程式。根據使用經驗, 中研院網站比其他兩站較易連結, 傳輸速度也較快。如

果你要下載 CTAN 上之軟體, 可直接至此網站取用。國內另一個複製 CTAN 內容之網站爲中正大學電算中心, ftp 網址爲:

```
ftp://ftp.ccu.edu.tw/pub1/tex
```

如果你購買使用手册, 其內附有光碟; 所有檔案依其性質存於下列檔案夾 (file folder, 亦稱爲子目錄):

- `\dos`
- `\windows`
- `\linux`

`\dos` 檔案夾內存放本書第一版所介紹的 emTEX, 目前應該很少人使用了。`\linux` 存放適用於 Linux 平台上之 cwTEX 系統, 此一版本仍在測試中。

`\windows` 檔案夾內進一步分 `\fptex` 與 `\miktex` 兩檔案夾。如果你的電腦使用的是 Win95/98/NT, 或者 Windows 2000, 應從這兩個檔案夾內擇一安裝。以 `\miktex` 檔案夾爲例, 其內又有下列幾個子檔案夾:

`\adobe`	Acrobat Reader 與 PostScript 列印驅動程式,
`\contrib`	使用者提供之程式, 如漢書軟體之介面程式,
`\cwtex`	cwTEX 系統,
`\gsview`	GSview/Ghostscript 軟體,
`\latex2html`	LaTeX2HTML 工具軟體,
`\miktex`	MiKTEX 系統,
`\wincmd`	輔助工具軟體,
`\winedt`	文字編輯軟體。

相對的, 在 `\fptex` 檔案夾內也有類似的子檔案夾; 其中, `\fptex` 子檔案夾存放 fpTEX 系統。

檔案夾 `\winedt` 內存放 WinEdt 文字編輯軟體。請注意, 此文字編輯軟體爲 shareware, 你可以免費試用約 30 天。若欲繼續使用, 須付費登記。你的電腦中可能已有一套文字編輯軟體。即使如此, 我們建議你仍選擇安裝 WinEdt。上一章曾說明如何透過 WinEdt 軟體功能鍵進行排版工作。系統安裝完成之後, 試用幾次, 然後在你原使用的文字編輯軟體中作類似的設

定, 以方便排版工作。\winedt 檔案夾內置放 Windows Commander, 這是一個好用的檔案管理工具, 也是 shareware。

在 Windows 平台上安裝排版軟體請按照: (1) 工具程式, (2) TEX 系統, (3) cwTEX 中文程式之順序; 先後順序若顛倒, 可能無法正確安裝。以下三節分別說明各部分之安裝方法。底下的說明假設你有光碟; 若無光碟, 請自網站下載各項軟體, 依原檔案夾結構存放於硬碟內, 再依序安裝。排版系統安裝完成之後, 4.5 節說明如何進一步的調整。

4.2 安裝工具程式

所謂工具程式是指 WinEdt 文字編輯軟體與 GSview 預視/列印等軟體。第一步是安裝文字編輯軟體。進入光碟 \winedt 檔案夾內, 直接執行其內之執行檔案 winedt32.exe, 軟體將解壓縮於 c:\windows\temp 檔案夾內; 其下有 \144mb\Disk1 子檔案夾。執行 setup.exe 即開始安裝。安裝完成之後, 所有暫存檔皆可刪除。依內定值, WinEdt 會安裝於 c:\Program Files\WinEdt 檔案夾內, 除非有必要, 請勿更改。WinEdt 的使用方法及設定, 請見 4.4 節。

安裝文字編輯軟體之後, 接下來是安裝 Ghostscript 與 GSview。若硬碟空間夠大, 也請選擇安裝 Acrobat Reader 軟體。你的電腦中可能已安裝有 5.50 或更舊版 GSview, 請解除舊版, 安裝新版。如果繼續使用舊版, 以下步驟之設定無法正確執行。新版 Ghostscript 某些功能是舊版所不能及的。欲安裝 Ghostscript 6.0 版, 請直接執行 \gsview 檔案夾內之 gs600w32.exe 即自動安裝。其次, 安裝 GSview 請執行同一檔案夾內之 gsv28w32.exe。若選用內定值, 以上程式將安裝於 c: 硬碟。若選擇其他硬碟, 底下的安裝中, 有些步驟須自行調整。

Acrobat Reader 用於預覽/列印 PDF 格式之檔案, 目前 (2000 年初) 的最新版本是 4.05 版。根據使用者的經驗, 新版之功能較舊版 (3.0) 爲佳。若你原安裝舊版, 請刪除舊版, 重新安裝新版。新版 Acrobat Reader 檔名爲 ar405eng.exe, 也是直接執行即可安裝。

除以上兩項工具之外, 你還可選擇安裝 Windows Commander。此工具程式其實與排版無關, 但複製/移動檔案很方便。若欲安裝, 請執行檔案夾內之 install.exe。最後, LATEX2HTML 軟體是用於製作 HTML 網路檔案, 安裝方法請見第 16 章。

4.3 安裝 TEX 系統

Windows 平台上的作業系統包括 Windows 95/98, Windows NT, 與 Windows 2000。以下的安裝說明大抵適用於任何一套系統, 但 Windows NT 與 2000 有幾項細節須注意, 請參考 4.4.3 節之說明。

在 Windows 平台上, MiKTEX 與 fpTEX 是兩套免費使用之 TEX 系統, 每一套系統各有優點。對於一般使用者而言, 兩者之差異不大。以目前的情況來看, MiKTEX 更新的速度較快, 新的工具程式似乎也較快地加入系統內。不過, 如果你要排版複雜的長篇書籍, 其中使用之字型檔可能超過256種, 則我們建議使用 fpTEX。不過, MiKTEX 所占硬碟空間較少, 如果你有硬容量的考慮, 請選用 MiKTEX。以下先介紹安裝 MiKTEX 的方法, 再介紹 fpTEX。

4.3.1 安裝 MiKTEX

在2000年初, MiKTEX 的最新版本是1.20。新版本與舊版 (1.11版) 的設定大不相同。如果你的電腦原安裝舊版, 請先執行 Windows 之 "我的電腦|控制台|新增/移除程式" 移除舊版。但此一程式並無法完全清除所有檔案, 因此請進一步將 `\texmf` 與 `\localtexmf` 內之檔案刪除。檔案夾內若有自己寫作之檔案或下載之巨集套件, 請先作備份。

欲安裝 MiKTEX, 請先進入 `\miktex` 檔案夾內, 執行 `setupwiz.exe` 安裝程式。安裝過程中, 電腦會詢問幾個問題, 包括要安裝於 `c:` 或 `d:`, 檔案夾名稱等等。原則上你可以選擇安裝於任一硬碟, 但下一步驟安裝 cwTEX 時, 安裝程式假設 TEX 系統是位於 `c:`, `d:`, 或 `e:`。因此, TEX 系統請勿安裝於 `f:` 之後。除了選定硬碟位置外, 其餘問題請直接按 [Enter] 鍵, 選用內定值。

安裝完成之後, 硬碟中將出現:

- `\texmf`
- `\localtexmf`

兩檔案夾, 前者存放 MiKTEX 主系統, 後者存放個人檔案及其他字型檔案; 兩檔案夾請安裝於同一硬碟。安裝程式執行完畢之後, 請以文字編輯軟體開啓硬碟 `c:` 根目錄之 `autoexec.bat` 批次檔, 請在該檔案末端加入新的一行指令:

```
set path=c:\texmf\miktex\bin;%path%
```

如果 MiKTEX 是安裝於 d:, 則指令行內之 c: 請改爲 d:。加上此一行指令的目的是讓 MiKTEX 程式執行時, 能自動找到所需之檔案。如果你電腦曾安裝舊版 MiKTEX, 此一步驟可省略。

此外, 請以文字編輯軟體開啓硬碟 c: 根目錄之 config.sys 檔案, 檢查其中是否類似下列之指令行:

```
shell=c:\command.com /e:2048 /p
```

如果沒有, 請加入。以上指令的目的是將儲存環境變數 (environment variable) 之空間加大爲 2048 bytes。若有必要, 此數字還可以再加大一些。以上之設定假設 command.com 檔案是置於 c:\ 根目錄內, 若置於他處, 請對應修正。

如果電腦中以往曾安裝 emTEX 系統, 則 autoexec.bat 批次檔內之 path 指令行中可能有:

```
c:\emtex\bin;c:\emx\bin;
```

之設定, 請將之刪除。此外, 同一檔案內可能還有下列一行指令:

```
call c:\emtex\bin\setctx.bat
```

請亦將之刪除。以上設定完成之後, **請重新啓動電腦**, 設定才能生效。

下一步驟是安裝中文系統, 但在此之前請先確定 MiKTEX 安裝是否正確。重開機之後, 進入 \texmf\doc\miktex 檔案夾內, 點選 miktex.dvi 檔案, 片刻之後, 電腦會詢問使用那一個程式開啓, 此時應選用:

```
\texmf\miktex\bin\yap.exe
```

第一次執行時, 畫面會出現訊息, 說明正在產生字型檔; 片刻之後, 排版結果即出現在顯示器上。YAP 程式全名爲 Yet Another Previewer, 其功能是預覽/列印排版結果。欲列印文稿, 請先設定印表機機種, 方法如下: 啓動 YAP 軟體視窗上方工作列中之 “View|Options|Printer” 即可設定印表機。同一畫面 Printer 選項左邊爲 Display, 點選此項, 在 Mode 空格中請選用同型印表機。

若能正確預覽/列印排版結果, 表示 MiKTEX 已安裝成功。排版時, 個人文稿或圖形檔案可能置放於硬碟特定位置。爲了讓 MiKTEX 自動找到所需檔案, 我們可以設定讓程式自動搜尋, 請見4.5節說明。

MiKTEX 附有很多說明檔, 置於 \texmf\doc\miktex 檔案夾內。欲了解 MiKTEX 之運作, 請閱讀各說明檔。部分檔案之延伸檔名爲 HTML, 直接點選即啓動瀏覽器閱讀。部分檔案之格式爲 PDF, 須使用 Acrobat Reader 或 GSview 預覽/列印。MiKTEX 附有許多巨集套件, 但新的巨集套件不斷出現。欲使用自己下載之巨集套件, 應將巨集套件檔案複製於 \localtexmf\tex\latex 檔案夾內。**請注意**, 複製檔案之後, 務必執行下列指令以更新檔案資料系統:

```
c:>initexmf -u
```

否則排版程式可能找不到新移入之檔案。

4.3.2 安裝 fpTEX

欲使用 cwTEX 排版系統, 你僅須安裝 MiKTEX 或 fpTEX 兩者之一。如果同時安裝兩套系統, 執行時會發生衝突。如果你的電腦以往曾安裝 MiKTEX, 現欲改用 fpTEX, 請先將 MiKTEX 刪除; 方法是將 \texmf 與 \localtexmf 兩檔案夾內之檔案全數刪除, 但自己製作的檔案請留著。此外, 還須把硬碟根目錄 autoexec.bat 內相關之 path 設定刪除。如果電腦中以往曾安裝 emTEX, 請亦將之刪除。相關細節, 請見上一小節之說明。

在2000年初, fpTEX 最新版本是0.3版, 全部檔案合計約70MB, 安裝完成之後, 占硬碟空間約120MB。若加上中文字型檔, 可能達170MB。爲了配合下一階段的 cwTEX 安裝程序, 請注意兩項要點:

- 請將 fpTEX 安裝於根目錄,
- 個別使用者之檔案夾請設定爲 \localtexmf。

若不照以上設定, 下一階段所安裝之 cwTEX 無法執行。安裝方法很簡單, 自光碟 \fptex 檔案夾內執行 setup.exe。安裝程式開始執行後, 第3個畫面是要選擇軟體安裝之位址 (Destination Folder), 內設值爲 c:\tex, 請去掉 tex, 變成 c:\。或者, 你也可以選擇安裝於 d:\ 或 e:\。再提醒一次, 請選擇安裝於根目錄。

下一個畫面是選擇 local tree 之安裝位置。若主程式選擇安裝於 c:\, 此處之選項會變爲 c:\texmf.local, 請將之改爲 c:\localtexmf。**請注意**,

若不更改, 下一階段之 cwTEX 安裝無法成功。若上一畫面選擇安裝於 d:\, 此處則改爲 d:\localtexmf。

接下來, 你可選擇安裝之內容, 選項由最精簡的 Basic 到最完整的 Full 共計 4 種。我們建議你挑選的 Recommended。安裝程式的最後一個畫面會詢問是否安裝 “Ghostscript/Ghostview” 等 4 項程式。若依本章前面之步驟, 這些程式在安裝工具程式的階段已安裝於硬碟內, 故此時應直接跳過不選。

fpTEX 安裝程式執行完畢之後, 會自動在 autoexec.bat 檔案內加入 4 行指令。若選擇安裝於硬碟 c:\, 最後二行將爲:

```
SET FPTEX=c:\
PATH=%FPTEX%\bin\win32;%path%
```

請將第一行末端之反斜線去除, 改爲:

```
SET FPTEX=c:
```

此外, 還須在檔案最末端加入:

```
PATH=c:\gstools\gs5.50;%path%
```

經過以上步驟之後, 請重新開機, 讓設定生效。

安裝完成之後, 硬碟內新增下列 5 個資料夾:

- \bin\win32 存放執行檔,
- \texmf 存放 TEX 系統,
- \localtexmf 存放個人檔案,
- \man 存放 Unix 格式之說明檔,
- \info 存放說明檔。

在進行下一步驟之前, 請先測試 fpTEX 是否安裝成功。請進入 c:\texmf\doc\programs 檔案夾內, 點選其中之 dvips.dvi, 電腦將啓動預覽/列印軟體 Windvi, 並開始創造描點字型。此程式是透過 Ghostscript 創造描點字型檔, 如果電腦中並未安裝 Ghostscript 軟體, 或者前面所述之 path 指令設定有誤, 即無法產生描點字型檔。

欲列印結果, 請先設定描點字型之密度, 點選軟體視窗上方之 “view | options”, 畫面右上角之 MF mode 請選擇印表機。若印表機機型不在其中, 請選

用接近的機型。譬如, 若印表機爲 HP1100A, 可選用 “HP Laser Jet 5”。畫面左上角 Pixels per inches 之數字爲描點字型之密度。若印表機爲 300dpi 密度 (如一般之 HP 印表機), 可選用 300, 若爲日系噴墨印表機 (如 Canon 或 Epson 機型), 解析度應選用 360。

若能正確預覽/列印排版結果, 表示 fpTEX 已安裝成功。排版時, 個人文稿或圖形檔案可能置放於硬碟特定位置。爲了讓 fpTEX 自動找到所需檔案, 我們可以設定讓程式自動搜尋, 請見4.5節說明。fpTEX 附有許多巨集套件, 但新的巨集套件不斷出現。欲使用自己下載之巨集套件, 應將巨集套件檔案複製於 \localtexmf\tex\latex 檔案夾內。**請注意**, 複製檔案之後, 務必執行下列指令以更新 fpTEX 檔案資料系統:

```
c:>mktexlsr
```

否則排版程式可能找不到新移入之檔案。

fpTEX 附有很多說明檔, 置於 \texmf\doc 檔案夾內。欲了解 fpTEX 之運作, 請大略閱讀各說明檔。

4.4 安裝 cwTEX

安裝 cwTEX 中文程式分兩步驟, 第一步驟是安裝程式、輔助工具與5種基本字體。若安裝成功, 可進一步安裝其他字型。cwTEX 程式與相關檔案置於光碟 \cwtex 檔案夾內, 中文字體檔案則置於其下之 \fonts 子檔案夾內。

開始安裝之前, 請先確定英文 TEX 已能執行。此外, 如果 WinEdt 爲開啓狀態, 請先關掉; 因爲安裝程式會須重新設定 WinEdt。直接執行 \cwtex 檔案夾內之 setup.bat, 即開始安裝工作。安裝過程中須創造中文字體之對應檔案, 此一過程須花上數分鐘時間, 請耐心等候。安裝完成之後, **請重新啓動電腦**。

cwTEX 安裝完成後, 硬碟 \texmf 檔案夾下將新增 \cwtex 子檔案夾, 存放 cwtex 執行檔及相關工具; 中文字型檔則置於 \texmf\fonts 下。本書所介紹的各巨集套件則置於 \texmf\tex\latex 檔案夾下。爲方便參考, 我們將部分套件的原說明檔置於 \texmf\cwtex\doc 檔案夾內。請注意, fpTEX 或 MiKTEX 系統內可能已含有某些套件; 譬如, fpTEX 即提供 crop 套件。必要時請自行刪除其中較舊的版本。

WinEdt 軟體每一次執行結束時, 都會將所有的設定儲存於 WinEdt.ini 內, 此檔案是置放於 \Program Files\WinEdt 檔案夾內。安裝 cwTEX 系統時, 安裝程式會更新此檔案, 原檔改名爲 WinEdt.ori。如果你要使用原始設定, 只要將 WinEdt.ori 改回 WinEdt.ini 即可。

4.4.1 測試與調整

重新開機之後, 應先測試安裝是否正確。啓動 WinEdt 軟體, 開啓 test.ctx 測試檔, 視窗上方工作列上應出現 cwTEX 圖像 (icon), 以滑鼠點選 (或直接按 [F9]), 電腦即執行 cwtex 將測試檔之中文字轉換爲 TEX 之控制碼。萬一出現錯誤訊息, 請參考下文的排除錯誤說明。

轉換中文之後, 以滑鼠點選 LATEX 圖像 (或按 [F10]), 即啓動排版程式。接下來, 點選 DVIPS 圖像 (或按 [F11]), 排版結果將轉換爲 PostScript 格式。最後, 點選 GSview 圖像 (或按 [F12]), 即啓動預覽/列印軟體。除了以上的途徑之外, 我們也可以直接使用 YAP/Windvi 軟體預覽/列印; 或以 pdfLATEX 將文稿直接排版爲 PDF 格式, 再以 Acrobat Reader 預覽/列印; 請參見上一章之說明。

如果你對於 WinEdt 或 GSview 視窗顯示之精細度不滿意, 可自行調整。請見 4.5 節之說明。

4.4.2 安裝更多的中文字型

cwTEX 安裝程式會安裝5種中文字型, 若硬碟尚有空間, 你可選擇安裝其他中文字型檔。欲安裝中文光碟內之字型, 請自光碟之 \cwtex 檔案夾內執行 addfont.bat 安裝程式。該批次檔置於光碟 \cwtex 檔案夾內。若欲安裝垂直中文字型, 請執行 addfontv.bat。

目前, cwTEX 中文字體計23種, 字體樣本請見圖 6.3 (頁 74)。安裝中文字型時, 畫面上半部分顯示可選用之字體名稱, 下半部分爲執行選項。譬如, 要顯示硬碟內已安裝之中文字型, 請按 s。欲安裝粗明體, 請鍵入 mb; 欲安裝隸書體, 請鍵入 l; 按 [Enter] 之後, 光碟內之字型檔即複製於硬碟內。複製字型檔之後, 還須製作字型對應檔案。因此, 選擇安裝之字體、複製了字型檔之後, 完成安裝還須按 q 鍵。最後一個步驟可能要花上十幾分鐘時間, 請耐心等候。

實際上, \addfont 批次檔案作兩個動作, 一是將字型複製於硬碟, 二是製作字型對應檔 (font mapping file)。必要時, 我們可以手動將字型檔複製於硬碟內, 之後執行:

```
d:\cwtex>addfont map
```

選項 map 指示僅產生字型對應檔。

4.4.3 Windows NT 與 Windows 2000

根據幾位使用者的報告, 以上的安裝方法也適用於 Windows NT 與 Windows 2000。不過, 安裝之後, 請再確認 autoexec.bat 之 path 路徑設定是否正確。

如果你安裝 MiKTEX, 而且可能會使用 htex386 排版文稿, 請以文字編輯軟體開啓 \texmf\cwtex 檔案夾內之 hlatex.bat, 將第 6 行之 &latex 改爲 ^&latex, 否則 latex 程式無法執行。在 NT 與 Windows 2000 平台上, & 有特別的意義, 故須作以上更動。

4.4.4 常見的安裝問題

相較於工具程式與 TEX 系統, cwTEX 之安裝過程似乎較容易出現問題。本小節列舉常見的安裝問題及解決之道:

- 無法安裝成功
  請檢查硬碟空間是否足夠。整套 MiKTEX 加上 cwTEX (含中文字體5套) 約占用 50MB 之硬碟空間。若是 fpTEX 加上 cwTEX, 需更大空間。

- 安裝時出現 "... out of environment space" 訊息
  請在 c: 根目錄的 config.sys 檔案中加入下面一行指令:

  ```
  shell=c:\command.com /e:2048 /p
  ```

  重開機, 再重新安裝。

- 按 [F9] 時, 出現 "Can not run cwtex..." 訊息
  可能是 autoexec.bat 之 path 指令設定錯誤, 請再仔細檢查一遍。

- 按 [F9] 出現 "Cannot Run: cwtex.exe -d=c:\xtemp ..." 訊息
  請將 c:\texmf\cwtex\doc\test.ctx 測試檔案複製於 c:\xtemp 檔案夾內, 進入 DOS 模式, 執行底下指令:

  ```
  c:\xtemp>cwtex test
  ```

  若能順利執行, 請再執行

  ```
  c:\xtemp>latex test
  ```

  若以上兩步驟都能順利執行, 表示排版程式已正確複製於硬碟內, 問題是在於 WinEdt 軟體功能鍵未能正確設定。

  正確設定 WinEdt 功能鍵的關鍵是安裝的順序, WinEdt 某些功能鍵是由 cwTEX 安裝程式自動設定的。如果安裝 cwTEX 系統之前沒有先安裝 WinEdt, 後面的安裝步驟即無法正確設定。請試將 \texmf\cwtex 檔案夾內之 WinEdt.ini 取代 c:\Program Files\WinEdt 檔案夾內之同名檔案。

- 排版測試檔時出現 "I can't find m0.tfm" 訊息
  可能原因有二: 第一是字型檔案未複製於硬碟內, 第二是未正確設立字型對應 (font mapping) 檔案。中文字型規格檔案置於下列檔案夾:

  ```
  c:\texmf\fonts\tfm\cwtex
  ```

  請檢查檔案夾內是否有 m0.tfm 字型規格檔。若字型檔存在, 則問題可能是在於字型對應檔。中文字型對應檔有兩個, 一般正常字型對應檔名爲 cwtex.map, 變形字對應檔爲 cwtex1.map, 應置於 c:\texmf\dvips\cwtex 檔案夾內。若硬碟尙安裝直排字型, 其對應檔名爲 cwtex2.map。

- 順利使用一段時間之後, 突然無法執行
  可能原因是 WinEdt 之設定檔被改掉。WinEdt 是 shareware, 安裝之後可使用約一個月。過了期限未註册, 軟體會自動回復原始的設定, 因此無法以 [F9] 功能鍵執行 cwtex。請將 c:\texmf\cwtex 檔案夾內之 WinEdt.ini 覆蓋 c:\Program Files\WinEdt 檔案夾內之同名稱檔案。前一檔案內已設定執行排版程式之功能鍵。若欲長期使用 WinEdt 軟體, 請註册。

4.5 設定調整

安裝成功之後, 我們可以進一步作細節調整, 以使排版工作更輕鬆。本節說明各軟體的細節調整方法。

4.5.1 顯示密度與印表機

大部分的 Win95/98 桌面之顯示密度都設定爲 800×600, 或者 1024×768。若是後者, WinEdt 之內設值應該是令人滿意的結果。如果桌面顯示密度爲 800×600, 你可以考慮將桌面顯示密度提高; 或者將 WinEdt 視窗之設定調整一下。由視窗上方工作列開啓 "Options|Preferences|Font", 右下方四個數字使用於調整行距與字距, 原先數字爲 16,20,2,10; 請試改爲 16,16,1,8。左上方可選用字體, 我們的建議如下: 先點選 Font, 選取「細明體」, 按「確定」之後; 再度點選 Font, 選取 Fixedsys, 再按「確定」。

使用 GSview 軟體須調整顯示密度, 並選用印表機。顯示密度之設定方法如下: 點選視窗上方之 "Media|Display settings", 將 Text Alpha 之值由 1 改爲 4, Graphics Alpha 之值亦由 1 改爲 4; 顯示畫面會變得比較精細。

印表機設定方法如下: 由表單之 "File|Print", 可開啓 Printer Setup 畫面, 左上方之 Device 可選擇印表機。請自其中選用最接近你的印表機之機種。實際列印時, 可由 Resolution 選擇列印密度; 由 Pages 選擇列印之範圍。紙張種類可以由 Media 選單內選擇。萬一你的印表機與表單上所列出者毫無相似之處, 請試在 Derive 空格處選用 mswinpr2 驅動程式。此一驅動程式將列印工作直接交由 Windows 之印表機處理, 列印速度可能慢一些, 但任何印表機應該都可以列印。譬如, 如果你有 HP1200 印表機 (解析度達 1200dpi), 即可透過此一方式列印。GSview 所支援之印表機隨時可能更新, 請見:

```
http://www.cs.wisc.edu/~ghost/printer.html
```

4.5.2 重新設定 WinEdt 功能鍵

WinEdt 對大部分常用之編輯指令都設有功能鍵。上一章表 3.1 說明安裝程式所作之設定。不過, 每一使用者各有其偏好與習慣。若你不喜歡以上之功能鍵設定, 可重新設定。舉例來說,「再搜尋同一字串」若要重新定義爲 Shift+[F5], 方法如下: 經由 WinEdt 視窗上方之:

Options|Menu setup|Search|Search Again

將游標移至中欄下方之 Shortcut 方格, 按下 Shift 鍵不放, 再鍵入 [F5], 新設定即生效。

再舉另一個例子。功能鍵 Alt+[F12] 是設定爲啓動 Acrobat Reader 軟體, 如果你購買了完整的 Acrobat 軟體套件, 希望以 Adobe Acrobat 替代 Acrobat Reader, 設定方法如下: 開啓 WinEdt 視窗上方之:

Options|Menu setup |&Accessories

畫面左欄下方有 Acrobat Reader, 點選此項, 中間欄位的上方有 Utility 方格, 更改其內之設定即可選用 Adobe Acrobat。

排版複雜中文稿時, latex 與 DVIPS 可能出現錯誤訊息, 原因是文稿內使用太多的字型檔。前面曾說明, 我們可使用 emTEX 之 htex386 替代 latex。另外一個辦法是使用 lambda 程式。lambda 程式是 omega 程式集的一部分, 後者的設計目的之一是擴充 TEX 之功能, 以應付國際上各種語文之排版, 原始作者包括 John Plaice 與 Yannis Haralambous。要將 latex 功能鍵改爲執行 lambda 程式, 請開啓 WinEdt 視窗之:

"Options|Menu setup |Accessories|&LaTeX"

將中間上方 Utility 空格內之 latex.exe 改爲 lambda.exe。依同樣方法, dvips 程式也可以 odvips 程式替代, 後者也是特別用來處理複雜的文稿。

WinEdt 軟體彈性甚大, 幾乎任何功能或表單都可重新設定, 你可以經由功能表單上之 Help 了解設定方法。除此之外, 存放 WinEdt 軟體的檔案夾下有 \doc 檔案夾, 內有進一步的說明文件。

4.5.3 工作檔案夾

測試文稿 test.ctx 檔案原置於 c:\texmf\cwtex\examples 檔案夾內, 但在 WinEdt 視窗內按 [F9] 功能鍵之後, cwtex 會將 test.ctx 轉換爲 test.tex, 並將後一檔案移入 c:\xtemp 工作檔案夾內。接下來按 [F10] 功能鍵時, latex 會在工作檔案夾內進行排版, 結果將存爲 test.dvi。設定工作檔案夾的好處是排版過程中所產生的暫存檔全部集中於此。排版完成之後, 只要保存原始 test.ctx 檔案, 其餘暫存檔並無保存之必要。因此, 每隔幾個星期, 我們可以將 c:\xtemp 內之檔案全部刪除。

LaTeX 系統在排版/預覽/列印時, 可能須引用其他檔案。舉例言之, 第2章 examp2.ctx 例子引用了一外製圖形 examp.eps, 排版/預覽/列印程式若找不到圖形檔, 即出現錯誤訊息。如果只是偶爾使用圖形, 則最簡單的方法是將圖形檔複製一份於工作檔案夾 c:\xtemp 內, 排版程式一定找得到。但如果引用許多外製圖形, 每一圖形檔都要複製於工作檔案夾內並不方便。因此, 最好是設定讓排版/列印程式能搜尋圖形檔。兩套 TeX 系統搜尋檔案之設定方法並不同, 安裝 MiKTeX 者, 請見下一小節之說明; 安裝 fpTeX 者, 請見再下一小節。

4.5.4 MiKTeX 檔案搜尋

MiKTeX 各程式依一定的順序搜尋檔案, 如果要改變順序, 只要修改設定檔案 miktex.ini 即可; 此檔案置於 \texmf\miktex\config 檔案夾內。以文字編輯軟體開啓此一檔案, 可發現其內有許多分區, 每一分區都有標題。舉例言之, 在 [LaTeX] 標題下有下列設定:

```
Input Dirs=.;%R\tex\latex//;%R\tex\generic//
```

排版時, latex 即依上面之設定搜尋圖形檔 examp.eps。首先, 等號後面是英文句點, 代表工作檔案夾; 因此排版程式先在工作檔案夾內尋找。若找不到, 上述設定中第 1 個分號後面的 %R\tex\latex// 是下一個搜尋對象。若 MiKTeX 是安裝於硬碟 c: 之 \texmf 與 \localtexmf 檔案夾內, 則 %R 符號代表 c:\texmf 與 c:\localtexmf。設定最後之 // 符號代表所有之子檔案夾。因此, latex 程式首先搜尋 c:\texmf\tex\latex 及其下之所有子檔案夾; 若找不到圖形檔案, 則進一步搜尋 c:\localtexmf\tex\latex 及其下所有檔案夾。如果仍找不到, 即依設定行指示至下一個檔案夾內搜尋: %R\tex\generic。

事實上, examp.eps 圖形檔案是置放於 c:\texmf\cwtex\examples 檔案夾內, 因此原始之設定無法讓程式找到此圖檔。那麼, 設定應如何更改? 在 \localtexmf\miktex\config 檔案夾也有一個名爲 miktex.ini 之設定檔, 其內容與主設定檔類似, 但大幅簡化。其中, [LaTeX] 標題下之設定如下:

```
Input Dirs=.;%R\tex\latex//;%R\tex\generic//;%R\cwtex//;c:\tex//
```

相對於 MiKTEX 之內設值, 安裝程式增加了兩項設定, 一爲 %R\cwtex//, 此一設定讓排版程式可以找到 examp.eps。最後一項設定是假設使用者所有之排版文檔與圖形檔全部置於 c:\tex 或其下之檔案夾。假設你的排版檔案全部是置放於 d:\mydoc 或其下之檔案夾內, 則 c:\tex 應改爲 d:\mydoc。

除了 latex 程式外, 其他程式也須搜尋檔案。例如, 本例之圖檔爲 EPS 格式, 以 GSview 預覽/列印之前須先將排版結果轉換爲 PostScript 檔案。因此, DVIPS 也須搜尋圖檔。安裝程式也會在 miktex.ini 檔案內之 [dvips] 標題下加入以下之設定:

```
GraphicsPath=.;%R\dvips//;%R\tex//;%R\cwtex//;c:\tex//
```

此外, YAP 預覽/列印 PostScript 圖形檔時, 會先將圖形檔轉換爲特定格式之描點圖形, 再顯示於畫面上。YAP 搜尋圖形的設定是在 miktex.ini 檔案內 [MiKTeX] 標題下, 其設定如下:

```
GraphicsPath=.;%R\dvips//; ... ;%R\tex//;%R\cwtex//;c:\tex//
```

根據以上的說明, 可能有人會覺得最簡單而且萬無一失的設定如下:

```
Input Dirs=.;c://;d://
```

換言之, 我們要求排版程式搜尋硬碟 c: 與 d: 所有檔案夾內之檔案。此種設定的確萬無一失, 但成本太高。搜尋檔案需要時間; 若硬碟存放許多檔案, 搜尋時間就長。譬如, latex 排版 test.tex 時會搜尋一輔助檔案 test.aux, 這是上一輪排版時所產生的檔案。但文稿若是首次排版, test.aux 根本不存在。如果硬碟內有許多檔案, 搜尋將會花上很長時間, 且一無所獲。如果將搜尋範圍設得小一些, 執行速度會加快許多。

以上例子假設個別使用者之設定檔名爲 miktex.ini。實際上, 個人設定檔可任意取名, 置於任選檔案夾內。若自行選定檔名與檔案夾, 安裝完成之後請記得執行下列指令:

```
c:>initexmf --personal=c:\localtexmf\miktex\config\miktex.ini
```

當然, 檔案夾與檔名須替代以自行選用者。

4.5.5 fpTEX 檔案搜尋

fpTEX 設定搜尋順序之檔案名爲 texmf.cnf, 置於 \localtexmf\web2c 檔案夾內。欲設定 LATEX 搜尋路徑, 請找出下列一行:

```
TEXINPUTS.latex = .;$TEXMF/tex/... ;$TEXMF/cwtex//;c:/tex//
```

在其末端加入路徑設定。例如, 若個人所有之檔案都置於 c:\mydoc 或其下之檔案夾內, 則上一行末端應加入 ;c:/mydoc。請注意, 其前端有一區分用之分號; 而且檔案夾名稱是以右斜線起頭, 而非反斜線。

除了 LATEX 之外, 我們還須設定 DVIPS 搜尋路徑, 請在同一檔案中找出下列一行指令, 依同樣方法設定。

```
TEXPSHEADERS = .;$TEXMF/{dvips, ... }//;$TEXMF/cwtex//;c:/tex//
```

4.6 cwTEX 系統運作原理

本章以上各節說明自動安裝 cwTEX 的方法。自動安裝的好處是避免麻煩; 缺點是較缺乏彈性。當然, 你可以自行手動安裝。爲方便手動安裝者, 本節略述 cwTEX 之運作原理。

國內外目前有好幾套中文化 TEX 排版系統。這些系統表面上看來差異甚大, 但背後的原理其實很類似。對於 TEX 程式而言, 任何文字都是一個一個的符號, 排版的工作就是將特定文字符號置放於版面適當位置。從這個角度來看, 英文與中文的唯一差別是在於字數的多少。普通的英文稿件使用的英文字母符號約一百個, 而常用的中文單字約兩三千個。中文字數多, 表面上看來排版工作複雜許多; 不過, 只要有符合 TEX 規格之中文字型檔, TEX 就把中文字當成英文字編排, 毫無困難。

要讓 TEX 排版中文有兩個條件: 第一是中文字型檔案, 第二是將文稿內之中文字轉換爲 TEX 指定之規格。cwTEX 中文字型是 PostScript 格式; 中文字轉換工作則是利用 cwtex 前階處理程式。對照以上之系統架構, 安裝 cwTEX 系統的工作也分兩部分, 第一是把中文轉換程式與中文字型複製於 TEX 系統內; 第二是設定讓排版與預覽/列印程式使用中文字型。

中文轉換程式主要是 cwtex.exe。安裝時, 此一程式及一些輔助檔案皆複製於 \texmf\cwtex 檔案夾內。爲了使作業系統能自動搜尋找到工具程式, 安裝程式並在 autoexec.bat 檔案內設定搜尋路徑。

早期 TEX 系統大都使用 METAFONT 字體。譬如, 描點字型檔 cmr12.pk 是由 cmr12.mf 所產生; 後者是描邊字型檔案。不過, 現已有人將 cmr12.mf 轉換成 PostScript 格式之描邊字型檔 cmr12.pfb。cwTEX 安裝程式會設定使用 PostScript 格式之字型檔, 主要原因是使用彈性較大。

cwTEX 提供的中文字是 PostScript 描邊字型規格, 每一個字型檔內含256個中文字。一般 Windows 系統內之中文字形檔案含有一萬三千多個中文字。以明體字爲例, cwTEX 將所有的中文字拆開置放於52個字型檔案內, 檔名由 M0 排至 M51。以明體字型 M0 爲例, 字體檔案其實共有三個, 分別名爲 m0.pfb, m0.afm 與 m0.tfm。因此, 明體字共計 156 個字型檔。安裝時, 這些檔案分別複製於 \texmf\fonts 下之檔案夾內。

5 指令與文字

TEX 是幕後排版系統, 所有的排版指令必須與文稿內容同時輸入於檔案內。換言之, 我們在文稿內所輸入的中英文字, 若不是文件內容, 就是排版指令。文稿版面是以排版指令編排, 排版指令可以設定版面大小、排版表格、設計標題等等。有時候, 我們也使用指令**變更字體**以強調某一句話, 或者排版數學符號 $\sqrt{2}$。從本章開始, 我們將陸續介紹 LATEX 排版指令。本章首先說明輸入文字及排版指令的一般原則。

5.1 符號與註銷指令

文稿內容的輸入很簡單, 英文或數字可直接鍵入, 中文則須以中文輸入法輸入。但是, 有些符號在鍵盤上並沒有對應的字鍵。譬如, 商標符號 © 或者英鎊符號 £ 在鍵盤上就沒有對應的字鍵。一般的文書處理軟體通常是以內碼輸入法鍵入特殊符號。但是, 在 LATEX 中這些符號是以指令排版。譬如, 要排版 £ 記號, 我們應鍵入 `\pounds`; 要排版商標符號, 我們應鍵入 `\copyright`。LATEX 的排版指令絕大部分是以反斜線開頭, 英鎊符號用 `\pounds` 來表示, 就是一個例子。LATEX 排版數學式子的能力特別強, 大部分的數學符號也都是用指令來代表。

LATEX 指令由英文字母組成, 不能使用數字; 而且大小寫字母是有分別的。譬如, 要排版 LATEX 這幾個字, 指令是 `\LaTeX{}`。其中, 字母大小寫不可弄錯。如果輸入 `\LaTEX{}`, 將會出現錯誤。

你可能注意到 `\LaTeX{}` 指令末端加上左右大括號。爲什麼呢? LATEX 排版時, 必須分辨那些是文字, 那些是指令。指令是以反斜線 \ 起頭, 判斷上不困難。但指令結尾如何判斷呢? 原則上, 從指令的第一個字母開始, 到第一個非字母的字元爲止, 就構成指令。非字母字元包括空白、標點符號、數字等。因此, 若輸入 `the \LaTeX logo`, 排版結果爲: the LATEXlogo。標誌與其後的文字之間未隔開, 原因是指令後面的空白被當成是指令的結束, 因此 logo

表 5.1: 重音符號

ò	\‘{o}	õ	\~{o}	ǒ	\v{o}	o̧	\c{o}
ó	\’{o}	ō	\={o}	ő	\H{o}	ọ	\d{o}
ô	\^{o}	ȯ	\.{o}	o͡o	\t{oo}	o̲	\b{o}
ö	\"{o}	ŏ	\u{o}				

即緊接著編排。爲了避免此種錯誤, 我們應輸入 `the \LaTeX{} logo`, 連續的左右大括號標示指令結束, 緊接其後的空白才會排成一眞正的空格。我們也可以輸入 `the \LaTeX\ logo`, 其中反斜線之後加上空格是排版空格的指令。第三種輸入方式是 `the {\LaTeX} logo`。

LaTeX 提供許多符號可供排版德文、法文或其它語文。表 5.1 列出重音符號 (accents) 及其排版指令。 要排版 õ 符號, 我們應輸入 `\~{o}`。此一指令把一波浪符號加在字母 o 的上面。爲了易於分辨起見, 字母以大括號括起來。如果大括號中只有一個字母, 則括號可以省略。譬如, 直接輸入 `\~o` 也可得到結果。

表 5.1 的例子是把重音加在小寫字母 o 上頭, 當然這些重音可以加在任何字母或符號上面。例如, `\v{z}` 指令可得到 ž, 而 `\={A}`, 得到 Ā。除了重音符號之外, 其它的特殊符號列於表 5.2。利用表中的符號, 我們可以排版德文句子:

```
Die Höhe der Steuer muß sich durch die Wertung ...
Die H\"ohe der Steuer mu\ss{} sich durch die Wertung ...
```

事實上, 若要排版長篇德文或法文稿件, 應使用 babel 巨集套件, 請參見 15.7 節之說明。

文稿內經常須記錄排版當天的日期與文稿檔名。欲排版當天日期, 可使用 `\today` 指令。在文稿內任何地方鍵入此一指令, 排版後即出現當天的日期, 但月份是以英文字排出。譬如:

```
今天是 March 6, 2000.
今天是 \today。
```

第 15 章之圖 15.1 將說明如何修改指令使日期以中文形式排版。

欲記錄文稿檔名, 可使用 `\jobname` 指令。此項指令僅記錄主檔名, 附加檔名須自行填入。譬如, 排版本書之主檔名爲 `cxbook.ctx`, 我們在文稿末端

表 5.2: 特殊符號

§	\S	¡	!‘	¿	?‘	†	\dag	‡	\ddag
¶	\P	ø	\o	Ø	\O	ł	\l	Ł	\L
æ	\ae	å	\aa	ß	\ss	Æ	\AE	Å	\AA
œ	\oe	Œ	\OE	©	\copyright			£	\pounds

以下列指令記錄檔名與排版日期:

cxpdf.ctx (March 6, 2000) `\jobname.ctx (\today)`

輸入文稿時, 有時候我們必須在某處加註註銷說明。這些說明並不是排版文稿的一部分, 只是作提醒或補充說明之用。此時, 我們應使用 % (註銷指令, comments) 指令。第2章例1的第1行爲:

```
% file: examp1.ctx
```

這是用以註明檔案名稱。因爲其開頭有 % 指令, 排版時 % 符號及其後的文字與指令將完全略過。LaTeX 的指令大都是以反斜線起頭, 但註銷指令是一個例外。

若註銷指令出現在一行文字中間, 指令之前的文字會出現, 但指令本身連同後面所有的文字都不會排版出來。因此, 你可能馬上想到一個問題: 如何排版百分比 % 符號? 底下是一個例子:

```
台灣政府支出占 GDP 比率約27和其他國家相差不大。
台灣政府支出占GDP比率約27%,和其他國家相差不大。
```

因爲數目字 27 之後緊接著註銷指令, 其後的逗號即略過不處理。要排版百分比 % 符號, 我們必須輸入 \%。因此, 正確的指令及排版結果如下:

```
台灣政府支出占 GDP 比率約27%, 和其他國家相差不大。
台灣政府支出占GDP比率約27\%,和其他國家相差不大。
```

除了註銷指令 % 外, 另外有 9 個符號在 LaTeX 中各有其特殊用途。因此, 符號本身也須以指令方式輸入, 我們將這些符號列於表 5.3, 以供參考。除了 % 及 \ 之外, 最常用的符號是左右大括號, 它們是用來界定指令的範圍。

表 5.3: 以指令方式輸入之英數符號

符號	符號輸入指令	功能
%	\%	註銷
\	$\backslash$	定義指令
{	$\{$	左大括號
}	$\}$	右大括號
&	\&	表格排列
$	\$	數式模式
#	\#	記錄巨集指令參數
~	\~{}	加入空白或防止分割字串
^	\^{}	數式上標
_	_{}	數式下標

若欲排版第1欄之符號, 我們必須輸入第2欄所示之指令。例如, 要排版百分比符號, 輸入指令爲 \%。第3欄說明第1欄符號本身在 LaTeX 中之特殊用途。

譬如, 上面所介紹的重音符號, 即以大括號標示字母範圍。其餘各符號的用途, 我們將陸續介紹。

5.2 英文稿件輸入原則

LaTeX 是幕後排版, 版面的控制由指令爲之。輸入文字時, 我們只須顧慮文字與指令是否正確, 不必考慮輸入之文稿是否排列整齊。文稿排版是 LaTeX 的工作, 不管輸入文稿在顯示器上看來有多雜亂, 只要指令正確, 最後的結果一定令人滿意。有人在輸入文稿時刻意排得整整齊齊, 這是不必要的。

輸入文字時要注意一些基本原則。首先, 在英文的段落之間, 就像在打字機打字時一樣, 單字之間必須留有空白。用打字機打字時, 兩個字中間如果鍵入兩個空白鍵, 紙上會出現連續兩個空白。但在輸入 LaTeX 文稿時, 多個空白和一個空白的作用完全相同。換句話說, 碰到多個空白時, LaTeX 會自動將它們刪減成一個空白。另外, 新的一行開頭之處, 不須再留一空白。

由圖 5.1 例子可知, 不管輸入多少空白鍵, 其作用和一個空白鍵相同。輸入時第2行的 by 和 spaces 兩個單字之間雖留有三個空白, 排版之後其間距和正常間距相同。其次, 第1行尾巴的 sentences 和第2行開頭的 are 之間並未留有空白, 但是排版結果兩個單字之間也有正確的間隔, 原因是

The ends of words and sentences are marked by spaces. It doesn't matter how many spaces you type; one is good as 100.
 	
Note also that one or more blank lines denote the end of a paragraph.
	`The ends of words and sentences are marked by spaces. It doesn't matter how many spaces you type; one is good as 100.`
	
	`Note also that one or more blank lines denote the end of a paragraph.`

圖 5.1: 控制空白與段落

sentences 之後換新行。在顯示器上雖然第 1 行末端看不到任何符號, 但電腦檔中該處隱藏一個換行指令。排版時, 碰到此換行指令, LaTeX 就當作一空白處理。

上面例子中兩段文字之間有一空行, 其作用是確認以上段落結束, 以下爲新起一段之開頭。如果不留空行, 我們必須在第一段之末加上 \par 指令。指令 \par 爲英文 paragraph 之縮寫, 意義爲段落。和空白鍵的情形類似, 空一百行和空一行的作用完全一樣, 都是表示要另起新的段落。如果沒有另外設定, 新段落開始的一行, 行首會自動內縮一點。在英文排版中, 這稱爲 indent。

排版之後, 英文單字之間留有適當間隔, 標點符號之後也留有空白。某些標點符號, 如句點或問號, 代表一段話的結束, 讀者的眼睛在此地方應該有小停頓。空白多一點表示停頓的時間長一些。因此, 句點之後的空白應該比逗點之後的大一些, 因爲前者表示整句話的結束, 後者只是一長句子中的小休。爲了使 LaTeX 能正確處理標點符號之後的間隔, 輸入文稿時標點符號之後應留一空白, 或者換新行。

爲了提高可讀性, LaTeX 在英文句點 '.', 問號 '?', 冒號 ':', 及驚歎號 '!' 之後所留的空白, 比單字之間的空白或者逗點之後的空白都要大一些。但是, 有時候英文句點並不代表句子結束。舉一個簡單的例子, 如果你輸入 `Mr. know-all`, 排版之後將如第 1 行所示。

Mr. know-all	`Mr. know-all`
Mr. know-all	`Mr.\ know-all`

Mr. 單字的小點並不是英文句點, 但 LaTeX 誤把它當作句點, 因此 Mr. 與 know-all 之間的空白拉大了一些。若句點之後不特別加多空白, 排版結果

應該是第二行所示。有人會認爲這一點差異沒有那麼了不得。不過, 在專家的眼中這個差別是很重要的。要使句點之後的空白大小正確, 我們應使用 \␣ 指令, 亦即, 反斜線之後加上一空白。因此, 正確的輸入方法是第2行所示: `Mr.\␣know-all`。

英文句點之後若緊接著右圓括號或者英文引號, LaTeX 也會將之解釋成是句子結束而加大空白。但是, 偶而會有例外情況, 如:

Famous coffee beans (Santos, etc.) are expensive.	`Famous coffee beans (Santos, etc.)\ are expensive.`

輸入時, 右圓括號之後加上強制空白指令, 以免空白太大。

5.3 中文稿件輸入原則

輸入中文稿時, 文字之間是否留空白, 以及在何處更換新行, 須特別小心。首先, 輸入時中文字之間不應該留有空白。如果中文字之間插入空白鍵, 排版之後兩字之間距會變大。底下是一個例子:

輸入中文文 稿時, 我們對於空白 及換新行要很小心。	輸入中文文 稿時，我們對於空白及換新行要很小心。

因爲「文 稿」兩個字之間有一空隔, 排版之後兩字之間距並不正確。此外, 上一節曾說明更換新行和留空白的作用相同。本例中「空白」兩字與下一行開頭的「及」之間因爲更換新行, 排版之後字距也加大了。

那麼中文稿應該在那裡更換新行呢? 如前所述, 標點符號之後本來就要留一空白, 以使排版結果美觀。因此, 中文稿也應該在標點之後更換新行。綜合以上所述, 輸入中文稿請謹記下列兩項原則:

- **句子內的中文字之間請勿留下空白**;
- **換新行請盡量在標點符號之後。**

根據這兩項原則, 上面句子中「文稿」兩個字中間不應留空白;「空白」兩個字之後不應該換行; 換行應該在逗點之後。

不過, 如果你的中文輸入已養成習慣, 在固定長度就按 [Enter] 鍵換行, 則解決問題的方法是在執行 cwtex 時加入 `-c` 選項。譬如, 若文稿檔名爲 `test.ctx`, 轉換中文字時應執行下列指令:

```
c:\xtemp>cwtex -c test
```

加入選項之後, 若遇有兩個中文字間插入換行指令之情況, cwtex 會將換行指令刪除。若是使用 WinEdt 軟體, 我們可將以上之設定加入 [F9] 功能鍵內, 以方便使用。設定方法如下, 開啓 WinEdt 視窗上方之:

Option|Menu Setup|&Accessories|cwtex

將 Utility 空格內之 cwtex.exe 改爲 cwtex.exe -c 即可。

有時候, 我們會碰到非常長的句子, 若要等到標點之後再換行, 輸入時不很方便。此時, 我們可以使用註銷指令 %。如前所述, 一行文字中若出現註銷指令, 該指令及其後所有的文字符號都不處理。譬如,

很長的英文單字, 例如 percholoethylene, 可以用註銷指令切成兩行; 中文亦然。

```
很長的英文單字，例如 percholoe%
thylene，可以用註銷指令切成%
兩行；中文亦然。
```

因爲第 1 行之末有 % 指令, 這等於是第 2 行開頭的 thylene 直接緊接在第 1 行尾巴的 percholoe 之後, 因此可排版出正確結果。同理,「切成」兩字之後緊接著加上 % 指令, 因此排版結果不會出現多餘的空白。

5.3.1 標點符號

LaTeX 所定義的標點符號共有下列 16 個:

```
，　．　：　；　？　！　‘　’　（　）　［　］　-　/　*　@
```

在 cwTeX 系統中, 上列16個標點符號是中英文共用的。但中文另外有 10 個標點符號是英文沒有的:

```
、　。「　」『　』〈　〉《　》
```

大部份的中文輸入法都有輸入中文標點符號的方法。萬一沒有, 我們可以用內碼輸入。各標點符號之內碼如下:

```
、A142　「A175　」A176　〈A16D　〉A16E
。A143　『A179　』A17A　《A16D　》A16E
```

另外, 中文數字「〇」之內碼爲 A2AF。(在倉頡輸入法下, 鍵入全型的 0 亦可得到結果。) 若使用 WinEdt 文字輸入軟體, 請依表 3.1 所列, 以功能鍵輸入標點符號較爲方便。

目前各式各樣的中文輸入方法中, 有不少是採所謂的「全型輸入」方式輸入標點符號。譬如, 中文句點在顯示器上占用相當於兩個英文字母的位置。這是所謂的「全型」。以上所列出的中文標點符號, 全部都必須以全型方式輸入。相對的, 中英文共用之逗點或者問號, 可能採全型輸入, 也可能是採半型輸入。所謂「半型」是指直接英數狀態下的英文逗號, 因此它所占用的空間和英文字母或阿拉伯數字相同。

使用全型輸入中文標點符號時, 雖然顯示器上看起來該標點之後有一空白, 其實不然。你必須再鍵入一空白才算數。輸入全型標點符號時若未留空白, 排版結果中文字與標點符號之間距可能仍然是正確的。原因是 cwtex 中文轉換程式會辨識常見的錯誤, 並作適當的修正。雖然 cwTEX 有此功能, 使用者最好還是養成正確的輸入習慣。

LaTeX 會在英文句點, 問號, 冒號及驚歎號之後多加一點空白, 以提高可讀性。同樣的, cwTEX 也在中文句點之後多加上一點空白。但是, 有些標點符號之前或之後是不須留空白的。譬如:

小明說:「公主大叫『救命啊!』。」　小明說：「公主大叫『救命啊！』。」

本例中, 驚嘆號之後的三個標點符號之間, 不應加上任何空白。如果插入空白, 結果反而不佳。

5.3.2 英文引號與破折號

英文標點符號用於中文稿中唯一要注意的是引號。如果你要排版 ‘測試’ 或 ‘test’, 輸入之指令分別是: ‘ 測試 ’ 與 ‘ test ’。請特別注意, 左單引號和右單引號是不同的。個人電腦中, 左單引號字鍵通常置於鍵盤左上角; 右單引號鍵則置於鍵盤的右下角。

如果要排版雙引號, 譬如, “測試”, 應如何輸入呢? 一般鍵盤的右下方有一個雙引號的字鍵, 因此很多人會直接輸入 "測試"。但是, 排版結果卻變成: ”測試”。正確的輸入是: ‘‘ 測試 ’’。亦即, 左雙引號是連續輸入兩個左單引號; 右雙引號則是連續輸入兩個右單引號。左右引號未分辨清楚, 是初學者常犯的錯誤, 應特別小心。

表 5.4: 破折號與引號

名稱	指令	例子
hyphen (-)	`-`	motor-car, mother-in-law
en-dash (–)	`--`	頁 12–35, 1981–1990 年
em-dash (—)	`---`	I saw them — three of them
減號 (−)	`$-$`	$5 - 2 = 3$

如果你必須同時使用單引號與雙引號呢? 譬如, 你碰到下面的句子:

“‘Five’ or ‘Hive?’” she asked.

若輸入 ‘‘‘Five’ 我們不曉得到底是前兩個單引號構成雙引號, 或者是後面兩個。解決此一困擾的方法是使用 \, 指令, 在兩個引號之間加入一小小的空白。因此, 正確的輸入方法如下:

```
‘‘\,‘Five’ or ‘Hive?’\,’’ she asked.
```

英文使用的破折號由短到長共有三種。最短的稱爲 hyphen 是用於連接英文單字, 如 mother-in-law。次長的破折號稱爲 en-dash, 通常用於連接數目字。譬如, 頁 34–56, 或者, 1945–1950 年。要產生 en-dash, 你必須接連著輸入兩個 hyphen: `--`。最長的破折號是由三個 hyphen 組成, 其用法相當接近中文之破折號: 台電公司即將限電 — 如果天氣不轉涼的話。

以上的分類適用於 Knuth 所設計的 Computer Modern 字體。如果我們使用 PostScript Type 1 字體排版, 其區分並無如此細緻。這些字體通常並未區分 hyphen 與 en-dash。本書正文使用的是 Minion 字體, 排版頁碼範圍時若直接使用 hyphen 指令, 結果爲: 頁 12-35; 若使用 en-dash, 結果爲: 頁 12–35。

還有一個和破折號樣子很像的符號是減號, 不同的是其筆劃較粗。要排版減號, 應該輸入: `$-$`。前後加上兩個錢號 `$` 的作用是使 LaTeX 進入數學模式。譬如, 輸入 `-100`, 排版結果爲: −100, 若不進入數學模式, 排版結果爲 -100。以上所述標點符號的用法歸納於表 5.4; 數學式排版將於第 9 章說明。

有些人以連續三小點 (甚或六小點) 代表一段省略的話。在 LaTeX 中, 你可以用 `\dots` 指令產生三小點。例如指令 `\dots\dots` 產生: ……。最後, 謹列舉輸入文稿之原則, 以供參考。

- 逗號或句號不宜與前接文字分開;因此,「鍵入雙引號 ,」爲錯誤, 因爲逗號之前不應留空白。正確的排版是:「鍵入雙引號,」。相反的, 逗號或句號之後應留空白。因此,「譬如,下面的句子 ...」爲錯誤, 正確應該是「譬如, 下面的句子 ...」。
- 同理, 段落前端之標點符號也不宜和後接文字分開。譬如,「昭和 20 年 (1945 年) 日本投降」比「昭和 20 年 (1945 年) 日本投降」爲佳, 阿拉伯數字 "1945" 之前應緊接圓括號,「年」之後也不應留空白。另外,「昭和 20 年(1945 年)日本投降」也不佳, 左圓括號前面與右圓括號後面應留出空白。
- 阿拉伯數字前後有無空白間隔皆可, 但以不留空白爲佳。譬如,「西元 1945 -1950 年間的台灣」, 輸入時中文與阿拉伯數字間並未留空白; 但 cwtex 會自動在阿拉伯數字與中文字之間加入一點小空白。若自行留空白, 結果變成「西元 1945-1950 年間的台灣」。
- 英文字夾於中文句子中, 前後則以留出空白爲宜。譬如,「最短的稱爲 hyphen 是用於」比「最短的稱爲hyphen是用於」易讀。

以上之細節看似瑣碎, 但對於排版結果大有影響。欲排版高品質文稿, 請多用心觀察、欣賞好的排版, 學習其風格。專業排版對於標點符號之規範, 請見 Bringhurst (1996)。

除了以上所述之外, 有關於中文數學文稿之輸入原則, 請見9.2.1 節 (頁 153)。

6 版面設計與選用字體

開始著手排版一本書或一篇文稿之前, 應先想好版面之整體設計, 譬如, 版面尺寸、正文字體、章節標題格式、註解的字級 (font size) 等等。使用 LaTeX 的好處是我們可以選用現成的版面格式, 不須花太多時間在版面設計上。舉例來說, 如果正文使用 12pt 字體, 則文稿中註解之字體會自動選用 10pt, 行距也會小一些。又如文稿中若區分章、節, 其標題會自動選用較大的字體, 且前後會空出適當空白。

LaTeX 所設計的版面已考慮了專業排版的規範。除非你有專業排版的知識, 否則使用現成的格式排版, 其結果通常優於自行設計者。使用現成的格式, 一方面節省時間, 另一方面免於犯一些常見的排版錯誤, 這是其優點。但是, 也有人認爲 LaTeX 所排版之書籍其版面格式都很接近, 不易顯出個人風格。要變更內定之版面並不難; 特別是在新版 LaTeX 中, 使用者有相當大的調整空間。

本章首先介紹以設計文稿整體版面的基本概念, 下一章說明文稿結構與章節安排之指令; 第 8 章則說明段落編排的指令。

6.1 版面設計

書籍或文稿版面設計的第一步是決定版面大小。一般的簡短文稿通常列印於 A4 或 letter size 紙張上。在 LaTeX 中, 一旦選定紙張尺寸, 版面大小即有內定值。當然, 我們也可以自行設定版面的尺寸。由第3章的簡單例子可知, 任何 LaTeX 文稿至少含有下列三道指令:

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

第一行指令 `\documentclass` 稱爲「文件類別指令」, 這通常是文稿的第一

表 6.1: 紙張尺寸

a4paper	21 × 29.7 公分	letterpaper	8.5 × 11 吋
a5paper	14.8 × 21 公分	legalpaper	8.5 × 14 吋
b5paper	17.6 × 25 公分	executivepaper	7.25 × 10.5 吋

道指令。本例中, 文件類別指令之後接著 article 選項, 設定以短文格式排版。若是排版書籍, 我們可選用 book 選項。

本例中另外兩行指令 \begin{document} 與 \end{document} 是成對出現的, 兩道指令之間即爲排版文稿內容。後一道指令是文稿最後一行指令; 其後的任何文字或指令 LaTeX 皆不處理。此種成對出現的指令所涵蓋的範圍稱爲「指令環境」(environment)。本例之指令即稱爲 document 指令環境。LaTeX 提供許多指令環境, 譬如排版表格可使用 tabular 指令環境。

6.1.1 設定版面大小

動手排版之前應先設計好整篇文稿的版面。文件類別指令的目的即是在選定最基本的版面設計。在文件類別指令中加入選項, 即可選用特定的版面尺寸或字級。對 LaTeX 而言, 文稿版面之組成元素如圖 6.1 所示。台灣目前同時通行美國與歐洲的紙張規格。美國常用的紙張尺寸是 letter size 與 legal size; 歐洲或日本通用的尺寸則是 A4, A5, B5 等。如果不作選擇, LaTeX 自動選用 letter size, 其大小爲 8.5 × 11 英吋。

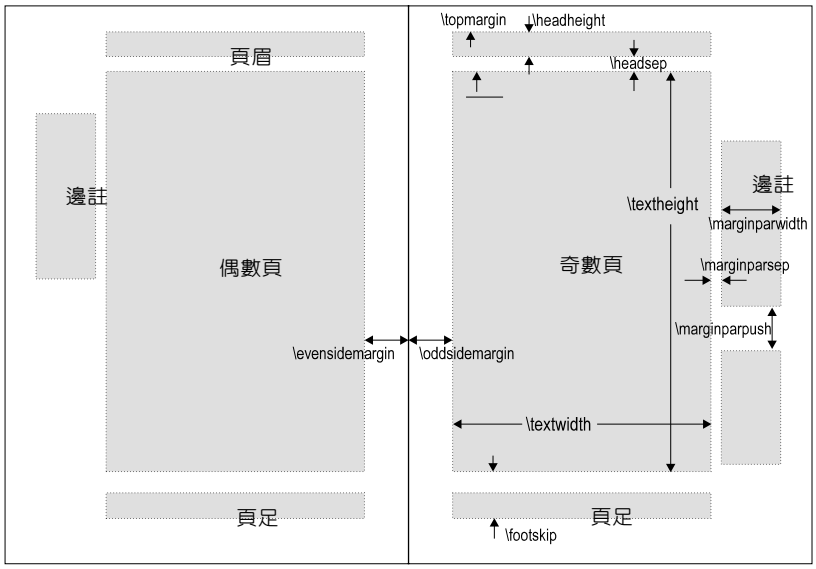
選擇紙張規格很簡單, 只須在 \documentclass 指令中加入選項即可。紙張選項如表 6.1 所示。若選用 A4 紙張排版短文 (article), 字體選用 12pt, 指令爲:

```
\documentclass[12pt,a4paper]{article}
```

\documentclass 指令之後的方括號內爲選項, 若不加選項, LaTeX 將使用內設值。因此, 若指令爲:

```
\documentclass{article}
```

LaTeX 將選用 letter size 紙張, 並以 10pt 字體排版。大括號中之選項並無預設值, 因此一定要加上。



`\textwidth`	正文方格寬度。	`\marginparsep`	正文方格與邊註的距離。
`\textheight`	正文方格高度。	`\marginparwidth`	邊註的寬度。
`\linewidth`	正文一行之寬度。	`\marginparpush`	連續兩個邊註之間的最小垂直距離。
`\headheight`	頁眉高度。		
`\topmargin`	頁眉上方之空白。	`\columnseprule`	正文有兩欄或以上時，兩欄間分隔直線的寬度。
`\columnsep`	正文有兩欄或以上時，兩欄間的距離。	`\columnwidth`	正文有兩欄或以上時，每一欄的寬度。

`\oddsidemargin` 取 `twoside` 選項時，奇數頁左邊的空白。若未取 `twoside` 選項，此項設定值即爲各頁左邊的空白。

`\evensidemargin` 取 `twoside` 選項時，偶數頁右邊的空白。

圖 6.1: LaTeX 之版面與控制指令

如果要使用特別的紙張尺寸, 須自行設定。文稿的版面設計必須在正文之前決定, 因此紙張尺寸之指令須置於 \begin{document} 之前, 此一區域稱爲「全文設定區」(preamble)。譬如, 若選用的版面 (含文字與四周之空白) 高 26 公分, 寬 20 公分, 須在全文設定區加入下列兩行指令:

```
\paperheight=26cm
\paperwidth=20cm
```

紙張大小選定之後, 版面正文方格及其他部分之規格都會隨之調整。

如圖 6.1 所示, 文稿的每一頁除了正文文字之外, 還有頁足詞、頁眉詞、邊註等。在 LaTeX 中, 排版於正文文字方格上方之資訊稱爲 header, 傳統中文排版稱之爲「天頭」, 本書將稱之爲「頁眉詞」; 相對而言, 正文文字方格下方之資訊稱爲 footer, 傳統排版稱之爲「地脚」, 本書則稱之爲「頁足詞」, 正文方格的大小、正文與頁眉的距離、邊註的寬度等, 都有內設值, 但也都可以重新設定。正文方格是指 \textheight 與 \textwidth 所形成之長方形, 如果要改變其大小, 須在全文設定區加入下列兩行指令:

```
\textheight=20cm
\textwidth=16cm
```

以上指令重新設定正文方格之寬度爲 16 公分, 高度爲 20 公分。其他各項距離的設定也以類似方式爲之。

前面的 \paperheight 指令雖然選擇紙張大小, 排版結果可能還是列印於 A4 紙張上。列印時, 列印程式將選取白紙左上方的某特定點爲開始列印的參考點。參考點通常距離紙張上沿及左沿各一公分。因爲正文方格的寬度及高度是自行設定, 列印之後正文方格四周的空白可能並不對稱。此時, 我們可以利用 \voffset 與 \hoffset 指令調整之。例如, 要將正文方格上移 0.5 公分, 右移 1 公分, 可以在全文設定區加入下列指令:

```
\voffset=-0.5cm
\hoffset=1cm
```

6.1.2 geometry 巨集套件

上一小節所介紹的指令可用以控制版面大小與位置, 但使用上仍嫌複雜。本小節將介紹巨集套件 geometry, 使用上較方便, 作者爲 Hideo Umeki。

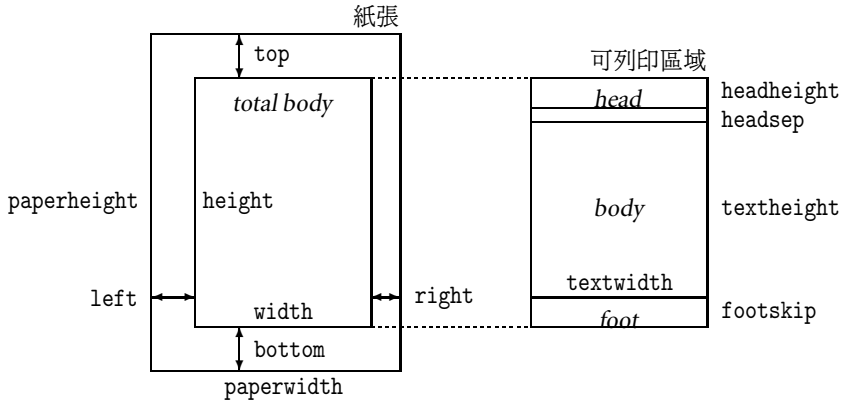


圖 6.2: geometry 巨集套件設定指令圖示

說明: 此圖取自 geometry 巨集套件說明檔。

介紹指令之前, 先簡單說明幾個概念。從紙面列印的角度來看, 一張白紙上包含可列印區域 (printable area) 及四周邊緣 (margins); 可列印區域 (又稱爲 total-body) 進一步區分爲正文方塊 (text area), 頁眉, 頁足, 與邊註 (marginal notes); 而四周邊緣可分上下左右四部分。

如圖 6.2 所示, 設定正文方塊大小之指令是 `textwidth` 與 `textheight`。請注意, 在 geometry 巨集指令內, 設定指令之前端並無反斜線。以上兩道設定指令對應 LaTeX 之 `\textwidth` 與 `\textheight`。相對於 `textwidth` 指令, `width` 指令爲 `textwidth` 加上邊註 (marginal notes) 之寬度。

根據圖 6.2, 若要以 geometry 巨集套件指令設定正文方格區域之尺寸, 僅須在全文設定區加入下列一行指令:

```
\usepackage[textwidth=5in,textheight=8in]{geometry}
```

如果要進一步設定左邊緣爲 1in, 上邊緣爲 1.5in, 設定指令爲:

```
\usepackage[textwidth=5in,textheight=8in,
   left=1in,top=1.5in,nohead]{geometry}
```

請注意, geometry 巨集套件之指令可拆爲兩行以上, 各行末端不須加入註銷指令 %。本例中還加入 `nohead` 設定, 這等於是把 `headheight` 與 `headsep` 都設爲零。

由圖 6.2 可知, 以上之設定並不完整, 我們至少還須加 paperheight 與 paperwidth。不過, 若使用者不設定, geometry 會取用內定值。例如, 若紙張種類不設定, 巨集指令即取用 letterpaper, 並進一步決定 paperheight 與 paperwidth 之值。如果是使用 A4 尺寸紙張, 應加入 a4paper 設定項; 或者也可以直接加入於 \documentclass 選項內。

除了圖 6.2 所介紹的設定指令外, 還有許多其他指令可供使用, 以下僅介紹其中較常用者。欲了解所有細節, 請參考說明檔。假設選用 A4 紙張, 可列印區域 (正文方塊加上頁眉與頁足) 之高度設爲 9in, 左右邊緣又留 1in 空白, 指令爲:

```
\usepackage[a4paper,height=9in,hmargin={1in,1in}]{geometry}
```

其中, hmargin={1in,1in} 是用於設定左右邊緣之空白。以上之設定方法是直接將各設定項加入於 \usepackage 指令內, 另一種方法是將所有之設定另外輸入於 \geometry 指令內, 如下例所示:

```
\usepackage{geometry}
\geometry{a4paper,height=9in,hmargin{1in,1in}}
```

指令效果完全相同。本例中, 長度直接設定爲英吋單位, 但我們也可以下列方式設定:

```
\usepackage[hmargin{0.1\textwidth,0.1\textwidth}]{geometry}
```

以下列出一些較常用的設定指令:

nohead	等於 headheight=0pt,headsep=0pt。
nofoot	等於 footskip=0pt。
hscale	可列印區域之寬度與紙張寬度之比率, 例如 hscale=0.8。
marginparwidth	邊註 (marginal notes) 之寬度。
marginparsep	正文方塊與邊註之間距。
width	等於 textwidth + marginparsep + marginparwidth。
twoside	奇數頁 (右頁) 之列印區域會稍靠左, 雙數頁 (左頁) 之列印區域則稍靠右; 參見下一指令。

表 6.2: 常用的長度單位

cm : 公分	pc : pica = 12 點
mm : 公厘 = 0.1 公分	bp : big point = 1/72 吋
in : 吋 = 2.54 公分	em : 約爲大寫字母 M 之寬度
pt : 點 = 1/72.27 吋	ex : 約爲小寫字母 x 之高度

`twosideshift` 以 `twoside` 方式列印時, 可用此一指令設定奇數頁左移 (與雙數頁右移) 之距離。若不自行設定, 內設值爲 20pt。譬如, 若奇數頁要左移 0.5 公分, 指令爲: `twosideshift=-.5cm`。

`hoffset` 同 TEX 之 `\hoffset` 指令。

`voffset` 同 TEX 之 `\voffset` 指令。

本書所使用之指令如下:

```
\geometry{paperheight=23.5cm,paperwidth=15.25cm,
   textwidth=10.75cm,textheight=18.5cm,nohead,left=2.25cm,
   twosideshift=-.75cm,top=2cm,footskip=1.75cm,
   marginparwidth=2cm}
```

除了設定版面尺寸之外, 本書頁眉空白, 故加入 `nohead` 設定。另外, `twosideshift` 設定是用以調整單雙頁之版面位置。我們還使用 `\hoffset` 與 `\voffset` 指令以控制列印位置, 但這兩項指令是單獨輸入。

6.1.3 長度與間距

版面設計的第一步是設定版面的寬度與高度、訂定行距等等。所有這些工作都須使用長度的單位。此外, 字體大小也是長度的一種。專業字體設計家在設計字體時, 都設想實際排版時最美觀的大小, 稱之爲設計尺寸 (design size)。LETEX 的英文及數學符號字體, 大部分的設計尺寸都是 10 點 (printing points, 簡稱爲 pt)。1 點等於 1/72.27 吋, 或大約 0.0351 公分。簡單來說, 英文字體 10 點之大小大約是各字母中的最高點 (如字母 h 之頂端) 到最低點 (如字母 y 之底) 的距離。爲了與英文字的大小配合, cwTEX 中文字體的設計尺寸也都是10點。專業排版常以 printing point 作爲長度的單位, 但我們也可以使用一般的長度單位。表 6.2 列出常用的長度單位。

上一小節的例子中, 設定長度的方法是在指令之後直接以等號定義。事實上, LaTeX 設定長度的標準指令是 \setlength。要把版面寬度定爲20公分, 指令如下:

```
\setlength{\textwidth}{20cm}
```

若以上的指令稍嫌麻煩, 我們也可以使用簡化的指令格式, 上面的例子即使用下列的簡化指令:

```
\textwidth=20cm
\textwidth20cm
```

第2行指令連等號都省略掉, 更爲簡化。

以上所設定的長度稱爲固定長度 (fixed length), 有時候我們必須設定所謂的彈性長度 (rubber length)。 舉例來說, 排版時我們須設定版面的高度與行距。版面高度通常是固定長度, 如果行距也是固定, 可能出現的情形是版面最底下無法剛好把最後一行擺進去。如果行距設爲彈性長度, 萬一最後一行擺不進去時, LaTeX 可以把行距都縮小一些, 使最後一行可以排入。底下是一個例子:

```
\setlength{\textwidth}{20cm}
\setlength{\baselineskip}{15pt plus0.4pt minus0.2pt}
```

第2行指令設定行距爲一彈性長度, \baselineskip 爲行距之指令, 15pt 爲行距之正常大小, plus0.4pt minus0.2pt 爲彈性大小。因此, 行距最大可以是15.4pt, 最小是14.8pt。

\setlength 指令用以直接設定長度, 但某些變數已有內定值, 如果只是要加長或縮短, 可以使用 \addtolength 指令。譬如,

```
\addtolength{\baselineskip}{10pt}
```

可將標準行距加大10pt。若 10pt 改爲 -5pt, 標準行距將減小5pt。

以上介紹的是長度單位及設定方法。有時候我們須在文稿內產生水平或垂直空白。產生垂直空白的指令爲 \vspace, 相對而言, 橫向空白之指令爲 \hspace。譬如, 要在版面中空出垂直3公分, 指令爲 \vspace{3cm}; 同理, 要在某行文字內加入水平2公分的空白, 指令爲 \hspace{2cm}。但是,

\vspace 指令的位置若恰好是在一頁版面的最上端, 指令將失效。但如果是使用 \vspace* 指令, 不管是下於何處, 指令都有效。\hspace 也有類似的情況; 指令若恰位於一行之最左端, 指令也是無效。若使用 \hspace* 指令形式, 則下於任何地方都有效。

以上控制間距之指令須在大括號內加上間距之長度, 這是所謂的強制變數 (mandatory argument)。相對而言, 有一些指令除了強制變數之外, 還可加入選項變數 (optional argument)。強制變數是加於大括號內, 選項變數則加於中括號內。顧名思義, 選項變數可加可不加。譬如, LaTeX 劃直線的指令爲:

\rule[*lift*]{*width*}{*height*}

以上指令中, *width* 設定直線長度, *height* 設定線的粗細, 這兩個都是強制變數。相對而言, *lift* 屬於選項變數, 設定直線往上抬高的距離。因此, 要畫出長度 1 公分, 粗細 0.1 公分的直線: , 指令爲 \rule{1cm}{0.1cm}。如果要將此直線抬高 0.1 公分, 可以加上選項變數: \rule[1mm]{1cm}{0.1cm}, 畫出之直線爲: 。

6.2 字型規格

介紹字體指令之前, 我們先簡單說明字體設計的概念。所謂字體 (font), 簡單來說是指文字的某一種書寫方法。在排版中, 字體選擇占著舉足輕重的地位。Knuth 教授當初發展 TeX 排版系統時, 同時又發展一套設計字體的軟體 METAFONT。原始 TeX 與 LaTeX 系統所使用之英文字體全部都是利用 METAFONT 軟體設計出來的。這些字體之字型檔名大部分是以 cm 開頭, 代表 Computer Modern 字體。譬如, 羅馬字體 10pt 之字型檔就稱爲 cmr10。

傳統的鉛字排版中, 字型直接刻在鉛塊上。電腦排版發展之後, 每一種字體內各單字或字母之形狀則儲存爲電腦檔案。過去一、二十年中, 字體設計之技術有相當大的進步。其中, 美國 Adobe 公司所發展之 Type 1 字體 (又稱爲 PostScript 字體) 是最重要者。Type 1 字體是所謂的描邊字體 (outline font), 每一個單字或字母之形狀都以數學式描繪出來, 列印時再填爲實心字。描邊字體的好處是可以隨意放大, 而不會嚴重失眞。自 1980 年代中期以來, Type 1 字體日益普及, 幾乎變成是電腦排版的字體標準。因爲如此,

新版 LaTeX 一個重要的發展方向就是讓排版者可以取用 Type 1 字體。我們將在第 13 章介紹 Type 1 字體的使用。

中文電腦字型之技術近幾年來也有很大的進步, 特別是在 Windows 系統開始推廣之後。Windows 主要使用另外一種字型規格, 稱爲 True Type, 但也可以使用 Type 1 字型。中文電腦字型的設計面臨許多困難, 最主要的是中文字數實在太多。在英文中, 一套字體通常不超過 256 個字母, 其中包括大小寫字母、數字、標點符號等等。但是, 一套中文明體字可能超過一萬字。以 Big-5 內碼爲例, 常用字有 5,401 字, 次常用字則有 7,652 字。因此, 創造一套中文字體所耗人力物力相當驚人。這是爲什麼英文字體的選擇很多, 而中文字體的選擇卻很有限。

在 LaTeX 中, 我們可以選用不同的英數字體, 也可以將字體放大或縮小。不過, LaTeX 之字體指令只能處理英文與數字字體, 對於中文字體並無作用。中文字體的變換與放大縮小必須靠 cwTeX 的中文字體指令。以下兩節進一步說明之。

6.3 選用英數字體

Knuth 一共造出七十多種字體。除了英文與數字之外, 還有不少特殊符號是爲了排版數學用的。排版時, 如果你不指定字體, LaTeX 就選用 10 點字級 (font size) 之羅馬字體編排, cwTeX 內定的中文字體則爲10點明體字。

在舊版的 LaTeX 系統中, 選用字體的指令彈性較小。1990 年代初期, 兩位德國專家 Frank Mittelbach 與 Rainer Schöpf 設計了一套「新式字體選用法」(new font selection scheme, 簡稱爲 NFSS)。此套方法出版之後, 頗受歡迎, 現在已經變成新版 LaTeX 系統之一部分。本節以下所介紹的是新版之字體指令。**請特別注意**, LaTeX 之字體指令只能改變英文及數字字體, 中文字體必須以 cwTeX 指令才能變更之。初學者很容易忘記這一點, 請特別注意。

英文字體式樣甚多, 但依 NFSS 分類, 字體可以三種特徵區分:

- 字形 (font shape),
- 字體序列 (font series),
- 字體族 (font family)。

字形又進一步區分爲: 直形 (upright), 意大利斜體 (*italic*), 數學斜體 (*slant*), 與 SMALL CAPITAL 等四類。字體序列則包括中體序列 (medium series) 與

表 6.3: 英文字體指令

字體分類	標準字體指令	宣告字體指令	簡化宣告指令
字形 *(shape)*			
Upright	`\textup{`*text*`}`	`{\upshape` *text*`}`	
Italic	`\textit{`*text*`}`	`{\itshape` *text*`}`	`{\it` *text*`}`
Slant	`\textsl{`*text*`}`	`{\slshape` *text*`}`	`{\sl` *text*`}`
SMALL CAPS	`\textsc{`*text*`}`	`{\scshape` *text*`}`	`{\sc` *text*`}`
字體序列 *(series)*			
Medium	`\textmd{`*text*`}`	`{\mdseries` *text*`}`	
Boldface	`\textbf{`*text*`}`	`{\bfseries` *text*`}`	`{\bf` *text*`}`
字體族 *(family)*			
Roman	`\textrm{`*text*`}`	`{\rmfamily` *text*`}`	`{\rm` *text*`}`
sans serif	`\textsf{`*text*`}`	`{\sffamily` *text*`}`	`{\sf` *text*`}`
Typewriter	`\texttt{`*text*`}`	`{\ttfamily` *text*`}`	`{\tt` *text*`}`

粗體序列 (**bold series**)。字體族則包括以下三類: 羅馬字族 (Roman family), sans serif (無裝飾邊) 與打字機 (typewriter) 字族。以上三種分類中, 字體族的範圍最大。譬如, 在羅馬字族之下可區分爲中體序列與粗體序列, 而中體序列之下可進一步區分爲直形、意大利斜體形等; 請見表 6.3。

文稿中任何地方都可以下指令改變英數字體。字體指令有兩種格式, 第一種指令格式與一般的指令類似, 亦即將欲改變字體之段落置於大括號內, 我們將稱之爲「標準字體指令」。譬如, 英文 text 要改變爲粗體字 **text**, 指令爲 `\textbf{text}`; 要變更爲意大利斜體 *text*, 指令爲 `\textit{text}`。如果是要改變爲數學斜體字 *text*, 指令爲 `\textsl{text}`。請注意, 數學斜體與意大利斜體並不相同, 前者主要用於數學式子內。

除了使用標準字體指令之外, 我們也可利用「宣告字體指令」(declaration command) 變更英文字體。事實上, 這是舊版 LaTeX 改變字體的標準方法。宣告字體指令提供三種格式, 以下以粗黑體爲例說明之。

- 第一種格式是在文稿中任何地方直接下 `\bfseries` 指令, 從該點開始所有的英數文字都變成粗體字。
- 第二種下指令的方式與第一種指令相同, 但指令 `\bfseries` 簡化爲 `\bf`。大部分宣告字體指令都可以使用簡化指令。譬如, `\itshape` 指令可以簡化爲 `\it`, `\rmfamily` 可簡化爲 `\rm`。

- 第三種方法是以指令環境之方法改變字體, 譬如:

  `\begin{bfseries}` *text* `\end{bfseries}`

  指令環境內所有英數文字都變成粗體字。

以上三種指令格式中, 第二種格式最爲簡單。

上面所介紹的標準字體指令格式是以大括號界定字體變更的範圍, 宣告字體指令也可以大括號界定範圍, 但是, 大括號必須將字體指令本身也涵蓋在內。例如, 以下指令將大括號內三個英文字改用數學斜體編排:

```
{\sl switch to slant}
```

右大括號之後又回復原先的英數字體。底下是一個較複雜的例子, 但字體指令之運作原理不難理解。

Started with italic, switch to Roman, *switch to slant*, back to Roman, make one word **bold**, simulate `typewriter`.

```
\it Started with italic,
\rm switch to Roman, {\sl switch
to slant}, back to Roman,
make one word \textbf{bold},
simulate {\tt typewriter}.
```

本例是以 `\it` 指令選用斜體字形; 但我們也可以用 `\em` 替代 `\it`, 排版效果完全相同。

本例子內之字體指令主要是變更字體序列與字形。如果硬碟內安裝了他種 PostScript 字型檔, 如 Times 或 Garamond, 我們也可以在文稿內選用另一字體族。安裝 PostScript 字型檔的方法, 請見第 13 章; 選用字體之指令, 請見 13.3.2 節 (頁 269)。

一般的 LaTeX 宣告指令, 包括字體指令在內, 如果置於大括號內, 指令之效力僅限於大括號內。LaTeX 提供許多指令環境, 例如排版表格的 `tabular` 指令環境。宣告指令如果置於特定指令環境內, 指令環境結束之後, 宣告指令之效力也消失。如果希望指令之效力僅限於某一範圍, 最簡單的方法是以左右大括號界定其範圍。

在英文打字稿中常以加底線的方式強調文句。不過, 正式排版中加底線的作法較少見, 通常是以改變字體的方式爲之。在 LaTeX 中, 加上底線的指令是 `\underline`。例如:

表 6.4: 字體級數對照表

字體級數	放大倍數	字體點數
0	$1.2^0 = 1.0$	10
h	$1.2^{0.5} \cong 1.095$	10.95
1	$1.2^1 = 1.2$	12
2	$1.2^2 = 1.44$	14.4
3	$1.2^3 = 1.728$	17.28
4	$1.2^4 \cong 2.074$	20.74
5	$1.2^5 \cong 2.488$	24.88

說明: 字體級數 h (half), 代表半級。因此, 對應之放大倍數爲 $1.2^{0.5} \cong 1.095$。

此句話加上底線。 `\underline{此句話加上底線。}`

底線指令可用於中文, 亦可使用於英文。

6.3.1 英數字體相對大小指令

除了更換字體之外, 我們也可以改變字級。前面已說明, 大部分字體之設計尺寸爲 10 點。不過, 我們可以將 10 點之字體放大成 14 點, 或者縮小爲 8 點。所謂 14 點的字體, 其橫寬約爲 10 點字體的 1.4 倍; 相對的, 8 點字體的橫寬爲 0.8 倍。依原先設計, 大部分字體都具有下列的級數 (由小至大): 5, 6, 7, 8, 9, 10, 10.95, 12, 14.4, 17.28, 20.74 與 24.88。

你或許會覺得奇怪, 爲什麼字體點數爲 10, 10.95, 12, 14.4, . . . , 而卻沒有 13 點或 16 點? TEX 字體在放大或縮小時, 其倍數是以 1.2 的次方爲單位, 而次方數爲 0, 0.5, 1, 2, . . . 。若放大次方數爲 1, 字體點數爲 $10 \times 1.2^1 = 12$。因此, 12 點之字體也可以說是放大 1 級之字體。若放大次方數爲 2, 則字體點數爲 14.4, 簡稱爲 14 點。因此, 當我們說 14 點 (或 14 級) 字體時, 實際上它是 14.4 點之字體。同樣的, 若放大次方數爲 4, 則字體點數爲 $10 \times 1.2^4 \cong 20.74$, 簡稱爲 20 點字體。因此, 20.74 點之字體也可以說是放大 4 級。表 6.4 列出字體級數對照表。

選擇英文字級有兩種方法, 一種是直接選定字級; 另一種則是以相對大小指令選擇字級。我們首先介紹相對大小之指令。字級相對放大之指令, 依序爲 `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`:

normal, large Large LARGE huge Huge

其中, \Large 字級比 \large 大一些; \LARGE 又比 \Large 大一些; 餘此類推。相反的, 字級相對縮小指令依序爲: \small, \footnotesize, \scriptsize, \tiny。其中, \tiny 所產生的字體最小。

normal, small footnotesize scriptsize tiny

相對大小指令所產生的字體, 其實際尺寸上並不是固定的, 而是相對於正文之字級而定。正文字體之字級稱爲 \normalsize。如果我們選定正文字級爲10點, \normalsize 意指10點之羅馬字體。此時, \large 指令將選用10.95點之字體。但是, 若正文字級爲12點, 則 \large 指令將選用14.4點之字體。正文字體之選定由 \documentclass 指令設定, 例如以下指令選用12點字體與短文格式:

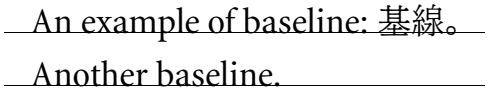
```
\documentclass[12pt]{article}
```

如果文稿某處選用了較大字體, 則 \normalsize 指令可以變回內定的正文字級。

6.3.2 選擇字級與行距

選用較大字體時, 須同時選用較大的行距, 版面才會美觀。另外, 中文字之筆劃較英文字複雜。因此, 英文稿之行距可以小一些, 但中文稿之行距必須加大。如果中文稿中夾有一段英文, 其行距必須仔細調整才會美觀。

「行距」指的是本行到下一行的距離。但是, 從本行的那一點到下一行的那一點呢? 版面的每行文字都有一條所謂的「基線」(baseline), 行距即指兩條基線的距離。下例中的兩條橫線即爲基線, 我們設定之行距爲22點, 也就是這兩條細橫線的垂直距離。



由此例子亦可看出來, 英文小寫字母恰坐落在基線上, 而中文字的底部則比基線還要低一些。

行距可以由 \baselineskip 定義之, 但每一種字級各有內定之行距。當我們使用相對放大指令時, 行距也隨之而變更。因此, 在英文稿中, 我們幾乎不須擔心行距調整的問題。但是, 有些時候我們需要改變行距。譬如, 文章初稿之行距通常會大一些以利修改。又如, 中文字之筆劃複雜, 因此中文稿之行距通常也大一些, 以利閱讀。

欲更改全文之行距, 最簡單的方法是在文章一開頭即加以設定。譬如, 要將全文之行距放大爲內定值之 1.15 倍, 可使用以下的指令:

```
\renewcommand{\baselinestretch}{1.15}
```

此一指令通常是置於全文設定區, 也就是在 \begin{document} 指令之前。請注意, 此一指令將把各式大小字體之內定行距通通放大爲原來的 1.15 倍。因此, 若文稿中有註解, 其行距也會加大。如果希望將各種大小字體之行距一律改爲某固定值, 應使用下列指令:

```
\setlength{\baselineskip}{15pt}
```

以上指令將全文行距一律變更爲 15pt。

上述兩項指令也可以出現在正文中, 但是其發生效果的時刻不同。在同一段落中若變更 \baselineskip 數次, 只有最後一次之變更值有效。而且, 每當字級改變時, 行距即變回內定值。相反的, 若 \baselinestretch 指令是下於行文當中, 則只有在下一次字級變更時, 它才發生效果。如果希望它即時產生效果, 我們可以在指令之後加上 \small\normalsize, 其功能是把字體縮小一級之後, 又立刻回復原來大小。因爲字級已變更, 行距指令即可發揮作用。

另外一種改變字級與行距的方法是使用 NFSS 之 \fontsize 指令。假設要選用 14.4 點字體, 同時行距要變成 20 點, 指令爲:

```
\fontsize{14.4}{20pt plus.5pt minus.4pt}\selectfont
```

第一個大括號中之 14.4 是選用 14.4 點字體; 第二個大括號中之數字選定行距爲 20 點, 伸縮彈性爲正 0.5 點與負 0.4 點。伸縮彈性部分之指令可有可無, 但最好是有, 因爲 LaTeX 較容易編排版面。不過, 行距之伸縮彈性也不宜太大, 否則每一頁版面之行距可能大小不一, 版面看起來的感覺並不好。

表 6.5: 選用字體與行距之巨集指令: mymacro.tex

```
\newcount\fs
\def\sz#1#2{\fs=#1#2
 \ifnum\fs=10\fontsize{10}{12.5pt plus.3pt minus .2pt}\selectfont
 \else\ifnum\fs=11\fontsize{10.95}{16pt plus.3pt minus.2pt}\selectfont
 \else\ifnum\fs=12\fontsize{12}{18pt plus.4pt minus .3pt}\selectfont
 \else\ifnum\fs=14\fontsize{14.4}{20pt plus.5pt minus .4pt}\selectfont
 \else\ifnum\fs=17\fontsize{17.28}{22pt plus.5pt minus .4pt}\selectfont
 \else\ifnum\fs=20\fontsize{20.73}{28pt plus.6pt minus .5pt}\selectfont
 \else\ifnum\fs=25\fontsize{24.88}{33pt plus.6pt minus .5pt}\selectfont
 \fi\fi\fi\fi\fi\fi\fi}
```

利用 \fontsize 指令可以同時改變字級與行距。如果經常須變動字級與行距, 使用表6.5之巨集指令較方便。此巨集指令名爲 mymacro.tex, 已置於 cwTEX 系統內。欲使用此巨集指令, 請在全文設定區加入下列一行指令:

```
\input mymacro
```

若要選用 17.28 點之字體 (行距設定爲 22pt), 只要鍵入 \sz17 即可; 如果要選用 12 點之字體 (行距爲 18pt), 則鍵入 \sz12; 餘此類推。mymacro 巨集指令之例子中設定 12 點字體之標準行距爲 18 點, 另再加上伸縮彈性。不過, 行距及伸縮彈性都可以自行修改。

有時候我們使用 \fontsize 指令的目的不在改變字級, 而只是要變更行距。底下的例子裡, 我們以 \sz12 指令選用 12pt 之字體。(爲了節省空間, 此例子之 \sz12 指令行距設定爲 15pt。)

I start with 12 points roman, *switch to italic type, switch to slant type,* switch back to Roman and make one word **bold**.

```
\input mymacro
\sz12
I start with 12 points roman,
\em switch to italic type,
\slshape{switch to slant type},
\rm switch back to Roman and
make one word {\bf bold}.
```

6.3.3 選用任意點數之英數字體

依原先設計, LaTeX 僅能使用特定尺寸之英數字體。不過, Knuth 所設計之

Computer Modern 字體已經被轉換成 Type 1 描邊字型格式。若使用新格式之字型排版, 我們即可選用任意點數之字體。其方法是在全文設定區引用 David Carlisle 之 type1cm 巨集套件:

```
\usepackage{type1cm}
```

引入此巨集套件之後, 英數字級可以透過 \fontsize 指令任意選擇。譬如, 下列指令

```
\fontsize{13.5}{20pt}\selectfont
```

即選用 13.5pt 之英文/數字字體。

6.4 選用中文字體

上一節所介紹之字體指令只能變換英文及數字之字體, 對於中文字並無作用。中文字體的變換必須利用 cwTEX 之指令。中文字體指令有簡要格式與完整格式兩種。簡要格式使用較方便, 完整格式則可以作細節控制。

6.4.1 簡要中文字體指令

中文字體指令也是以反斜線起頭, 指令內容包含兩個部分: 第一部分是中文字體名稱, 第二部分是指定字級。字級大小是以點數計算。譬如, 要選用明體字14.4點, 指令爲 \m14.4; 選用黑體字 12 點的指令爲 \b12, 楷體 12 點爲 \k12, 圓體 10 點的指令爲 \r10。圖 6.3 以 14 點字級爲例, 列示 cwTEX 現有的中文字體及引用字體之指令。

圖 6.3 所列每一種字體尙可作水平縮小或傾斜之變形。例如, \bbs12 指令選用 12 點斜粗黑體字, \bbe12 爲狹長粗黑體字, 而 \bbes12 則選用狹長斜粗黑體字, 請見圖 6.4 (頁 75)。同理, 選用 12 點狹長粗黑體字體之指令爲 \bbe12; 14 點斜楷體之指令爲 \ks14。中文變形字之式樣, 如傾斜角度與水平縮小比例, 皆可以自行設定, 相關之討論請見圖 13.1 (頁 258)。

以上的所介紹的字體是用於橫排之文稿。除了橫排字體外, cwTEX 尙提供直排字體。譬如, \vm12 選用直排12點明體字; \vbb14 則選用 14 點之粗黑體。換言之, 橫排字體指令前端加上英文字 v, 即選用直排字。直排例子請見第2章例6。直排字體尙在測試階段, 因此, 目前僅提供明體、粗黑體、楷體、圓體、仿宋體等5種; 而且僅有常用字。

\ml14	石創際手祝迴	\rl14	石創際手祝迴
\m14	石創際手祝迴	\r14	石創際手祝迴
\mb14	石創際手祝迴	\rb14	石創際手祝迴
\mu14	石創際手祝迴	\ru14	石創際手祝迴
\mx14	石創際手祝迴	\rx14	石創際手祝迴
\bl14	石創際手祝迴	\fl14	石創際手祝迴
\b14	石創際手祝迴	\f14	石創際手祝迴
\bb14	石創際手祝迴	\kl14	石創際手祝迴
\bu14	石創際手祝迴	\k14	石創際手祝迴
\bx14	石創際手祝迴	\ku14	石創際手祝迴
\l14	石創際手祝迴	\kx14	石創際手祝迴
\lb14	石創際手祝迴		

圖 6.3: 中文字體 (以 14 點字體爲例)

cwTEX 中文字體指令之有效範圍與 LATEX 英文字體宣告指令相同, 若指令下於大括號或指令環境內, 該指令僅於該範圍內有效。譬如, 以下三行例子中都在文字當中改變字體。因爲中文字體指令都是包圍在大括號或指令環境內, 故字體指令僅於該範圍內有效。

```
... \footnote{\m10 ...} ...
... \author{... \k12 ...} ...
... \begin{tabular} \bb12 ...\end{tabular} ...
```

第 2 行例子以 \author 指令排版作者名字。在 \k12 指令之前, 仍使用原先之字體, 之後則更改爲楷體 12 點。但遇到右大括號之後, 字體又回復原先之選擇。

以第 2 章例 2 來說明, 文稿標題字體選用明體 17 點, 因此標題 \title 指令大括號內加入 \m17 指令。中文字體指令與英文相同, 有效範圍是在大括號內。出了大括號之後, 中文字體恢復原先之設定。本例中, \title 指令之前並無任何中文指令, 因此標題之後回到內定之明體 10 點字體。

```
\title{\m17 台灣長期總產出之變動}
\author{\m11 吳聰敏\thanks{\m10
  作者任教 ...}}
```

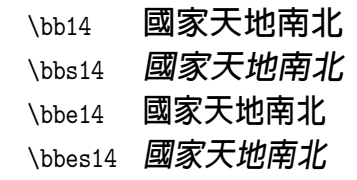


圖 6.4: 中文變形字: 以粗黑體字爲例

本例之作者名字要以明體11點編排, 因此 \m11 指令下於 \author 指令大括號內。作者名字之後的 \thanks 指令是用以排版致謝詞, 其內容是以明體字10點排版, 因此我們加上 \m10 字體指令。

同一例子內有下列一行文字:

```
... 總督府推動{\bb11 資本主義化}的政策,
```

行文中改變字體一定要加上左右大括號, 以確定字體變更的範圍。

有時候, 我們須在 LaTeX 指令範圍內變換中文字體。譬如, 中文章節標題通常會使用較大之字體:

```
\section{{\m14 耕者有其田}}
```

外圈的左右大括號是 \section{...} 指令的一部分, 內圈的左右大括號是 {\m14 ...} 中文字體變動的範圍。在此情況下, 我們可以省略一圈的左右大括號, 簡化成: \section{\m14 耕者有其田}。

依原先之設計, LaTeX 之英數字體是以1.2的次方放大。譬如, 放大1.2的2次方之後, 字級爲14.4點; 與此對應之中文明體字應爲 \m14.4。同樣的, 如果是放大1.2的0.5次方倍, 字級爲10.95點。爲了簡化輸入, 指令 \m14.4 可以簡化爲 \m14; 類似的, \b10.95 可以簡化爲 \b11 等等。以上之簡化指令雖然使用上較方便, 但卻產生另一個問題: 因爲 \b11 指令自動轉爲 \b10.95, 因此無法使用眞正的11點字體。如果不希望將 \b11 自動轉換爲 \b10.95, 執行 cwtex 指令時應加上 -n 選項:

```
c:\xtemp>cwtex -n examp2
```

此時, 選用明體10.95點字體, 須輸入之指令爲 \m10.95; 而明體11點字體之指令則爲 \m11。

6.4.2 完整中文字體指令

完整中文字體指令較複雜, 好處是可以自行設定中文字之間距及中文與阿拉伯數字的間距。從排版的角度來看, 中文和英文不同地方在於中文每一個字自成一個單位, 英文則以單字 word 爲單位。在純英文稿中, 爲了使版面每一行的右邊能夠切齊, LaTeX 會在英文單字之間加多或減少一些空白。遵循此一原理, cwTeX 也在中文單字之間插入一個可調整大小之空白。如此一來, 如果某行文字末端必須擺進一個標點符號, TeX 可以調整減少該行各中文字之間距。反之, 如果某一行的文字必須排得寬鬆一些, 也不難辦到。

中文字之間距有內定值。若你覺得內定之字距太大或太小, 可以改變之。完整中文字體指令之格式如下:

```
\ctxf{b}{12}{0.5}{0.8}{1.2}
```

此一指令的前半部分 `\ctxf{b}{12}`, 相當於簡要字體指令格式中之 `\b12`。後半部分中, `{0.5}` 選項的作用是將中文字之間距比內定值拉大 0.5 點; `{0.8}` 則是將中文字與阿拉伯數字之間距拉大 0.8 點; 最後的 `{1.2}` 則使中文句點之後的空白比內定值加大 1.2 點。以下的例子比較兩種指令格式之排版結果。首先是簡要指令格式:

台灣在 1945 年發生惡性物價膨脹; 美援進來之後, ...

```
\m11
台灣在1945年發生惡性物價膨脹;
美援進來之後, ...
```

若以完整格式指令改變字距, 排版結果爲:

台灣在 1945 年發生惡性物價膨脹; 美援進來之後, ...

```
\ctxf{m}{11}{0.8}{0.8}{1.0}
台灣在1945年發生惡性物價膨脹;
美援進來之後, ...
```

如果只要改變中文字距及中文與數字之間距, 最後之選項可以省略。

在 cwTeX 系統中, 英文與數字是直接取用 TeX 字體。那麼, 中文與阿拉伯數字的間距如何控制呢? 文稿內若有阿拉伯數字, 輸入時中文字與數字之間並不須特別留一空格, 因爲 cwTeX 會自動加入適當的間距。但如果你認爲 cwTeX 所留下的空白太小, 可以使用 `\ctxf` 指令擴大之, 如上例所示。另一個辦法是在數字 1945 前後各留一空格。此時, 排版結果將變成:

台灣在 1945 年發生惡性物價膨脹; 美援進來之後, ...

```
\m11
台灣在 1945 年發生惡性物價膨脹;
美援進來之後, ...
```

有些使用者抱怨 cwTEX 內定之中文字距太小, 排版時常利用完整字體指令加大字距。事實上, 自從文書處理軟體普及之後, 中文排版的最大問題是字距太大, 行距太小。結果使得排版文稿變得難以閱讀。在你動手改變字距之前, 請收集幾本品質較佳的雜誌, 看看其字距與行距之安排, 想想其中的道理。

以上的例子是以指令改變文稿某一段落之字距。若整篇文稿的字距都要改變, 最簡單的方法是在執行 cwtex 時加入選項。譬如, 中文字距要加大 0.5 點, 中文與數字之間要加大 0.8 點, 中文句點之後空白要加大 1.2 點, 執行指令時之選項爲:

```
c:\xtemp>cwtex -z+0.5 -Z+0.8 -zZ+1.2 test
```

請注意, 選項數字之前須加上 + 號。若要縮減小字距, 則加入 - 號。

除了調整字距之外, 中文字還可以上下移動, 以配合一些特別的英文字體。要把文稿內全部的中文字 (含中文標點符號) 下移 0.5 點, 執行 cwtex 時應加入選項 -l0.5:

```
c:\xtemp>cwtex -l0.5 test
```

如果是要上移中文字, 選項數字應爲負值, 如 -l-0.3。

6.4.3 中文字距之細節調整

上一小節所介紹的字體指令雖然可改變中文字距, 但僅適用於調整整篇文稿或某個段落之字距。如果只是要改變章節或表格標題之中文字距, 可使用 \csp 或 \cspp 巨集套件, 其定義與應用請見 194 節之說明。

排版中文大字標題時, 有時候須調整某兩個字之間距。譬如, 若以 40 點之仿宋字體排版「排版系統」四個字, 我們會發現「版」與「系」兩字之間距顯得太大。欲調整某兩個字之間距, 可使用 TEX 的 \kern 指令。譬如, 以下指令:

```
排版\kern-2pt 系統
```

可將「版」與「系」兩個字之間距縮小 2 點; 若取用正值, 間距將加大。

6.5 選擇字體與行距

要排出高品質文稿, 除了選用適當字體之外, 字距與行距的選擇也很重要。英文排版的原則是單字 (words) 應盡量靠近; 而行距應大於字距。心理學的視覺研究發現, 人們在閱讀英文文章時, 眼睛注視的並不是一個一個的字母, 而是整個單字, 或幾個單字合併而成的詞。如果單字的距離太大, 眼睛移動不順暢, 閱讀的速度將受影響。

一般英文排版的原則是: 若正文選用 10 點字體, 行距則設爲 12 點, 亦即字級的 120%。若正文使用 12 點的字體, 行距則設爲 14 點或 14.5 點。不過, 這項規則只供作參考, 並非一成不變的定則。事實上, 有些字體本來就設計得比較大, 因此使用不同的字體時, 即使字級相同, 行距可能也須改變。另外一個重要的考慮因素是行寬。一行的寬度越大, 行距通常也須加大。否則, 閱讀者的視線從上一行的末端, 不容易找到下一行的開頭。

中文字筆劃較英文字複雜, 因此中文行距的設定不能完全依照英文的原則。首先, 中文字體的高度和寬度和英文/數字並不相同。cwTEX 的英文與數字直接取用 TEX 之字體, 仔細比較之後, 你會發現中文字高度比英文字母的最高點還高一些, 底部則略低於基線。因爲中文字較高而且筆劃複雜, 如果行距仍設爲字體點數的 120%, 版面會擠得密密麻麻; 因此, 中文排版的行距應大於英文稿。至於大到什麼地步才適當, 目前似乎也沒有定則, 排版者應該用自己的眼睛去判斷。改變行距只需一道指令。因此, 我們儘可以去嘗試不同的行距, 選取自認爲最適當者。本書正文選用 11 點字體, 行距則設爲 16.5 點。

6.5.1 選用什麼字體?

字體選擇是排版首要考慮之一。我們從英文排版文獻整理一些原則, 以供參考。這些意見或許不能直接用於中文排版中, 不過, 舉一反三我們還是可以從中得到一些有用的提示。

首先是有關於字體的概念。一般的報章雜誌, 幾乎全部是以明體字編排。英文書籍或報紙也絕大部分是以羅馬字族排版。爲什麼呢? 西方心理學者的實驗研究發現, 羅馬字族的特點是易讀 (legible)。在專業字體設計的

術語中, 字族可大略區分爲 serif (裝飾邊) 及 sans serif (無裝飾邊) 兩種。英文羅馬字族和中文明體字一樣, 都是屬於 serif 型態, 其特徵是在每一筆劃的尾端有特別的勾勒。因爲這項特徵, 閱讀時字母較容易分辨, 字母容易接續起來形成單字。而且, 讀者容易從字母的上半部分辨識出該字母。因爲易讀, 這種字體普遍使用於報章、書籍與雜誌的排版中。

相對於 serif 字族的, 就是所謂的 sans serif, 其特徵是筆劃粗細較一致。譬如, 英文字體中的 sans serif 及 typewriter 字體; 或者中文字的黑體及圓體。在法文中, sans 表示「沒有」, 因此 sans serif 的意思是筆劃尾端沒有特別的勾勒。相對於 serif 字族而言, sans serif 字族較不易辨識, 因此也較不適用於排版文稿的正文。但是, 這種字體很醒目, 適合用於強調某段文字, 也適合用於排版章節的標題。

排版中英文夾雜的文稿時, 請注意字體搭配之問題。譬如, 如果中文使用明體或仿宋體, 英文應使用 serif 字體。反之, 若中文使用圓體或黑體, 英文以使用 sans serif 字體爲宜。底下第一行文字爲明體加上 serif 英文字體; 第二行爲圓體加上 sans serif:

中文明體字加上英文 serif 字體
中文圓體字加上英文 sans serif 字體

如果將中文明體字於與英文 sans serif 字體共用, 排版結果看來並不相稱。

在英文或其它西方國家的文字中, 每一套字體中的字母數目大都不超過 256 個, 因此專業的字體設計家設計出數以千計的字體可供選用。相對的, 常用的中文字就有三、四千字, 設計一套幾千字的中文字體要花費相當大的工夫。因此, 中文字體的選擇就相當有限。但不管是中文或英文, 正文內容和章節標題的字體如何搭配是一門學問。在純英文稿中, 雖然可供選用的字體數以千計, 但專家的建議是, 書籍一頁版面上不應超過三種字體。

排版的目的是把作者的意見清楚、扼要地傳達給讀者。一個版面中使用太多的字體, 讀者的注意力分散在花花綠綠的字體上, 作者的觀點反而無法有效的傳達。初學排版者往往傾向於使用多一點的字體, 這是應該避免的。

TEX 的排版能力強, 品質甚佳, 但如果使用不當, 可能排出慘不忍睹的版面。有興趣於排版者, 請參考 Bringhurst (1996) 與 Taylor (1995)。以下謹列出一些常見的錯誤, 請儘可能避免之。

- 行距太小

  行距太小, 版面文字密密麻麻, 讀者閱讀時壓力太大。適當的行距應該是多少? 若是排版純英文書籍, 直接使用 LaTeX 之內定值, 效果不錯。但若是中文稿件, 因爲中文字筆劃複雜, 行距須加大。行距與字級有關。字體小者, 行距可以小一些。行距與行長 (版面寬度) 也有關係。行長較大, 行距須加大。版面太寬時, 閱讀壓力也上升。行距沒有標準數值。如果是一般文稿, 正文選用12點字體, 行距請試用 18-20 點。

- 行長太大

  行長太大, 閱讀時壓力大, 其道理與行距太小一樣。如果排版結果要印在 A4 紙張上, 紙面寬度爲 21 公分。很多人把行長設爲 17 公分, 左右各只留 2 公分的空白。事實上, 左右兩邊至少應各留 3-4 公分的空白。如果左右各留 4 公分, 行寬變成 13 公分, 閱讀時會感覺更爲順眼。絕大部分的英文雜誌, 如 The Economist 或 TIME 等, 都是以 2-3 欄型式編排, 其道理就是在減少行長。另外, 看看國內的報紙、雜誌, 我們也會發現每一行的長度都不大。

- 字距太大

  早期簡陋的文書處理軟體常把每一個字排得斗大, 文字之間的空格也加得很大。研究人員的實驗發現, 人在閱讀時, 看的並非以一個文字爲單位, 而是以詞句爲單位。字距太大, 詞句的長度也增加, 眼睛感到吃力, 吸收能力也下降。很多人認爲 cwTeX 所設定的字距太小, 有興趣者不妨自行實驗, 同一文稿分別以內定字距與加大字距排版, 再比較結果, 看看哪一種字距最適合閱讀。

- 選用不當的正文字體

  文稿章節標題之字體通常不同於正文之字體。一般而言, 章節標題字體的要點是顯目; 正文字體之選擇要點是易讀。我們經常可看到一些以楷體、隸書體、仿宋體等排版正文之文稿。事實上, 這些字體的特徵是醒目, 而非易讀。翻開報紙或任何較具水準的雜誌, 內文字體毫無例外都是明體, 原因是明體字是最易讀的字體。因此, 除非文稿簡短或性質特殊, 選用非明體字排版正文之前, 請三思而後行。

6.5.2 避頭點

仔細觀察英文書籍, 我們發現逗點、句點等標點符號不會出現在一行之首。在專業排版中, 這稱爲「避頭點」。傳統的中文排版也有類似的作法, 其中的道理並不難理解。標點符號是作者議論或語氣停頓之處, 也是讀者眼睛稍微休息時。就一本書的版面來看, 讀者的眼睛從上一行之尾端轉到下一行開頭時, 他預期小停頓之後有一個新的起頭。因此, 如果一行之首竟然碰到標點符號, 顯然不合讀者的直覺與預期。這是爲什麼排版時要「避頭點」的原因。不幸的是, 很多中文桌上排版系統, 都沒有把這項因素納入考慮。

有些排版系統所使用的中文字及標點符號橫寬一致。排版之後, 每一行的中文字上下對得整整齊齊的, 但卻無法避頭點。有些人甚至認爲在橫排的版面中, 文字上下對齊才好看。但是, 好的排版是要使文章或書籍容易閱讀。在橫排的書籍或文章中, 讀者並不會從上往下看一本書。因此, 中文字上下對齊並無任何意義。cwTEX 原則上可以處理避頭點的問題, 但偶而仍會有標點符號出現於一行開頭。遇有此種情況, 請將前一兩行的文字增減一字, 以解決避頭點的問題。

7 文稿結構與章節設計

使用 LaTeX 排版須了解文稿結構 (document structure) 的概念。文稿結構是指書籍或文稿各部分的組合方式。以書籍而言, 組成部分包含標題、目錄、序言、各章節、附錄、索引等等; 相對而言, 一般的短文可能只有標題、節與小節、參考文獻等。文稿有長有短, 長篇文稿的版面與短文的版面也可能不同。譬如, 若排版長篇文稿, 標題可能單獨排爲一頁; 而一般的短文標題之下即直接排版正文。

LaTeX 提供現成的指令讓排版者選用特定之文稿結構。排版文稿開端的第一道指令通常是以 `\documentclass` 指令選用排版文件之類別。文件類別 (document class) 一經選定, 全篇文稿之結構就同時決定。譬如, 若選用 `book` 文件類別, 則以 `\title` 指令排版標題時, 標題文字將自成一頁。反之, 若文件類別是選用短文 `article`, 則同樣的 `\title` 指令會選用較小的字體, 而標題之下直接排版正文, 不獨立成一頁。

除了一兩頁的短文之外, 文章通常以章節區分段落。因爲章節設計是文稿結構最重要的一部分。本章除了說明 LaTeX 的文稿結構指令之外, 也將介紹一些用以重新設計文稿結構的巨集套件。

7.1 文稿結構

LaTeX 的文件類別指令 `\documentclass` 通常是文稿內的第一行指令, 其功能是選定排版文件的結構。譬如, 若選定 `book` 文件類別, LaTeX 即自動處理各種排版細節, 包括版面長度與寬度; 選用特定字體排版標題與正文文字; 單數頁與雙數頁版面的位置對稱於書脊; 每一章從單數頁開始編排; 節標題文字之前會留下較大空白, 標題之後第一行文字不內縮 (indent) 等等。

LaTeX 所提供的文件類別有適用於編排整本書者, 也有適用於編排短文或信函者。以 `article` 文件類別來說, 其章節結構包含標題、摘要、正文、參考文獻、附錄幾部分。正文則進一步細分爲節、小節、次小節等等。`book`

的章節結構和短文類似, 但內容更複雜一些。

7.1.1 短文

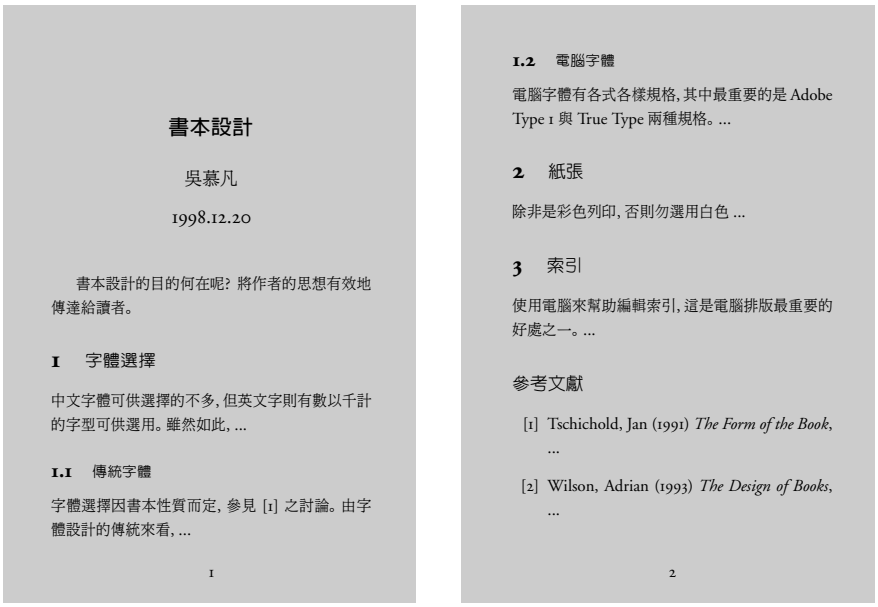
圖 7.1 是短文的例子, 上半部分爲排版結果, 其下爲輸入之指令與文字。文稿第一行爲文件類別指令 \documentclass, 此道指令設定以短文 article 編排文稿, 並以選項 [12pt] 選用 12 點字體。若不選字體, 內定值爲 10 點。第 4 行 \begin{document} 指令開始即進入 document 指令環境, 其內爲文稿內容, 文稿最後一行爲 \end{document}。本例子之檔案名爲 exam-art.ctx, 置於 \texmf\cwtex\examples 檔案夾內, 請試自行排版。

在 LaTeX 中, 以 \begin{...} 及 \end{...} 所涵蓋的內容稱爲「指令環境」(environment)。因此, 從 \begin{document} 開始到 \end{document} 結束的全部文字及指令, 即構成「文稿指令環境」(document environment)。在 \begin{document} 指令行以上的區域稱爲「全文設定區」(preamble)。本例中, 全文設定區僅含一道指令, 其作用是重新定義參考文獻之中文標題與字體大小, 底下將進一步說明。

文稿一開始通常先排版文章題目、作者姓名、稿件日期等, 這是第 5-7 行指令所排版。第 5 行以 \title 指令排版標題, 並以 \bb17 中文字體指令選用 17 點粗黑體。第 6 行以 \author 指令排版作者名字。第 7 行之 \date 指令若省略不加, LaTeX 將自動填入排版當天的日期。第8行 \maketitle 指令之作用是指示在此之前的文字應排版於標題區域內。在 article 文件類別下, 標題區域之下即接著排版正文內容; 但在 book 文件類別下, 標題將獨占一頁。

文稿正文之節、小節是以 \section 與 \subsection 指令編排, LaTeX 會自動將各節、小節編上號碼。圖 7.1 中, 文稿計分 3 節, 都是以 \section 指令編排。章節指令除了自動編號之外, 還會選用較大之英數字體。因此, 若排版中文稿, 我們須加入中文字體的指令, 以使數字編號與中文標題之大小對稱。本例中, 我們以 \r14 選用 14 點之圓體字排版節標題。第 19 行與 23 行之 \subsection 指令是編排小節標題, 因爲位於第 1 節之內, 因此自動編號爲**1.1** 與**1.2**。小節之中文標題選用 12 點圓體字排版。

專業文章經常引用參考文獻。如果參考文獻多, 徵引時容易出現錯誤。LaTeX 提供 thebibliography 指令環境以排版參考文獻。本例38-43 行說明如何使用此一指令環境。每一徵引文章或書本皆以 \bibitem 起頭, 並任取



書本設計

吳慕凡

1998.12.20

書本設計的目的何在呢? 將作者的思想有效地傳達給讀者。

1 字體選擇

中文字體可供選擇的不多, 但英文字則有數以千計的字型可供選用。雖然如此, ...

1.1 傳統字體

字體選擇因書本性質而定, 參見 [1] 之討論。由字體設計的傳統來看, ...

1

1.2 電腦字體

電腦字體有各式各樣規格, 其中最重要的是 Adobe Type 1 與 True Type 兩種規格。...

2 紙張

除非是彩色列印, 否則勿選用白色 ...

3 索引

使用電腦來幫助編輯索引, 這是電腦排版最重要的好處之一。...

參考文獻

[1] Tschichold, Jan (1991) *The Form of the Book*, ...

[2] Wilson, Adrian (1993) *The Design of Books*, ...

2

```
\documentclass[12pt]{article}
\renewcommand{\refname}%
 {\r14 參考文獻}
\begin{document}
\title{\bb17 書本設計}
\author{\m14 吳慕凡}
\date{1998.12.20}
\maketitle
\fontsize{12}{17pt}\selectfont
\m12
書本設計的目的何在呢?
將作者的思想有效地傳達給讀者。

\section{\r14 字體選擇}
中文字體可供選擇的不多,
但英文字則有數以千計的字型%
可供選用。雖然如此, ...

\subsection{\r12 傳統字體}
字體選擇因書本性質而定,
參見 \cite{tsc} 之討論。
由字體設計的傳統來看, ...

\subsection{\r12 電腦字體}
電腦字體有各式各樣規格,
其中最重要的是 Adobe Type 1
與 True Type 兩種規格。
...

\section{\r14 紙張}
除非是彩色列印, 否則勿選用白色
...

\section{\r14 索引}
使用電腦來幫助編輯索引,
這是電腦排版最重要的好處之一。
...

\begin{thebibliography}{99}
 \bibitem{tsc} Tschichold, Jan (1991)
  {\it The Form of the Book}, ...
 \bibitem{wilson} Wilson, Adrian (1993)
  {\it The Design of Books}, ...
\end{thebibliography}
\end{document}
```

圖 7.1: 短文 article 之文稿結構

一簡名。例如, 第 39 行列出 Tschichold 的著作時, 指令爲 \bibitem{tsc} , 其中 tsc 爲自取之簡名。正文第 21 行引用此著作時, 使用 \cite{tsc} 指令。排版時, LaTeX 自動從參考文獻中找到 tsc 項, 並在正文中代入其排序號碼。因爲此項著作爲徵引文獻的第 1 篇, 排版之後變成「參見 [1] 之討論」。如文稿 38 行所示, 輸入 thebibliography 指令環境時其末端加上 {99}, 目的是爲參考文獻的序號預留 2 位數字之排版空間。如果文稿引用之參考文獻超過 100 篇, 則 {99} 應改爲 {999}。

在短文中, 使用 thebibliography 指令環境時, LaTeX 會自動在文獻項目之前加上英文字 **References**。若是中文文稿, 我們可以更改此項標題爲中文字。在 LaTeX 中, **References** 一字是以 \refname 指令代表。因此, 我們在全文設定區使用 \renewcommand 指令將 \refname 之內容重新定義爲「參考文獻」, 並選用 14 點圓體字。

以上說明如何使用 thebibliography 指令環境排版參考文獻。實際上, LaTeX 雖然提供排版參考文獻的指令環境, 但並不強制非得使用不可; 我們也可以用一般的指令排版參考文獻。不過, 文稿結構指令 \documentclass 與 document 指令環境等三道指令是一定要有的。

7.1.2 書籍

上一小節介紹的 article 文件類別主要用以排版篇幅較短之文稿。本小節進一步介紹 book 文件類別, 這通常用於排版書籍。不過, 用來排版較長篇論文也無不可。

圖 7.2 是書籍 book 文稿的一個例子, 雖然指令比圖 7.1 所示之短文要複雜一些, 但基本邏輯是相同的。以章節架構而言, 整本書可分 \frontmatter, \mainmatter, 及 \backmatter 三部分。其中, \frontmatter 包含序言、目錄、書名、作者名字等; \mainmatter 包含正文內容, \backmatter 包含文獻索引等等。使用這些指令的好處之一是它們可以幫忙控制頁碼之字體。不過, 本例中並未使用這三道指令。如果在本例的第 30 行加入 \mainmatter 指令, 則第一章之前的目錄與書名頁之頁碼將自動改以小寫羅馬字排版; 第 1 章開始則以阿拉伯數字編頁碼, 並且重新由 1 開始計算。

書籍的正文之前須排版書名、版權頁、目錄等等。選用 book 文件類別排版時, 書名與作者名字將排版於獨立的標題頁中, 其後通常接著排版章節目錄。章節目錄可以使用 \tableofcontents 指令編排; 欲排版圖目錄, 指

令爲 \listoffigures。同理, \listoftables 指令則用以編排表目錄。本例子並無圖形, 例子內加入以上指令僅提供參考。圖 7.2 所示例子中, 第 24 行指令 \tableofcontents 即用以排版目錄。

LaTeX 會自動取用各章節標題, 算出其頁碼, 並在 \tableofcontents 指令處排出一目錄頁; 目錄頁上方並自動加上英文字 **Contents**。若是中文文稿, 應將此標題改爲中文。本例第 9-10 行之 \renewcommand 指令, 即是將英文標題改爲中文「目錄」兩字, 並選用 14 點圓體。

LaTeX 如何取得目錄之內容? 它根據的是使用者輸入的章節標題。譬如, 圖 7.2 第 31 行指令爲

```
\chapter[{\m12 導論}]{\bb17 導論}
```

其中, \chapter 爲排版章標題指令; {\bb17 導論} 選用中文粗黑體17點字體排版本章標題。在編排目錄內容時, LaTeX 將自此行指令取用標題文字。但是, 標題頁之中文是選用17點粗黑體字, 因此目錄中也將以同樣大小之字體排版。這顯然不太適合。若希望目錄以較小字體編排, 例如, 明體 12 點, 我們須在 \chapter 指令中加入 [{\m12 導論}] 選項, LaTeX 將取用選項中之文字排版目錄頁。方括號內之選項除用以排版目錄頁之外, 也用於排版每一頁之頁眉 (header) 或頁足 (footer), 底下將有進一步說明。

英文的章節標題是 LaTeX 自行設定的, 因此從頭到尾都會選用同一字體。本例之中文標題則是一個一個分別以中文字體指令控制。排版長篇書籍時, 這很容易出錯。爲解決此一問題, cwTeX 提供指令以設定全篇文稿的章節標題之中文字體, 請見 7.7 節之說明。

圖 7.2 的書籍例子計有 4 章, 每一章之標題都是以 \chapter 指令排版, 分別位於第 31、46、74 與 78 行。最後的參考文獻也是利用 thebibliography 指令環境排版。在 book 文件類別下, 欲更改參考文獻之標題字變成中文, 指令如第 6-7 行所示。請注意, 在短文類別中, 所更動之指令爲 \refname; 在書籍類別中指令爲 \bibname。

本例子第 2 章內有 2 節, 2.2 節內有一表格。本書第 10 章將詳細介紹排版表格的方法。本例子的主要目的是說明如何排版表目錄。較大的表格通常須加上標題。第 65 行 \caption 指令之功能即是排版此表格標題「字體規格」四個字。我們可以利用 \listoftables 指令 (第 28 行), 要求LaTeX 收集全書各 \caption 指令之內容, 排版出表目錄。如果書籍內有圖形, 我們

也可以利用 \listoffigures 指令 (第26行) 排版圖目錄。最後, 11-14行之指令的功能是將圖目錄與表目錄之標題改爲中文字。

以上大體介紹了書籍排版常用之指令。一般書籍常利用頁眉或頁足排版頁碼或節標題。此例中第 4-5, 25, 27, 29, 與 32 行的指令即是控制頁眉與頁足之排版, 我們將於 7.8 節詳細說明。最後, 2-3 行, 60 行, 與第 89 行指令是用以排版索引。

第60行兩個 \index 指令是用以標誌索引名詞。有關於排版索引的方法, 請見 15.5 節。圖 7.2 的例子檔名爲 exam-bk.ctx, 置於 \texmf\cwtex\examples 檔案夾內。請試以第 3 章之方法排版。執行 cwtex 與 latex 之後, c:\xtemp 檔案夾內將產生幾個輔助檔案。其中, exam-bk.idx 檔案記錄各索引名詞之頁碼。欲在文稿末端自動排出索引, 請進入 DOS 模式, 執行下列指令:

```
c:\xtemp>cwidx exam-bk
```

cwidx 是一批次檔, 其功能是將各索引項排序。最後, 再執行一次 latex, 索引名詞即正確地排版於文稿末端。

7.1.3 文件類別

圖 7.1 與 7.2 中 \documentclass 指令選用之 article 與 book 稱爲文件類別 (document class)。短文與書籍類別之間, 還有報告 report 類別, 其章節結構比短文複雜, 但比書籍簡單一些。常用之文件類別包括 article, report, book, slides 與 letter 等五種。其中, letter 是用於排版信函, slides 用於排版投影片, 其餘三種則用於排版各類之長短稿件。

以上之文件類別可以靈活應用。若書籍之內容單純, 我們可以直接使用 article 排版。反之, 若文章的內容複雜, 則以 book 編排可能較方便。如果文稿中含有許多數學式, 我們還可以使用 amsart 與 amsbook 文件類別, 這是美國數學學會 (American Mathematical Society) 爲了排版數學文稿所發展出來的, 請見第 9 章之說明。

以上 5 種文件類別所提供之指令雖然已能滿足多數人之需求, 但任何事情都有更上一層樓的空間, 排版也不例外。因此, 各國的 LaTeX 專家與使用者又寫出許多應付特別需求之巨集套件 (package)。譬如, LaTeX 本來就提供排版表格之指令, 但是有人又寫了一套功能更強之 array 巨集套件。又如,

```
\documentclass[12pt]{book}
\usepackage{makeidx}
\makeindex
\usepackage{fancyhdr}
\pagestyle{fancy} \lhead{}
\renewcommand{\bibname}%
  {\r14 參考文獻}
\renewcommand{\indexname}{\r14 索引}
\renewcommand{\contentsname}%
  {\r14 目錄}
\renewcommand{\listfigurename}%
  {\r14 圖目錄}
\renewcommand{\listtablename}%
  {\r14 表目錄}
\renewcommand{\chaptername}{}
\renewcommand{\tablename}{\r12 表}
\renewcommand{\figurename}{\r12 圖}
\textwidth=8cm \voffset=-2cm
\begin{document}
\title{\bb20 書本設計}
\author{\m14 吳慕凡}
\date{1998.12.20}
\maketitle
\tableofcontents
 \clearpage \rhead{\m12 目錄}
\listoffigures
 \clearpage \rhead{\m12 圖目錄}
\listoftables
 \clearpage \rhead{\m12 表目錄}

\chapter[{\m12 導論}]{\bb17 導論}
\fancyhead[RE]{\rightmark}
\fontsize{12}{17pt}\selectfont
\m12
書本設計的目的何在呢?
將作者的思想有效地傳達給讀者。

\newpage
感謝許多朋友的協助幫忙,
使本書得以順利完成。...

\vspace*{3cm}\hspace*{.4\textwidth}
\parbox{4cm}{
吳慕凡 \\ \today}

\chapter[{\m12 字體選擇}]{\bb17 字體選擇}
中文字體可供選擇的不多,
但英文字則有數以千計的字型可供選用。
雖然如此, ...

\section[{\m12 傳統字體}]{\r14 傳統字體}
字體選擇因書本性質而定,
參見 \cite{tsc} 之討論。
由字體設計的傳統來看, ...

\section[{\m12 電腦字體}]{\r14 電腦字體}
電腦字體有各式各樣規格,
其中最重要的是 Adobe Type 1
與 True Type 兩種規格。
\index{Type 1} \index{True Type}
...

\begin{table}
\centering
 \caption{字體規格}
 \begin{tabular}{rl}
  \hline
  描邊字體 & Type 1 \\
  描點字體 & \TeX{} pk 字體 \\
  \hline
 \end{tabular}
\end{table}

\chapter[{\m12 紙張}]{\bb17 紙張}
除非是彩色列印, 否則勿選用白色。
...

\chapter[{\m12 索引}]{\bb17 索引}
使用電腦來幫助編輯索引,
這是電腦排版最重要的好處之一。
...

\begin{thebibliography}{99}
 \bibitem{tsc} Tschichold, Jan (1991)
  {\it The Form of the Book}, ...
 \bibitem{wilson} Wilson, Adrian (1993)
  {\it The Design of Books}, ...
\end{thebibliography}
\printindex
\end{document}
```

圖 7.2: 書籍 book 文件類別之結構

LaTeX 中已有指令可以排版頁眉詞, 但 fancyhdr 巨集套件功能更強, 使用更方便。我們如何可使用這些巨集套件呢? 以 array 爲例, 欲使用巨集套件內之指令, 首先必須在全文設定區以 \usepackage 指令引用之,

```
\documentclass[11pt]{article}
\usepackage{array}
```

如果要同時使用兩種以上之巨集套件, 引用指令爲:

```
\documentclass[11pt]{article}
\usepackage{fancyhdr}
\usepackage{array}
```

或者

```
\documentclass[11pt]{article}
\usepackage{array,fancyhdr}
```

請注意, 各選項之間不得留有空白。

以上所介紹的文件類別指令名爲 \documentclass, 這是新版 LaTeX2ϵ 之名稱; 舊版 (2.09版) 稱之爲 \documentstyle。新版 LaTeX 中, 巨集套件是由 \usepackage 指令引入; 但在舊版中則直接將之作爲文件類別指令之選項。如果你仍使用舊版 LaTeX2.09, 引用巨集套件之指令爲:

```
\documentstyle[11pt,array,fancyhdr]{article}
```

新版的 LaTeX 是在 1994 年開始流通。如果你已由舊版轉爲新版, 但以往許多的文稿是以舊版指令排版, 這些舊指令絕大部分在新版中仍然可以使用, 不過執行速度會慢一些。

在 CTAN 網站上有許多現成的巨集套件可供下載使用。若能善用網路上的資源, 再怎麼特別的排版需求幾乎都可以找到現成的巨集套件來解決。本書將介紹一些我們認爲有用的巨集套件。

7.1.4 指令選項

許多巨集指令都可以設定選項 (options)。舉例來說, 以 article 文件類別編排文章時, LaTeX 內定以 10 點字體排版正文。如果想要使用較大的 12 點字體, 可以加上 [12pt] 之選項。此時文件類別指令變成:

```
\documentclass[12pt]{article}
```

指令選項須以方括號括起來, 以有別於以大括號括起來之強制選項。事實上, [12pt] 選項除了改變字體大小外, 也改變版面其它部分。譬如, 註解文字之字體會隨著加大一些; 文稿中的數式若有上下標, 其大小也會配合正文字體而改變。若嫌 12pt 之字體太大, 我們也可以選用 11pt。

除了選擇字體大小之外, 文件類別指令還可以加入許多其他選項。我們選擇其中較重要的簡單介紹如下。首先, 紙張的選擇可使用下列選項:

```
letterpaper   11×8.5 英吋
a4paper       29.7×21 公分
a5paper       21×14.8 公分
```

其中, letterpaper 爲內定值。其次, 大多數的排版是將文稿內容印於垂直的紙張上, 這稱爲 portrait (畫像) 模式, 也是 LaTeX 內定之模式。但是, 若在文件類別指令中加入 landscape 選項, 則文稿將以橫向或所謂的 landscape (風景) 模式排版。

在 article 文件類別中, 文稿每一頁將排版於紙張相同的位置。但如果選用 book 格式, 單雙頁將對稱於書脊排版。換言之, 紙面上右頁的版面會較靠近左邊; 左頁的版面則較靠右邊。本書是以 book 文件類別排版, 你現在閱讀的這一頁與對面一頁就是對稱於書脊。依原始設定, article 與 report 文稿的單雙頁版面都是位於紙面同樣位置。如果希望單雙頁對稱於書脊編排, 可在文件類別指令中加入 twoside 選項。

此外, 以 book 文件類別排版書籍時, 每一章開頭之標題頁會自動從右頁 (單數頁) 開始編排。如果要取消此項設定, 文件類別指令中應加入下列選項:

```
openany
```

最後, 數學式之排版格式也可以用選項控制。依原始設定, 數學式將居中編排, 數式編號則排於右邊。若數式編號要置於左邊, 應在文件類別指令中加入下列選項:

```
leqno
```

如果是數式要靠左編排, 選項指令爲:

`fleqn`

我們還可以進一步使用 `\mathindent` 指令讓靠左編排之數學式從特定位置開始排版。譬如, 下列兩行指令:

```
\documentclass[12pt,fleqn]{article}
\mathindent=1.5cm
```

即設定讓每一行數學式從距離文字版面邊緣 1.5 公分處開始排版。

除了以上所述之外, 文件類別指令其他選項如下:

`twocolumn` 以兩欄型式編排,
`titlepage` 使文章標題、作者名字等自成一頁。

`twocolumn` 選項雖然可以設定排版兩欄式版面, 但使用上限制較多。欲排版多欄位版面, 請見 8.10 節之說明。

7.2 文稿標題

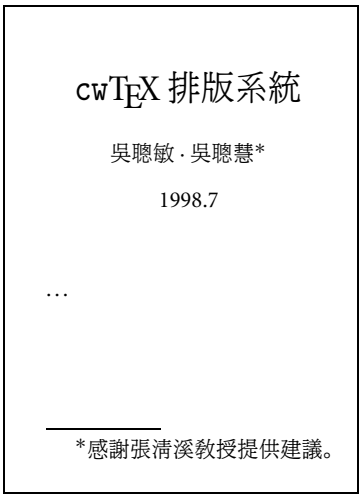
一般書籍或文章的標題頁包含題目、作者、出版日期與謝詞四部分。`book` 與 `report` 文件類別之標題獨占一頁; 但短文 `article` 中, 標題區域之後即緊接著排版正文 (或摘要), 標題並未獨占一頁。不過, 爲了簡化文字說明, 以下都以標題頁稱之。

標題頁可以自行設計排版, 也可以利用現成之指令。若自行設計, 指令與文稿內容可以置於指令環境中:

```
\begin{titlepage}
 ...
\end{titlepage}
```

若利用現成指令, 排版標題的指令爲 `\title`, 排版作者名字可使用 `\author`, 日期指令爲 `\date`, 如圖 7.3 所示。各項指令都會自動選用適當大小的英數字體, 但中文字體必須由 cwTEX 指令設定。本例以 `\m17` 指令排版中文標題, 謝詞則使用10點明體字, 作者名字以 12 點明體字排版。

輸入標題之後, 接著輸入 `\maketitle` 指令, 指示以上爲標題頁內容。在 `\author` 指令之後, 我們可使用 `\thanks` 指令排版致謝詞, 其內容將以註




```
\documentclass[11pt]{article}
\renewcommand{\thefootnote}%
  {\fnsymbol{footnote}}
\m17
\title{{\tt cw}\TeX 排版系統}
\m12
\author{吳聰敏 $\cdot$ 吳聰慧\thanks{
\m10
感謝張清溪教授提供建議。}}
\date{1998.7}
\begin{document}
\maketitle
\renewcommand{\thefootnote}%
  {\arabic{footnote}}
\m11
...
```

圖 7.3: 編排標題

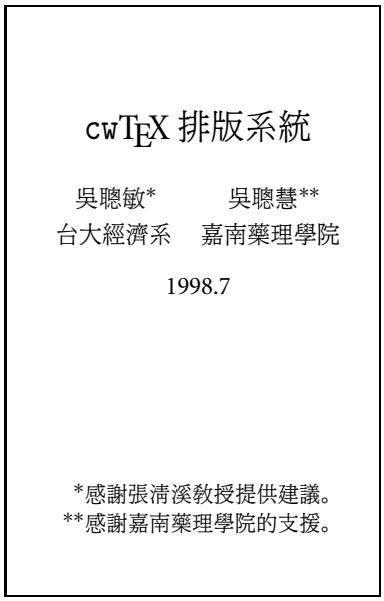
解形式出現在標題頁下方。本例中, 誌謝詞之註解是以星號標示。事實上, 依 LaTeX 之設定, 誌謝詞是以阿拉伯數字標示。圖 7.3 例子的第 2-3 行利用 \fnsymbol 指令將阿拉伯數字改爲星號。標題頁排版完成之後, 我們使用類似的指令將註解標示改回阿拉伯數字, 見本例第 13-14 兩行指令。有關於註解格式之變更, 詳見 8.6 節之說明。

本例之致謝詞是選用 \m10 字體。你或許會感到疑惑, 如何選出適當大小的中文字體? 答案很簡單: 多試兩次。若正文字體是以 12 點字體編排, 則註解與致謝詞內容會自動選用 10 點字體。若不能確定, 直接以中文字體指令改變字體, 試看看排版結果如何; 一兩次之後, 很快就可以得到答案。

如果你對於英文/數字字體之格式不滿意, 也可以改變之。譬如, 本例中若嫌日期指令 \date 所設定之英文字體太大, 我們可以用 \footnotesize 指令縮小之, 例如:

```
\date{\footnotesize 1998.7}
```

如果不下 \date 指令, LaTeX 將自動填入排版當天的日期。若不希望列出日期, 可以使用 \date{} 指令。兩個大括號緊接在一起, 表示日期指令內容空白。同樣的, 如果我們下 \author{} 指令, 作者名字將不出現。但如果完全省略 \author 指令, 排版時將產生錯誤訊息。




```
\documentclass[11pt]{book}
\renewcommand{\thefootnote}%
  {\fnsymbol{footnote}}
\m17
\title{{\tt cw}\TeX 排版系統}
\m12
\author{吳聰敏\thanks{\m10
 {感謝張清溪教授提供建議。}}\\
 台大經濟系\\
\and
 吳聰慧\thanks{
 {\m10 感謝嘉南藥理學院的支援。}}\\
 嘉南藥理學院}
\date{1998.7}
\begin{document}
\maketitle
\renewcommand{\thefootnote}%
  {\arabic{footnote}}
...
```

圖 7.4: 作者與感謝詞

如上所述, 標題、作者等指令輸入完畢後, 還必須下 \maketitle 指令。這道指令通常是緊接在 \begin{document} 指令之後。如果使用 book 或 report 文件類別, 標題、作者與日期將排版在單獨的一頁上, 正文則從下一頁開始, 請見圖 7.4 之例子。

作者有兩人或兩人以上時, 名字可以並列, 圖 7.3 是最簡單的方法。事實上, 這無異於排版單一作者的情況, 唯一不同的是我們以 $\cdot$ 指令產生一小點, 加在兩位作者名字中間。圖 7.4 的例子稍爲複雜一些, 輸入方法是把第二位作者的名字及感謝詞加於第一位作者名字與謝詞之後。但是, 兩位作者名字之間需加入 \and 指令。我們以換行指令 \\ 指令區隔作者名字及其服務單位。

基本上, 作者名字及其所屬單位可以看成 \author 指令之內容。若不拆開, 作者名字及所屬單位將排版成一長串文字。換行指令 \\ 的作用是使服務單位移到下一行, 並對準作者名字。

如果文章標題很長, 一行排列不下, 我們可以用換行指令 \\ 拆成兩行甚或三行。譬如, 經濟學名著《國富論》的英文原名甚長, 輸入時可以拆成

兩行:

```
\title{An Inquiry Into The Nature And\\
       Causes Of The Wealth of Nations}
```

排版之後, 兩段文字都將居中排列。

如前所述, 如果對於 LaTeX 所編排之格式不滿意, 我們可以自行設計標題頁之版面。此時, 可以將排版標題、作者、名字等指令置於 titlepage 指令環境中。請注意, 在此指令環境內, 上述之現成指令, 如 \title, \author 等皆不能使用, 也不須下 \maketitle 指令。

7.3 頁碼

文稿若無特別設定, LaTeX 會自動編入頁碼。依專業排版之規範, 書籍每一章標題頁通常不排版頁碼; 傳單之類的簡單文稿通常也不加頁碼。LaTeX 會自動取消每章標題頁的頁碼, 但欲取消傳單的頁碼, 須以指令消除。

要更改目前正在編排的這一頁的頁碼, 須使用 \thispagestyle 指令。相反的, 如果是要改變文稿中的每一頁, 應使用 \pagestyle 指令。譬如, 在全文設定區加入下列指令:

```
\pagestyle{empty}
```

文稿每一頁的頁碼都不出現。反之, 在特定點加上 \thispagestyle{empty}, 該頁之頁碼即不出現。事實上, 以上兩道指令不僅取消頁碼, 也使版面上下方的頁眉與頁足資訊全部變成空白。第 7.8 節將詳細介紹排版頁眉與頁足之指令。

7.3.1 重新編頁碼

在正常的情況下, 頁碼是從 1 開始編排。如果頁碼要重新定爲其他數字, 例如 20, 須透過頁碼計數器 (page counter)。計數器是 LaTeX 記錄章節編號、頁碼數目等之變數。頁碼數目是記錄於 page 計數器內; 欲改變頁碼, 可使用 \setcounter 指令:

```
\setcounter{page}{20}
```

表 7.1: 頁碼數字之格式

`arabic`	阿拉伯數字 (內定),
`roman`	小寫羅馬字, 如 i, ii, . . . ,
`Roman`	大寫羅馬字, 如 I, II, . . . ,
`alph`	小寫英文字母, 如 a, b, . . . ,
`Alph`	大寫英文字母, 如 A, B, . . . 。

以上指令將本頁頁碼改爲 20。若某書共有 9 章, 排版時每一章分開編排。假設第 2 章是從第 20 頁開始, 我們即可在第 2 章檔案前端加入上述指令。

頁碼之數字是以阿拉伯數字編排, 但也可以改成羅馬數字甚至英文字母。舉例言之, 欲改用小寫羅馬數字編排頁碼, 請在全文設定區輸入下列一行指令:

```
\pagenumbering{roman}
```

其他可用之選項如表 7.1 所示。`\pagenumbering` 指令除了改變頁碼數字格式之外, 頁碼計數也會由 1 重新開始。請注意, 小寫字母之指令爲 `alph`, 而非 `alpha`。這可能是爲了與排版希臘字母 α 之指令 `\alpha` 有所區別。同理, 設定大寫字母之指令爲 `Alph`。

某些洋文書的目錄或序文是以小寫羅馬字排頁碼, 正文之頁碼才使用阿拉伯數字。若欲追隨此項傳統, 可使用 `\frontmatter` 與 `\mainmatter` 指令。若在標題頁內容之前加上 `\frontmatter` 指令, 頁碼數字將改以羅馬數字編排。開始排版正文之前應下 `\mainmatter` 指令, 頁碼將改以阿拉伯數字編排, 並重新從 1 起算。`\frontmatter` 指令除了改變頁碼數字格式之外, 還有其他效果。

排版書籍時, `\chapter` 指令可用以編排每章標題文字, LaTeX 會自動加入章編號。因此, 文稿中第一道 `\chapter` 指令即爲本書第1 章。但是, 在 `\frontmatter` 與 `\mainmatter` 指令之間所下之 `\chapter` 指令, LaTeX 仍以一般章標題格式編排, 但不加入數字編號。此項特別設定讓我們可以利用 `\chapter` 指令排版序言或目錄。不過, 文稿中一旦碰上 `\mainmatter` 指令, 其後之 `\chapter` 即恢復自動編碼功能。書籍末端若欲排版索引或參考文獻, 可先加入 `\backmatter` 指令, 其後之 `\chapter` 指令也不自動編號。

頁碼編排除了計數問題之外, 還有一個問題是排版位置。一般而言, 頁

碼通常排版頁眉/頁足。第7.8將介紹如何使用 fancyhdr 巨集套件以控制頁碼之編排位置。

7.3.2 換頁

排版時, LaTeX 會自動選擇適當的地方換頁。但是, 若希望把某一段文字排在同一頁裡, 而 LaTeX 所選擇換頁的地方剛好是在這一段文字中間, 我們可以在此段文字之前下換頁指令 \newpage, 其下的整段文字將移至下一頁。我們也可以使用 \clearpage 指令, 兩者的差別在於後者除了換頁之外, 還會把尚未排出的圖表列印於本頁之後。若是使用 \newpage 指令, 圖表將移於文稿最末端。

如果 \newpage 指令恰出現於一頁之頂端, 該指令變成無效, 亦即不會再空出一頁。因此, 如果一定要空出一頁, 應使用下列指令:

```
\mbox{}\newpage
```

其中, \mbox{} 創造出一虛擬字元, 因此跳頁指令即可產生效果。

文稿若要排版成兩欄格式, 我們可以在文件類別指令中加入 twocolumn 選項。在兩欄式版面中, 換頁應使用 \cleardoublepage 指令。此一指令的功能與單欄版面中之 \clearpage 指令類似, 但下接之文字會跳至單數頁開始編排。換言之, 如果本頁的頁碼爲單數, 下一頁 (雙數) 將爲空白, 文字會出現於再下一頁中。

排版書籍時, 是否跳頁常依本頁是單頁或雙頁而定。譬如, 若每一章標題是排於單數頁, 而上一章結束於單數頁, 則下一頁 (雙數頁) 須留爲空白。反之, 若上一章是結束於雙數頁, 本章恰好接著由單數頁開始編排, 不須跳頁。將下列一行指令加於每一章之前, 即可出現以上的效果:

```
\ifodd\count0 \else \mbox{}\clearpage \fi
```

以上的 TeX 指令中, \ifodd 是判斷數字是否爲奇數; \count0 則記錄頁碼。因此以上指令之意義如下: 若本頁頁碼爲奇數, 不作任何動作; 反之, 若爲偶數, 則強迫跳一頁。指令最後之 \fi 代表定義結束。

以上指令解決了先判斷單雙頁再決定是否跳頁的問題。不過, 跳空的那一頁雖然空白, 版面正下方仍然會排出頁眉/頁足詞。如果跳空的那一頁希望全頁空白, 上面之指令須修改如下:

```
\ifodd\count0 \else \thispagestyle{empty}\mbox{}\clearpage \fi
```

定義內 `\thispagestyle{empty}` 指令之功能是設定本頁之頁眉/頁足皆爲空白, 頁碼自然也不會出現。

以上是以跳頁方法避免某段文字拆散於兩頁之中。除此之外, 我們也可以將 `\samepage` 指令加在文字段落之後。LaTeX 會試著把整段文字全部擠入本頁。頁面調整指令應在排版最後階段再加入。否則, 文稿一經修改, 又須重新調整, 徒然浪費時間。

7.4 摘要

文章正文之前可能排版摘要, 摘要可以 `abstract` 指令環境編排。LaTeX 會在摘要文字之前加上 **Abstract** 英文字。如果是中文稿, 我們須將此一英文字轉換爲中文。LaTeX 是以 `\abstractname` 指令設定摘要之標題文字。因此, 只要重新定義其內容即可解決問題, 辦法是在全文設定區加入下面指令:

```
\renewcommand{\abstractname}{\r12 摘要}
```

此一指令重新設定摘要標題爲中文「摘要」兩字, 字體採 12 點圓體字。

在 `article` 文件類別裡, 摘要將排版於標題下面; 在 `report` 裡, 摘要自成一頁, 而且不編上頁碼; `book` 文件類別裡則不能使用摘要指令。

7.5 章節標題

一般文稿通常區分章節, 章節的編排設計必須前後一致。LaTeX 提供現成的指令以編排章節; 一般而言, 這些指令已可滿足大多數人的需求。不過, 中文與英文究竟不同。譬如, 英文常使用 “Chapter 1” 作爲第一章之編號, 中文則使用「第 1 章」或「第一章」。如何編排適當的中文標題, 對於 LaTeX 使用者而言是一個挑戰, 主要原因是直接修改標題指令並不容易。幸運的是, 我們有一些功能甚佳的巨集套件可資使用。

7.5.1 章節標題之層級

首先, 我們介紹 LaTeX 排版章節標題之指令。使用這些指令排版時, 章節標題上下會留出適當空白, 小節以上之標題會自動編入號碼, 標題內之英文/數

字選用粗體字排版。文稿若設定排版頁眉/頁足, 章節標題與編號會自動出現於頁眉或頁足。此外, 若以 LaTeX 之指令排版目錄, 章節標題/編號也會自動編入目錄內。LaTeX 之章節指令如下:

```
\part（部）          \chapter（章）
\section（節）       \subsection（小節）    \subsubsection（次小節）
\paragraph（段）     \subparagraph（小段）
```

章節標題指令有其層級結構, 原則上先後順序不能倒置。例如, 某一章之內可含有好幾個節; 某一節之內又可分成幾個小節; 其下又分次小節等等。

章節標題指令與文件類別關係密切。若使用 book 文件類別排版, 以上各指令皆可使用; 不過, \part 指令可跳過不用, 直接從下一層級的 \chapter 指令開始用起。相對而言, 若是使用 article 文件類別排版短文, 文稿內不能使用 \chapter 指令。但 \part 指令可有可無; 若不使用, 章節層級結構即由 \section 開始。

圖 7.1 (頁 85) 的例子中, 各節標題是以 \section 指令編排, 小節標題則以 \subsection 排版。文稿中遇有節/小節標題指令時, 文稿將另起一行, 自動編上號碼, 選用稍大字體, 再排出標題文字; 正文內容將排於再下一行。節標題所選用之字體較小節字體大; 小節標題字體又比次小節字體大一些。請注意, 中文字體之大小與種類須以中文字體指令自行設定。若文稿甚長, 每個章節標題都須輸入中文字體指令不僅不方便, 也容易產生錯誤。7.7 節將說明如何定義整篇文稿之中文標題字體。

在 book 文件類別中, 若使用 \chapter 指令排版章標題, 章編號與標題文字將靠左, 並分上下兩行排出。下一層級的 \section 與 \subsection 指令所排版之節與小節標題, 其格式與 article 文件類別類似; 數字編號之後即接著標題文字, 正文內容將從再下一行開始。\subsection 以下之標題皆不會自動編號, 其中 \subsubsection 指令所排版之標題文字會再小一些, 也是單獨成一行靠左編排。\paragraph 指令所排版之格式在專業排版中稱爲 running-head 標題。標題文字與上一段之距離會稍大一些, 標題使用粗體字, 但大小與正文一樣。標題文字之後會留一點空白, 其後直接排版正文。\subparagraph 之格式與 \paragraph 類似, 但前者之標題會內縮 (indent); 後者之標題左邊則切齊版面左緣。

如果文稿一開始就跳過 \section 指令, 直接使用 \subsection (小節), 文稿仍可編排, 但第 1 小節的編碼將變成 **0.1**, 第 2 小節變成 **0.2**, 等等。因此, 除非有特別理由, 章節指令應按順序使用。次小節 \subsubsection 以下之標題不會自動編號, 一方面是避免編號過於複雜, 另一方面則是讓段落編排更有彈性。譬如, 排版者在某一節或小節之內可以跳過 \subsubsection 指令, 直接使用 \paragraph 指令以 runing-head 格式排版特定之段落。

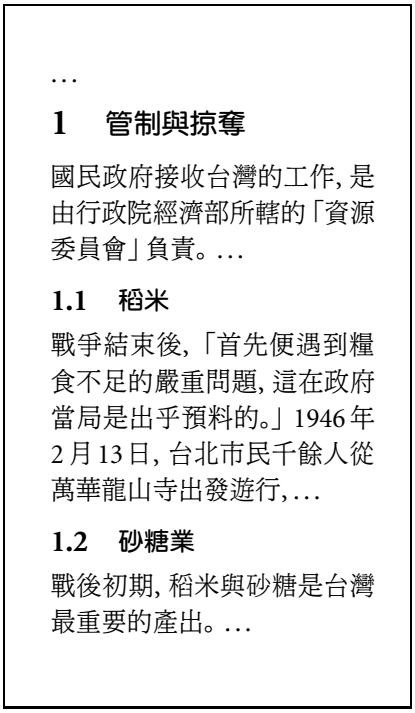
以 LaTeX 的指令排版中文時, 面臨兩個問題。第一是中/英文字體之搭配是否適宜, 第二個問題是中英文章節編號之格式不同, 以下進一步說明解決的方法。章節標題指令中, \part 指令適合於排版書籍或較長報告之標題。\part 指令之排版格式與 \chapter 類似。例如, 如果我們以下列指令排版標題: \part{On Movies}, 版面上將先排出 **Part I**, 下一行再以粗體字排版 **On Movies**。文稿內第二次使用 \part 指令時, 版面上將出現 **Part II**, 其下再排出標題文字。換言之, LaTeX 自動編上的序號是英文字的 **Part I** 與 **Part II**。如果是排版中文書, 英文序號與中文標題可能並不搭配。使用節指令時, LaTeX 也自動編上號碼, 不過是以阿拉伯數字編碼, 用於中文稿並無不妥。

如果要使用章節標題指令, 但不要自動編號, 應在章節指令之後加上 * 號, 譬如 \section* 或 \subsection*。 使用加 * 號之指令編排節標題, 雖然不再編上號碼, 但標題前後仍留出空白, 標題之數字或英文選用較大字體, 而且首段文字不內縮。使用 * 號節指令的一個問題是, 節標題不會自動編入目錄與頁眉之中。如果節標題不想編號, 但又希望編入目錄與頁眉中, 須使用 \addcontentsline 指令, 請參見第 7.6.6 節之說明。

7.5.2 設定章節標題之字體

LaTeX 之章節指令除了留空白與編號碼之外, 並選用放大的粗體字。但是, LaTeX 之字體指令無法改變中文字體。因此, 我們必須自行下指令改變中文標題的字體及大小。圖 7.5 的例子採用 article 文件類別, 中文標題則採用放大之黑體字, 但結果並不十分理想, 因爲數字是羅馬字族粗體。如果中文標題使用中黑體字, 較搭配之英文/數字應該是 sans serif 字族。我們如何改變標題之英/數字體呢? 最簡單的方法是使用下一節所介紹之 titlesec 巨集套件。不過, 此處先介紹一般性之概念。

在 LaTeX 中, 章節之編號是由章節編號計數器 (counter) 自動調整。以節編號爲例, 其數值是由 section 計數器決定。文稿一開始, section 之值爲




```
\documentclass[11pt]{article}
\begin{document}
...
\section{\rb12 管制與掠奪}
\m11
國民政府接收台灣的工作，
是由行政院經濟部所轄的
「資源委員會」負責。
...
\subsection{\rb11 稻米}
戰爭結束後，
「首先便遇到糧食不足的嚴重問題，
這在政府當局是出乎預料的。」
1946年2月13日，
台北市民千餘人從萬華龍山寺出發遊行，
...
\subsection{\rb11 砂糖業}
戰後初期，
稻米與砂糖是台灣最重要的產出。
...
```

圖 7.5: 章節指令與中文標題

0。碰到第 1 個 \section 指令時, 計數器值成爲 1。碰到第 2 個 \section 指令時, 計數器再加 1 變成 2。不過, section 計數器只是計算節之編號數字, 實際之排版指令爲 \thesection。要將節編號之字體改變爲 sans serif, 可在全文設定區加入下列指令:

```
\renewcommand{\thesection}{\textsf{\arabic{section}}}
```

如果小節編號也要作同樣改變呢? 由以上的例子可知, 第 1 小節之編號爲 1.1, 第 2 小節之編號爲 1.2。因此, 小節之編號用上兩個數字: 節編號與小節編號。要改變小節編號之字體, 我們不能只變更小節編號部分, 必須同時調整兩者才能得到正確的結果:

```
\renewcommand{\thesection}{\textsf{\arabic{section}}}
\renewcommand{\thesubsection}%
  {\thesection.{\textsf{\arabic{subsection}}}}
```

第 1 行指令之作用與上例完全相同, 第 2 行指令定義小節數字編號之排版方式。我們先以 \thesection 指令排入節編號, 其後緊接一英文句點, 接著再排版小節編號。

根據以上原理, 節編號可以作進一步的變化。LaTeX 的指令 \alph 可以將阿拉伯數字變成對應的小寫英文字母。因此, 上面第 2 行指令若改爲:

```
\renewcommand{\thesubsection}{\thesection.{\alph{subsection}}}
```

小節編號將變成 1.a, 1.b, ... 等。另外一個指令 \Alph 是將阿拉伯數字變成對應的大寫英文字母。如果第 2 行維持不變, 但第 1 行指令之 \arabic 改爲 \Alph, 節編號將變成 A, B, C, ...; 第 1 節之下的小節編號將變成 A.1, A.2, ... 等等。

LaTeX 改變數字排版格式之指令計有下列 5 個:

\arabic	阿拉伯數字,	\alph	小寫英文字母,
\roman	小寫羅馬字,	\Alph	大寫英文字母,
\Roman	大寫羅馬字。		

以上指令可以和表 7.1 改變頁碼字體之指令對照。

以上例子是以 article 文件類別爲例。如果是 book 或 report 類別, 編號是從章開始。因此, 第 1 章第 1 節之編號爲 1.1, 第 2 節編號爲 1.2; 而第 1 章第 1 節第 1 小節之編號則爲 1.1.1; 餘此類推。

7.6 titlesec 巨集套件

要改變標題之排版方式, 最簡單的方法是使用現成的巨集套件。本節將介紹 titlesec (2.3 版) 巨集套件, 這是由 Javier Bezos 所寫, 提供簡易與進階兩種指令方式以變更章節標題。如果你只是要改變標題字體與大小, 或者只是要改變標題之排版位置, 簡易指令即可滿足需求。如果要進一步更改標題設計, 則須使用進階指令。

7.6.1 簡易指令

欲使用簡易指令, 僅須在引用巨集套件時直接加入控制格式之選項即可, 表 7.2 列出簡易指令之選項。譬如, 若標題文字要改爲居中編排, 僅須在全文設

表 7.2: titlesec 巨集套件簡易指令

\usepackage[*options*]{titlesec}

- 字體: `rm sf tt md bf up it sl sc`
- 字級: `big medium small tiny`
- 標題位置: `center raggedright raggedleft`
- 間距: `compact`

定區加入下列兩行指令:

```
\usepackage[center]{titlesec}
\renewcommand{\chaptername}{}
```

第 1 行指令中 `center` 選項之作用是將章節標題全部改變成居中排版。若文稿是採用 `book` 文件類別, 使用 `\chapter` 指令編排章標題時, 第 1 章標題文字之上方將自動加入 **Chapter 1**, 第 2 章自動加入 **Chapter 2** 等等。

在LaTeX 中, `\chaptername` 指令代表 **Chapter** 英文字, 本例之第 2 行指令即將 `\chaptername` 設爲空白, 因此章標題之 **Chapter** 一字即不出現。排版之後, 第 1 章標題分上下兩行, 第 1 行爲 **1**, 第2行爲標題文字。以上指令雖可去掉 **Chapter** 一字, 但版面上方卻會顯得空盪盪。必要時, 我們可以使用 `\vspace*` 指令將標題往上移一些。

爲方便參考, 我們將 titlesec 簡易指令之選項全部列於表 7.2。控制標題位置之選項除了 `center` 之外, 尚有 `raggedleft` (標題靠右) 與 `raggedright` (標題靠左)。若選項有兩個以上, 須以逗號分隔。字體/字級之指令選項僅對英數字有效; 中文字體與字級之選擇須以中文字體指令另行設定。LaTeX 章節標題之英數字是以粗黑體排版。如果中文標題是選用圓體或粗黑體, 則英文或數字改用 sans serif 字體搭配較適當。此時, 可使用下列指令:

```
\usepackage[sf,small]{titlesec}
```

其中, `small` 選項指示使用較小一點的字體; 內設值爲 `big`。此外, 選項中若加入 `compact`, 則標題文字與上下文之間距會縮小一些。

傳統洋文書常在章節數字編號前後加上裝飾符號, titlesec 巨集套件所提供之 `\titlelabel` 指令可用以排版裝飾符號。LaTeX 是以 `\thetitle` 指令

代表整個章節之數字編號, 若要在章節編號之前加上 § 符號, 僅須在全文設定區加入底下一行指令即可:

```
\titlelabel{\S\ \thetitle\quad}
```

此行指令中, \S 指令的作用是加入裝飾符號 § 於 \thetitle 之前, \quad 指令是用以加大章節編號與標題文字之間距。排版後, 7.5 節之標題編號將變成: § **7.5**。以上指令雖然方便, 但它會更動文稿中全部章節之標題。除非文稿結構單純, 否則不一定適用。

7.6.2 進階指令

若上一小節之簡易指令不能滿足需求, 則須使用進階指令。進階指令主要有 \titleformat 與 \titlespacing 兩項, 前者設計章節標題式樣, 後者控制間距。此外, 還有一些配合使用之控制指令。底下以幾個例子說明使用方法, 這些例子係參考巨集套件之說明檔而來。

爲求文稿風格統一, 各章節標題應採用相同設計, 因此, 改變章節標題之指令通常是放在全文設定區。不過, 必要時 \titleformat 指令也可以用在特定章節標題之前。章節標題分兩部分: 第一部分爲章節編號數字, 第二部分爲標題文字。因此, \titleformat 指令選項一部分用於控制數字編號, 一部分用於控制標題文字; 表 7.3 列出進階指令之格式, 我們將以幾個例子說明其用法。

假設我們以 book 文件類別排版, 每一章之標題要居中編排, 章編號希望以「第 2 章」格式出現; 節與小節之標題沿用原有格式, 但使用 sans serif 字體。下列指令可以排版出以上的設計:

```
\usepackage[sf]{titlesec}
\titleformat{\chapter}[display]{\centering\LARGE\sf}
   {\rb20 第\ \thechapter\ 章}{0.2cm}{}
```

第 1 行指令爲 titlesec 巨集套件之簡易指令。除了選用 sans serif 字體外, 其餘皆依 LaTeX 原有格式編排。第2-3行是 \titleformat 指令加上選項, 因爲選項太長, 故拆爲兩行。請注意, 依 LaTeX 之規範, 指令拆爲兩行時, 第1行末端應加上 % 符號。不過, titlesec 巨集指令之語法特別, 指令拆爲兩行以上時可以不加上 % 符號。

表 7.3: titlesec 巨集套件進階指令

\titleformat{*command*}[*shape*]{*format*}{*label*}{*sep*}{*before*}[*after*]

- *command*: 欲重新設計之指令, 如 \chapter, \section, \subsection 等。
- *shape*: 設定標題編排方式。可用選項如下: hang, block, display, leftmargin, rightmargin, runin, drop, wrap, frame。
- *format*: 此項內之指令將同時控制章節數字編號與標題文字之排版方式。
- *label*: label 是指章節編號及附加符號。此項內之指令是用以控制 label 之排版。
- *sep*: 控制章節數字編號與文字標題之間距。
- *before*: 用於控制章節標題之排版。
- *after*: 附加於標題文字後之控制指令。

\titlespacing{*command*}{*left*}{*beforesep*}{*aftersep*}[*right*]

- *command*: 欲重新設計之指令, 如 \chapter, \section 等。
- *left*: 章節標題「左邊」之空白, 因 *shape* 選項之不同而異。若選用 runin, 此項設定是指內縮 (indent) 之大小。
- *beforeskip*: 排版章節標題之前所留之空白。
- *afterskip*: 章節標題之後所留之空白。
- *right*: 選用 hang, display, block 或 display 時, 可進一步加此選項於標題末端。

其他控制指令

- \titlerule[*height*]: 畫橫跨正文寬度之線條; *height* 設定粗細, 若不設定, 內定值爲 0.4pt。此指令通常置於 *shape* 選項內。
- \titleline[*height*]{*text*}: 同上一指令; 但線條替代以自行選定 *text*, 如細點。

說明: \titlespacing 指令若採 * 號形式, 變成 \titlespacing*, 則標題後之段落不內縮 (indent)。反之, 若指令不加 * 號, 段落首行文字將內縮。

巨集套件 titlesec 之指令相當複雜, 爲便於參考, 我們將較重要者列於表 7.3。第 1 選項 \chapter 指示欲更改章之標題。如果要更改節標題, 應填入 \section, 餘此類推。第 2 選項稱爲 *shape*, 設定章節編號與標題文字之排版方式。本例中使用 display 選項, 這是 LaTeX 原來排版章標題之格式: 編號與標題文字分兩行排出。除了 display 之外, 其他可能的設定包括 runin, hang 等, 以下將陸續介紹。若不選用 *shape* 選項, 內設值爲 hang, 這是 LaTeX 排版節標題之格式, 亦即標題文字接於數字編號之後, 排爲一行。

第3選項爲 *format*, 大括號內所置放之指令對於編號與標題文字都會產生效果。本例使用 3 個指令, \centering 使標題與編號居中編排, \LARGE 選用較大字體, \sf 則選用 sans serif 字體。第 4 選項 *label* 是用於排版章節標題編號。本例選用粗圓體把第 2 章標題編號排版爲「**第 2 章**」, 字體大小爲20點。第 5 選項 *sep* 控制編號與標題文字之間距。在 display 格式下, 此間距表示上下兩行之距離; 若是 hang 或者 runin 等格式, 編號與文字排於同一行, 此選項即設定水平間距。

第 6 選項 *before* 是用於控制標題文字之排版。換言之, 此部分之指令僅施用於標題文字, 編號不受影響。此選項可能含一個或多個指令, 最後一個指令可帶有一個參數, 此參數即代表標題之文字。底下將以例子說明使用方法。最後一個選項 *after* 之內容將附加於標題文字後面。譬如, 要在章標題文字之後加上一短線, 可利用此一選項爲之。本例中, 6-7 選項內容皆爲空白。排版結果每一章之標題將居中, 編號與標題文字之間距爲 0.2cm。

有人認爲中文標題內之數字應以國字編排。本例中, 若第 4 選項之內容改爲:

```
{\rb20 第一章}
```

則標題文字中之編號將改爲國字數字。但因爲國字之數字編號不會自動跳加, 第 2 章之標題編號仍然排爲「**第一章**」。如果不怕麻煩, 一個解決的方法是在每一章 \chapter 指令之前重新定義 \titleformat 指令, 即可排出正確之國字編號。

除了下指令的困難之外, 以國字編號還有一個困難的問題。中文數字不容易表現章節層級, 若一本書區分章、節、小節, 以國字編章節號碼其實很不容易排得美觀。

7.6.3 彩色或灰階色標題文字

上面例子中, \titleformat 指令第 6 選項空白。此選項主要是用於排版標題文字。在 LaTeX 中, 我們可以使用 color 巨集套件之 \textcolor 指令將文字段落變成灰階色。譬如, 先定義 light 爲灰階度 0.60 之灰階色:

```
\usepackage{color}
\definecolor{light}{0.60}
```

之後, 以下指令即可將「章節標題」四個字變成灰階色。

```
\textcolor{light}{章節標題}
```

色彩/灰階色之定義方法請參見第 12 章之說明。

不過, 以上之指令事實上並未完全解決問題, 原因是標題文字須直接置於 \titleformat 指令的大括號內; 但是每一章節之標題文字並不相同。titlesec 巨集套件提供的解決辦法如下: 定義一新巨集指令, 其內可自動填入章節標題文字, 再將此巨集指令引入 \titleformat 指令選項內。實際作法如下: 首先, 在全文設定區定義指令 \mytitle:

```
\newcommand{\mytitle}[1]{\textcolor{light}{#1}}
```

接下來再將 \mytitle 指令填入 \titleformat 指令第 6 選項內。再以上一個例子來說明, \titleformat 指令將變成:

```
\titleformat{\chapter}[display]{\centering\LARGE\sf}
   {\rb20 第\ \thechapter\ 章}{0.2cm}{\mytitle}
```

排版時, 每一章之標題文字將自動代入, 以灰階色排版。

第 10 章說明表格排版時, 將介紹一巨集指令 \cspp, 其功能是將中文字間距拉大爲選定之距離; 請見圖 10.9 (頁 195) 之說明。利用此一巨集指令, 我們可將章節標題之中文字距拉大。欲創造此一效果, 首先定義 \cspp 巨集指令, 並將 \mytitle 修改如下:

```
\newcommand{\cspp}[1]{\let\zori=\z
  \def\zx{\hskip 2mm plus0.2pt minus0.1pt}
  \let\z=\zx \mbox{#1}\let\z=\zori}
\newcommand{\mytitle}[1]{\textcolor{light}{\cspp{#1}}}
```

再與上述之 \titleformat 指令合併使用即可。

以上指令僅將章標題文字轉爲灰階色, 但章編號部分並不受影響。如果章編號也要以同一灰階色排版, 必須另作處理。\titleformat 排版章節編號之指令是置於第 4 選項 *label* 內。因此, 若將第 4 選項之內容改爲:

```
\textcolor{light}{\rb20 第\ \thechapter\ 章}
```

即可得到滿意的結果。

7.6.4 設計章節標題

在標準的 LaTeX 格式中, 章編號與標題文字是分兩行排版, 此一格式稱爲 display。下一個例子則採用 hang 格式, 編號與標題文字將排於同一行。這是 LaTeX 排版節編號與標題文字之標準方式, 指令如下:

```
\titleformat{\chapter}[hang]{\hspace*{.2\textwidth}%
  \fontsize{45}{20pt}\selectfont\bf}
  {\thechapter}{4mm}{\mytitle}
```

第 3 選項排版章編號時, 先右移 0.2\textwidth 距離, 再選用 45pt 之粗體字。第 4 選項內僅有 \thechapter 一項指令, 表示除了阿拉伯數字編號之外, 不加入其他文字。第 5 選項將編號與標題文字之間距控制爲 4mm。排版標題文字之指令如下:

```
\newcommand{\mytitle}[1]{\raisebox{2mm}%
  {\parbox{.6\textwidth}{#1}}}
```

標題文字置於 \parbox 內, 並上移 2mm。使用 \parbox 指令的目的是爲了處理多行之標題。因爲文字置於 \parbox 內, 不管標題文字有多少行, 第 2 行開始將與上一行切齊, 不至於凸出至版面左緣。

利用以上指令, 若某書第 2 章之標題是以下列指令輸入:

```
\chapter{\rb15 國際貿易與\\ 國際借貸市場}
```

排版之後章標題如下:

**2 國際貿易與
國際借貸市場**

請注意, 鍵入標題文字時須自行拆爲兩行。

使用 LaTeX 指令排版章節標題時, 章節會自動編入號碼。以章標題爲例, 標題編號是以 \thechapter 代表, 其來源是 chapter 計數器。具體言之, 若本章爲第 7 章, LaTeX 自 chapter 計數器中取出 "7", 設定以阿拉伯數字格式編排。若章編號要改以大寫羅馬字, 僅須在 \titleformat 指令之前加入下列指令即可:

```
\renewcommand{\thechapter}{\Roman{chapter}}
```

其他控制數字排版格式之指令請見表 7.3。

以上是章標題的例子。底下我們舉幾個例子說明節標題之設計, 這些指令大多也可以應用於排版章或小節標題。第一個例子是 running-head 標題, 指令如下:

```
\usepackage{titlesec}
\titleformat{\section}[runin]{\normalfont\sffamily}%
  {\S\ \thesection}{.5em}{}[---]
\titlespacing{\section}%
  {0pt}{1.5ex plus .1ex minus .2ex}{\wordsep}
```

如上所述, \titleformat 指令共有 7 個選項, 其中第 2 與第 7 選項可有可無。第 1 選項爲 \section, 表示欲更改節標題。第 2 選項設定標題之排版方式, 此選項若不加入, titlesec 巨集套件將自動代入 [hang] 選項; 編號與標題文字將獨立排爲一行。本例使用 [runin] 設定, 在洋文書排版術語中, 這稱爲 running-head, 表示節標題文字並不單獨排成一行, 而是直接置於本節第一行文字之前端。第 3 個選項內之指令同時控制節數字編號與標題文字。本例中, 此選項內有兩道指令, \normalfont 先將英文/數字回復標準字體, \sffamily 則選用 sans serif 字體。

第 4 選項是排版章節數字編號之指令。本例中之指令爲:

```
{\S\ \thesection}
```

其作用是在節數字編號之前加上裝飾符號 §, 其後空一格, 最後再加上節編號 \thesection。以本書 7.5 節爲例, 數字編號將排爲: § **7.5**。第 5 選項控制節數字編號與文字標題之間距。若數字編號與標題文字排於同一行, 間距表示水平距離, 本例間距爲 {0.5em}。第 6 選項內是用於控制標題文字。譬

如, 若要在標題文字之前或上方加入任何線條或符號, 指令應置於此選項內。本例中, 此選項空白。最後一選項內爲標題文字後之控制指令, 此例之指令爲 [---], 因此標題文字之後將加上一短橫線。以本書 7.5 節之標題爲例, 若原排版指令爲 \section{\r12 章節標題}, 排版結果如下:

> § 7.5 章節標題— 一般文稿通常區分章節, 章節的編排設計必須前後一致。LaTeX 提供現成的指令以編排章節; 一般而言, 這些指令已可滿足大多數人的需求。...

除了 \titleformat 指令外, 本例中另使用 \titlespacing 指令, 其功能是控制間距。如表 7.3 所示, 此一指令有 5 個選項, 第 1 選項也是選定所要更改的章節標題。第 2 選項設定「左邊空白」(left space);「左邊空白」的意義因 \titleformat 第 2 選項之不同而異。本例中以 [runin] 排版節標題, 此時左邊空白指的是標題前之內縮 (indent) 空白。若 *shape* 爲 wrap 或 drop,「左邊空白」設定標題文字寬度。本例子設定爲 0pt, 因此標題不內縮。

第 3 選項設定標題上方之間距。第 4 選項設定標題文字與下接正文文字之間距。若標題文字單獨爲一行, 此選項爲垂直間距; 若標題文字之後直接排版正文, 此一選項控制水平間距。本例中設定爲 \wordsep, 代表英文單字間之正常間距。第 5 選項爲附加於標題後之文字或符號, 本例中從缺。

依 LaTeX 之原始設計, 章標題之數字編號與文字標題分兩行編排; 節標題則是兩者排於同一行。下一個例子是設計把節標題分爲兩行, 第1行排版數字編號, 第 2 行爲標題文字:

```
\titleformat{\section}[display]{\normalfont\sffamily}%
  {\bb12 第 \large\thesection ~節}{0.5em}%
  {\vspace*{-15pt}\hrulefill\\[.2ex]}
```

第 2 選項爲 [display], 指示將節編號與標題文字當作一個段落編排, 因此可排版爲兩行或多行。這是 LaTeX 章標題之原始標準格式。第4 選項內之 \thesection 指令是節編號數字, 前面加上 \large 指令選用較大英數字體; 前後又加入「第」與「節」兩個中文字。

最後的第 6 選項內之指令是用於排版標題文字。此一例子中, 利用 TeX 之 \hrulefill 指令畫一橫線。其前之 \vspace* 指令用以控制橫線與上面

文字之間距; 其後之換行指令亦加上控制指令以設定行距。再以本書 7.5 節爲例, 排版後節標題將爲:

第 7.5 節

章節標題

一般文稿通常區分章節, 章節的編排設計必須前後一致。LaTeX 提供現成的指令以編排章節; 一般而言, 這些指令已可滿足大多數人的需求。...

本例並未使用 `\titlespacing` 指令, 因此所有間距將使用內定值。

7.6.5 嵌入標題

第 3 個例子係西文排版中所謂的 drop 標題, 本書稱之爲「嵌入標題」。「嵌入標題」之設定通常用於節或小節, 標題並非單獨成一行, 而是直接置於節段落前端之左沿, 首段文字的前兩三行則內縮一些。一個簡單的例子如下:

7.5
章節標題

一般文稿通常區分章節, 章節的編排設計必須前後一致。LaTeX 提供現成的指令以編排章節; 一般而言, 這些指令已可滿足大多數人的需求。...

設計標題之指令如下:

```
\titleformat{\section}[drop]%
  {\normalfont\fontseries{b}\selectfont\filright}%
  {\thesection}{.6em}{\mbox{}\\[5pt]}
\titlespacing{\section}%
  {5pc}{1.5ex plus .1ex minus .2ex}{1.5pc}
```

第 2 選項 `[drop]` 指示選用嵌入標題。第 3 選項最後一個指令爲 `\filright`, 作用和 LaTeX 之 `\raggedright` (靠左編排) 類似, 但取消英文單字之音節斷字 (hyphenation) 功能。第 6 選項內指令爲 `\mbox{}\\[5pt]`, 作用是換行, 並加大行距 5pt。因此, 標題文字將排於編號之下。如果只有換行指令, 排

版時將出現錯誤訊息, 因此我們先加入 \mbox{}, 目的是填入一不占空間之虛擬文字。

使用 [drop] 選項時, \titlespacing 指令第 2 選項即設定標題文字所占空間。此例中設定之空白爲 5pc。因爲每一節之標題長度不一, 若設定之長度小於標題文字之長度, 標題將自動排版爲兩行甚或三行。

與 drop 格式類似的是 wrap。此一格式會自動計算並調整標題寬度。若標題寬度小於給定之長度, 巨集指令自動把標題空間縮小; 反之, 若大於給定之長度, 標題也是自動折爲兩行或多行。一個簡單的例子如下:

```
\titleformat{\section}[wrap]
  {\normalfont\fontseries{b}\selectfont\filright}
  {\thesection}{.6em}{}
  \titlespacing{\section}
  {8pc}{1.5ex plus .1ex minus .2ex}{1.5pc}
```

以上指令選用 wrap 格式, \titlespacing 指令第 2 選項設定標題寬度爲 8pc。輸入標題時, 我們自行將文字折爲兩行如下:

```
\subsection{\r12 章節標題\\ 排版方法}
```

排版時, LaTeX 自動計算標題之寬度, 並在右邊留下一點空白:

7.5 章節標題 排版方法 一般文稿通常區分章節, 章節的編排設計必須前後一致。LaTeX 提供現成的指令以編排章節; 一般而言, 這些指令已可滿足大多數人的需求。...

以上所介紹的是 titlesec 巨集套件 2.3 版, 如果你使用 2.2 版或更早版本, wrap 格式可能無法正確運作。此外, 爲了避免和 fancyhdr 巨集套件衝突, 舊版曾提供 nops 選項, 但在新版中此一指令已取消。

7.6.6 序言與索引等之標題

排版書籍時, 正文每一章之標題是以相同格式編排; 但目錄、序文、附錄、索引等之標題可能採用不同格式。譬如, 正文每一章之標題可能編上號碼, 但序文、索引之標題通常不編號, 因此其標題之編排需另作處理。

以目錄爲例, 若使用 \tableofcontents 指令排版目錄, 則其標題格式與 \chapter* 指令相同, * 號表示不編號。如果我們使用 \titleformat 指令設計章標題之編排, 則目錄頁之標題可能須另作調整。不過, 一篇文稿內, \titleformat 指令可使用多次。如有必要, 我們可另定義一格式, 專用於排版目錄或序文之標題。

序文之標題通常也不加上編號, 因此我們須以 \chapter* 指令編排。不過, 此一指令有一個副作用, 其所排版的標題不會自動納入目錄中。欲解決此問題, 我們可使用下列指令:

```
\addcontentsline{toc}{sec-name}{text}
```

以上指令中, *text* 爲欲出現在目錄中之文字。*sec-name* 爲章節指令名稱, 但前端不加上反斜線, 例如 chapter, section。如果我們以 \chapter* 指令排版序言標題, 則指令之前加上:

```
\addcontentsline{toc}{chapter}{\bb12 序文}
```

排版後, 「序文」兩字將出現於目錄內, 並以粗黑體 12 點字體編排。

除了出現於目錄與否的問題之外, 目錄頁與序文頁之頁眉/頁足詞通常也須特別處理。請見下一節之說明。

7.7 設定中文標題之字體

使用 LaTeX 編排章節時, 標題內之英數文字會自動選用較大的粗黑字體。對某些人來說, 章節標題不須另行設計是 LaTeX 排版系統的優點。但對於想自行設計章節標題的人來說, 固定格式反而是缺點。上一節所介紹的 titlesec 巨集套件讓更改文稿標題之設計變得很容易; 不過, 其中仍然有一個不方便之處, 那就是中文標題須在每一個章節標題處下指令更改。

爲了讓使用者能一舉設定章節標題之中文字體, cwTeX 提供一簡單的巨集指令: \ctxfdef。舉例言之, 若文稿內使用 \section 與 \subsection 指令排版節與小節標題。假設節標題之中文字選用 \bb14 字體; 小節標題選用 \bb12 字體, 則在全文設定區加入下列兩行指令即可:

```
\ctxfdef{\section}[\m11]{\bb14}
\ctxfdef{\subsection}[\m11]{\bb12}
```

第 1 行指令設定節標題之中文字以 14 點粗黑體字排版, 中括號內之 \m11 指令設定章節標題文字排版於頁眉/頁足或目錄處之字體。(請見下一節之說明。) 若不排版頁眉或目錄, 中括號選項可以省略。同理, 第 2 行標題設定小節標題之中文字體。

經過以上設定, 排版本節標題之指令可簡化爲:

```
\section[設定中文標題之字體]{設定中文標題之字體}
```

請注意, 中文標題須鍵入兩次, 中括號內者是用於排版頁眉與目錄, 大括號內則用於排版實際標題。定義 \ctxfdef 指令之後, \section 指令前後或者指令大括號內不須再加入中文字體指令。如果某一節是以 \section* 指令排版, 則 \ctxfdef 所定義之中文字體仍然有效。若文稿前端以 \ctxfdef 指令設定中文字體, 而文稿內某特定章節欲改用不同字體, 可在該章節指令處再加入中文字體指令。例如, 若 7.5 節欲改用楷體 14 點字體, 指令爲:

```
\section[章節標題]{\k14 章節標題}
```

除了章節標題之中文字之外, 我們也可以設定註解之中文字體。譬如, 以下之指令將文稿註解內之中文字全部選用明體 10 點字:

```
\ctxfdef{\footnote}{\m10}
```

具體言之, 指令之後大括號內僅能接受下列 LaTeX 指令:

\part	\chapter	\section
\subsection	\subsubsection	\paragraph
\subparagraph	\footnote	\caption

若加入其他的排版指令, 執行時將出現錯誤。\ctxfdef 字體設定指令是在 cwtex 11.2 版才開始提供。若你仍使用舊版, 請更新版本。

7.8 編排頁眉與頁足

LaTeX 提供一些指令排版頁眉與頁足, 但功能較簡單。本節將介紹功能較強之 fancyhdr 巨集套件, 作者爲 Piet van Oostrum。請注意, 此一巨集套件舊版名稱爲 fancyheadings, 2.0 版開始改用新名字。此巨集套件功能完整, 本節僅介紹較常用之指令, 其他細節請參考該巨集套件所附說明檔。

LaTeX 提供 \pagestyle 及以下選項以排版頁眉/頁足:

empty	頁眉與頁足全部空白,
plain	頁碼置於頁足正中央, 頁眉空白,
headings	頁碼與章節標題等資訊自動排版於頁眉, 頁足空白,
myheadings	與 headings 類似, 但章節標題的排版可以自行控制。

若不加任何指令, LaTeX 將選用 plain 格式, 頁碼自動加於每頁下方, 頁眉爲空白。

如果要控制某一頁之頁碼, 應使用 \thispagestyle 指令。譬如, 依排版規範, 書籍每一章開頭頁之頁眉與頁足應該是空白的。欲作此設定, 我們可在 \chapter 指令之後加入下面一行指令:

```
\thispagestyle{empty}
```

相反的, 學術期刊論文開頭頁之頁足通常會刊出版權、出版單位等資訊。此時, 我們也可以另行定義頁眉/頁足格式。

LaTeX 之頁眉/頁足指令雖然可以重新設定, 但彈性較小, 使用上也不方便。若格式複雜, 應考慮使用 fancyhdr 巨集套件。在介紹指令之前, 我們先說明幾個相關的概念。

圖 7.6 說明版面上頁眉與頁足詞之位置。在洋文書的排版中, 以 book 文件類別所排版的書籍, 左右頁之版面格式不同。通常, 左邊頁碼爲偶數 (even), 右邊爲奇數 (odd)。圖中之 LH-even 代表偶數頁左邊頁眉詞, 其中 LH 代表左邊頁眉, even 代表偶數頁。同理, CF-odd 代表奇數頁中央頁足詞。若以短文類別排版文稿, 但不選用 [twoside], 則每一頁皆視爲右頁。下一小節先說明左右頁版面相同之情況。

7.8.1 單雙頁相同版面

上面說明, LaTeX 之 \pagestyle 指令提供 plain 等 4 種格式, fancyhdr 巨集套件則提供另一種頁眉/頁足格式, 稱爲 fancy, 其內容接近 headings, 但很容易重新設定。欲使用 fancyhdr 巨集套件, 首先在全文設定區鍵入下列 2 行指令:

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

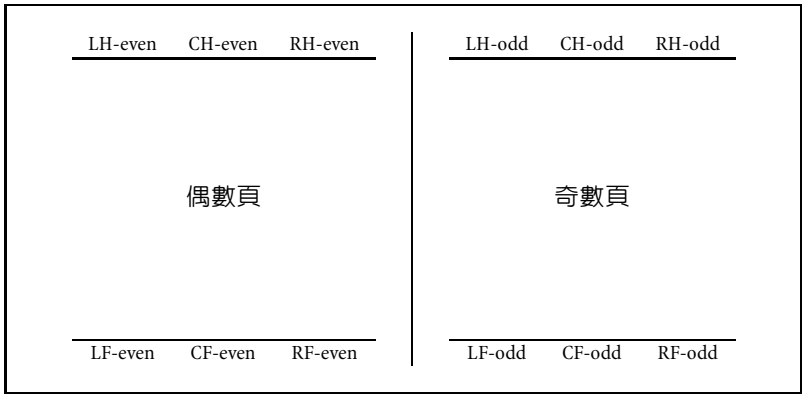


圖 7.6: 頁眉與頁足版面圖示

第 2 行指令指示以 fancy 格式排版頁首/頁足, 因此, 接下來我們須輸入定義 fancy 格式之指令。

圖 7.7 是短文 article 文件類別的例子, 其左右頁格式完全相同。版面上頁眉與頁足共有 6 個地方可供排版。假設小丸子要向亂馬市長報告交通改善之績效, 報告之左上角排版報告對象, 右上角排版題目, 左下角排版作者, 頁碼則排版於頁面下方中央。定義格式之指令可置於文稿任何地方, 不過, 若整篇文稿之格式都相同, 定義指令可置於全文設定區。本例之定義指令如下:

```
\lhead{\r11 致: 亂馬市長}
\chead{}
\rhead{\r11〈交通改善績效報告〉}
\lfoot{\r11 報告人: 小丸子}
\cfoot{\thepage}
\rfoot{}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0pt}
```

本例中, \lhead 指令用於排版頁眉左欄, \rhead 用於排版右欄, \lfoot 則用於排版頁足左欄, 餘此類推。\chead{} 指令表示頁首中央爲空白; \cfoot 指令內之 \thepage 指令用於排版頁碼。

上述指令倒數第 2 行之 \headrulewidth 是用於在頁眉畫一粗細 0.4pt

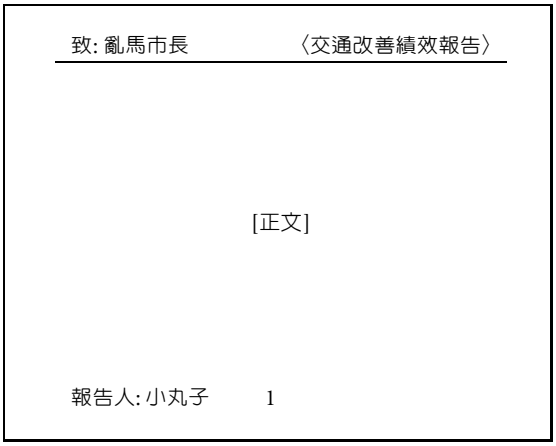


圖 7.7: fancyhdr 巨集套件應用例子

之橫線; 最後一行之 \footrulewidth 定爲 0pt, 因此頁足詞沒有長線。本例中, \rfoot 爲空白; 如果要在頁足右欄排版當天日期, \rfoot 指令內容應改爲:

```
\rfoot{\today}
```

以上所定義之 fancy 格式用於排版文稿中大部分版面之頁眉/頁足; 但不是每一頁, 因爲某些頁面之頁首/頁足須使用特別的格式。譬如, 標題頁通常不排版頁眉/頁足詞。如果標題頁只要頁碼, 其餘的資訊都不出現, 僅須在 \begin{document} 指令之後加入下一行指令即可:

```
\thispagestyle{plain}
```

不過, 若文稿第 1 頁已使用 \maketitle 指令, 以上之指令事實上可以省略, 因爲 LaTeX 會自動在 \maketitle 指令之後加入上述指令。

如果標題頁連頁碼都不要, 應使用 empty 格式:

```
\thispagestyle{empty}
```

一般正文的頁面是以 fancy 格式設計。若標題頁版面之頁眉/頁足要與正文其他各頁相同, 則標題頁可以下列指令選用相同格式:

```
\thispagestyle{fancy}
```

請特別注意, 若文稿使用 \textwidth 指令設定正文寬度, 該項指令應置於 fancyhdr 巨集套件設定指令之前, 否則頁眉詞之位置與正文文字方塊可能無法對齊。

7.8.2 單雙頁版面不同

若單頁與雙頁之版面格式不同, 則單雙頁版面都須定義。最基本的兩項設定指令如下:

```
\fancyhead[RO]{...}
\fancyfoot[CE]{...}
```

第一行指令用於設定頁眉詞之格式,第二行指令則用於設定頁足詞之格式。方括號內須填入所欲設定之欄位。本例中, RO 代表奇數頁 (odd page) 右欄, CE 代表偶數頁 (even page) 中間欄。同理 LE 爲偶數頁左欄, LO 則代表偶數頁左欄。

底下的例子比圖 7.7 更複雜一些。開始設定之前, 我們先清除各欄位原有之內容, 這是第 1-2 行指令的作用; 第 3 行開始再填入各設定文字。假設偶數頁頁眉左欄 (LE) 與奇數頁頁眉右欄 (RO) 要排爲「致: 亂馬市長」, 下列指令第 3 行即作此項定義。同理, 4-5 行指令之作用分別爲: 偶數頁頁眉右欄與奇數頁頁眉左欄爲〈交通改善績效報告〉; 偶數頁頁足左欄與奇數頁頁足右欄爲「報告人: 小丸子」。單雙數頁之頁足之中間欄皆爲頁碼; 最後兩行指令則在頁眉下沿及頁足上沿各畫出 0.4pt 粗細之直線。

```
\fancyhead{}
\fancyfoot{}
\fancyhead[RO,LE]{\r11 致: 亂馬市長}
\fancyhead[RE,LO]{\r11〈交通改善績效報告〉}
\fancyfoot[RO,LE]{\r11 報告人: 小丸子}
\fancyfoot[CO,CE]{\thepage}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
```

前面曾說明, 我們可使用 \thispagestyle{plain} 指令讓頁眉空白, 頁碼排於版面下方中央。使用 book 文件類別時, 每一章開頭之 \chapter 指令出現處, LaTeX 自動加入上述指令, 因此我們不須再自行輸入。雖然 LaTeX

會自動將每一章之首頁以 plain 格式編排, 但其格式可能不是我們所想要的。譬如, 學術期刊內每一篇論文之首頁的正下方通常會排出版權聲明。如果我們是以 book 文件類別排版整本期刊, 則每一章首頁之格式必須調整。fancyhdr 巨集套件提供 \fancypagestyle 指令, 其功能即是重新設計 plain 之格式。

若要把首頁之頁足中央改爲版權聲明, 我們須重新定義 plain 之格式。若使用 fancyhdr 巨集套件, 僅須在全文設定區輸入下列指令即可:

```
\fancypagestyle{plain}{%
\fancyhf{}    % clear all header and footer fields
\fancyfoot[C]{\copyright{} \textit{TER}}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}}
```

第 1 行指令表示要重新定義 plain 格式, 第 2 行指令清除頁眉與頁足原有之內容, 使之變爲空白; 第 3 行指令設定頁足中央排版出下列標誌:

© *TER*

第 4-5 行設定頁眉/頁足都不要橫線。

7.8.3 頁眉/頁足之章節標題

書籍排版常把章節標題置於頁眉或頁足, 一般洋文書的習慣是把章標題排於雙數頁之頁眉或頁足, 節標題排於單數頁。短文 (article) 文件類別不能使用 \chapter 指令, 但若引用 twoside 選項, 則節標題通常排版於雙數頁之頁眉或頁足, 小節標題則排於單數頁。

LaTeX 會自動記錄章節標題, 因此要將章節標題排於頁眉並不困難。最直接的方法是使用 \leftmark 與 \rightmark。在 book 文件類別下, 前一指令記錄章標題; 後一指令記錄單數頁之資訊。相對的, 在 article 文件類別下, \leftmark 存錄節標題, \rightmark 則記錄小節標題。 當頁碼增加, 由第 1 節變爲第 2 節, 或由第 3 章變成第 4 章時, 以上兩項指令所記錄之內容也隨著更新。

以 book 文件類別排版書籍時, 若偶數頁頁眉之左欄要排版章標題, 奇數頁右欄要排版節標題, 而頁碼要排在偶數頁頁眉之右欄與奇數頁頁眉之左欄, 則 fancy 格式可定義如下:

```
\fancyhead{}
\fancyfoot{}
\fancyhead[LE]{\leftmark}
\fancyhead[RO]{\rightmark}
\fancyhead[RE,LO]{\thepage}
```

加入以上指令之後, 版面上方會出現一條橫線。這是 fancyhdr 巨集套件之原始設定。

在 book 文件類別下引用 fancyhdr 巨集套件時, 若僅選取 fancy 格式而不加入任何設定指令, 內定之格式如下:

```
\fancyhead[LE,RO]{\slshape \rightmark}
\fancyhead[LO,RE]{\slshape \leftmark}
\fancyfoot[C]{\thepage}
\headrulewidth=0.4pt
\footrulewidth=0pt
```

第 1 行指令設定偶數頁頁眉左欄與奇數頁右欄爲節編號與標題; 第 2 行指令設定偶數頁右欄與奇數頁左欄爲章編號與標題。以上的內定格式雖然完美, 但中文稿並不適用。以第 7 章爲例, 頁眉之標題除了章名之外, 還會自動加上英文 *CHAPTER* 與數字編號, 格式爲 "*CHAPTER 7.*"。不過, 以上的格式也可以重新設定。

LaTeX 是以下列三個指令儲存頁足/頁眉之章節編號與標題文字格式:

```
\chaptermark
\sectionmark
\subsectionmark
```

以上指令分別儲存章、節、小節之頁眉/頁足標題。欲重新設計頁足/頁眉之標題格式, 我們須重新定義以上的指令。第 1 個指令 \chaptermark 用於設定章標題之格式; 後兩個指令分別用於節與小節標題。例如, 若要去掉章編號之前的 "*CHAPTER*", 須在全文設定區加入指令:

```
\renewcommand{\chaptermark}[1]{\markboth{\thechapter.\ #1}{}}
```

以上指令使用 \markboth 指令重新設定 \chaptermark 指令所定義頁眉/頁足詞內之章標題格式。\markboth 指令存放兩項資訊, 分別置於指令之後兩

對大括號內。在 book 文件類別中, 第一對大括號內存放欲出現在左頁章標題處之文字; 第二對大括號內則存放欲出現在右頁節標題處之文字。

本例中, 節標題不更動; 因此第二對大括號內爲空白。章標題處之文字可能包括章編號與標題, 另外, 我們也可能加入諸如 "Chapter" 之文字。LaTeX 將章編號儲存於 \thechapter 指令內; 而 \chaptername 指令可用於定義英文字 "Chapter"。本章爲第 7 章, 因此在本章範圍內, \thechapter 之值即爲 "7"。在中文稿件中, 我們不想加入 "Chapter" 文字, 因此 \chaptername 可省略不用。本例中, \thechapter 指令之後緊接一英文句點, 其後以 \␣ 指令加入一空格, 最後的 #1 參數則代表章名。

經過以上之重新設計, \chaptermark 指令即將左頁之章標題定義爲:

如果頁眉之章編號要改爲「第 7 章」, 上一指令應更改如下:

```
\renewcommand{\chaptermark}[1]{\markboth{%
   第\thechapter 章\ #1}{}}
```

其中, 中文字之字體與大小尙可以中文字體指令設定。

出現在頁眉與頁足中之英文/數字, fancyhdr 巨集套件會自動選用特定字體, 但是這些指令並不能改變中文字體或其大小。欲改變中文章標題之字體, 可於下 \chapter 指令時直接鍵入字體指令, 或者以 \ctxfdef 指令作全文設定。若章標題是以中文粗黑體 20 點編排, 則章標題出現於頁眉時其大小即爲 20 點字體。欲解決此一問題, \chapter 指令應加入選項, 例如:

```
\chapter[文稿結構與章節設計]{文稿結構與章節設計}
```

同時, 全文設定區再以 \ctxfdef 指令選用中文字體。經以上設定, LaTeX 將取用方括號中之內容排版頁眉/頁足。

事實上, 選項方括號內之文字可以和大括號內之標題文字不同。譬如, 若章標題文字太長, 利用選項可使出現於頁眉內之文字簡短一些。另外, 若使用加 * 號之章節指令, 如 \section* 等, 標題內容不會自動編入頁眉中, 但我們可使用選項指令編排頁眉/頁足詞。

除了章標題之外, 單數頁之頁眉/頁足上之節與小節標題也可以使用類似的指令更改。以節標題爲例, 儲存節標題之指令爲 \sectionmark; 編號

則儲存於 \thesection 指令內。舉例言之, 若頁眉之節標題只要文字不要編號, 指令爲:

```
\renewcommand{\sectionmark}[1]{\markright{#1}}
```

請注意, 改變節標題須使用 \markright 指令, 而該指令僅有一選項。同理, 以類似的指令重新定義 \subsectionmark 之內容, 即可變更頁眉/頁足之小節標題之格式。

本書之頁眉爲空白, 頁碼與章節標題全部置於頁足。使用之指令列舉於下, 以供參考:

```
\fancyhf{}
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]%
   {\markright{\thesection~ #1}}
\fancyfoot[LE]{$\cdot$~\thepage~$\cdot$\rule{4mm}{0pt}\leftmark}
\fancyfoot[RO]{\rightmark\rule{4mm}{0pt}$\cdot$~\thepage~$\cdot$}
\renewcommand{\headrulewidth}{0pt}
```

以上指令在左頁頁足之左欄排版章標題, 但不編號; 右頁頁足之右欄則排版節標題。其中, \rule{4mm}{0pt} 是用以在頁碼與標題之間產生 4mm 之間距; 頁碼兩旁之小點是以 \cdot 指令排版。

以上所介紹的指令適用於排版正文各章節之頁眉/頁足, 而文稿正文以外的頁眉/頁足之排版可能須另作處理。以本書爲例, 1-19 章依序編號, 章標題與節標題分別出現於左右頁下端。但是,「序」是以 \chapter* 指令編排; 第 19 章之後的參考書目與索引也是以 \chapter* 指令編排。序文標題並編號, 內容也不分節, 索引亦然, 因此正文各章之頁眉/頁足格式並不適用。fancyhdr 之設定指令可在文稿內多次使用, 故排版時, 我們設計兩種標題格式, 一種用於排版正文各章之標題, 另一種則用於排版序、參考書目、與索引。

巨集套件 fancyhdr 所提供的功能甚多, 以上僅介紹其中較常用者, 完整的功能介紹請見該巨集指令之說明檔。該檔案說明如何讓頁眉長度大於正文方格之寬度; 也說明如何讓空白頁之頁眉/頁足空白。有關於 \leftmark 與 \rightmark 指令之詳細說明, 請見 Goossens, Mittelbach, and Samarin (1994), 4.3.1 節。最後, 圖 7.2 (頁 89) 的例子使用了 fancyhdr 巨集套件, 例

子檔案置於 \texmf\cwtex\examples 檔案夾內。請嘗試編排, 以了解各項指令之作用。

7.9 目錄

LaTeX 能自動記錄章節與圖表標題之文字內容及其頁碼, 因此排版目錄甚爲簡單。欲自動排版目錄, 章節標題須使用 \chapter, \section 等指令編排, 圖表標題則須以 \caption 指令編排, 否則 LaTeX 無法取得所需資訊。

根據 7.5 節之說明, 章節指令有好幾層。依內定值, book 文件類別所排版文稿之目錄將排版到 \subsection 層級, 短文 article 文件類別之目錄則排版到次小節。若增加或減少目錄之層級內容, 須改變 tocdepth 計數器之值。譬如, 於全文設定區加入下列指令:

```
\setcounter{tocdepth}{3}
```

則目錄將編排到 \subsubsection 層級。

章節標題若是以加 * 格式之指令編排, 如 \section*, 其內容不會自動排版於目錄內。欲將其內容自動編入目錄, 可使用 \addcontentsline 指令:

```
\addcontentsline{toc}{sec-name}{text}
```

請見 7.6.6 節之說明。此外, 如果要在目錄內特定地方加入排版指令或文字, 可使用以下指令:

```
\addtocontents{toc}{text}
```

目錄頁通常是排版於正文與序言之前。在文稿中選定目錄頁所要出現之位置, 鍵入 \tableofcontents 指令, 以 latex **編排文稿兩次**, 章節目錄內容即出現於指定位置。目錄首頁將出現 **Contents** 標題, 此英文標題可以變更爲中文字。例如, 在全文設定區加入下列指令:

```
\renewcommand{\contentsname}{\b20 目錄}
```

標題將以 20 點之中黑體字排出。

目錄是以內定之格式編排, 若不滿意其格式, 也可以自行設計。若文稿檔名爲 test.ctx, 第一次以 latex 編排之後, 工作檔案夾內會出現 test.toc 輔助檔案, 其內容即爲各章節之標題與頁碼。若爲中文稿件, 章節標題之中

文字是經過 cwtex 程式轉換後的格式, 而非原始輸入之中文字。但我們可以利用 cwTEX 提供的工具程式 tex2xtc 轉回中文字碼:

```
c:\xtemp>tex2xtc test.toc
```

執行指令之後, 將產生 test.xtc 檔案, 其內含有中文章節標題與頁碼。將此檔案修改爲理想的格式後, 即可引入原檔案內進一步排版。

若要排版圖標題之目錄, 文稿中須下 \listoffigures 指令; 同理, 排版表格標題目錄之指令爲 \listoftables。 排版之後, 工作檔案夾內分別產生 test.lof 與 test.lot 兩個檔案。圖標題目錄頁上方會自動排出 **List of Figures**, 表目錄頁上方則爲 **List of Tables**。若欲改爲中文標題, 請倣照上面的作法, 使用 \renewcommand 指令重新定義 \listfigurename 之內容與 \listtablename 指令之內容。或者, 我們也可以利用 tex2xtc 程式, 將 .lof 與 .lot 檔案轉換爲中文檔, 再進一步編輯排版。

7.10 附錄

排版附錄章節可使用 \appendix 指令。在 book 與 report 中, 此一指令之層級與 \chapter 指令層級相同, 但編號改爲 A, B, ...。譬如, 正文之後第一道 \appendix 指令將使附錄標題變成 **Appendix A**, 第二個 \appendix 指令則使標題變成 **Appendix B**。若是中文稿件, 我們可以使用以下指令將 **Appendix** 一字改爲「附錄」或其它中文標題:

```
\renewcommand{\appendixname}{\b14 附錄}
```

若使用 article 文件類別排版, 文稿內不能使用 \chapter 指令。因此, \appendix 指令之層級對應的是 \section。文稿內 \appendix 指令之下若出現 \subsection, 其標題編號將變成 A.1, A.2, ...等等。

8 段落編排

上一章說明文稿結構與章節設計,本章將說明排版局部段落之指令,如迷你版面、註解、列舉項目等等。從排版的角度來看, 版面上最基本的元素是單字。由單字組成句子、句子再組成段落、衆多的段落則構成章節。因此, 段落編排是很重要的一部分。

要編排普通的文字段落很簡單, 我們在文稿中留出一行空白, 表示上段落結束, 以下爲新段落開始。LaTeX 會將此段文字排得整整齊齊。但有些段落需要特別的設計, 譬如, 引述一段話、排版詩歌、條列說明等。這些特別的段落都是利用指令環境來編排。

除此之外, 本章也將介紹照列原文 (verbatim) 之指令環境, 對於排版原始程式碼而言, 此一工具非常方便。另外, comment 指令環境可用於讓某些段落之文字不要排版出來。排版長篇文章時, 各章節之間有時候要互相引述。譬如, 第 5 章之內文可能引述第 3 章之文字或第 4 章之圖表。LaTeX 所提供的 \label, \ref, 與 \pageref 三道指令即作此用途, 我們將於 8.8 節介紹。最後, muliticol 巨集套件可用於排版多欄式版面。

8.1 段落格式

一般文稿在新段落開始處會內縮一小段距離, 在英文中這稱爲 indent。此一工作可以由 \indent 指令來完成。輸入文稿時, 空一行 (或多行) 即表示要起新段落。當 LaTeX 碰到一空白行時即結束本段, 同時在下一段開頭自動加上 \indent 指令。因此, 除非有特殊的情況, 我們不須再下 \indent 指令。如果不希望句子內縮, 我們可以在段落開始之處使用 \noindent 指令。

若以 12 點之字體排版, 則段落開頭內縮之距離約爲 0.6 公分。不過, 此一距離可以自行選擇。譬如, 要將段落開頭內縮之距離拉長爲 0.8 公分, 只要在全文設定區加入下列指令即可:

```
\parindent=0.8cm
```

上面說明, 若要起新段落, 輸入文稿時必須留一空行。另外一個辦法是在上一段落結束處加上 \par 指令, 其作用和留空行完全相同。另外, 上一段落最末一行與本段落第一行間的行距和一般的行距相同, 並不特別加大。如果要加大段落間的行距, 可以使用 \parskip 指令。譬如, 若在全文設定區加上以下的指令:

```
\parskip=20pt
```

段落間的行距將改成 20 點。

段落內若要新起一行, 可以將換行指令 \\ 加於本行之末; 下一行之文字將齊頭編排, 不會內縮。本行與下一行之間的行距若要比正常行距加大0.5公分, 可以加上選項: \\[.5cm]。如果行距要縮小, 則選項應爲負值, 例如 \\[-3pt] 可以把行距減小 3pt。另一個對應的指令是 *。這個指令的作用也是換行, 不過它可以禁止 LaTeX 在指令處換新頁。\newline 指令與 \\ 功能完全相同, 但前者不能加上變更行距之選項。

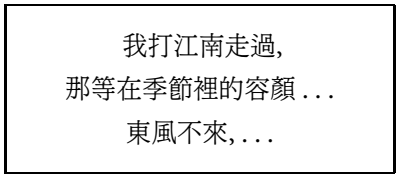
8.1.1 居中與靠邊

排版時, 我們常須把文字段落居中、靠左或靠右編排。LaTeX 提供各式各樣的指令控制文字段落之位置。

首先, 要使一段文字居中, 可使用 \centering 指令。若要靠左, 可使用 \raggedright。英文 raggedright 的意思是版面右邊不切齊, 因此指令之作用是段落靠左。反之, 文字段落靠右須可使用 \raggedleft 指令; 請見底下例子。


```
\centering{標題居中}
\raggedright{可靠左}
\raggedleft{或靠右}
```

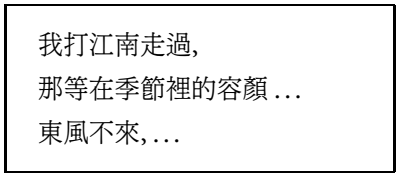
以上指令只移動一小段文字。若要將整段或數段文字、圖表等居中或移邊, 則使用指令環境較方便。譬如, 欲將某段落居中排版, 可使用 center 指令環境, 靠邊則使用 flushright 或 flushleft 指令環境。底下以鄭愁予的詩簡單說明指令之用法。




```
\begin{center}
我打江南走過,\\
那等在季節裡的容顏 ...\\
東風不來, ...
\end{center}
```

本例中, center 指令環境內全部爲文字, 但其中也可以含有其他的指令環境。第 10 章將說明排版表格之指令, 若要把表格排版於版面中央, 只要將整個表格之指令與文字置於 center 指令環境內即可。

如果要靠左排版, 應使用 flushleft 指令環境:




```
\begin{flushleft}
我打江南走過,\\
那等在季節裡的容顏 ...\\
東風不來, ...
\end{flushleft}
```

若要靠右編排, 則使用 flushright 指令環境。

以上所說明之居中排版是指將段落安排於橫向的中間位置。如果是要將某一段落安排於縱向的中間位置, 我們可以使用 \vfill 指令。首先, 在上一段落結束加上 \newpage 指令, 底下之文字段落將另起新的一頁。然後, 在段落文字之前後各加上 \vfill 指令, 即可產出縱向居中之段落。\vfill 指令的作用是將文稿段落往上頂或往下擠。

舉例來說, 若文稿最後一頁的文字並未填滿整頁, 而我們希望在該頁底部的左方記錄文稿檔名及日期, 一個簡單的方法是在 \end{document} 之前一行輸入底下的指令:

```
\par\vfill\noindent \jobname.ctx (\today)
```

以上指令中, \par 指令用於結束上一段落, \vfill\noindent 指令則把檔名及當天日期往下擠壓至最底端, 且靠左編排。\jobname 是 TEX 指令, 代表排版文稿的主檔名, 我們須在其後自行加上延伸檔名 .ctx。

8.2 調整間距

排版時, 我們經常須控制單字、句子或表格之位置, 或者段落間之距離。上

面已說明我們可以使用 \parskip 指令控制段落之間距; 本節將進一步介紹調整段落內間距之指令。

8.2.1 插入空白

LaTeX 有三個現成的垂直空白指令: \bigskip, \medskip, 與 \smallskip。其中, \smallskip 所產生之間距最小, \medskip 所產生之間距爲 \smallskip 的兩倍; \bigskip 所產生之間距又爲 \medskip 的兩倍。事實上, \smallskip 指令相當於是: \vspace{\smallskipamount}, 其中 \smallskipamount 是一個內定之數值。在正常情況下, 其值爲 3pt。我們之所以說「在正常情況下」, 原因是此數值事實上是在 2pt 到 4pt 之間伸縮。換言之, 其伸縮彈性爲 1pt。當 LaTeX 碰到 \smallskip 指令時, 它會先考慮整個版面之空間, 情況許可的話, 它將空出 3pt 之垂直空白。若加入 3pt 之垂直空白會使版面變成太擁擠, 它可能只空 2.4pt 的空白。反之, 若 3pt 之垂直空白使版面顯得太空盪, 則它可能把空白加大到 3.8pt。

若要自行設定間距大小, 可直接使用 \vspace 或 \vspace* 指令。 前一項指令如果下在本頁末端, 或者新一頁之起頭處, 指令皆無效。相對的, 後一指令不管是下在什麼地方, 都產生垂直空白。要注意的是, 垂直空白指令若下在段落中間, 則空白是在本行之後才出現。我們在本段結尾加上 \vspace*{.5cm} 指令, 因此段落間的空白比正常間距多了 0.5 公分。

對應垂直空白指令, LaTeX 亦提供幾個現成的水平空白指令, 較常用的是 \quad 與 \qquad。前者所產生之水平空白恰等於正文字體之點數。正文若使用 10pt 字體, 則 \quad 產生 10pt 水平空白; \qquad 則產生兩倍之距離。

若要直接控制間距, 可直接使用 \hspace 與 \hspace* 指令。譬如:

距離 1公分
距離 1公分
距離 1公分
後退0.5公分

```
距離\hspace{1cm}1公分\\
距離 \hspace{1cm}1公分\\
距離 \hspace{1cm} 1公分\\
\hspace{-.5cm}後退0.5公分
```

若距離值爲負數, 文句將左移。此例中, 第 4 句開頭 \hspace{-.5cm} 指令讓句子左移 0.5 公分。

仔細比較例子的前三行, 可發現句子中間空白的大小和 \hspace 指令前後是否留有空白有關。如果指令之前留一空白, 版面上所產生的空白爲 1 公分加上空白鍵之距離。如果後面也留空白, 空白爲 1 公分加上兩個空白鍵之距離。和 \hspace 指令相近的是 \hspace*。前一指令如果恰好出現在一行開頭處, 則指令無效, 本行文字開頭並不會右移。反之, 後一指令不論是在什麼地方出現, 都將產生水平空白。

有時候, 我們要把一行文字拆開爲兩段, 並儘量往左右兩邊靠, 或者是要將垂直空白插入一頁當中; 此時可應用 \hfill 或 \vfill 指令。我們可將 \hfill 指令想像成是一個有力的彈簧, 如果加在一行中央, 它會將前後的文字往左右兩邊推擠。例如, 若輸入下列指令:

```
1996年7月\hfill 台灣大學\\
1996年7月\hfill 台灣經濟史\hfill 台灣大學\\
```

排版結果爲:

1996年7月		台灣大學
1996年7月	台灣經濟史	台灣大學

本例中, \hfill 指令事實上等於是 \hspace{\fill}。\hspace 指令是留出水平空白, 但我們設定留出的空白是一個會往外伸張的彈簧 \fill。第一行指令中, 彈簧左右兩邊的文字都被往外推。第 2 行指令裡有兩個彈簧, 因此「台灣經濟史」5 個字就從兩邊往中間擠壓。上一節曾說明居中排版指令: \centering, 由以上例子可知, 兩個 \hfill 指令也可產生同樣效果。

若要將文字靠左排版, 可以利用下列指令:

```
\noindent ... \hfill\par,
```

其中 ... 代表排版文字。以上指令相當於上一節所介紹之 \raggedright。類似的, 靠右排版 \raggedleft 的對應指令是: \hfill ...\par。

8.2.2 填入細點或直線

在一行文字中加入水平方向之空白, 可使用 \hfill 指令; 相對的, \dotfill

指令可以在加入的空白中填入細點。如果要在產生的空白中畫出一水平線，應使用 \hrulefill 指令。以上的指令中，\dotfill 的應用之一是排版目錄。例如，下列的指令：

```
1. 前言 \dotfill 1\\
2. 排版例子 \dotfill 5
```

排版結果是：

1. 前言 1
2. 排版例子 5

若使用 \hrulefill，點線將變成實線，請見以下例子：

```
名字：\hrulefill
職稱：\hrulefill\hspace{1cm} 公司：\hrulefill
```

本例中，「職稱」之後有兩個 \hrulefill 指令，其中夾著 1 公分的空白及「公司」兩個中文字。排版結果爲：

名字: _____

職稱: _____ 公司: _____

8.3 引文與詩詞

TEX 雖然是以排版數學文稿著稱，但也有很多人用它來排版文學作品。LATEX 提供 verse 指令環境，可用於排版詩詞；引文指令環境 quote 可用於徵引他人文章。

8.3.1 引文指令環境

文稿中若引用他人的文句，可以使用 quote 與 quotation 指令環境。請見以下的例子：

在國慶文告的最後, 偉大的領袖一定會說:

> 讓我們一起高呼, 反共必勝! 建國必成! 中華民國萬歲! 蔣總統萬歲! 萬歲! 萬萬歲!

全國民衆歡欣鼓舞不已。

```
在國慶文告的最後,
偉大的領袖一定會說:
\begin{quote}
讓我們一起高呼, 反共必勝!
建國必成! 中華民國萬歲!
蔣總統萬歲! 萬歲!
萬萬歲!
\end{quote}
全國民衆歡欣鼓舞不已。
```

指令環境 quote 通常用於簡短的引文。如果引文甚長, 分開成好幾段, 則使用 quotation 指令環境較爲方便。在此環境之下, 各文字段落也是以空白行區隔。排版之後, 引文的每一新段落開始將內縮一些。相對的, 在 quote 指令環境下, 新起之段落行首不會內縮。

8.3.2 詩詞指令環境

若要排版詩詞歌謠, 可以使用 verse 指令環境。底下例子取自呂泉生作曲的〈阮那打開心裡的門窗〉:

〈阮那打開心裡的門窗〉是由呂泉生作曲 ...

> 阮那打開心裡的門,
> 就會看見故鄉的田園。
> 雖然路途千里遠,
> 總是 ...

```
〈阮那打開心裡的門窗〉
是由呂泉生作曲 ...
\begin{verse}
阮那打開心裡的門,\\
就會看見故鄉的田園。\\
雖然路途千里遠,\\
總是 ...
\end{verse}
```

我們在每句之後加上 \\ 指令以確定分行。一首詩歌若區分數個段落, 段落間應以空行分開。排版之後, 詩詞每一行會自動內縮一點。

8.4 條列指令環境

文稿中常常出現條列式文字, 或作列舉, 或作摘要式說明。LaTeX 提供三種條列指令環境: itemize, enumerate, 與 description。每一種條列指令環

境各有特定格式。必要時, 我們還可以自行定義格式。不過, 以下僅說明三種現成的指令環境之用法。

條列指令環境內還可以再使用另一層的條列指令環境, 最多可以使用 6 層。不管是那一種指令環境, 每一列舉項都是以 \item 指令起頭。排版之後, 每一條列項句子之前會有一文字或符號標籤 (label)。在 itemize 與 enumerate 指令環境下, 條列項之標籤有內定之符號, 但也可以自行設定。description 指令環境之標籤則由使用者自行輸入。

若使用 itemize 指令環境, LaTeX 會先在各條文之前加上一個圓點, 其後再排版條文內容。若 itemize 環境下還有一層 itemize 指令環境, 則內層之各項條文將以短線作爲標籤。內層若使用 enumerate 指令環境, 條文之標籤是阿拉伯數字。再下一層的 enumerate 環境中, 標籤是小寫英文字母 (a), (b), (c) 等等。

圖 8.1 之例子先以 itemize 指令環境起頭, 其下是 enumerate 指令環境。在 enumerate 環境之下, 含有兩個次一層的 enumerate 環境。在以上三個 enumerate 指令環境之後, 我們使用另一個 itemize 指令環境。指令環境是以 \begin{...} 與 \end{...} 的形式成對出現。當同時使用多個條列指令環境時, 須小心起頭與結尾的配對指令不可搞錯。排版時, 每一列舉項之上下各會留出較大的空白。如果要調整空白的大小, 必須另外下控制指令。底下會有進一步的說明。

以上所介紹兩個條列指令環境, LaTeX 將自動在列舉條文之前加上標籤。但是, 我們也可以自行選用符號或文字。例如, 圖 8.1 中第一個 \item 若改爲 \item[A], 排版之後標籤將改爲 **A**。我們還可以加上字體控制指令。例如, 若標籤要變成粗體字的 A, 指令爲 \item[{\bf A}]。標籤也可以是中文字, 例如, \item[甲、]。

如果要改變全部標籤符號, 可以使用 \labelitemi 或 \labelitemii 指令。前者代表最外層之 item 指令環境; 後者代表次一層之指令環境。再下一層之指令則爲 \labelitemiii 等等。譬如, 文稿中若使用下列指令:

```
\renewcommand{\labelitemii}{+}
```

則第 2 層指令環境之標籤將由 - 改變成 +。

同理, enumerate 指令環境也可以用類似的指令變更。此一指令環境之標籤是以數字英文字母依序標出, 控制指令有: \arabic, \roman, \Roman, 與

...
工友總聯盟的組織如下:

- 聯盟的組織
  1. 意志機關
     (a) 聯盟代表大會
     (b) 各工友會會員大會
  2. 執行機關
     (a) 執行委員會
     (b) 各部的組織
- 構成分子
  - 蘭陽總工大會
  - 基隆船炭工大會

```
...
 工友總聯盟的組織如下:
\begin{itemize}
\item 聯盟的組織
  \begin{enumerate}
  \item 意志機關
    \begin{enumerate}
    \item 聯盟代表大會
    \item 各工友會會員大會
    \end{enumerate}
  \item 執行機關
    \begin{enumerate}
    \item 執行委員會
    \item 各部的組織
    \end{enumerate}
  \end{enumerate}
\item 構成分子
  \begin{itemize}
  \item 蘭陽總工大會
  \item 基隆船炭工大會
  \end{itemize}
\end{itemize}
```

圖 8.1: 條列指令環境

\Alph。譬如,

```
\renewcommand{\labelenumi}{\arabic{enumi})}
```

將把 enumerate 指令環境最外層之標籤改變成 1), 2), ... 之形式。同理, 下列一行指令:

```
\renewcommand{\labelenumi}{\Roman{enumi}}
```

把標籤變成大寫羅馬字: I., II., ...。

以上之定義指令若加於全文設定區, 其效果將及於文稿中每一個條列指令環境。反之若指令是下於某指令環境內, 則僅該處之設定改變, 其他指令環境不受影響。一般而言, 改變行距、字體、或其他設定之指令若是下於指令環境中, 其影響效果只在該指令環境內。若希望其效果延伸至文稿每一部分, 指令應置於全文設定區。

第三種條列指令環境爲 description。使用此一指令時, 必須自行輸入

條文標籤符號, 並以方括號括起來, 請參見底下的例子:

執行委員會權限如下:

甲、執行各決議案

乙、管理聯盟的財政

丙、決定偶發事件之對策

```
執行委員會權限如下:
\begin{description}
\item 甲、執行各決議案
\item 乙、管理聯盟的財政
\item 丙、決定偶發事件之對策
\end{description}
```

標籤符號並不限定是一個字或符號, 也可以是一個名詞甚或一段文字:

政治 政客玩的遊戲,

政客 玩政治的人,

政治家 長得好看, 而且聲音好聽的政客。

```
\begin{description}
\itemsep=-1pt
\item[{\b11 政治}]政客玩的遊戲,
\item[{\b11 政客}]玩政治的人,
\item[{\b11 政治家}]長得好看,
    而且聲音好聽的政客。
\end{description}
```

其中, `\itemsep=-1pt` 指令的目的是把各列舉項之間的距離縮小一些。

在 `description` 指令環境下, 若某條列項之文字超過一行寬度, 第 2 行開頭文字會略往內縮一特定距離, 如上例中之第 3 條列項。但是在 `itemize` 與 `enumerate` 指令環境下, 列舉條文之長度若超過行寬時, 第 2 行開始每一行之起頭將內縮於標籤相對位置之後, 與條文第一行起頭對齊。

使用條列指令環境時, 我們可以設定每一層列舉項條文之內縮距離。第一層列舉項之內縮距離是以 `\leftmargini` 控制, 最後的字母 `i` 代表第一層。因此, 第二層列舉項以 `\leftmarginii` 控制。在圖 8.1 的例子內, 我們設定每一層列舉項內縮之距離都是 6mm。亦即, 在第一個 `\begin{itemize}` 指令之前加上下列三行指令。

```
\leftmargini=6mm
\leftmarginii=6mm
\leftmariniii=6mm
```

除了以上三種指令環境外, LaTeX 還提供 `list` 指令環境, 讓使用者自行設定特別的列舉格式。詳細說明, 請見 Kopka and Daly (1995, 頁 73-79)。

8.5 迷你版面指令環境

文稿版面上, 字母或單字是最小的組成要素。一個中文字或英文字母在版面上都是一個小方格。在橫排文稿中, 由單字或字母所串接而成的一行, 也是一個方格。段落是由數行文字組成, 在版面上它是一個更大的方格。因此, 文稿版面可以看成是由一堆大大小小的方格組合而成。

在簡單的文稿中, LaTeX 自動把一個一個文字按順序組合成方格。但有時候我們必須自己控制方格的大小及位置。舉一個例子來說, 如果我們在某一頁的左上角引用一個外製圖形 (見第 11 章), 其寬度爲正常行寬的三分之二, 則右邊剩下來的三分之一空白可以用來排版圖形說明文字。因此, 圖形是一個大方格, 說明文字是一個小方格。另外一個例子, 在信函結尾處會寫出發信人姓名, 其下爲其頭銜。假設姓名與頭銜共占 3 行, 且排於版面中間靠右之處, 我們可以將此 3 行所占空間視爲一方格, 再以指令控制其位置。

底下先介紹迷你版面指令, 下一小節則介紹文字方格 (box) 指令。

8.5.1 迷你版面

欲將某段文字控制於特定之小方格中, 可以使用迷你版面指令 \parbox, 或 minipage 指令環境。這兩道指令在段落編排上的用途甚廣, 值得細心了解。通常, \parbox 指令是用於處理較短的段落, minipage 指令環境則用於編排較長的文字。最簡單的指令形式如下所示:

```
\parbox[t]{5cm}{...}
```

若使用 minipage 指令環境, 指令如下:

```
\begin{minipage}[t]{5cm}
 ...
\end{minipage}
```

其中, 5cm 設定迷你版面之寬度爲 5 公分。

排版時, 迷你版面所形成之方格不會被拆開。換言之, 此方格和一個單字一樣, 變成是版面構成的最基本元素。排版時, 單字或字母是將其下沿對齊基線 (baseline)。但是, 迷你版面之高度可能是單字的數倍, 其對齊基線之點可以自行設定。上例中, t 爲迷你版面之對齊選項, 對齊選項共有三個選擇:

代表隊員名單如下:

天龍: 柯達
韓曼　地虎: 張德培
里歐斯

```
代表隊員名單如下:\\[.6cm]
 天龍: \begin{minipage}{1cm}
        柯達\\  韓曼
       \end{minipage}
\hspace*{.5cm}
 地虎: \begin{minipage}{1.2cm}
        張德培\\ 里歐斯
       \end{minipage}
```

圖 8.2: 迷你版面指令環境例子 1

- `t` 迷你版面之上沿對準基線,
- `c` 迷你版面之中央對準基線,
- `b` 迷你版面之下沿對準基線。

其中, `c` 爲內定之選項, 可以省略不加。圖 8.2 是迷你版面指令環境一個應用例子。

上面說明, 迷你版面可以視爲一小方格。因此, 本例之隊員名單事實上是由 4 個方格組成: 隊名爲其中兩個, 其餘兩個方格各由兩個人名所組成。輸入文字的第一行之後加上 `\\[.6cm]`, 其作用是強迫換行, 並加大行距。下一行的基線對齊由「天龍」、「地虎」等文字之下緣。兩個迷你版面都沒有選擇對齊點, 因此 LaTeX 選用內定值, 以迷你版面之中央點對齊文字之基線。

若輸入之文字大於迷你版面一行之寬度, LaTeX 會先把本行填滿, 再把其餘文字排至下一行。在此例子中, 我們希望兩個人名上下排出, 因此第 1 個人名之後加上換行指令, 使第 2 個名字排於下一行。本例中, 迷你版面寬度分別爲 1 公分與 1.2 公分。實際上,「柯達」兩個字所占寬度大約只有 0.7 公分左右, 其後約 0.5 公分在版面上變成空白。萬一我們設定之迷你版面寬度太小容不下三個字, 超過長度之文字將移至下一行。

迷你版面內亦可使用段落指令, 以 `\par` 指令或者空白一行爲之。但是, 新段落的頭一行不會內縮 (indent), 這是和全頁版面不同之處。如果希望迷你版面內每一段落開頭也內縮, 可以在文字之前加入以下指令:

```
\parindent=.6cm
```

每一段落開頭內縮之距離將爲 0.6 公分。

代表隊員名單如下:

隊長	**隊員**
陳一軍	山普拉斯 阿格西

```
代表隊員名單如下:\\[.5cm]
\hspace*{.2cm}
\begin{minipage}[t]{1.2cm}
{\bb11 隊長}\\ 陳一軍
\end{minipage}
\hspace{.5cm}
\begin{minipage}[t]{1.8cm}
{\bb11 隊員}\\ 山普拉斯\\ 阿格西
\end{minipage}
```

圖 8.3: 迷你版面指令環境例子 2

圖 8.3 迷你版面另一個應用例子, 其中對齊指令之選項爲 `t`。第3行的 `\hspace*` 指令使隊員名單右移 0.5 公分。

> 再舉另一個迷你版面的應用例子。你目前所閱讀的這段文字是納入在一個 `minipage` 指令環境中, 其寬度設定爲正常寬度的 0.9, 並居左編排。請注意, 在迷你版面中段落開頭文字並不內縮。我們所使用之指令如下:

```
\begin{flushright}
\begin{minipage}{.9\textwidth}
 ...
\end{minipage}
\end{flushright}
```

舊版 LaTeX 中, 迷你版面只能設定對齊基線之位置, 新版則提供另外兩個選項。第一個選項用以設定迷你版面的高度; 第二個選項則設定版面內文字段落之高低位置。因此, 完整的迷你版面指令爲:

```
\parbox[pos][height][in-pos]{width}{ ... }
```

若使用 `minipage` 指令環境, 格式爲:

```
\begin{minipage}[pos][height][in-pos]{width}
 ...
\end{minipage}
```

第 1 個 *pos* 選項設定迷你版面與基線對齊之位置; 第 2 個 *height* 選項設定版面高度。輸入指令時, 可以直接選擇高度, 如 `3cm`, 也可以使用下列之單位:

`\width`	文字方格的寬度,
`\height`	文字方格頂端到基線之距離,
`\depth`	文字方格的深度,
`\totalheight`	`\height` 加上 `\depth`。

譬如說, 若 *height* 設定爲 `[0.8\width]`, 則文字方格之高度爲寬度的 0.8。第 3 個 *in-pos* 選項設定文字在迷你版面內之編排位置。選擇 `t` 將使文字排於上端; `b` 排於下端; `c` 則居中編排; `s` 選項則使文字均勻排於迷你版面內。

在迷你版面內, 若下指令改變行距或選用特定英文字體、或作其他設定, 其作用只在指令環境內有效。因此, 若在迷你版面內一開始即選用 sans serif 字體。或者特定中文字體, 離開指令環境之後將自動恢復原先之字體。

8.5.2 文字方格 (box)

如上所述, 文稿版面是一個一個方格組合而成, 上一小節介紹的迷你版面是文字方格的一種。LaTeX 有三種文字方格的概念: LR 方格, 段落方格, 及線條方格 (rule boxes)。 在 LR 方格中文字只能由左至右編排; 段落方格則是一行一行之文字由上而下垂直疊起來的方格; 而線條方格則指由線段本身所構成的方格。上一小節介紹的迷你版面即爲段落方格, 因爲其版面是由一行一行的文字垂直堆積而成。

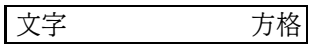
LR 方格中之文字只能由左至右水平相接。欲產生 LR 方格, 可以使用下列指令:

`\mbox{...}`	`\makebox[`*width*`][`*pos*`]{...}`
`\fbox{...}`	`\framebox[`*width*`][`*pos*`]{...}`

左邊兩項指令, `\mbox` 指令將大括號內之文字形成一無外框之方格; `\fbox` 則自動加上方形外框。例如, `\fbox{文字方格}` 指令, 將產生 文字方格 。`\makebox` 指令類似 `\mbox`, 但可以選擇寬度及方格內文字之排版方式; 同樣的, `\framebox` 指令類似 `\fbox`, 但也可以選擇寬度及方格內文字之排版位置。請見底下的例子:

文字方格	`\framebox[4cm][l]{文字方格}`
文字方格	`\framebox[4cm][r]{文字方格}`

選項 l 設定文字靠左, r 設定文字靠右。我們還可以選擇 s 選項, 代表均勻延伸 stretch, 再使用 \hfill 或 \dotfill 指令讓文字均勻排於方格內, 請見底下的例子:




```
\framebox[4cm][s]{文字\hfill 方格}
```

LR 方格之 *width* 選項除了直接選定長度之外 (如本例之 4cm), 還可以使用下列之單位: \width, \height, \depth, 及 \totalheight, 請參見上一小節之說明。

因爲 \makebox 或 \framebox 爲 LR 方格指令, 若文字之長度超過 width, 排版後文字將凸出方格外框, 不會自動拆爲兩行或三行。如果希望自動拆行, 可以使用下列指令:

```
\fbox{\begin{minipage}{4cm}
 ...
\end{minipage}}
```

在 minipage 指令環境中之所有文字段落, 不管有多少行, LaTeX 都視爲單一字母, 因此其外加上 \fbox 指令時, 即可產生方形外框。

使用文字方格指令畫方形外框時, 可以用 \fboxrule 設定外框線條的粗細; 而外框與其內文字之距離則以 \fboxsep 控制。例如:


```
\fboxrule=1pt
\fboxsep=15pt
\fbox{文字方格}
```

\fboxrule 之值若爲零, 則外框線條將隱而不見。

要將一小段文字上下移動, 可以使用下列指令:

```
\raisebox{lift}[height][depth]{ ... }
```

此道指令將大括號內之文字形成一 \mbox, 並使之垂直移動 *lift* 之距離。若 *lift* 爲負值, 方格向下移動。選項 *height* 與 *depth* 分別代表文字方格之高度與深度。

8.5.3 線條方格

線條方格就是一條直線本身, 指令如下:

```
\rule[lift]{width}{height}
```

其中, *width* 爲線條長度, *height* 爲其粗細; 而 *lift* 選項則將線條上下移動。因此, `\rule{1cm}{0.1cm}` 產生: ▬; 長度爲 1 公分, 粗細爲 0.1 公分。

在某些應用上, 我們可以將長度或高度設爲零。譬如, `\rule{2cm}{0pt}` 將產生 2 公分的水平空白; 反之, `\rule{0cm}{1cm}` 將產生 1 公分之垂直空白。底下指令:

```
\framebox{\rule{3mm}{0pt}\rule{0pt}{3mm}}
```

產生一 3mm 之正方形: □。

8.5.4 儲存方格

若一組文字在文稿中會重覆出現, 我們可以先將之儲存爲方格, 再於適當地方引用之。儲存方格之前, 須先取一名字; 新取之方格名字不可與現有巨集指令名稱相同。方格命名之指令爲 `\newsavebox`; 儲存方格可以使用 `\sbox` 或者 `\savebox` 指令。舉例言之, 若上一小節之正方形取名爲 `\sqe`, 指令爲:

```
\newsavebox{\sqe}
\sbox{\sqe}{\framebox{\rule{3mm}{0pt}\rule{0pt}{3mm}}}
```

利用以上定義, 我們即可使用 `\usebox` 指令三次:

```
\usebox{\sqe} \usebox{\sqe} \usebox{\sqe}
```

即可排版出三個連續方格: □□□

若使用 `\savebox` 指令儲存方格, 指令爲

```
\savebox{\boxname}[width][pos]{...}
```

其中, `\boxname` 爲自行選定之方格名稱; *width* 與 *pos* 選項之意義與上一小節所介紹之 `\framebox` 指令選項相同。

8.6 註解與邊註

文稿內排版註解與邊註 (marginal notes) 很容易。註解會自動編上號碼; 其內容以較小之字體排於當頁底下, 邊註則排於版面兩旁。

8.6.1 註解

註解是以 `\footnote{...}` 指令編排, 此道指令應緊接在引述註解之正文文字或標點符號之後, 中間不留下空白, 而註解內容則輸入於大括號之中。排版之後, 註解內容會出現在該頁正文下方, 正文中將以上標阿拉伯數字標示該註解之編號。若註解內容甚長, 本頁之剩餘空間無法容納, LaTeX 會自動將後半部分移至下一頁。

註解內之英文與數字將以較小字體編排。若爲中文稿, 我們須以指令改變中文字體之大小。第2章之例2爲一排版註解之實例, 在 `\footnote` 指令之後, 輸入註解文字之前, 我們以 `\m10` 指令選用10點明體字, 使中英文字體之大小得以適當搭配。若文稿使用許多註解, 比較方便的作法是以 `\ctxfdef` 指令設定中文字體。請見 7.7 節之說明。

LaTeX 會自動幫註解編上阿拉伯數字號碼。在 `book` 或 `report` 文件類別下, 每一章之註解將重新由1起編。但必要時使用者可以自行編號。若要將某一註解自行邊號爲4, 指令爲 `\footnote[4]{...}`。註解之編號是透過計數器 (counter)。因此, 另外一個自行編號的方法是透過註解編號計數器 `footnote`。要改變計數器之內容, 可以使用下列指令:

```
\setcounter{footnote}{4}
```

經過以上設定, 下一次再遇到 `\footnote` 指令時, 註解編號將變成5。

正常情況下, 註解編號爲阿拉伯數字。但亦可改爲 `*`, `**` 等符號。註解之計數器爲 `footnote`, 但實際排版號碼之指令爲 `\thefootnote`。LaTeX 有一特別用來設定註解符號字體之指令, 稱爲 `\fnsymbol`。 如果我們使用下列指令:

```
\renewcommand{\thefootnote}{\fnsymbol{footnote}}
```

則原來阿拉伯數字1-9之註解號將變成:

* † ‡ § ¶ ‖ ** †† ‡‡

作此改變之後, 在正文第 9 個註解之後, 使用者應重新將計數歸零, 使下一個註解再從第一個符號起編。如果註解編號是要以 a, b, c, ... 等英文字母排版, 指令爲:

```
\renewcommand{\thefootnote}{\roman{footnote}}
```

除了 \roman 之外, 尙可使用 \Roman, \alph, \Alph 等指令。

註解將編排於版面下方, 以一橫線與正文隔開。若欲改變此分隔橫線之格式, 可以使用以下之指令:

```
\renewcommand{\footnoterule}{\rule{w}{h}\vspace{-h}}
```

其中, \rule{*w*}{*h*}\vspace{-*h*} 重新定義分隔線格式; *w* 爲長度, *h* 爲粗細, \vspace 指令可用以控制橫線所占空間。若欲取消橫線, 直接以空白分隔正文與註解, *h* 值應設爲0pt。

同一頁中若有兩個以上之註解, 其間隔是以 \footnotesep 控制。例如:

```
\footnotesep=5mm
```

指令設定兩個註解之間距爲 5mm。

在數學模式或表格中, 不能使用 \footnote 指令。萬一有必要在數學式中或表格內加上註解, 我們可以使用下列兩道指令間接排版註解:

```
\footnotemark[n]
\footnotetext[n]{...}
```

第 1 道指令可以在正文中標誌註解編號, *n* 爲自行選擇之編號; 第2 道指令則是用於將註解內容排版於版面下方。

舉例來說, 若某數學式中須加上註解, 因爲不能使用 \footnote 指令, 我們只好以 \footnotemark 替代。若不加選項 [*n*], 此一註解仍將依續原有之編號。若加上 [9], 註解編號將變成 9。請注意, 以上指令純粹用於加註編號, 無法編排註解內容。排版完數學式之後, 底下再以 \footnotetext[*n*]{...} 指令排版註解內容。註解文字將和其他註解一樣, 出現於版面下方。註解文字若有中文, 須自行以中文字體指令調整大小。

LaTeX 會把註解文字排版於版面下方。但是, 有些學術期刊則要求把註解內容排版於文稿最後面, 此時我們可以使用 John Lavagnino 之 endnotes 巨集套件。指令使用方法請見該套件內附之說明檔。

x	y	z
111	222	333[a]
444	555	666[b]

[a] First note.
[b] Second note.

```
\usepackage{booktabs}
\begin{minipage}{2.8cm}
\renewcommand{\footnoterule}%
 {\rule{1cm}{0cm}\vspace{-2mm}}
\begin{tabular}{rrr}
\toprule
x & y & z\,\,\,,
\midrule
111& 222& 333\footnote{First note.}
444& 555& 666\footnote{Second note.}
\bottomrule
\end{tabular}
\end{minipage}
```

圖 8.4: 表格註解

8.6.2 表格註解

表格中若欲加入註解, 最簡單的方法是利用迷你版面指令環境。在迷你版面指令環境中使用註解, 編號將自動變爲 a, b, c, . . . , 而非阿拉伯數字。註解之文字將排版於迷你版面內之下方, 而非正常版面底下。

要在表格中排版註解, 可以將整個表格指令環境置於迷你版面內, 再設定取消註解之分隔橫線。圖 8.4 的例子中, 我們將排版表格的 tabular 指令環境置於迷你版面內, 同時設定將分隔橫線取消。如此一來, 迷你版面內之註解即接在表格底下排版。此種方法有一個不方便之處, 迷你版面之寬度須數度調整, 註解文字之長度及位置才會與表格寬度配合。本例中, 迷你版面之寬度爲 2.8 公分。在變數 z 之後, 我們加上 3 個空白指令 \, 使該欄之文字上下對齊。

表格註解之排版也可以使用 threeparttable 巨集套件, 作者爲 Donald Arseneau, 請見 10.6.3 節之說明。有關排版表格之指令, 請見第 10 章。

8.6.3 邊註

註解之內容是排版於版面下方, 相對而言, 邊註之內容則排版於版面兩側。排版邊註之指令爲 \marginpar, 註解內容將置於一 1.9 公分寬的迷你版面內, 排版於版面之兩旁。因爲迷你版面寬度甚小, 版面右緣不易對齊。解決此問題的辦法是設定讓每一行依文字之正常寬度排列, 右緣不必對齊。

舉一個例子來說, 排版本頁邊註之指令如下:

```
... 迷你版面內加上 \rightskip 指令,\marginpar{
\setlength{\rightskip}{0pt plus 3pt}
\m10
邊註之內容將排版於正文方格旁。}
...
```

邊註之內容將排版於正文方格旁。

本例中, 我們在迷你版面內加上 \rightskip 指令, 其作用是讓右沿之空白有一伸縮彈性。本例中, 彈性大小爲向外延伸3pt。

本書版面寬度較小, 因此我們使用下列指令將容納邊註文字之迷你版面寬度縮小爲1.3公分:

```
\marginparwidth=1.3cm
```

此項更動設定之指令必須下於邊註指令之前。

除了寬度可以更改之外, 我們還可以改變邊註版面與正文方格之距離, 兩個連續邊註之上下距離也可以自行設定。欲將前者距離設爲0.5cm, 後者設爲1cm, 指令分別爲:

```
\marginparsep=0.5cm
\marginparpush=1cm
```

8.7 照列原文

本書列舉許多例子說明 LaTeX 的排版指令。要在文稿內照列排版指令事實上並不容易, 譬如要排版數學符號 β, 指令爲 β, 但是在文稿內鍵入指令時, LaTeX 將直接排版出數學符號。同理, 排版 LaTeX 標誌符號之指令爲 \LaTeX, 但鍵入指令時, 標誌符號將直接排版出來, 無法看到原始指令。

要將原始排版指令直接列出, 可使用 verbatim 指令環境, 或者 \verb 巨集套件。英文 verbatim 的意義是照列原文的意思。指令環境 verbatim 主要用於排版電腦程式或原始排版指令; 不過, 此指令環境僅能用於英文/數字, 無法處理中文字。欲排版中文電腦程式或者像本書衆多的例子, 可使用 Timothy van Zandt 之 fancyvrb 巨集套件內之 Verbatim 指令環境。以下首先說明 LaTeX 照列原文之指令, 之後再說明 fancyvrb 巨集套件之功能。

在 verbatim 指令環境內之任何文字或指令都將照原輸入樣式列出, 並選用 typewriter 字體排版。譬如, 下例中含有 \LaTeX 指令文字, 正常情況下排版結果爲 LaTeX。但因爲全部文字段落置於 verbatim 指令環境內, 因此指令將照原樣列出, 不會排版爲 LaTeX 標誌符號。

```
A test of verbatim
environment of \LaTeX.
```

```
\begin{verbatim}
A test of verbatim
environment of \LaTeX.
\end{verbatim}
```

請注意原輸入指令有兩行, 排版結果也列爲兩行。若使用 verbatim* 指令環境, 則空白鍵將以 ␣ 符號出現。

如果照列之文字不超過一行, 我們可以使用 \verb 指令, 例如:

```
{\bb11 實質利率}%
等於名目利率減物價膨脹率。
```

```
\verb+{\bb11 +實質利率\verb+}%+\\
等於名目利率減物價膨脹率。
```

\verb 指令之後兩個 + 號中間所夾之文字或指令將原文照列出來。本例中, 第一個 \verb 指令內含左大括號, 中文字體指令 \bb11 及一個空白; 第 2 個 \verb 指令內含右大括號與百分比符號。如果照列之文字中有 + 號, 指令分隔符號須改用其他字元, 如 - 號或 # 號。要特別注意的是, 中文字不得出現於 verbatim 指令環境內或者 \verb 指令中。中文字體指令, 如本例之 \bb11, 若出現於指令環境內, cwTeX 也不作轉換動作, 而是直接列出。

除了不能用於排版中文外, \verb 指令或 verbatim 指令環境還有一些限制, 例如不能使用於註解指令之內。如果要照列中文, 或者要使用於註解指令內, 我們可使用 fancyvrb 巨集套件。此一巨集套件功能甚強, 它除了解決以上所述之問題外, 還有其他的功能。譬如, 我們可以設定在每一行文字之前編上號碼。

巨集套件 fancyvrb 主要提供 \Verbatim 指令環境, 此指令環境之基本功能與 LaTeX 原有之 verbatim 指令環境相同, 但是若加上選項, 即可產生特別的功能。舉例言之, 指令環境之後若加上適當的 commandchars 選項, 即可處理中文, 請見圖 8.5 之例子。

首先, 選項指令須置於方括號內, 本例之選項指令爲:

```
commandchars=+\[\]
```

```
\usepackage{fancyvrb}
\r11
\begin{Verbatim}[commandchars=+\[\]]
欲在 \LaTeX 內照列中文，
可使用 Verbatim 指令環境。
\end{Verbatim}
```

欲在 \LaTeX 內照列中文，
可使用 Verbatim 指令環境。

圖 8.5: 照列原文

等號右邊定義 3 個符號, 第一個符號 + 替代 LaTeX 之反斜線 \ ; 第二個指令符號 \[爲左中括號, 替代原有之大括號 { ; 第三個指令符號 \] 爲右中括號, 替代 LaTeX 原來之右大括號 } 。

定義以上三個符號之後, 原先之反斜線與左右大括號已失去原始之功能, 故 LaTeX 原有之排版指令, 例如 \textit{test}, 須更改爲 +textit[test]。本例中, \LaTeX 之反斜線已失去原來定義指令之功能, 故排版時不會產生 LaTeX 標誌符號, 而是依原鍵入字元列出。請注意, cwTeX 會特別處理 Verbatim 指令環境內所有之中文字, 故排版之後仍正確地出現。

以上例子重新以加號與左右中括號替代原有之指令符號。若 Verbatim 指令環境之文字內容含有以上 3 個符號, 替代符號須選用其他字元, 例如 *!\#。換言之, 我們以星號替代反斜線, 驚嘆號替代左大括號, \# 字號替代右大括號。除了 commandchars 指令之外, Verbatim 還提供許多控制選項, 例如, frame=single 設定在照列出之文字段落加上單一線條之方格, 方格與其內文字之間距可以用 LaTeX 之 \fboxsep 指令設定。此外, numbers=left 則在每一行文字之前自動編上號碼, 如底下例子所示。

```
\usepackage{fancyvrb}
\begin{Verbatim}[frame=single,%
  numbers=left,commandchars=*!\#]
欲在 \LaTeX 內照列中文，
可使用 Verbatim 指令環境。
\end{Verbatim}
```

1 欲在 \LaTeX 內照列中文，
2 可使用 Verbatim 指令環境。

圖 8.6的例子使用 Verbatim 指令環境排版 C 程式語言。因爲程式內使用了 [,], # 等符號, 故選用替代符號時須避開。fancyvrb 巨集套件另外一個功能是簡化 \verb 指令。例如, 下列指令定義兩個垂直線區域內爲 \verb 指令之範圍:

```
#include <iostream.h>
void main()
{
  char s1[12]="Hello Class";
  char *s2="Hello Class";
  cout << "s1=" << s1 << endl;
  cout << "s2=" << s2 << endl;
  // s1++;  錯誤, s1 為常數
  s2++;
  cout << "s2=" << s2 << endl;
}
```

```
\begin{Verbatim}[commandchars=@!?]
#include <iostream.h>
void main()
{
  char s1[12]="Hello Class";
  char *s2="Hello Class";
  cout << "s1=" << s1 << endl;
  cout << "s2=" << s2 << endl;
  // s1++;  錯誤, s1 為常數
  s2++;
  cout << "s2=" << s2 << endl;
}
\end{Verbatim}
```

圖 8.6: fancyvrb 巨集套件

```
\DefineShortVerb{\|}
```

經過上述定義之後, 欲照列 \LaTeX 標誌指令, 僅須在指令前後加上垂直線即可: |\LaTeX|, 不須使用較複雜的 \verb+\LaTeX+ 指令。如果照列之文字段落內含垂直線, 我們可以使用其它字母如 # 替代。此時, 上一行指令須更改如下:

```
\DefineShortVerb{\#}
```

最後, 如果 \verb 指令要用於註解之內, 須在文稿內執行下列宣告指令:

```
\VerbatimFootnotes
```

另一個具有照列原文功能之巨集套件爲 Donald Arseneau 所寫之 url。它提供 \url 指令。此巨集指令原先設計之目的是用於排版網路網址, 但也可以用來照列原文; 使用方法與 \verb 指令類似。\url 指令的使用彈性比 \verb 來得大, 譬如, 前者可置於其他巨集指令之內, 後者則不行。\url 指令內之文字若遇到一行末端, 可以自動拆爲兩行, \verb 則無換行功能。另外, 我們還可以設定是否將 \url 指令內所有的空白去除。

通常, \url 指令之內容須置於兩個相同的符號之間, 但我們也可以將之置放於左右大括號內。請特別注意, \url 指令內也不能輸入中文。詳細的指令使用說明直接置於 url.sty 檔案內, 請自行參閱。

8.8 引述其他章節

長篇文稿或書籍常在文稿某處引述其他章節或圖表, 這稱爲 cross-reference (引述)。LaTeX 會自動對章節編上號碼, 因此引述章節時通常是引述其編號。欲以 LaTeX 指令引述其他章節很簡單, 首先在被引述章節的標題指令處下 \label 指令作標誌, 接下來即可使用 \ref 或 \pageref 指令引述。譬如, 本節標題指令爲:

```
\section{引述其他章節段落}\label{refer}
```

則文稿任何地方鍵入下列指令:

```
... 請參見\Z\ref{refer}\Z 節（頁\Z\pageref{refer}）之說明。
```

排版之後, LaTeX 會自動算出章節編號與頁碼:

... 請參見 8.8 節 (頁 148) 之說明。

請注意, 排版時須連續執行 latex 兩次才能得到正確結果。除了章節之外, 若將 \label 指令置於某一註解 \footnote 的大括號之內, 文稿內其他地方以 \ref 指令即可引述該項註解, \pageref 指令則引述出現註解之頁碼。引述指令也可用於數學式中, 請見第 9 章之說明。

8.9 comment 巨集環境

文稿寫作經常須反覆修改增刪。修改過程中, 我們可以使用註銷指令 % 暫時去掉某一段落。若段落甚長, 較方便的方法是使用 comment 巨集套件, 作者爲 Victor Eijkhout。

首先在全文設定區引入巨集套件:

```
\usepackage{comment}
```

接下來, 文稿任何地方可加入 comment 指令環境:

```
\begin{comment} ... \end{comment}
```

指令環境內所有之文字與指令, 排版時即略過不處理。我們也可以直接使用下列指令:

```
\comment ... \endcomment
```

其效果相同。進一步的指令細節, 請見 comment 巨集套件之說明檔。

8.10 多欄位版面

一般的雜誌常以多欄方式排版。在 LaTeX 文件類別指令內加入 `twocolumn` 選項, 即自動以兩欄格式排版。但此法在排版上有一些限制, 譬如, 若文稿前端原以單欄方式編排, 中間改爲二欄時, 必須另起一頁。

有鑑於以上之限制, Frank Mittelbach 寫了 multicol 巨集套件以供排版多欄位文稿。此巨集套件功能較強, 使用上也很方便。請注意, multicol 巨集套件與 `\multicolumn` 指令之名稱接近, 但功能完全不同; 後者是用於排版表格內橫跨多欄之文字。巨集套件 multicol 提供 `multicols` 指令環境, 格式如下:

```
\begin{multicols}{columns}[preface][skip]
...
\end{multicols}
```

其中, *columns* 設定版面之欄位數, *preface* 選項可用以排版多欄位上方之標題。若 `multicols` 指令出現於接近版面下端位置, 則多欄位之文字會跳到下一頁才開始排版。第 3 個選項 *skip* 用於調整空間, 若設爲 6 公分, 則當版面所剩空間少於 6 公分時, 多欄位之段落將移至下一頁才開始編排。

底下的例子將第 2 章之例 2 排版爲兩欄格式:

```
\usepackage{multicol}
\raggedcolumns
\begin{multicols}{2}[\section*{\k12 國民所得兩萬美元}]
台灣的生活水準在提升嗎?
...
這不難理解。
\end{multicols}
```

排版結果如圖 8.7 所示。

多欄位指令環境可置於其他指令環境內, 例如 `minipage`。事實上, 本例子即置於 `minipage` 指令環境內, 寬度設爲原版面寬度的 90%。開始多欄位版面之前, 我們尚加入以下指令:

國民所得兩萬美元

台灣的生活水準在提升嗎? 若從大街小巷充斥著賣名牌的商店來看, 答案是肯定的。但如果從生活環境品質日益惡化來看, 答案則是否定的。

三十年前, 台灣的所得尚低。一般民衆最關心如何提升所得; 因此政府施政也以提升所得爲主要目標, 這不難理解。

圖 8.7: 多欄式版面

```
\raggedcolumns
```

此項指令的目的是設定各欄最底下一行不須強迫對齊。如果不下設定指令, 巨集套件將自行加上 \flushcolumns 指令, 儘可能讓各欄底端對齊。

多欄位版面之格式可以自行控制。首先, 欄位之間距是由 \columnsep 決定, 內定值爲 10pt, 使用者可以自行調整。其次, LaTeX 開始排版多欄位文字段落之前, 會與前面段落隔出一段距離; 與下接文字之間也會空出一點距離, 此間距是由 \multicolsep 指令所控制, 內定值爲 12pt, 並有一些伸縮彈性, 必要時可更改之。另外, 欄位之間可以加上垂直線以更明顯區隔。欲加入垂直線, 應在多欄位指令環境之前加入下列指令:

```
\setlength{\columnseprule}{0.4pt}
```

其中, 0.4pt 設定垂直線之粗細。

從 1.5u 版開始, multicol 巨集套件新增加 \columnbreak 指令。此一指令可強迫結束本欄之排版, 底下文字將排版於下一欄。相對而言, 如果我們使用文件類別指令之 twocolumn 選項排版兩欄式版面, 則結束本欄排版之指令爲 \pagebreak。

除了以上所介紹的之外, mutlicol 巨集套件尚提供不少指令以控制欄位之編排, 請參見該套件之說明檔。最後, 如果要將多欄位文稿之註解全部排版於右欄下方, 可使用同一作者所寫的 ftnright 巨集套件。

9 數學式子

TEX 以排版數學文稿著稱。Knuth 當初就是爲了排版自己的專業著作, 發現一般的排版系統處理數學式子的能力不佳, 而開始發展 TEX 的。LATEX 是由 TEX 的巨集指令組合而成, 它同樣有排版數學式子的優越能力。在專業排版中, 數學式的編排有特別的規範。譬如, 上下標須使用較小的字體; 數學符號須以數學斜體字編排; 數學函數如 log, max 等須使用正體字。所有這些細節, LATEX 都能夠自動處理。

若數式相當複雜, 或者其中用到一些符號是 LATEX 所沒有的, 我們還可使用美國數學學會所發展的數學符號與巨集套件。本章主要介紹 LATEX 的指令, 但也將簡單說明 $\mathcal{AMS}$-LATEX 巨集套件之功能。

9.1 科技文稿之排版規範

LATEX 排版數學或專業文稿的能力甚強, 但這並不保證我們的排版一定能達到專業水準。優美排版的條件之一是注意排版規範。排版的一般性規範請見 6.5 節, 本節進一步說明排版科技文稿的規範及原則; 主要內容係參考 Goossens, Rahtz, and Mittelbach (1997) 之 6.1 節。

排版任何文稿首須注意一致性 (consistency), 科學專業文稿更不例外。以數學文稿爲例, 大部分的數學符號須使用數學斜體字。排版時, 我們只要將符號置於數式模式中, LATEX 即自動選用數學斜體字。但是, 若一個符號在數學式子中是以數學斜體如 MPL 出現, 行文中卻以正體字 MPL 或意大利斜體 *MPL* 出現, 讀者可能滿頭霧水。因此, 行文中之符號、變數、甚至是數字, 必須與出現在數式內者相同。

依排版規範, 大部分的數學函數都以正體字編排, 例如 sin, max 等。這些函數符號可以使用現成的指令排版, 如 `\sin`, `\max` 等。相對的, 大部分之數學符號是以斜體排版, 但也有例外。例如, 度量衡單位通常以正體字排版。爲方便參考, 底下列出幾項較重要之規範:

- 單位, 如 cm, g, KeV 等以正體排版。但, 物理學中之常數, 如光速 c, 通常以斜體字排版, 例如: GeV/c。
- 化學元素, 如 Ne, O, Cu, 通常使用正體字。物理學中基本粒子, 如 p, K, g, H 等, 亦然。
- 常用之函數, 如 sin, tan, min 等, 以正體字排版。
- 數字使用正體字。
- 簡寫符號亦使用正體字。譬如, 以 exp 代表 experimental; max 代表 maximum; GNP 代表 Gross National Product。
- 微分符號使用正體字, 例如 $\mathrm{d}Y/\mathrm{d}k$。

9.2 數式環境

數學式可能以兩種型式出現, 一是隨文數式 (*in-text* formula); 另一種是數學式自成一行或一個段落, 我們稱之爲展示數式 (*display* formula)。隨文數式是夾在文章中的數學式, 例如: 若 $\alpha = 2$, 則 $\alpha^3 = 8$。展示數式則是獨立成行的數學式, 如:

$$\phi = \sum_{t=0}^{\infty} \beta^t U(c_t, x_t).$$

排版數式的方法和一般文字不同。欲排版數式, 首先須進入數式模式 (math mode), 或稱數式環境 (mathematical environment)。

隨文數式之數式環境很容易定義, 只須在數式前後加上 $ 符號即可, 排版時 LaTeX 即自動處理所有字體與間距的細節。上列隨文數式的輸入方法如下:

```
若 $\alpha = 2$, 則 $\alpha^3 = 8$。
```

如果忘了輸入數式模式符號, LaTeX 將出現錯誤訊息。欲進入數式模式, 除了使用 `$ ... $` 指令外, 我們也可以使用 `\begin{math} ... \end{math}`, 或者 `\( ... \)`。顯然, 第一種指令型式最簡單, 也因此最常使用。

排版展示數式須進入展示數式環境。底下的三種方法都可以達到目的:

```
\begin{equation} ... \end{equation}
\begin{displaymath} ... \end{displaymath}
\[ ... \]
```

若以 equation 指令環境排版, 每一行數學式都將自動編上號碼。相對的, displaymath 指令環境中之數學式則不編上號碼。第三種指令方式爲第二種指令之簡化型式; 我們也可以使用更簡化之指令型式: $$ ... $$。

欲排版上面舉例之展示數式, 指令如下:

```
\[  \phi = \sum^\infty_{t=0}\beta^t U(c_t,x_t).  \]
```

輸入以上指令時, 進入與離開數式模式之指令: \[與 \] 可以各自單獨占一行, 以方便辨識。展示數式上下, LaTeX 會自動留出適當空白, 不須再空一行。若特別留下空白行, 版面反而不正確。

較複雜的數學式可能橫跨兩行以上, 例如:

$$\begin{aligned} m_e &= m_e c \simeq \frac{1}{2} M_e V \\ &= m_e c^2 \simeq \frac{1}{2} M_e V \end{aligned}$$

跨行數式之排版方法, 將在9.6節介紹。

9.2.1 數學文稿輸入原則

第5.3節曾說明中文稿之輸入原則, 其中特別重要的是在何處換行、中文字間之空白, 及中文與標點符號之間隔。如果排版中文數學文稿, 還須注意以下兩點:

- 隨文數式前後請留一空格。因此, "若α之值" 是不對的; 應改爲 "若 α 之值"。請注意數式模式前後之空格。
- 展示數式上下不須多留一空行, LaTeX 會自行調整理想間距。

TeX 排版數學式子的能力特別強, 但輸入文稿時如果不注意以上的細節, 排版結果會很不理想。

9.2.2 簡單運算符號與上下標

數學式是由數字、數學符號、與加減乘除等運算符號組成。數字與普通運算符號可直接由鍵盤上鍵入。下列符號是直接由鍵盤鍵入:

```
+  -  =  <  >  /  :  !  '  |  [  ]  (  )
```

要注意的是, 左右大括號 { } 在 LaTeX 中有特殊用途。欲排版左大括號, 指令爲 \{, 右大括號之指令爲 \}。利用以上的符號可排版下列數式:

$$a < b + c = |d| \qquad \texttt{\$ a < b+c = |d| \$}$$
$$y'' = f\{y', y(x)\} \qquad \texttt{\$y'' = f\textbackslash\{y', y(x)\textbackslash\}\$}$$

其中, 二次微分符號須連續鍵入兩個英文引號 ''。若欲排版三次微分符號, 則須鍵入三個引號。譬如, 鍵入 $f'''(x)$, 排版結果爲 $f'''(x)$。

數學上標是以 ^ 指令輸入, 下標是以 _ 指令輸入。因此, 輸入 $x^{1/2}$ 可得到 $x^{1/2}$; 鍵入 a_{n} 可得 a_n; 而鍵入 x^2_t 或者 x_t^2 均可產生 x_t^2。最後一個例子顯示, 輸入上下標指令之順序並不重要。

上下標若只有單一符號或數字, 則大括號可以省略。最後一例中, 上下標均爲單一變數或數字, 輸入時大括號即省略之。輸入上下標時, 若忘記加上大括號, 很容易發生錯誤。例如, 若我們想排版的是 x^{2a}, 但輸入時只鍵入 x^2a, 排版結果爲: x^2a。底下是幾個較複雜的例子。

x_t^{2a}	`$x^{2a}_t$`	$A^{x^2_j}_{j^{2n}_{n,m}}$	`$A^{x^2_j}_{j^{2n}_{n,m}}$`
x_{y^2}	`$x_{y^2}$`		
x^{y_1}	`$x^{y_1}$`	$A^{-n/2}_{i,j,k}$	`$A_{i,j,k}^{-n/2}$`

9.2.3 分式

分式有兩種表現方式。簡單的分式, 如 $n/2$ 或者 $m/(m+n)$, 輸入方法分別爲: `$n/2$` 與 `$m/(m+n)$`。隨文數式之分式通常是以此一方式輸入。較複雜的分式常置於展示數式中。此時, 分式的輸入是以 \frac 爲之。例如:

$a/(m+n)^2$ 乘上 $1/(x+y)$ 等於 `$a/(m+n)^2$ 乘上 $1/(x+y)$ 等於`

$$\frac{a}{(m+n)^2(x+y)}. \qquad \texttt{\textbackslash[\textbackslash frac\{a\}\{(m+n)\^{}2(x+y)\}. \textbackslash]}$$

如本例所示, 分式指令 \frac 包含兩項數式, 第一項爲分子, 第二項爲分母。兩項都須以大括號括起來。本例亦顯示, 展示數式指令之前並不須空一行, LaTeX 會在正文與數式之間加入適當間距。

輸入複雜的分式時, 我們應在適當的地方加上大括號, 否則容易出現錯誤。底下是展示數式之分式:

$$\frac{x+y}{1+\frac{y}{x+y}}$$

```
\[
  \frac{x+y}{1 + \frac{y}{x+y}}
\]
```

分式指令 `\frac` 也可以用於行文中, 以產生上下層的分式, 如 $\frac{x}{x+y}$。但如此一來版面會顯得擁擠, 因此此種用法比較少見。

9.2.4 開根號

排版開根號的指令很簡單。譬如, $\sqrt{x+y}$ 的指令是 `$\sqrt{x+y}$`。開根號上面的橫線涵蓋的範圍是大括號內所有的符號。如果誤將左右大括號輸入成圓括號, 你等於是輸入 `$\sqrt{(}x+y)$`, 排版結果將變成 $\sqrt{(}x+y)$, 根號的橫線只涵蓋了左圓括號。

欲排版三次方以上的開根號, `$\sqrt$` 指令之後必須加入次方選項。例如 $\sqrt[5]{x+y}$ 的指令是: `$\sqrt[5]{x+y}$`。底下是一個較複雜的例子:

$$u = \sqrt[3]{-q+\sqrt{q^2+p^3}}$$

```
\[ u = \sqrt[3]{-q+\sqrt{q^2+p^3}} \]
```

本例中, 數式模式指令直接置於數式之前後。爲了易於分辨, 數式模式指令也可以單獨成一行, 數學式則輸入於兩行指令中間。

9.3 數學符號

數學文稿中經常使用各種數學符號,這些符號必須以指令輸入,而且必須置於數式環境中。本節將依序介紹排版各種數學符號之指令。

9.3.1 希臘字母符號

數學文稿經常使用希臘字母符號, LaTeX 直接採用希臘字母之音標爲其指令名稱, 很容易記住。表 9.1 列出所有的希臘字母符號。利用表中所列指令, 欲排版 X_b^β 須鍵入: `$X^\beta_b$`。欲排版:

$$(\gamma^\mu - m)\psi = 0.$$

須鍵入: `$(\gamma^\mu - m)\psi = 0.$`。在數式模式中, LaTeX 會自行調整各符號的間距。鍵入文稿時, 符號指令之間留不留空白對於排版結果並無

表 9.1: 希臘字母符號

α	\alpha	θ	\theta	o	o	τ	\tau
β	\beta	ϑ	\vartheta	π	\pi	υ	\upsilon
γ	\gamma	ι	\iota	ϖ	\varpi	ϕ	\phi
δ	\delta	κ	\kappa	ρ	\rho	φ	\varphi
ϵ	\epsilon	λ	\lambda	ϱ	\varrho	χ	\chi
ε	\varepsilon	μ	\mu	σ	\sigma	ψ	\psi
ζ	\zeta	ν	\nu	ς	\varsigma	ω	\omega
η	\eta	ξ	\xi				
Γ	\Gamma	Λ	\Lambda	Σ	\Sigma	Ψ	\Psi
Δ	\Delta	Ξ	\Xi	Υ	\Upsilon	Ω	\Omega
Θ	\Theta	Π	\Pi	Φ	\Phi		

影響。因此, 輸入上例時若將空白全部去除: `$(\gamma^\mu-m)\psi=0.$`, 排版結果相同。

除了希臘字母之外, LaTeX 另外有 26 個大寫字母的數學花體字 (script letters 或 calligraphic):

$$\mathcal{A}\mathcal{B}\mathcal{C}\mathcal{D}\mathcal{E}\mathcal{F}\mathcal{G}\mathcal{H}\mathcal{I}\mathcal{J}\mathcal{K}\mathcal{L}\mathcal{M}\mathcal{N}\mathcal{O}\mathcal{P}\mathcal{Q}\mathcal{R}\mathcal{S}\mathcal{T}\mathcal{U}\mathcal{V}\mathcal{W}\mathcal{X}\mathcal{Y}\mathcal{Z}$$

排版指令爲 \mathcal。譬如, 欲排版 $\mathcal{R}^n$, 應鍵入 `$\mathcal{R}^n$`。但如果使用舊版 LaTeX209, 指令爲 \cal。

數學符號也可以改變字體。上面所介紹的 \mathcal 指令是選用數學字體的一個例子。在 LaTeX 中, 選用數學字體之指令包括:

```
\mathrm       \mathtt       \mathbf
\mathsf       \mathit       \mathcal
```

數學字體指令只改變英文字母、數字、大寫希臘字母; 其他符號, 如小寫希臘字母或下一小節所介紹的函數符號, 並不受影響。請見以下的例子:

$$\mathbf{A}=\mathbf{B}^0(x) \qquad \texttt{\textbackslash[\textbackslash mathbf\{A\}=\textbackslash mathbf\{B\}\^{}0(x) \textbackslash]}$$

若要將式子中全部符號、變數等都變爲粗體字, 可使用 \boldmath 指令。欲取消數學粗體字, 指令爲 \unboldmath。請注意, 這兩道指令都必須下於數式模式之外, 否則將出現錯誤。

表 9.2: 函數符號

\arccos	\arcsin	\arctan	\arg	\cos
\cosh	\cot	\coth	\csc	\deg
\det	\dim	\exp	\gcd	\hom
\inf	\ker	\lg	\lim	\liminf
\limsup	\ln	\log	\max	\min
\Pr	\sec	\sin	\sinh	\sup
\tan	\tanh			

$\boldsymbol{A = B^0(x)}$

```
\boldmath
\[ A=B^0(x) \]
```

$A = B^0(x)$

```
\unboldmath
\[ A=B^0(x) \]
```

有些數學符號並無粗體字形, 若加上數學粗體字指令, LaTeX 會設法以普通字體替代之。

9.3.2 函數符號

依據專業排版規範, 數學變數應以數學斜體字編排, 但函數則應該以正字體排版。函數符號, 如 log, max 等, 若直接輸入, 如 `$log xy$`, 排版結果爲: $log xy$, 其中, log 函數及數學變數 xy 都以數學斜體字排版, 且兩者之間並未留有適當空白。在 LaTeX 中, 函數應以指令輸入: `$\log xy$`, 排版結果爲 $\log xy$。表 9.2 列出函數符號指令, 以供參考。

統計學中經常使用數學期望值 (expectation) 與變異數 (variance) 符號。根據排版規範, 函數符號應使用正體字, 因此排版結果應爲: $\mathrm{E}(x)$ 與 $\mathrm{var}(x)$。這兩個函數 TeX 並無現成的指令, 因此我們須自行控制。在數式模式中, 排版正體字可使用 \mbox 指令。因此以上兩個式子之輸入指令分別爲: `$\mbox{E}(x)$` 與 `$\mbox{var}(x)$`。如果文稿中經常使用這兩個符號, 我們可以定義兩個巨集指令, 以節省輸入時間, 並避免錯誤。巨集指令之定義方法, 請見第 14 章之說明。

9.3.3 積分與加總函數

某些數學符號在隨文數式中會比在展示數式中小一些, 積分與加總函數是

表 9.3: 積分與加總符號

$\sum$	\sum	$\prod$	\prod	$\int$	\int	$\oint$	\oint
$\coprod$	\coprod	$\bigcap$	\bigcap	$\bigcup$	\bigcup	$\bigvee$	\bigvee
$\bigwedge$	\bigwedge	$\bigodot$	\bigodot	$\bigotimes$	\bigotimes	$\bigoplus$	\bigoplus
$\biguplus$	\biguplus	$\bigsqcup$	\bigsqcup				

兩個例子。輸入積分符號之指令爲 \int, 加總符號指令爲 \sum。在行文當中, 積分函數之排版爲 $\int_0^1 f(x)dx$, 加總函數爲 $\sum_{i=1}^{n} x_i$。若是獨立一行之展示數式, 排版結果分別爲:

$$\int_0^1 f(x)dx \qquad \sum_{i=1}^{n} x_i$$

```
\[  \int^1_0 f(x)dx  \]

\[ \sum^n_{i=1} x_i  \]
```

除了符號大小有所差異之外, 函數上下標之位置也不同。在隨文數式中, 爲了不占太大的空間, 上下標是置於符號本身的右上方及右下方; 展示數式則置於上下方。如果要改變設定, 希望隨文數式中上下標也置於正上下方, 可以使用 \limits 指令。例如, 隨文數式 $\sum\limits_{t=0}^{t}$ 是特別加入控制指令排版而成: `$\sum\limits^t_{t=0}$`。請注意, 使用 \limits 指令時, 上下行距會自動加大一些, 效果不見得好。相反的, 若展示數式中之上下標希望置放於右上下方, 則將 \nolimits 指令加在上下標指令之前即可。

除了積分與加總函數之外, 表 9.3 所列各函數都會因隨文數式或展示數式而調整其排版方式。

9.3.4 箭號與相對關係符號

相對關係符號是指諸如大於或等於之類的符號, 數學文稿中經常使用。以下, 我們分箭號 (arrow symbols), 相對關係符號 (relation symbols) 與雙元運算符號 (binary operators) 三大類分別介紹之。

首先, 表 9.4 列出箭頭符號。這些符號除了用於數學式子之外, 也可以和直線、橫線連接, 用於排版線條形、流程圖等等。其中, $\leftarrow$ 符號之指令爲 \leftarrow, 但也可以使用較簡單的 \gets 指令。同樣的, $\rightarrow$ 右箭頭符號之

表 9.4: 箭頭符號 (arrow symbols)

$\leftarrow$	\leftarrow	$\longleftarrow$	\longleftarrow	$\uparrow$	\uparrow
$\Leftarrow$	\Leftarrow	$\Longleftarrow$	\Longleftarrow	$\Uparrow$	\Uparrow
$\rightarrow$	\rightarrow	$\longrightarrow$	\longrightarrow	$\downarrow$	\downarrow
$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow	$\Downarrow$	\Downarrow
$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow	$\updownarrow$	\updownarrow
$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow	$\Updownarrow$	\Updownarrow
$\mapsto$	\mapsto	$\longmapsto$	\longmapsto	$\nearrow$	\nearrow
$\hookleftarrow$	\hookleftarrow	$\hookrightarrow$	\hookrightarrow	$\searrow$	\searrow
$\leftharpoonup$	\leftharpoonup	$\rightharpoonup$	\rightharpoonup	$\swarrow$	\swarrow
$\leftharpoondown$	\leftharpoondown	$\rightharpoondown$	\rightharpoondown	$\nwarrow$	\nwarrow
$\rightleftharpoons$	\rightleftharpoons	$\leadsto$	\leadsto		

指令爲 \rightarrow, 但可以使用較簡化的 \to 指令。如果是要排版向量符號, 也可直接使用 TEX 之 \overrightarrow{a+b} 指令, 排版結果爲: $\overrightarrow{a+b}$。反之, \overleftarrow{x+y} 指令可排出 $\overleftarrow{x+y}$。表 9.4 列舉 TEX 系統所提供的箭號指令。

雖然由表 9.4 之指令已能排版各式各樣的箭號, 但是再多的符號似乎也永遠無法滿足所有人的需求。爲了應付數學文稿排版的需求, 美國數學學會 (American Mathematical Society) 又設計一些符號, 並提供 amssymb 巨集套件以方便引用。巨集套件 amssymb 提供的符號中, 有一些是數學領域以外的人也有機會使用者。譬如, 一般的文稿中也可能使用 $\because$ 符號代表「因爲」, 以 $\therefore$ 符號表示「所以」。但 LATEX 並未提供這兩個符號。若我們在全文設定區引入 amssymb 巨集套件:

```
\usepackage{amssymb}
```

文稿中數式模式內即可使用 \because 與 \therefore 指令排版以上這兩個符號。

巨集套件 amssymb 提供的符號可簡單區分爲 5 類, 以下將依序介紹。首先, AMS 箭頭符號全部列於表 9.5 以與 TEX 原有的箭號對照。其次, 相對關係符號列於表 9.6。相對關係符號中, \leq 指令用以排版 $\leq$, 但指令簡化爲 \le; 而 \geq 指令可以簡化爲 \ge。另外, $\neq$ 之指令爲 \neq, 但是, \not= 也產生同樣的結果。最後一個例子說明, 在相對關係符號指令之前

表 9.5: AMS 箭號 (amssymb 巨集套件)

⇛	\Rrightarrow	⇝	\rightsqugarrow	⇇	\leftleftarrows
↔	\leftrightarrow	⇐	\Lleftarrow	↞	\twoheadleftarrow
⇉	\rightrightarrows	⇄	\rightleftarrows	↠	\twoheadrightarrow
↢	\leftarrowtail	↣	\rightarrowtail	↫	\looparrowleft
↬	\looparrowright	⇋	\leftrightharpoons	⇌	\rightleftharpoons
↶	\curvearrowleft	↷	\curvearrowright	↺	\circlearrowleft
↻	\circlearrowright	↰	\Lsh	↱	\Rsh
⇈	\upuparrows	⇊	\downdownarrows	↿	\upharpoonleft
↾	\upharpoonright	↾	\restriction	⇂	\downharpoonleft
⇂	\downharpoonright	⊸	\multimap	↭	\leftrightsquigarrow
↚	\nleftarrow	↛	\nrightarrow	⇍	\nLeftarrow
⇏	\nRightarrow	↮	\nleftrightarrow	⇎	\nLeftrightarrow

加上 \not 指令, 將產生否定之關係符號。譬如, 鍵入以下指令 \not\in 與 \not>; 排版結果分別爲 $\notin$ 與 $\not>$。不過, \not\in ($\notin$) 與 \notin ($\notin$) 排版結果並不完全相同; 排版專家認爲後者較佳。

表 9.6 所列符號大部分是 TEX 系統所原有的。但是, 其中的 $\sqsubset$ 與 $\sqsupset$ 是 LATEX 特別製造之符號, 指令分別爲 \sqsubset 與 \sqsupset。欲使用這兩個符號, 我們必須在全文設定區引用 latexsym 巨集套件:

```
\usepackage{latexsym}
```

若不事先引入巨集套件, 文稿中使用這兩個符號指令時會出現符號指令未曾定義 (undefined) 之錯誤訊息。

表 9.6 中雖然有大於與小於符號, 但有些文稿使用大於、小於並列符號。我們可以由以上兩個符號合併而成。在全文設定區定義下列指令:

```
\def\gtls{\mbox{$_{<}$}\llap{\mbox{$^{>}$}}}
```

接下來在文稿內輸入 $a\gtls b$ 即可排版: $a \gtrless b$。不過, 直接以 amssymb 巨集套件所提供之符號排版, 品質更佳。表 9.7 列出 AMS 之雙元關係符號, 其中排版大於、小於並列符號之指令爲: \gtrless。類似的符號還有很多, 譬如大於、等於、小於並列符號之指令爲: \gtreqless。

除了雙元關係符號之外, AMS 另有 negated binary symbols, 我們稱之

表 9.6: 相對關係 (relation) 符號

$\leq$	\leq	$\geq$	\geq	$\equiv$	\equiv	$\models$	\models
$\prec$	\prec	$\succ$	\succ	$\sim$	\sim	$\perp$	\perp
$\preceq$	\preceq	$\succeq$	\succeq	$\simeq$	\simeq	$\mid$	\mid
$\ll$	\ll	$\gg$	\gg	$\asymp$	\asymp	$\parallel$	\parallel
$\subset$	\subset	$\supset$	\supset	$\approx$	\approx	$\bowtie$	\bowtie
$\subseteq$	\subseteq	$\supseteq$	\supseteq	$\cong$	\cong	$\Join$	\Join
$\sqsubset$	\sqsubset	$\sqsupset$	\sqsupset	$\neq$	\neq	$\smile$	\smile
$\sqsubseteq$	\sqsubseteq	$\sqsupseteq$	\sqsupseteq	$\doteq$	\doteq	$\frown$	\frown
$\in$	\in	$\ni$	\ni	$\propto$	\propto	$\vdash$	\vdash
$\dashv$	\dashv	$\notin$	\notin				

爲負雙元關係符號。這些符號全部列於表 9.8, 以供參考使用。此表內之符號有很多是表 9.7 內符號之相反。譬如, \leqq 指令產生 $\leqq$ 符號; 而 \nleqq 則產生 $\nleqq$。

接下來, 表 9.9 列出雙元運算符號。其中, \times 用以排版乘號, \div 用以排版除號。即使是一般文稿, 也常有機會使用加減乘除符號。在 LaTeX 中, 使用加減乘除應先進入數式模式。因此, 欲排版加號 $1+2=3$, 指令爲: \$1+2=3\$。若未進入數式模式, 排版結果爲 1+2=3。減號亦然, \$3-1=2\$ 排版結果爲 $3-1=2$。若未進入數式模式, 排版結果爲: 3-1=2。除了 TeX 的雙元運算符號外, AMS 又提供更多的雙元運算符號, 如表 9.10 所示。其中, \centerdot 比 LaTeX 之 \cdot 要稍大一些。

請注意, 在數學式子內, \dagger 指令可排版 † 符號; \ddagger 指令可排版 ‡ 符號。但在非數式模式內, 我們可使用 \dag 與 \ddag 指令排版同樣的符號。

除了以上各表所列之外, TeX 還有一些不易歸類的符號。有些用於數學或自然科學中, 有些是音符, 有些則用於一般文稿。我們將它們全部列於表 9.11。此表中, 下列 9 個符號是由 LaTeX 的 latexsym 巨集套件所提供:

$$\Box \quad \Diamond \quad \mho \quad \bowtie \quad \lhd \quad \rhd \quad \unlhd \quad \unrhd \quad \rightsquigarrow$$

連同前面所介紹的 $\sqsubset$ 與 $\sqsupset$, latexsym 巨集套件共提供 11 個符號。欲使用這些符號, 須先引用 latexsym 巨集套件。或者, 若引用了 amssymb 巨集套件, 我們也可以使用這些指令。

表 9.7: AMS 雙元關係符號 (amssymb 巨集套件)

$\leqq$	\leqq	$\leqslant$	\leqslant	$\eqslantless$	\eqslantless
$\lesssim$	\lesssim	$\lessapprox$	\lessapprox	$\approxeq$	\approxeq
$\geqq$	\geqq	$\geqslant$	\geqslant	$\eqslantgtr$	\eqslantgtr
$\gtrsim$	\gtrsim	$\gtrapprox$	\gtrapprox	$\eqsim$	\eqsim
$\lessdot$	\lessdot	$\gtrdot$	\gtrdot	$\lll$	\lll
$\ggg$	\ggg	$\lessgtr$	\lessgtr	$\gtrless$	\gtrless
$\lesseqgtr$	\lesseqgtr	$\gtreqless$	\gtreqless	$\lesseqqgtr$	\lesseqqgtr
$\gtreqqless$	\gtreqqless	$\doteqdot$	\doteqdot	$\Doteq$	\Doteq
$\eqcirc$	\eqcirc	$\risingdotseq$	\risingdotseq	$\circeq$	\circeq
$\fallingdotseq$	\fallingdotseq	$\triangleq$	\triangleq	$\backsim$	\backsim
$\thicksim$	\thicksim	$\backsimeq$	\backsimeq	$\thickapprox$	\thickapprox
$\subseteqq$	\subseteqq	$\supseteqq$	\supseteqq	$\Subset$	\Subset
$\Supset$	\Supset	$\sqsubset$	\sqsubset	$\sqsupset$	\sqsupset
$\preccurlyeq$	\preccurlyeq	$\succcurlyeq$	\succcurlyeq	$\curlyeqprec$	\curlyeqprec
$\curlyeqsucc$	\curlyeqsucc	$\precsim$	\precsim	$\succsim$	\succsim
$\precapprox$	\precapprox	$\succapprox$	\succapprox	$\vDash$	\vDash
$\vartriangleright$	\vartriangleright	$\shortparallel$	\shortparallel	$\Vdash$	\Vdash
$\vartriangleleft$	\vartriangleleft	$\smallsmile$	\smallsmile	$\Vvdash$	\Vvdash
$\trianglerighteq$	\trianglerighteq	$\shortmid$	\shortmid	$\smallfrown$	\smallfrown
$\triangleleftteq$	\triangleleftteq	$\bumpeq$	\bumpeq	$\between$	\between
$\blacktriangleleft$	\blacktriangleleft	$\pitchfork$	\pitchfork	$\varpropto$	\varpropto
$\blacktriangleright$	\blacktriangleright	$\Bumpeq$	\Bumpeq	$\backepsilon$	\backepsilon
$\therefore$	\therefore	$\because$	\because		

表 9.8: AMS 負雙元關係符號 (amssymb 巨集套件)

$\nless$	\nless	$\nleq$	\nleq	$\nleqslant$	\nleqslant
$\nleqq$	\nleqq	$\lneq$	\lneq	$\lneqq$	\lneqq
$\ngtr$	\ngtr	$\ngeq$	\ngeq	$\ngeqslant$	\ngeqslant
$\ngeqq$	\ngeqq	$\gneq$	\gneq	$\gneqq$	\gneqq
$\lvertneqq$	\lvertneqq	$\gvertneqq$	\gvertneqq	$\lnsim$	\lnsim
$\gnsim$	\gnsim	$\lnapprox$	\lnapprox	$\gnapprox$	\gnapprox
$\nprec$	\nprec	$\nsucc$	\nsucc	$\npreceq$	\npreceq
$\nsucceq$	\nsucceq	$\precneqq$	\precneqq	$\succneqq$	\succneqq
$\precnsim$	\precnsim	$\succnsim$	\succnsim	$\precnapprox$	\precnapprox
$\succnapprox$	\succnapprox	$\nsim$	\nsim	$\ncong$	\ncong
$\nmid$	\nmid	$\nshortmid$	\nshortmid	$\nshortparallel$	\nshortparallel
$\nparallel$	\nparallel	$\ntrianglelefteq$	\ntrianglelefteq	$\nvDash$	\nvDash
$\nVdash$	\nVdash	$\nVDash$	\nVDash	$\ntriangleleft$	\ntriangleleft
$\ntriangleright$	\ntriangleright	$\nvdash$	\nvdash	$\ntrianglerighteq$	\ntrianglerighteq
$\nsubseteq$	\nsubseteq	$\nsupseteq$	\nsupseteq	$\nsubseteqq$	\nsubseteqq
$\nsupseteqq$	\nsupseteqq	$\subsetneq$	\subsetneq	$\supsetneq$	\supsetneq
$\varsubsetneq$	\varsubsetneq	$\varsupsetneq$	\varsupsetneq	$\subsetneqq$	\subsetneqq
$\supsetneqq$	\supsetneqq	$\varsubsetneqq$	\varsubsetneqq	$\varsupsetneqq$	\varsupsetneqq

表 9.9: 雙元運算符號

$\pm$	\pm	$\cap$	\cap	$\diamond$	\diamond	$\oplus$	\oplus
$\mp$	\mp	$\cup$	\cup	$\triangle$	\bigtriangleup	$\ominus$	\ominus
$\times$	\times	$\uplus$	\uplus	$\triangledown$	\bigtriangledown	$\otimes$	\otimes
$\div$	\div	$\sqcap$	\sqcap	$\triangleleft$	\triangleleft	$\oslash$	\oslash
$*$	\ast	$\sqcup$	\sqcup	$\triangleright$	\triangleright	$\odot$	\odot
$\star$	\star	$\vee$	\vee	$\lhd$	\lhd	$\bigcirc$	\bigcirc
$\circ$	\circ	$\wedge$	\wedge	$\rhd$	\rhd	$\dagger$	\dagger
$\bullet$	\bullet	$\setminus$	\setminus	$\unlhd$	\unlhd	$\ddagger$	\ddagger
$\cdot$	\cdot	$\wr$	\wr	$\unrhd$	\unrhd	$\amalg$	\amalg

表 9.10: AMS 雙元運算符號 (amssymb 巨集套件)

$\dot{+}$	\dotplus	$\ltimes$	\ltimes	$\smallsetminus$	\smallsetminus
$\rtimes$	\rtimes	$\Cap$	\CAP, \doublecap	$\leftthreetimes$	\leftthreetimes
$\Cup$	\Cup, \doublecup	$\rightthreetimes$	\rightthreetimes	$\barwedge$	\barwedge
$\curlywedge$	\curlywedge	$\veebar$	\veebar	$\curlyvee$	\curlyvee
$\doublebarwedge$	\doublebarwedge	$\boxminus$	\boxminus	$\circleddash$	\circleddash
$\boxtimes$	\boxtimes	$\circledast$	\circledast	$\boxdot$	\boxdot
$\circledcirc$	\circledcirc	$\boxplus$	\boxplus	$\centerdot$	\centerdot
$\divideontimes$	\divideontimes	$\intercal$	\intercal		

表 9.11: 其他符號

$\aleph$	\aleph	$\prime$	\prime	$\forall$	\forall	∞	\infty
$\hbar$	\hbar	$\emptyset$	\emptyset	$\exists$	\exists	$\Box$	\Box
$\imath$	\imath	∇	\nabla	$\neg$	\neg	$\Diamond$	\Diamond
$\jmath$	\jmath	$\surd$	\surd	$\flat$	\flat	$\triangle$	\triangle
ℓ	\ell	$\top$	\top	$\natural$	\natural	$\clubsuit$	\clubsuit
$\wp$	\wp	$\bot$	\bot	$\sharp$	\sharp	$\diamondsuit$	\diamondsuit
$\Re$	\Re	$\|$	|	$\backslash$	\backslash	$\heartsuit$	\heartsuit
$\Im$	\Im	$\angle$	\angle	∂	\partial	$\spadesuit$	\spadesuit
$\mho$	\mho						

表9.11列有幾個樂譜符號, 如 ♯ 與 ♭ 等, 這些符號當然不足以排版樂譜。不過, 近幾年來陸續有人發展幾套以 TEX 系統爲基礎的樂譜排版系統; 請見9.8節的簡單介紹。AMS 所提供的其他符號列於表9.12, 其中, \square 可排版一正方形: □。若希望方形稍小一些, 可使用 latexsym 巨集套件提供的 \Box 指令, 排版結果爲 □。另外, \blacksquare 可排版一實心小正方形: ■, 有人用於標示數學証明結束。

最後, 數學中之 mod 函數有兩種型式, 第一種爲雙元型式 (binary), 輸入指令爲 $\bmod$; 第二種爲括號型式 (parenthesized), 指令爲 $\pmod$。譬如, 輸入 $a\bmod b$ 指令結果爲: $a \bmod b$; 相對的, $\pmod{a+b}$ 指令則排版爲: $\pmod{a+b}$。

如果只是偶而排版數學式, 要記得以上這些符號之指令其實並不容易。此時, WinEdt 軟體之數學符號圖形功能很有幫助。點選 WinEdt 視窗上之

表 9.12: AMS 希臘字母與其他符號 (AMS miscellaneous)

$\hbar$	\hbar	$\backprime$	\backprime	$\hslash$	\hslash
$\varnothing$	\varnothing	$\vartriangle$	\vartriangle	$\blacktriangle$	\blacktriangle
$\triangledown$	\triangledown	$\blacktriangledown$	\blacktriangledown	$\square$	\square
$\blacksquare$	\blacksquare	$\lozenge$	\lozenge	$\blacklozenge$	\blacklozenge
Ⓢ	\circledS	$\bigstar$	\bigstar	$\angle$	\angle
$\sphericalangle$	\sphericalangle	$\measuredangle$	\measuredangle	$\nexists$	\nexists
$\complement$	\complement	$\mho$	\mho	$\eth$	\eth
$\Finv$	\Finv	$\diagup$	\diagup	$\Game$	\Game
$\diagdown$	\diagdown	$\Bbbk$	\Bbbk		
$\digamma$	\digamma	$\varkappa$	\varkappa	$\beth$	\beth
$\daleth$	\daleth	$\gimel$	\gimel	$\lrcorner$	\lrcorner
$\ulcorner$	\ulcorner	$\urcorner$	\urcorner	$\llcorner$	\llcorner

$\sum$ 圖像 (icon), 即可開啓數學符號工作列, 其上列出各種數學符號之圖樣。點選任一圖樣, WinEdt 即在文稿中輸入指令, 甚爲方便。使用完畢, 再點選 $\sum$ 圖像, 即關閉工作列。

9.3.5 數學重音符號、底線與上線

以上所介紹的數學符號可附加上重音符號 (accents)。例如, $\hat{\beta}$ 或者 $\vec{a}$。在數學指令環境中, 重音附加符號共有下列幾種:

$\hat{o}$	\hat{o}	$\acute{o}$	\acute{o}	$\bar{o}$	\bar{o}	$\dot{o}$	\dot{o}
$\check{o}$	\check{o}	$\grave{o}$	\grave{o}	$\vec{o}$	\vec{o}	$\ddot{o}$	\ddot{o}
$\breve{o}$	\breve{o}	$\tilde{o}$	\tilde{o}				

重音符號指令當然也可以加在希臘字母上面, 如 $\tilde{\beta}$。另外, \hat 符號若要加大一些, 可使用 \widehat。譬如, 欲排版 $\widehat{x+y}/\widehat{xy}$, 輸入指令爲:

```
$\widehat{x+y}/\widehat{xy}$
```

同樣的, 較大的 \tilde 指令是 \widetilde。

在專業排版中, 字母 i 與 j 之上若有附加符號, 其頂上之小點應去掉。要排版去掉小點的字母 i 與 j, 指令分別爲 \imath 與 \jmath。例如,

```
$i+j \neq \vec{\imath} + \bar{\jmath}$
```

排版結果爲 $i + j \neq \vec{\imath} + \bar{\jmath}$。

重音符號是在字母上面加符號, 與此類似的是底線與上線。在普通文字稿中, 我們使用 \underline 指令劃底線, 此一指令也可以用於數式中:

value is $\underline{3x}$. `value is $\underline{3x}$.`

若要加入上線, 我們可以使用 \overline 指令。例如:

$$\overline{\overline{x}^2 + 1}$$
$$\overline{\overline{x^2} + 1}$$

```
$\overline{\overline{x}^2+1}$
$\overline{\overline{x^2}+1}$
```

底線與上線是在符號的上下加一直線。有時候, 我們須加上大括號。上括號之指令爲 \overbrace, 底括號爲 \underbrace。例如:

$\overbrace{a+\underbrace{b+c}}$ `$\overbrace{a+\underbrace{b+c}}$`

以上是隨文數式例子。展示數式中, 上括號與底括號若再加上下標, 排版結果如下所示:

$$\underbrace{a + \overbrace{b + \cdots + y + z}^{24} + z}_{26}$$

```
\[
 \underbrace{a + \overbrace{b +
  \cdots+y+z}^{24}+z}_{26}
\]
```

9.3.6 上下重疊符號

要把一個符號疊在另一個符號的上面, 可以使用 \stackrel 指令:

$A \stackrel{a}{\rightarrow} B$
$\vec{x} \stackrel{\rm def}{=} (x_1, \ldots, x_n)$

```
$A \stackrel{a}{\rightarrow} B$
$\vec{x} \stackrel{\rm def}{=}
  (x_1,\ldots , x_n)$
```

\stackrel 指令有兩項變數, 排版之後第一項變數的字體將會縮小一些, 並移至第二項變數的上頭。

我們也可以直接使用 TEX 所提供的指令排版上下重疊之數學符號。第一個指令是 \choose, 應用例子是排版 binomial coefficient。例如:

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}$$

```
\[
{n+1 \choose k} =
  {n\choose k}+{n\choose k-1}
\]
```

請注意, \choose 指令會自動加上左右圓括號。

第二個指令是 \atop。顧名思義, 此一指令的功能是把一個符號加到另一個符號上面。譬如,

$$\sum_{\substack{0\le i\le n \\ 0\le t}} U(c_{it}, x_{it})$$

```
\[
\sum_{\scriptstyle 0\le i \le n
  \atop \scriptstyle 0\le t}
  U(c_{it}, x_{it})
\]
```

此例子中之 \scriptstyle 指令是用以控制下標符號之大小。若不加這項指令, LaTeX 將以內定字體大小編排, 結果會顯得太小。

統計學或計量經濟學的論文中經常出現迴歸方程式 (regression equation), 其中估計值的下方須列出標準差, 譬如:

$$y = \underset{(2.23)}{1.23}\ x_1 + \underset{(1.22)}{2.34}\ x_2 + \epsilon$$

欲排版此式, 我們可以使用下列指令:

```
\[ y = \tb{1.23}{(2.23)}x_1 + \tb{2.34}{(1.22)}x_2 + \epsilon \]
```

其中, \tb 爲一巨集指令, 其定義如下:

```
\def\tb#1#2{\mathop{#1\vphantom{\sum}}\limits_{\displaystyle #2}}
```

此項巨集指令須在數式環境內才能使用。欲使用 \tb 巨集指令, 我們須在全文設定區鍵入其定義。

使用 \tb 指令時必須填入兩項參數, 第一項爲迴歸式之數字, 如上式中之 1.23, 第二項爲排版於其下之數字, 如上式中之 (2.23)。第二項數字與數學式之距離可以自行調整。上面巨集指令的定義中使用了數學符號 \sum。該符號之高度即爲上下兩行之距離。若希望距離加大一些, 只要使用較大的符號, 如 \big(或者 \Big(, 替代定義式中之 \sum 即可。

9.3.7 連續點

數學文稿中, 有時候以三個連續的圓點表示中間省略的符號。排版連續點之指令爲 \ldots 或者 \cdots。前項指令所產生的三點位於基線 (baseline) 上; 後者的點位置則稍微高一些。例如,

把 $x_1, x_2, \ldots$ 各項相加, 得
$X = x_1 + x_2 + \cdots$

```
把 $x_1, x_2, \ldots$ 各項相加,
得 $X=x_1+x_2+\cdots$
```

本例中, \cdots 所產生的連續點, 其位置稍高於 \ldots。相對的, \cdots 通常是用於 $+$, $-$, 與 $=$ 號之間。變數之間, 如 $a, b, \ldots$, 或者逗點之後之連續點, 則使用 \ldots。

除了水平的連續點之外, 尚有垂直及對角方向之連續點:

$\vdots$ \vdots $\ddots$ \ddots

底下將舉例說明這兩個指令的用法。請注意, \cdots, \vdots, 與 \ddots 只能用於數式指令環境中; 但是 \ldots 可以用於數式環境, 也可以用於一般的文字排版中。

9.4 定義與定理

數學文稿常須排版定義、定理等, 而不同期刊書籍各有其排版定義與定理之格式。LaTeX 原提供 \newtheorem 指令, 讓使用者自行設定定義與定理的排版格式。Frank Mittelbach 寫了一套 theorem 巨集套件, 強化原有指令的功能。以上這兩套巨集指令雖然已有相當好的能力, 不過, 任何事情都有更上一層樓的空間。美國數學學會所發展的 $\mathcal{AMS}$-LaTeX 巨集套件讓排版數學文稿, 包括排版定義與定理, 更簡單, 也更精確、完美。

1980 年代初期開始發展時, 此巨集套件原稱爲 $\mathcal{AMS}$-TeX。其後, LaTeX 系統逐漸普及。美國數學學會乃商請 Frank Mittelbach 與 Rainer Schöpf 重新發展爲 $\mathcal{AMS}$-LaTeX。$\mathcal{AMS}$-LaTeX 含幾套巨集套件, 對於數學家而言, 其中最重要的可能是 amsmath 巨集套件。

$\mathcal{AMS}$-LaTeX 是由美國數學學會所發展, 一般人可能認爲它僅適用於複雜的數學文稿。其實, 即使是簡單的文稿也可從中獲得好處。譬如說, 數學

定理 1: 若 v 均等分佈於 $[a, b]$ 之間, $\mathrm{E}(v_{km}^{kn-1}) = a + (b-a)(n-m)/n$。

証明. 獨立抽取 $N-1$ 個均等分配於 $U[0,1]$ 的隨機變數, 令 z_M^{N-1} 代表其中第 M 大者。
...
由上式, $\mathrm{E}(v_{km}^{kn-1}) = \mathrm{E}(v_m^{n-1})$。 □

```
\usepackage{amsthm}
\theoremstyle{thmsty}
\newtheorem{thm}{\rb11 定理}
\renewcommand{\proofname}{\hspace*{\parindent}{\k11 証明}}

\begin{thm}
若 $v$ 均等分佈於 $[a, b]$ 之間, ...
\end{thm}
\begin{proof}
獨立抽取 $N-1$ 個均等分配於 $U[0,1]$ 的隨機變數,
...
由上式, $\mbox{E}(v^{kn-1}_{km})=\mbox{E}(v^{n-1}_m)$。
\end{proof}
```

圖 9.1: 定理與証明

式若想編號爲 (3a), (3b) 等, 可輕易辦到, 但在 LaTeX 中則難以解決。另外, LaTeX 雖然提供 eqnarray 指令環境排版多行數學式, 但是 = 號左右之空白卻顯得太大。若使用 amsmath 之指令, 結果會很完美。

下文將簡單介紹 amsmath 之功能, 本節先介紹 amsthm 巨集套件, 其主要功能是排版數學文稿的定理與定義。amsthm 巨集套件是以 LaTeX 原有之 `\theorem` 指令爲基礎所發展出來的。此巨集套件雖然是 $\mathcal{AMS}$-LaTeX 之一部分, 但也可直接用於一般的 LaTeX 文稿。

圖 9.1 是以 amsthm 巨集套件排版定理的一個例子。本例子分定理陳述與証明兩部分, 証明部分是以 `proof` 指令環境排版, 這是 amsthm 所提供的指令環境。定理陳述則使用 `thm` 指令環境編排。此一指令環境並非現成的, 而是由使用者利用 amsthm 巨集套件內之 `\newtheorem` 指令自行定義。因爲是自行定義, 使用者可以控制排版格式。譬如, 定理可以自動編上號碼, 也可以不編號。定理之編號數字可置於「定理」兩字左邊或右邊。利用同樣

方法, 我們也可以設定預備定理 (lemma) 或定義 (definition) 的指令環境。

我們先說明排版証明之指令。使用 proof 指令環境時, LaTeX 將於証明開始之前自動加上 *Proof.* 一字; 証明結束處則加上 □ 符號。本例爲中文, 因此我們設定將 *Proof.* 改爲中文。前 4 行之指令須輸入於全文設定區, 第 4 行指令設定將 *Proof.* 改爲「証明」, 其原理是更改 \proofname 之定義。指令中 \hspace*{\parindent} 的作用是將該行內縮 \parindent 距離, 這是一般文字段落首行內縮的距離。經此設定, 証明文字之首行也內縮同樣距離。

依 amsthm 巨集套件的原始設定, *Proof* 一字是以斜體字排版, 其後並加上英文句點。本例中, 我們將句點改爲冒號, 並選用正體字。遺憾的是, amsthm (1.2d 版) 並未提供更動此種設定之選項。因此, 我們須直接修改原始指令。在 amsthm.sty (1.2d 版) 檔案的倒數第 11 行末端有 \itshape 指令, 其目的是選用斜體英數字形。去掉此指令即選用正體字。同檔案倒數第 10 行有 \@addpunct{.} 指令, 將 {.} 改爲 {:}, 即改用冒號。請注意, 爲尊重作者版權, 同時也爲了避免產生混淆, 作此更動之後, 巨集套件檔案應另取新名, 如 mythm.sty。

証明結束處之符號也可重新設定。例如, 在全文設定區加入下列指令:

```
\newcommand{\qedsymbol}{$\Box$}
```

符號將爲 □, 比原始正方形符號小一些。

接下來, 我們說明排版証明陳述的方法, 基本指令是 \newtheorem。此項指令可用於設定排版定義、定理、預備定理之格式。圖 9.1 的例子中, 全文設定區有下列一行指令:

```
\newtheorem{thm}{\rb11 定理}
```

此項指令設定排版定理的指令環境: thm。接下來, 我們即可利用 thm 指令環境排版定理。排版時, 定理陳述將自成一段落, 與上下文之間的距離稍加大一些。定理陳述之前將自動加上「**定理**」兩字; 自動編上號碼 (以粗黑體排版); 數字編號之後加上一英文句點。文稿中第二次使用 thm 指令環境時, 定理號碼將編爲 **2**。

若選擇自動編號, \newtheorem 指令另有兩個選項可以控制編號方法。以 book 文件類別爲例, 若第 3 章的定理要編爲 **3.1**, **3.2** 等, 則定義 thm 指令環境之指令應改爲:

```
\newtheorem{thm}{\rb11 定理}[chapter]
```

指令最後面之 [chapter] 選項指示以章爲編號單位。

另外一個情況, 若預備定理與定理要混合編號, 譬如, 定理 2, 引理 3, 引理 4, 定理 5 等等, 只要把下列指令加入圖 9.1 例子之全文設定區即可:

```
\newtheorem{lem}[thm]{\rb11 引理}
```

以上指令定義 lem 指令環境, 其中之 [thm] 選項表示新定義之 lem 指令環境將與 thm 指令環境混合編號。定理編號是附加於「**定理**」之後, 如果在全文設定區加入 \swapnumber 指令, 則編號將排於「**定理**」、「**引理**」等之前。如果不要編號, 定義指令應改用 \newtheorem*。

有些定理或輔助定理有特別名稱, 如 Zorn's Lemma, 此時可以使用下列定義:

```
\newtheorem*{Zorn}{Zorn's Lemma}
```

除了排版證明之外, 我們也可以同樣方法設定排版定義、預備定理等之指令環境, 而且我們也可以選用不同的格式。amsthm 巨集套件提供三種格式: plain, definition, 與 remark。譬如, 在全文設定區輸入下列兩行指令:

```
\theoremstyle{definition}
\newtheorem{def}{\bb11 定義}
```

文稿中即可使用 def 指令環境排版定義, 並選定 definition 格式。

萬一 amsthm 所提供的三種格式都不適合, 使用者還可以自行設計。設計排版格式須使用 \newtheoremstyle 指令。圖 9.1 所列的定理與証明例子是以自行設計的格式編排, 名爲 thmsty, 格式設計指令請見圖 9.2。我們所更動的格式包括: 定理陳述首行內縮 \parindent 距離;「定理」兩字之後加上冒號; 標題之英數字體以 \sffamily 字體族排版; 但定理陳述內容則以正體字編排。如果希望「定理」兩字要單獨排爲一行, 定理陳述排於其下, 則倒數第 2 行大括號內應改填入 \newline。

9.5 矩陣與行列式

矩陣的特徵是多項符號或字母規則性的排列, 並且左右各有圓括號或方括號。介紹矩陣指令之前, 我們首先要了解如何排列變數或符號。下一章所

```
\newtheoremstyle{thmsty}%  取名
  {3pt}%           上方間距
  {3pt}%           下方間距
  {}%              選用定理、定義陳述之英數字體
  {\parindent}%    首行內縮距離
  {\sffamily}%     選用標題之英數字體
  {:}%             標題後加上冒號
  {.5em}%          標題與定理陳述之間距
  {}%              特殊設定，請見巨集指令說明檔。
```

圖 9.2: 設計 amsthm 排版格式

介紹的 tabular 表格指令環境, 可使用於一般文字中, 也可使用於數式環境中。所以, tabular 指令環境可用於排列數學符號或變數。不過, 我們通常使用數學表列 array 指令環境來排列數字或符號。

數學表列指令和排版表格之指令很類似, 請見底下的例子:

$$\begin{array}{clr} a+b+c & m+n & xy \\ a+b & p+n & yz \\ b+c & mn & xyz \end{array}$$

```
\[ \begin{array}{clr}\\
    a+b+c & m+n  & xy \\
    a+b   & p+n  & yz \\
    b+c   & mn   & xyz
  \end{array}  \]
```

請注意, array 指令環境只能用於數式模式內。如同表格指令環境, 橫行各單項之間是以 & 區分, 每行之末則加上換行指令 \\。要注意的是, 最後一行的末端不須再加上換行指令。利用 \begin{array} 之選項, 可以控制每一直欄內各數字或符號要居中、靠左或靠右排列。本例中, 純粹爲了說明起見, 我們使用 {clr} 選項, 表示第一欄各文字居中, 第二欄文字靠左, 第三欄文字靠右。

如果要排版矩陣呢? 很簡單, 只要在數學表列兩旁加上大括號即可。若爲方括號, 指令爲 \left[與 \right]。此項指令所排版的括號, 其大小會因應數學表列大小自行調整:

$$A = \left[\begin{array}{clr} a+b & mn & xy \\ a+b & pn & yz \\ b+c & mp & xyz \end{array} \right]$$

```
\[  A = \left[
  \begin{array}{clr}
    a+b  & mn  & xy \\
    a+b  & pn  & yz \\
    b+c  & mp  & xyz
  \end{array} \right]  \]
```

表 9.13: 界限符號 (delimiter)

(	(	{	\{	⌊	\lfloor	↑	\uparrow
)	)	}	\}	⌋	\rfloor	↓	\downarrow
[	[	|	|	⟨	\langle	⇑	\Uparrow
]	]	‖	\\|	⟩	\rangle	⇓	\Downarrow
⌈	\lceil	⌉	\rceil	/	/	↕	\updownarrow
				\	\backslash	⇕	\Updownarrow

第 6 章曾經說明, 版面上的每一行文字都有一條基線 (baseline)。當使用數學表列指令時, 全表之中央對準基線。本例中 '$A\ =$' 將對準表列之中間點。通常我們不會把基線對準中點外的其它地方。不過, 必要時 array 指令環境可以加上選項, 以決定基線的對應位置。字母 t 表示選擇表列頂點對齊基線; b 表示選擇底點; c 是內定值, 表示選擇中間點; 請見底下的例子:

$$\Delta = \left(\begin{array}{cc} \alpha+b & \phi \\ \gamma & m\beta \end{array} \right)$$

```
\[
  \Delta = \left(
   \begin{array}[c]{cc}
   \alpha+b & \phi \\
   \gamma   & m\beta
   \end{array} \right)
\]
```

上面兩個例子同時說明大括號的使用方法。在 LaTeX 中, 大括號或方括號稱爲界限符號 (delimiter)。表 9.13 列出全部的界限符號。界限符號指令可以直接使用, 也可以在其前面加上 \left 或 \right 指令。一旦加上 \left 或 \right 指令, 界限符號的大小將隨其所包圍之數式或符號的大小自動調整。須注意的是, \left 與 \right 指令原則上須成對出現。不過, 在特別的應用例子中, 界限符號左右可以不同。下例中, 左邊爲大括號, 右邊爲中括號。請注意, 大括號之輸入指令爲 \left\{。

$$\Delta = \left\{ \begin{array}{c} \alpha \\ \phi \\ \gamma \end{array} \right]$$

```
\[
 \Delta = \left\{
  \begin{array}{c}
   \alpha \\ \phi \\ \gamma
  \end{array} \right]
\]
```

以上所使用的界限符號號 (包括左右括號) 都是由 LaTeX 自行計算大小。必要時, 我們也可以自行決定界限符號之大小。譬如, 欲使用大除號, 指令爲 \big/。要大一些, 指令爲 \Big/; 再更大一些, 指令爲 \bigg/; 最大除號之指令爲 \Bigg/。有些界限符號是左右成對出現的, 如左右圓括號。爲了易爲分辨, 以上之放大指令可加入 l 與 r。例如, \big(與 \big); 又如, \Big\[與 \Big\]。

底下的例子較複雜, 但只要把段落分清楚, 各項指令並不難了解:

$$\left(\begin{array}{cc} \left| \begin{array}{cc} a & b \\ c & d \end{array} \right| & A \\ B & \left| \begin{array}{cc} m & n \\ o & p \end{array} \right| \end{array} \right)$$

```
\[ \left( \begin{array}{cc}
    \left| \begin{array}{cc}
          a & b\\ c & d
     \end{array}\right| & A \\
  B & \left| \begin{array}{cc}
     m & n\\ o & p \end{array}
     \right| \end{array}
  \right)  \]
```

界限符號原則上必須成對出現, 但有些數學式只用上左大括號。此時, 我們可以使用 \right. 指令代替右括號, 排版之後式子右邊將爲空白:

$$x = \left\{ \begin{array}{ll} y & \text{if } y > 0 \\ z + y & \text{otherwise} \end{array} \right.$$

```
\[ x = \left\{
    \begin{array}{ll}
     y & \makebox{if $y>0$} \\
     z+y & \mbox{otherwise}
    \end{array} \right. \]
```

本例中, 數學式內排版了英文字 “otherwise”。在數學式中排版一般文字時, 我們須以 \mbox 指令將文字圈入, 否則 LaTeX 會將英文字當作數學符號排版。同理, 數式中若要排版中文字, 也必須置於 \mbox 指令內。

如果數學式子只需右括號, 則使用 \left. 替代左括號, 例如,

$$\left. \begin{array}{rcl} \Gamma(z) & = & \displaystyle\int_0^\infty e^{-t}t^{z-1}dt \\ \Gamma(z+1) & = & z\Gamma(z) \end{array} \right\}$$

```
\[
\left. \begin{array}{lll}
 \Gamma(z) &=& \int^\infty_0%
  e^{-t}t^{z-1}dt \\
 \Gamma(z+1) &=& z\Gamma(z)
 \end{array} \right\}
\]
```

最後一個例子是以 \ldots 與 \ddots 指令排版矩陣:

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix}$$

```
\[
 A=\left(
  \begin{array}{cccc}
  a_{11} & a_{12} & \ldots & a_{1n}\\
  a_{21} & a_{22} & \ldots & a_{2n}\\
  \vdots & \vdots & \ddots & \vdots\\
  a_{n1} & a_{n2} & \ldots & a_{nn}
  \end{array} \right)
\]
```

9.6 多行數學式

以上所舉的例子大多是單獨一行的數學式。那麼多行並列的數學式子如何編排呢? LaTeX 特別提供了多行數式 eqnarray 指令環境。不過, 此一指令環境仍有一些限制。譬如, 數式之編號方式很難改變。本節首先介紹 eqnarray 指令環境之使用方法, 接下來介紹功能較佳的 amsmath 巨集套件, 此爲 $\mathcal{AMS}$-LaTeX 之一部分。

依排版規範, 若多行式子並排, 且每一行中各有一個等號 (或大於、小於符號) 時, 各行式子之等號應上下對齊。在 eqnarray 指令環境內, 上下對齊之符號是以兩個 & 符號圈於其中。下例中, 第一行數式之輸入指令較長, 我們以 % 指令拆成兩行。第一行排版式子結束之處以 \\ 指令換行; 但最後一行之結尾不須加換行指令。

$$\begin{aligned} 1+r &= \frac{U_c(c_t, x_t)}{U_x(c_t, x_t)}, \\ c_t &= y_t + \frac{y_{t+1}}{1+r}. \end{aligned} \tag{9.1}$$

```
\begin{eqnarray}
1+r &=& \frac{U_c(c_t,x_t)}%
 {U_x(c_t,x_t)}, \nonumber \\
c_t &=&
   y_t + \frac{y_{t+1}}{1+r}.
\end{eqnarray}
```

使用 eqnarray 指令環境時, 每一行式子都自動編上號碼。若某一行不要編號碼, 可以在其末端加上 \nonumber 指令。如果每一條式子都不要編號碼, 可以直接使用 eqnarray* 指令環境。同一文稿內, eqnarray 與 eqnarray* 可以交互使用。

若以 amsmath 巨集套件排版類似的多行數式, 可使用 align 指令環境。請見底下例子:

$$a_1 = b_1 + c_1 \qquad (9.2)$$
$$a_2 = c_2 + e_2$$

```
\usepackage{amsmath}
\begin{align}
a_1 &= b_1+c_1\\
a_2 &= c_2+e_2 \notag
\end{align}
```

首先, 於全文設定區引入 amsmath 巨集套件。在 align 指令環境內, 欲上下對齊等號, 僅須使用一個 & 符號即可。在 align 指令環境內, 各式子會自動編號。欲取消編號, 須在末端加上 \notag 指令。如果各式都不編號, 應使用 align* 指令環境。

若數學式自動編號之格式不符需求, 我們可以使用 \tag 指令以自選之格式編號。下例中, 第 1 式編爲 (3a), 第 2 式爲 (3b)。若使用 \tag 指令, LaTeX 自動加上左右括號; 若使用 \tag* 指令, 使用者須自行填入括號。

$$a_1 = b_1 + c_1 \qquad (3a)$$
$$a_2 = c_2 + e_2 \qquad (3b)$$

```
\usepackage{amsmath}
\begin{align}
a_1 &= b_1+c_1\tag{3a}\\
a_2 &= c_2+e_2\tag*{(3b)}
\end{align}
```

若數學式分爲幾組, 每一組在特定地方須上下對齊, 我們仍可以 align 指令環境編排。底下的例子裡, 每一行使用 3 個 & 符號, 使兩組式子能在三個地方上下對齊。

$$a_1 = b_1 \quad x_1 = y_1 \qquad (9.3)$$
$$a_2 = c_2 \quad x_2 = y_2 \qquad (9.4)$$

```
\usepackage{amsmath}
\begin{align}
a_1 &= b_1 &x_1 &= y_1\\
a_2 &= c_2 &x_2 &= y_2
\end{align}
```

另外一種情況是上下兩行或多行排列的數學式應視爲一組, 因此只須一個編號即可, 此時可以利用 split 指令環境。此指令環境之功能與下文介紹的 multline 指令環境相同; 不過, 拆開的各行仍可使用 & 指令以上下對齊。另外須注意者, split 指令環境並無自動編號功能。若要自動編號,

可將此指令環境置於 equation 指令環境內, 式子編號將出現於首行與末行中央。底下是一個簡單的例子:

$$\begin{aligned} a_1 &= b_1 + c_1 \\ a_2 &= c_2 + e_2 \end{aligned} \tag{9.5}$$

```
\usepackage{amsmath}
\begin{equation}
\begin{split}
a_1 &= b_1+c_1\\
a_2 &= c_2+e_2
\end{split}
\end{equation}
```

若上下式子並無對齊等號之必要, 可使用 gather 指令環境編排。排版之後, 每一式子將居中編排。最後, 若一個式子太長, 無法編排在一行之內, 我們可使用 multline 指令環境編排, 在欲換行處加上 \\ 換行指令即可。一長式子若排爲 3 行 (需兩個換行指令), 第 1 行靠左編排, 第 3 行靠右編排, 第 2 行則居中編排。各行之間距可由 \multlinegap 指令設定, 我們可以用 \setlength 或 \addtolength 重新設定其值。

排版多行並列數式, 或者是把一條長式子拆成幾行時, 爲了提高可讀性, 我們必須調整各行式子的相對位置, 底下是一個參考例子:

$$\begin{aligned} \sum_{i=1}^{\infty} x_i + \sum_{i=1}^{\infty} y_i = \\ x_1 + x_2 + x_3 + \ldots + \\ y_1 + y_2 + y_3 + \ldots \end{aligned}$$

```
\usepackage{amsmath}
\begin{align*}
\lefteqn{\sum^\infty_{i=1}%
  x_i+ \sum^\infty_{i=1}y_i=}\\
 & &\quad x_1 + x_2 + x_3 +
      \ldots +\\
 & &\quad y_1 + y_2 + y_3 +
      \ldots
\end{align*}
```

此例中, 我們使用 \lefteqn 指令將第一行式子全部括於其中。排版時大括號內全部式子將視爲一符號, 且寬度爲零。換言之, 第一行式子不具有任何寬度, 而第 2 行與第 3 行的兩個 & 符號所包含之空白將上下對齊。因爲每一行的兩個 & 符號本身會產生一點小空白, 因此版面上第 2 行與第 3 行將往右移動一點距離, 整段式子看來較清楚。

若有一很長式子拆成兩三行, 而且第二或第三行是以加減號起頭, 則等號不宜直接與加減號對齊。簡單的解決辦法是在第二行加入 \quad, 讓第一

行之等號對齊第二行開頭處之空白:

$$y = x_1 + x_2 + x_3 \\ \quad + x_4 + x_5$$

```
\usepackage{amsmath}
\begin{align*}
y &= x_1 + x_2 + x_3 \\
  &\quad + x_4 + x_5
\end{align*}
```

本例是以 `align*` 指令環境編排, 因此數式並不編上號碼。

9.7 細節調整與數式編號

在數式環境下, LaTeX 會自動處理許多排版上的細節, 例如數學符號以斜體字編排, 符號之間的距離適當調整, 上下標選用較小字體, 數學式自動編上號碼等等。但是, 還是有某些情況, LaTeX 的處理結果不盡理想, 必須進一步人爲調整。以下分別說明之。

9.7.1 調整符號間距

在數式環境中, 各符號之間距會自動調整。不過, LaTeX 終究只是電腦軟體, 在特殊的情況下, 它所設定的結果可能不理想。此時, 我們須作進一步調整。底下列出數式模式內調整間距的指令:

`\,`	加入小空白 (約 1.5pt)	`\:`	加入中等空白 (約 3pt)
`\!`	減去小空白 (約 1.5pt)	`\;`	加入大空白 (約 5pt)

指令 `\!` 可視爲是 `\,` 的相反。前者縮小空白, 後者增加空白。底下的例子中, 左邊是排版結果; 中間是輸入指令, 其中已加入調整空白之指令; 右邊是不加上調整指令時的排版結果。

$\sqrt{2}\, x$	`$\sqrt{2}\, x$`	$\sqrt{2}x$
$n/\log n$	`$n/ \! \log n$`	$n/\log n$
$\iint z\, dx\, dy$	`$\int\!\!\int z\,dx\,dy$`	$\int\int z dx dy$
$x^2/3$	`$x^2\!/3$`	$x^2/3$

通常, 我們要看到初步的排版結果之後, 才能決定那些地方須作調整。因此, 剛開始輸入時, 你不須擔心是否要加上調整指令。等到已屆完稿階段, 才作間距調整。

9.7.2 調整符號大小

依據排版規範, 數學式之上下標符號須縮小一些; 上下列分式之分子分母符號亦然。LaTeX 會儘量將每一個數學符號調整至正確大小。但是, 人爲調整仍不可免, 連分式 (continued fraction) 是一個有名的例子。在底下的例子中, 我們須使用三個 \displaystyle 指令強制分母以較大之字體編排。

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4}}}}$$

```
\[ a_0 + \frac{1}{\displaystyle a_1
       + \frac{1}{\displaystyle a_2
       + \frac{1}{\displaystyle a_3
       + \frac{1}{a_4}}}} \]
```

若不加 \displaystyle 指令, LaTeX 以內定之大小排版分母, 各分母項之字體會越來越小, 結果並不理想:

$$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{a_4}}}}$$

本例中, 我們使用 \displaystyle 控制數學符號大小; 全部的控制指令計有底下四個:

\displaystyle	展示數式符號之標準大小,
\textstyle	隨文數式符號之標準大小,
\scriptstyle	第一層上下標之標準大小,
\scriptscriptstyle	第二層以後上下標之標準大小。

9.7.3 數學式居中與靠左

依原始設計, 數學式將居中排版。若希望靠左編排, 應於 \documentclass 指令中加入 \fleqn 選項。例如:

```
\documentclass[11pt,fleqn]{article}
```

靠左之數學式若要靠版面左緣一點距離, 可在全文設定區以下列指令設定。例如,

```
\mathindent=1cm
```

即設定每一展示數式自左沿內縮 1 公分。

9.7.4 引述數學式

數學式會自動編上阿拉伯數字號碼。若使用 chapter 文件類別排版, 在第 9 章內, 數學式將依序編號爲 (9.1), (9.2), ...。文稿其他地方若要引述某數學式, 可以使用引述指令: \label 與 \ref。

以下的例子中, 數式之編號是以 \tag 指令自行加上。被引述之數學式須以 \label 指令標示, 本例中, 兩行數學式分別標示爲 xy 與 ab:

$$z = x + y, \qquad (2a)$$
$$c = a + b. \qquad (2b)$$

```
\usepackage{amsmath}
\begin{align}
z &= x+y,\tag{2a} \label{xy}\\
c &= a+b.\tag{2b} \label{ab}
\end{align}
```

經過標示之後, 文章他處即可以 \ref{xy} 指令引述式 (2a)。同理, \ref{ab} 即可徵引式 (2b)。具體言之, 文稿內若鍵入「...數式 (\ref{ab}) ...」, 連續執行 latex **兩次**之後, 排版結果將變成「...數式 (2b) ...」。換言之, LaTeX 將 ab 標誌, 替代文稿之數式編號 2b。

如果排版文稿須反覆修改, 則使用 \label 與 \ref 指令甚爲方便。譬如, 在初稿中數學式之編號爲 (2a), (2b); 而修正稿中已變成 (4a), (4b)。此時, 以 \ref 指令徵引數式之處會自動更新編號。請注意, 使用 \ref 指令時, 我們須連續執行 latex 兩次, 才能產生正確結果。

數式編號是以 equation 計數器爲之。 假設文稿一開始的 4 條數式由 LaTeX 自動編號, 接著的兩條式子以 \tag 指令自行編號爲 (10.5a) 與 (10.5b), 再下一條數式應爲 (10.6)。此時, 我們可以必須先調整數式之計數器, 才能得到正確結果:

```
\setcounter{equation}{5}
```

9.8 其他專業文稿

除了排版數學文稿之外, LaTeX 的巨集套件還可以排版化學式、電子電路, 甚至樂譜、棋譜等。請參考 Goossens, Rahtz, and Mittelbach (1997) 之介紹。以下僅列出各巨集套件名稱, 有意使用者, 請自行至 CTAN 網路上取用, 並參考其中所附之說明檔。

化學式　排版化學式可使用 chemsym, XYMTEX, 或者 PPCHTEX 巨集套件,

物理　feynMF 巨集套件可用以排版 Feynman 圖形,

電路　circ 巨集套件可以排版電路或光學圖形,

樂譜　MusiXTEX 巨集套件可排版樂譜,

西洋棋　chess 巨集套件可以排版西洋棋棋譜,

象棋　cchess 巨集套件可以排版象棋棋譜,

圍棋　圍棋棋譜可以使用 go 巨集套件排版,

橋牌　橋牌可以使用 bridge.tex 巨集套件排版。

10 表格

若安排得當, 一個簡單的表格勝過千言萬語。製作表格的目的是使讀者能夠迅速地掌握資料或訊息。因此, 表格應該簡單、乾淨、準確。若表格太複雜, 讀者無法很快地了解其內容, 反而失去以表格表現資訊的目的。對於初學者而言, 排版表格可能是 LaTeX 最困難的一部分。不過, 如果能善用巨集套件, 排版表格並不困難, 而且排版品質是一般文書處理軟體遠不能及的。

在LaTeX 中, 排版表格可以使用 tabbing, array 與 tabular 指令環境。若以上指令環境還不能解決問題, LaTeX 另外提供 array 巨集套件, 目的是強化 array 與 tabular 指令環境之功能。除此之外, 我們尚可使用 tabularx 與 dcolumn 等巨集套件排版表格。除了以上巨集套件之外, 本章也將介紹如何排版彩色/灰階表格, 以及如何處理超大型表格。

若熟悉本章介紹的巨集套件, 即使是複雜表格也不難處理, 但這並不表示我們一定能排版出「高排版品質表格」。事實上, 從排版的角度來說, 困難的問題不是「如何排版」, 而是了解什麼是「品質」。一般表格排版最常見的錯誤是加入太多不必要的橫線或直線。因此, 在介紹表格指令之前, 我們先簡單歸納專家眼中表格排版的要點。

根據專家的意見, 排版表格應注意下列要點:

- 表格中絕勿畫垂直線,
- 不要畫兩條緊鄰橫線,
- 數字單位應排於欄位上端, 而非欄位內,
- 小數點之前應加上 0, 例如 .5 應排爲 0.5,
- 本欄位數字與上一欄位相同時, 請勿使用「同上」; 應直接排出數字。

以上意見引自 booktabs 巨集套件, 作者是 Simon Fear。你或許覺得這些意見極端, 不過, 研究形形色色的表格之後, 你會發現以上各點都很有道理。舉例來說, 底下所排版的這兩個表格各有三欄資料, 左邊表格各欄之間是以

直線區分, 右邊表格則拿掉各欄之間的直線, 上線與底線較粗, 中間線較細。右邊表格所含資料並不減少, 但版面清爽、易讀。

國家	央行 獨立性	物價 上漲率 (%)
意大利	0.5	16.1
英國	2	12.3
加拿大	2	8.1
西德	4	4.1

國家	央行 獨立性	物價 上漲率 (%)
意大利	0.5	16.1
英國	2.0	12.3
加拿大	2.0	8.1
西德	4.0	4.1

因此, 排版之前, 我們應該先思考何種設計能提高表格的易讀性。如果表格複雜到一般的 LaTeX 指令都難以處理, 則與其絞盡腦汁思考排版方法, 倒不如想一想是否有更容易表達想法的列表方式。換言之, 表格排版的目的是希望精簡地傳達重要訊息, 如果表格的設計複雜到不易排版, 那表示讀者可能難以掌握作者所欲傳達的訊息。此時, 簡化表格設計是第一步應該作的事情。

以上例子中, 表格置於行文當中。事實上, 在專業排版中, 大表格通常不置於文章中間, 而是移放版面上方或下方。爲了此一目的, LaTeX 提供 table 與 figure (浮動版面) 指令環境, 其主要功能是自動尋找適當置放圖表之位置。在此指令環境內, 我們可使用 \caption 指令排版圖表標題, LaTeX 會自動編入圖表號碼。

排版表格主要是使用 tabbing 與 tabular 指令環境。在介紹指令之前, 我們先簡單說明兩者之差別。指令環境 tabbing 的主要功能是將文字/數字排列於版面適當位置, 它並無現成指令可供加入橫線或直線。欲排版具有橫線或直線的表格, 最好是使用 tabular 指令環境。不過, tabbing 指令環境所排版之表格可以跨越一頁以上, tabular 則不行。若表格中有橫線條, 而資料又多到無法擠進一頁當中, 我們可以使用 longtable 巨集套件, 其指令功能與 tabular 類似, 但可處理超長或超寬表格。

如果表格複雜, tabular 指令環境內可以包含另一個 tabular 指令環境。因此, 我們可以用它來排版一個大表格, 其內包含幾個小表格。相對的, tabbing 指令環境中則不能包括另一個 tabbing 指令環境。一般而言, 排版表格大多使用 tabular 指令環境; 不過下一節我們先介紹 tabbing 指令環境, 再下一節開則介紹 tabular 指令環境之各種應用。

	大	中	小
牛肉麵	120元	100元	80元
搾醬麵	60元	50元	40元
酸辣湯	40元	30元	20元

```
\begin{tabbing}
\k11
\hspace*{1.3cm}\=~~大\qquad\= %
  ~~中\qquad\=~~小\\
\m11
牛肉麵 \> 120元 \> 100元 \>80元\\
搾醬麵 \>~~60元 \>~~50元 \>40元\\
酸辣湯 \>~~40元 \>~~30元 \>20元
\end{tabbing}
```

圖 10.1: 以 tabbing 指令環境排版表格

10.1 tabbing 指令環境

排版表格的方法之一是使用 tabbing 指令環境, 此一指令環境和傳統打字機編排表格的原理類似。傳統打字機鍵盤, 左上方有一 [Tab] 鍵, 按下此鍵, 打字頭將往右方移動若干距離。LaTeX 的 tabbing 指令環境即模倣 [Tab] 鍵之功能。如果表格內容主要是一些上下對齊的文字符號, 其間沒有橫線或直線, tabbing 指令環境很適合用來排版。反之, 如果表格中有區隔之直線或橫線條, 使用 tabular 指令環境排版比較容易。

圖 10.1 之例子表格共有 4 欄, 各欄之間距須自行設定。本欄之數字或文字排版後, 要跳至下一個欄位須使用 \> 指令, 此項指令之功能與打字機上之 [Tab] 鍵類似。本例中, tabbing 指令環境內含數行指令, 第一行除了排版標題之外, 並且使用 \= 指令設定欄寬; 行末之 \\ 表示此行指令結束。

表格內之文字如何排列? 設想表格中每一行最前端隱藏第 0 個 [Tab]; 第 0 到第 1 個 [Tab] 之間即容納第 1 欄文字, 第 1 到第 2 個 [Tab] 之間即容納第 2 欄文字。圖 10.1 例子中, 第 1 欄內之「牛肉麵」、「搾醬麵」等文字即在第 0 個 [Tab] 位置排出; 第 2 欄之價格數字即排於第 1 個 [Tab] 位置。此例中, 有些價格是三位數, 有些是二位數; 排版時最好是個位數上下對齊。在 tabbing 指令環境中, 文字/數字原本是緊貼著 [Tab] 位置排出。爲了使個位數上下對齊, 輸入價格數字時, 我們特別在 60 或 40 等數字之前加上兩個調整空白之指令 ~~。

欄位寬度如何控制? 本例中, 輸入指令的 3-4 行含有控制欄寬之設定。第一欄是以 \hspace*{1.3cm} 指令設定欄寬爲 1.3cm, 其後接之 \= 指令即標示第 1 個 [Tab] 位置。接著排版「大」字, 其後以 \qquad 指令拉開一點空

股市	收盤	漲跌幅
美國道瓊	10913.3	−1.1
法國	4323.8	0.8
台灣	7576.6	1.4

```
\begin{tabbing} \k11
美國道瓊數\= 222222222\=\kill
~~股市 \> ~~收盤 \> 漲跌幅\\
\m11
美國道瓊 \> 10913.3\> $-$1.1\\
法國     \>~~4323.8\> ~~~0.8\\
台灣     \>~~7576.6\> ~~~1.4
\end{tabbing}
```

圖 10.2: 以樣本行設定欄寬

白; 接著是第 2 個 [Tab] 之位置。「大」字之前另外加上兩個 ~~ 指令, 以免標題太靠欄位左邊。第 2、3 欄之寬度也是以類似的指令設定。第 2 行開始, 鍵入「牛肉麵」三個字之後, 使用 \> 指令即跳至第一個 [Tab] 位置。若某欄位空白, 連續兩個 \> 指令即可跳到再下一個 [Tab] 位置。

本例中, 每一行的第 4 欄都有文字。若某一行的第 4 欄空白, 則該行第 3 欄文字之後直接加上換行指令 \\ 即可結束該行。另外, 最後一行尾端不須加上換行指令, 因爲 \end{tabbing} 指令即兼有結束最後一行之功能。

10.1.1 以樣本行設定距離

上例中, 設定欄寬的第一行文字是表格的一部分, 會排版出來; 另外一種方式是以一樣本行 (sample line) 設定欄寬, 排版時樣本行並不出現, 請見圖 10.2 的例子。指令第一行爲樣本行, 行末須以 \kill 指令作爲結束, 不能使用 \\ 指令; 最右欄之欄寬不須設定。因爲樣本行之文字不會排版出來, 我們可以鍵入任何文字或以距離指令設定 [Tab] 位置。譬如, 第 1 欄寬度是以「美國道瓊數」文字設定距離; 第 2 欄則以 9 個「2」字設定距離。這些文字/數字與表格內容並無任何關係。

本例中, 美國道瓊股票指數之漲跌幅度爲 −1.1。請注意, 排版減號應進入數式模式: $-$1.1。如果未進入數學模式, 直接鍵入指令 -1.1; 排版結果爲 -1.1; 代表減號之橫線顯得太短。此外, 爲了使各數字小數點第一位上下對齊, 數字前端加上數個控制空白之指令; 但結果仍不理想, 下文會介紹使上下數字對齊之方法。

在每一欄位中, 文字將從指定之 [Tab] 位置開始排版。換言之, 文字串的最左端將位於 [Tab] 位置。但我們也可以設定將文字串之右端擺於

年期	館藏地
大正11年	台灣分館
12	台大總圖
昭和元年	台大總圖
2	台大總圖

```
\tabbingsep=0.2mm
\begin{tabbing}
\hspace*{1cm}\= 22222222\=\kill
\k11
\>年\'期 \> 館藏地 \\ \m11
\>大正\'11年 \> 台灣分館\\
\>       12    \> 台大總圖\\
\>昭和\'元年 \> 台大總圖\\
\>       2     \> 台大總圖
\end{tabbing}
```

圖 10.3: tabbing 指令環境之應用

[Tab] 位置, 此時須在文字串尾端加上 \' 指令。若此一指令是加於一串文字中間, 指令左邊的文字將排於 [Tab] 位置之左, 其右再接著排出文字串右邊文字。

圖 10.3是一個應用的例子。此例中, 我們設定兩個 [Tab], 第 1 個 [Tab] 位置距離版面左邊緣 1 公分。利用 \' 指令, 我們將「大正」、「昭和」等文字挪於 [Tab] 左方, 右方則排版「11 年」、「元年」等文字。接下來的「台灣分館」、「台大總圖」等將從第 2 個 [Tab] 位置開始編排。第 1 個 [Tab] 左右兩邊文字之間隔可以用 \tabbingsep 指令控制。此外, 我們利用 \tabbingsep 指令將間距設定爲 0.2mm。

與 \' 指令相對的是 \‘, 其功能是將排版文字往右推擠到版面邊緣。如果某段文字之前加上 \‘ 指令, 而且其後沒有 \= 或者 \> 指令, 則此段文字將居右 (right-justified) 排版。

以上所介紹的 \=, \', 與 \‘ 等三個指令, 原本是用以排版重音符號。(參見第5.1 節。) 在 tabbing 指令環境中, 因爲它們各有特別作用, 因此喪失其原來的功能。如果在 tabbing 指令環境中要排版重音符號, 我們須分別以下列指令替代: \a=, \a', 與 \a‘。例如, 要排版 ō, 我們須鍵入 \a=o。

10.1.2 其他控制指令

在 tabbing 指令環境下, 若一行之首加上 \> 指令, 第 1 欄文字將從第 1 個 [Tab] 位置開始編排, 而非隱藏的第 0 個 [Tab]。若表格共有 5 行, 每一行之首都須加上 \> 指令。爲了簡化輸入, 我們可以在第一行之首以 \+ 指令替

代 \> 指令, 以下各行前端就不須再加上任何指令, 各行最左欄文字/數字就會從第 1 個 [Tab] 位置開始編排。

如果某一行之首加上 \+\+ 指令, 則從該行開始以下的每一行文字將從第2 個 [Tab] 位置開始編排。相反的, \- 指令則使下一行開始編排之欄位往左移一個 [Tab] 位置。最後, 若某一行之首加上 \< 指令, 則前面所加之 \+ 指令在該行即失效。

10.2 array 巨集套件

除了 tabbing 指令環境之外, 排版表格尙可使用 tabular 與 array 指令環境; 其中, array 指令環境主要是用於排版數學行列式與距陣, 請見 9.5 節之說明。因爲表格經常出現在文稿中, 而且式樣多變, 新版 LaTeX 中另外提供 array 巨集套件以加強 tabular 指令環境之功能。

因爲表格樣式變化甚多, 因此除了以上 LaTeX 所提供的指令, 許多專家又寫了各種巨集套件。譬如, Simon Fear 設計 booktabs 巨集套件, 以配合 LaTeX 之 tabular 指令環境使用。本節主要介紹 tabular 指令環境與 array 巨集套件之強化功能, 稍後並介紹 booktabs 巨集套件。

10.2.1 tabular 指令環境

使用 tabular 指令環境排版表格並不難, 但是, 要控制到表格的每一個細節, 我們需仔細了解各項指令。學習排版表格最好的方法是多研究一些例子, 因此底下提供衆多例子說明。

圖 10.4 是 tabular 指令環境的第一個例子, 控制格式之指令如下:

```
\begin{tabular}{lccc}
```

其中, {lccc} 選項用以設定各欄位文字/數字之排版位置。括號內 4 個字母表示表格共有 4 欄資料, 第一個字母 l 代表 left (左邊), 表示第一欄資料應靠左排列。2-4 個字母都是 c 代表 center (居中), 表示 2-4 欄資料排版時應居中。如果 2-4 欄資料靠右編排, 指令應改爲 \begin{tabular}{lrrr}, 其中字母 r 代表 right。

表格資料如何輸入呢? 每一橫行的 4 筆資料之間是以 & 符號分間, 行末則加上換行指令 \\。若要調整本行與下一行之行距, 可以在換行指令之後

國家	央行 獨立性	物價 上漲率	支出 比率
意大利	0.5	16.1	35.6
英國	2	12.3	28.4
加拿大	2	8.1	23.1
西德	4	4.1	29.3

```
\begin{tabular}{lccc}
\hline
  & 央行& 物價& 支出 \\[-2pt]
國家 & 獨立性 & 上漲率 & 比率\\
\hline
 意大利 & 0.5 & 16.1 & 35.6\\
 英國   & 2   & 12.3 & 28.4\\
 加拿大 & 2   &  8.1 & 23.1\\
 西德   & 4   &  4.1 & 29.3\\
\hline
\end{tabular}
```

圖 10.4: tabular 指令環境

設定。本例中第一行標題尾端的指令爲 \\[-2pt], 目的是將兩行標題之行距減小一些, 以求標題更清楚。若某一欄資料從缺, 該欄位就留爲空白, 不須輸入任何資料。

排版時, 每一欄的寬度如何決定呢? 在 tabular 指令環境中, 欄寬可以自行決定, 也可以讓 LaTeX 計算決定。本例中, 欄寬是由 LaTeX 計算決定, 計算方法如下: LaTeX 檢查某一欄下各項資料排版之後的寬度。選擇其中最寬者, 左右再加上適當空白, 結果就是該欄的寬度。本例中我們加入 3 條水平線, 畫水平線的指令爲 \hline。連續兩道 \hline 指令將排版出兩條緊鄰橫線。不過, 請記住專家的意見: 表格中避免畫兩條緊接著的橫線。

排版時, LaTeX 將整個表格視爲一個字元。因此若整套表格指令是緊接在一段文字之後, 表格會出現在行文中間, 表格中央將對齊緊接其前之文字。如果希望表格上端對齊前接文字, 應加上 [t] 選項:

```
\begin{tabular}[t]{lrrr}
```

相反的, 若表格下沿要對齊前接之文字, 設定項應爲 [b]。把表格當成是一大字母來處理有一個好處; 排版時, 表格不會被拆開爲兩部分, 上半部分在本頁底下, 下半部分在下一頁的開頭。最後, 如果表格要置於橫跨版面的中央, 僅須將 tabular 指令環境置於 center 指令環境內即可。

因爲整個表格當成是一個字母處理, 因此要將數個表格並排很容易。譬如, 本章開頭之並排表格即以下列指令排版:

國家	央行獨立性	物價上漲率	支出比率
意大利	0.5	16.1	35.6
英國	2	12.3	28.4
加拿大	2	8.1	23.1
西德	4	4.1	29.3

圖 10.5: 表格內加上直線

```
\begin{tabular}{|l|c|c|}
... [左邊表格之指令]
\end{tabular} \hspace{.05\textwidth}
\begin{tabular}{lcc}
... [右邊表格之指令]
\end{tabular}
```

我們使用 \hspace 指令將兩個表格之間隔定爲版面寬度的 5%。仔細觀察圖 10.4, 我們發現欄位標題第一行文字與其上橫線之距離稍嫌小了一些。同樣的, 表格內第一橫行文字與其上橫線之間距也稍嫌小了一些。主要原因是表內含有中文字。一般而言, 中文字較英數文字爲高。若利用現成的表格格式排版表格, 通常須作些許調整。

欲拉大表格內每一橫行之間距, 可以使用 \extrarowheight 指令。譬如, tabular 指令環境之前加上:

```
\extrarowheight=2pt
```

即可將行距加大 2pt。

圖10.4的表格並未加上任何垂直線。排版專家的意見是表格中不應該加入任何垂直線。如果非加上垂直線不可, 只須在 tabular 指令環境的設定項中加上代表直線的指令 | 即可。舉例來說, 若設定項變成:

```
\begin{tabular}{|l|r|r|r|}
```

如果是連續下兩個直線指令 ||, 排版後即出現兩條緊鄰的垂直線。排版之後, 各欄資料之前後將會出現一直線, 如圖 10.5所示。請注意, 我們已將 2-4 欄文字改成居右排版, 表格上端只畫一橫線, 而且行高已加大 2pt。另外, 標題文字之後我們輸入兩道 \hline 指令, 畫出兩條水平直線。

意大利	0.5	16.12
英國	2	12.3
加拿大	2	8.1
西德	4	4.15

```
\begin{tabular}{rlr@{.}l}
\hline
意大利  & 0.5 & 16& 12 \\
英國    & 2   & 12& 3 \\
加拿大  & 2   &  8& 1 \\
西德    & 4   &  4& 15 \\
\hline\\
\end{tabular}
```

圖 10.6: 表格數字上下對齊

10.2.2 控制欄位間距

表格中各欄位間之空隔距離內設值爲 12pt, 但此一間距可以更改之。要將間距減爲 6pt, 僅須在 tabular 指令環境之前加上下列指令:

```
\tabcolsep=3pt
```

換言之, \tabcolsep 指令設定欄位間距之一半值。

以上之欄距指令同時更動所有欄位之間距。如果我們只想要更動某兩欄位之間距, 可以使用 @{...} 指令。譬如, 圖 10.4 第 2 欄與第 3 欄之間距若想設定爲 0.1 公分, 指令爲:

```
\begin{tabular}{lc@{\hspace{1mm}}cc}
```

在此設定下, 第 2 欄位與第 3 欄原有之間距被取消, 代之以選定之距離。事實上, @{...} 指令大括號內除了設定欄位間距外, 也可以鍵入任何文字或指令。排版時, 括號內之文字或宣告指令即自動填入表格中對應的欄位間隔處, 原有之空白自動消除。底下舉一個例子說明其用途。

在專業排版中, 表格內的數字若有小數點, 排版時小數點應上下對齊。就此點而言, 本章一開頭之表格並不符合標準。若使用 @{...} 指令, 原來表格可重新排版如圖 10.6。本例中, 排版之後有 3 欄, 但第 3 欄之數字爲了特別處理小數點上下對齊的問題, 整數與小數部份各占用一欄排版。設定指令 @{.} 的作用是在表格的對應欄位處排版小數點。左欄使用 r 設定指令, 用以排版整數; 右欄使用 l 指令, 用以排版小數。因此兩欄位所鍵入之整數及小數, 連同小數點即組合成正確的數字。

要注意的是, 如果輸入時我們鍵入 @{ . }, 亦即小數點前後各留下一空白, 排版之後小數點前後也會出現空白。因此, @{...} 指令的功能是先

	經濟表現	
國家	物價 上漲率	政府支出 比率
意大利	16.1	35.6
英國	12.3	28.4
加拿大	8.1	23.1
西德	4.1	29.3

```
\usepackage{booktabs}
\begin{tabular}{@{}lrr@{}}
\toprule
& \multicolumn{2}{c}{經濟表現}\\
\cmidrule(l){2-3}
      & 物價    & 政府支出\\-2pt
國家  & 上漲率 & 比率 \\
\midrule
意大利  & 16.1  & 35.6 \\
英國    & 12.3  & 28.4 \\
加拿大  &  8.1  & 23.1 \\
西德    &  4.1  & 29.3 \\
\bottomrule
\end{tabular}
```

圖 10.7: booktabs 巨集套件

去掉原來內設之任何空白, 再將大括號中任何文字或指令原封不動地排版出來。以上的方法雖然可以得到正確的結果, 但稍嫌麻煩。第 10.6.1 節將介紹 dcolumn 巨集套件, 可以較容易地解決小數點對齊的問題。

10.2.3 booktabs 巨集套件

利用 tabular 指令環境內之 \hline 指令所畫之表格橫線, 粗細都相同。若要畫不同粗細之橫線, 可以使用 booktabs 巨集套件。巨集指令之作者 Simon Fear 爲一專業排版者, 利用此巨集套件排版之表格, 結果相當優美, 值得推薦使用。

在 booktabs 巨集套件下, 我們可以使用下列橫線指令:

```
\toprule[wd]
\midrule[wd]
\bottomrule[wd]
\cmidrule[wd](trim){a-b}
```

前三行指令分別用於畫表格上方、中間及底下之橫線; [*wd*] 選項用於設定線條之粗細。若上方橫線之粗細要定爲 1pt, 指令爲 \toprule[1pt]。若不加選項, 程式會自動選用較粗之線條畫 \toprule 與 \bottomrule。

圖 10.7 之表格有三欄資料, 我們使用 \toprule 等指令畫三條粗細不同之橫線, 其長度與表格寬度相同。請注意, 橫線上下方之空白會自動加大,

不須再調整。中間的短橫線只貫穿兩個欄位，我們使用 \cmidrule(l){2-3} 畫出。{2-3} 設定短線橫跨第 2-3 欄位；圓括號 (l) 選項表示橫線左方要切短一點。若短線左右端都要截短，指令爲 \cmidrule(lr){2-3}。

若連續使用兩個 \cmidrule 指令，排版結果兩條短線會接在一起，無法分辨。爲解決此問題，可將左邊短線之右端及右邊短線之左端各切下一小截。若表格共有 5 欄，2-3 欄及 4-5 欄上方各要畫一短線，指令爲：

```
\cmidrule(r){2-3}\cmidrule(l){4-5}
```

請注意，兩道指令之間不可留空格，否則兩條線無法水平對齊。同一欄位上若設定兩道指令，將排出緊接之兩條短線，例如：

```
\cmidrule(l){2-3} \cmidrule(l){2-3}
```

2-3 欄短橫線上方的「經濟表現」四個字占用兩個欄位。欲排版占用兩欄位以上之文字標題，可使用下列指令：

```
\multicolumn{n}{col}{text}
```

其中，*n* 代表占用欄位數；*col* 設定要居中、靠右或靠左。本例中，文字占用兩欄，並居中排版。

圖 10.7 之表格例子中，欄位指令選項之前後端各加上一個 @{} 指令，其目的是除掉多餘的空白。這是專業人士建議的作法。若不加上此設定，表格內容之寬度將比橫線小一些。本例中，\cmidrule 是 booktabs 巨集套件特別提供的指令。如果直接使用 tabular 指令環境排版，指令爲 \cline{2-3}。請注意，\cline 指令之後僅能設定橫跨欄位數，不能控制線條粗細，也無法像 \cmidrule 指令設定 *trim* 之功能。如果我們以 \hline 替代 \toprule 等橫線指令，以 \cline{2-3} 替代 \cmidrule 指令重新排版，結果如下：

	經濟表現	
國家	物價上漲率	政府支出比率
意大利	16.1	35.6
英國	12.3	28.4
加拿大	8.1	23.1
西德	4.1	29.3

同 學 會 通 訊 錄

姓　名	電　　話
陳思源	(02)2345-6789
吳　文	(02)3456-6789
黃子文	(02)2234-3789

```
\usepackage{booktabs}
\begin{center}
\csp{3cm}{\k11 同學會通訊錄}\\2pt
\begin{tabular}{lc}
\toprule
\csp{1cm}{姓名} & \csp{1.5cm}{電話}\\
\midrule
\csp{1cm}{陳思源} & (02)2345-6789 \\
\csp{1cm}{吳文}   & (02)3456-6789 \\
\csp{1cm}{黃子文} & (02)2234-3789 \\
\bottomrule
\end{tabular}
\end{center}
```

圖 10.8: 調整中文字距

10.2.4　控制中文字距

以上的表格例子並未加上標題。欲在圖表加上標題, 可使用 \caption 指令。不過, 此一指令僅能下於浮動版面指令環境內, 我們將在稍後說明使用的方法。底下先說明自行排版標題的方法。圖 10.8 的例子爲了使標題各中文字之間距稍微拉開一些, 我們特別定義一巨集指令 \csp 如下:

```
\newcommand{\csp}[2]{\let\zori=\z \let\z=hfill
 \makebox[#1]{#2}\let\z=\zori}
```

請注意, 巨集指令必須定義於全文設定區。此外, 此巨集指令僅能用於中文字, 英文字串無效。

巨集指令 \csp 之原理如下: cwtex 處理中文時, 會在兩個中文字中間加上指令 \z, 目的是在控制中文字之間距。而 \csp 巨集指令將 \z 重新定義爲一伸縮彈性無窮大之彈簧。因此, 當我們限制標題寬度之後, 各中文字之間距會盡量加大, 產生均勻空白之效果。使用巨集指令時須先填入標題長度, 之後再填入文字。圖 10.8 的例子中, 標題長度定爲 3cm, 並以楷體字排版。我們將標題與表格全部置於 center 指令環境內, 排版之後標題居於表格中央。除了標題之外, 本例中之人名也是以 \csp 指令排版, 寬度爲 1 公分。如果姓名只有兩個字, 排版時, 兩字間距會自動拉開。

依同一原理, 我們也可以設定固定之字距, 譬如, 下列巨集指令將字距拉大爲 2mm, 並給與一點的伸縮彈性:

同學會通訊錄

姓　名	電　話
陳思源	(02)2345-6789
吳　文	(02)3456-6789
黃子文	(02)2234-3789

圖 10.9: 加大中文標題之字距

```
\newcommand{\cspp}[1]{\let\zori=\z
\def\zx{\hskip 2mm plus0.2pt minus0.1pt}
\let\z=\zx \mbox{#1}\let\z=\zori}
```

圖 10.8 排版標題之指令若改爲:

```
\cspp{\k11 同學會通訊錄}\\2pt
```

排版結果將如圖 10.9 所示。

以上指令的另一個應用是重新設定文稿之中文字距。cwTEX 內定之中文字距爲 0pt, 再加上一點伸縮彈性, 但有人認爲此一字距太小。欲加大中文字距, 最簡單的方法是在全文設定區加上下列兩行指令:

```
\def\zx{\hskip 2pt plus0.2pt minus0.1pt}
\let\z=\zx
```

第 1 行定義一新指令 \zx, 設定標準字距爲 2pt; 第 2 行指令將內定之字距改爲新設定之字距。

以上方法雖然簡單, 但如果每一篇文稿都希望加大字距, 更簡單的方法是在執行 cwtex 時加上 -z 選項。譬如, 執行 cwtex 時加入下列選項, 其效果與上面兩行指令相同:

```
c:\xtemp>cwtex -z+0.2 test
```

如果你是使用 WinEdt 文字編輯軟體, 直接在 [F9] 功能鍵之設定中加入上述選項, 即可一勞永逸地重新設定字距。

10.2.5 表格內的文字段落

以上所舉的表格例子裡, 欄寬是由 LaTeX 所決定。有時候, 表格某一欄的內

項目	分數	評述意見
方法	85	本研究所使用的方法是由作者本人所發展出來的。因此，作者對此一方法的運用頗爲得心應手。
貢獻	88	從理論和實際應用來看，本研究可以說都很有貢獻。
文字	85	甚佳。

註: 以上爲虛構, 如有雷同, 純屬巧合。

```
\usepackage{booktabs}
\begin{tabular}{lcp{2.5cm}}
\toprule
 項目 & 分數 & 評述意見 \\
\midrule
方法 & 85 & 本研究所使用 ...
...
... 頗為得心應手。\\2pt
貢獻 & 88 & 從理論和實際應用來看，
本研究可以說都很有貢獻。\\2pt
文字 & 85 & 甚佳。\\
\bottomrule
\end{tabular}
\par\smallskip
\parbox{4.5cm}{註: 以上為虛構，
如有雷同，純屬巧合。}
```

圖 10.10: 表格內之文字段落

容是整段文字, LaTeX 無法計算其寬度, 欄寬必須自行設定。若想把某一欄位的寬度設爲2.5公分, 可以在 tabular 指令環境中選用 p{2.5cm} 選項, 請見圖 10.10之例子。

圖 10.10之表格例子共有 3 欄, 前兩欄的寬度由 LaTeX 計算決定, 第3 欄以 p{2.5cm} 指令固定爲 2.5 公分。在固定寬度的表格欄裡, 我們可以輸入多個段落。本例中第一欄位內僅包含一個段落。如果有兩個段落, 第一段結束處應加上 \par 指令。要注意的是, 第 2 段落行首並不會內縮; 若有必要, 我們可以在段落開始之處下 \hspace*{...} 指令, 以產生適當的空白。

圖 10.10同時說明如何以 \parbox{...} 指令在表格下面輸入註解說明。LaTeX 將 tabular 指令環境內全部文字視爲一個大字母,因此排版註解之前, 我們必須先以空白行或 \par 指令結束表格本身之段落。否則, 註解將會排版在表格的右方而不是底下。本例中, 段落指令之後先以 \smallskip 指令拉大空白; 其下之註解是以 \parbox 指令編排。因爲是新的段落, 因此起頭會自動內縮一點距離 (indent)。欲取消開頭之空白, 應加上 \noindent 指令。第 10.6.3節將介紹排版表格註解之 threeparttable 巨集套件。

圖 10.10例子所使用的 p{2.5cm} 指令事實上等於是 \parbox[t]{2.5cm}, 選項 [t] 設定將段落文字之上端對齊該行其他欄位之文字或數字。因此, 第

一行第 2 欄之數字 “85” 即和第 3 欄文字段落的首行對齊。我們也可以使用 m{2.5cm} 或者 b{2.5cm}。前者將段落文字的中央對齊其他各欄之文字或數字; 後者則將段落文字之底端對齊其他各欄文字。爲了對照比較, 底下的表格例子中之段落是使用 m{4cm} 排版:

```
\begin{tabular}{lcm{4cm}}
```

我們另以 \midrule 指令加上橫線以區隔三項資料。請特別注意, m{...} 與 b{...} 是 array 巨集套件所提供的指令。欲使用這個指令排版, 須先於全文設定區引用之。

項目	分數	評述意見
方法	85	本研究所使用的方法是由作者本人所發展出來的。
貢獻	88	從理論和實際應用來看, 本研究可以說都很有貢獻。
文字	85	甚佳。

10.2.6 其他控制與設定指令

除了上述指令之外, 表格排版的指令還包括畫線指令: \hline, \vline, 與 \cline, 及多欄位指令: \multicolumn。其中, \hline 指令用於在表格內畫橫線, \multicolumn 指令用於排版橫跨兩欄位以上之文字。\cline 指令與 booktabs 巨集指令之 \cmidrule 類似, 但功能較簡單。例如, \cline{2-3} 可用以排版橫跨 2-3 欄之短橫線。最後, \vline 指令用於在特定欄位處排版短垂直線, 高度與欄高相同。tabular 指令環境另外還有一些排版與設定指令, 合併列舉於表 10.1。請注意, 部分指令是由 array 巨集套件所提供, 欲使用這些指令須先引入巨集套件。

以下以幾個例子說明表格指令之用法。首先, 我們介紹 !{...}, >{...} 與 <{...} 指令之應用。以上三個指令是 array 巨集套件所提供, 故須於全文設定區引用之。!{...} 指令是垂直線 | 指令的擴充; 在大括號中填入的任何文字或指令將替代原垂直線。圖 10.11 之表格例子中, 輸入指令爲 !{$\Box$}, 因此三個小正方形出現於表格左邊原垂直線之位置。

表 10.1: tabular 排版與設定指令 (array 巨集套件)

排版指令	
l	將文字資料排版於欄位左方,
c	將文字資料排版於欄位中央,
r	將文字資料排版於欄位右方,
|	加上垂直直線,
p{...}	排版文字段落, 其頂端對齊其他各欄,
m{...}	排版文字段落, 其中央對齊其他各欄,
b{...}	排版文字段落, 其底端對齊其他各欄,
@{...}	將文字或宣告指令加於兩欄位中間, 並去掉原有之欄位間距,
>{...}	此指令可以加於 l,r,c,p,m,b 選項之前, 其功能是將括號內之宣告指令施用於欄位內文字之前,
<{...}	功能類似上記指令, 但將宣告指令施用於欄位內文字之後,
!{...}	此指令擴充上記 | 之功能。若以此項指令替代 |, 排版之後出現的不是直線, 而是括號中之文字或符號。
設定指令	
\tabcolsep	tabular 指令環境下兩欄位間距之一半值, 內設爲 6pt。
\arraycolsep	array 指令環境下兩欄位間距之一半值, 內設爲 5pt。
\arrayrulewidth	直線與橫線條之粗細, 內設爲 0.4pt。
\doublerulesep	連線兩直線 (||) 或兩橫線 (\hline\hline) 之間距, 內設值爲 2pt。
\extrarowheight	此設定值用以加大橫行之間距, 但其深度 (depth) 維持不變。換言之, 僅基線上方之距離加大。
\arraystretch	設定行距之倍數。譬如, 若設定爲 1.5, 則橫行間距將爲原有之 1.5 倍。內設值爲 1。

☐ Brazil	33.92%
☐ China	8.82%
☐ India	8.37%

```
\usepackage{array,latexsym}
\begin{tabular}{!{$\Box$}%
  >{\sffamily}lr<{\%}}
\toprule
Brazil & 33.92 \\
China  &  8.82 \\
India  &  8.37 \\
\bottomrule
\end{tabular}
```

圖 10.11: 表格指令之應用

相對而言, `>{...}` 指令的作用不是替代特定欄位原有之內容, 而是附加於欄位內容之前; `<{...}` 指令則是將大括號內之指令或文字附加於欄位內容之後。本例中, 附加於前之指令爲 `>{\sffamily}l`, 因此表格第一欄位居左, 且以 sans serif 字體排版。附加於後之指令爲 `r<{\%}`, 因此本例中第2欄位內之數字靠右, 其後自動附加 % 符號。

在 `tabular` 指令環境中, 利用 `|` 設定指令可畫出垂直線, 但其粗細不能自己控制。利用上圖介紹的 `!{...}` 指令可以畫出任意粗細之垂直線。譬如, 上例中之指令若改爲:

```
!{\vrule width 2pt}
```

則表格左邊三個小正方形將改變爲粗細爲 2pt 之垂直線。指令中之 `\vrule` 爲 TEX 指令, 2pt設定粗細。

垂直線指令 `|` 或 `!{\vrule width 2pt}` 所畫出之直線由上至下貫穿整個表格。如果只要在某一橫列中間畫短直線, 可使用 `\vline` 指令。圖10.12例子中使用兩個 `\vline` 指令, 畫出貫穿兩橫列之垂直線。

表 10.1 指令中, 除了 `\arraystretch` 指令必須以 `\renewcommand` 指令重新定義之外, 其餘各指令可以在 `tabular` 指令環境之前直接設定新值。例如, 若欄高要加大 2pt, 兩欄位之間距要加大 2pt, 指令如下:

```
\extrarowheight=2pt
\addtolength{\tabcolsep}{1pt}
```

如果要把橫距變爲原來的 1.2 倍, 請在全文設定區加入下列指令:

```
\renewcommand{\arraystretch}{1.2}
```

單位	國立台灣大學	
主持人	陳大川	322-4433
聯絡人	張小河	733-3344
地址	台北市羅斯福路	

```
\extrarowheight=2pt
\begin{tabular}{|l|l|}
\hline
單位 & 國立台灣大學\\
\hline
主持人 & 陳大川~~\vline%
  ~~322-4433\\
聯絡人 & 張小河~~\vline%
  ~~733-3344\\
\hline
地址   & 台北市羅斯福路 \\
\hline
\end{tabular}
```

圖 10.12: 表格中短直線

10.3 tabularx 巨集套件

對應於 tabular 指令環境, 另有一個 tabular* 指令環境, 兩者不同的地方在於後者可以設定表格的寬度。若表格有 3 欄, 而排版指令爲:

```
\begin{tabular*}{6cm}{lrr}
```

則表格寬度將爲 6 公分, 每一欄之寬度仍由 LaTeX 各欄之文字或數字自動計算大小。但是, 因爲表格寬度已設定爲6 公分, 因此 LaTeX 自動把欄距拉大, 以使表格寬度等於設定之值。顯然, 如果寬度設定值不適當, 排版結果不會太理想。

要排版固定寬度之表格, 另外一個辦法是使用 tabularx 巨集套件, 作者爲 David Carlisle。此巨集套件亦讓排版者指定表格寬度, 但它的特點是可以自動算出特定欄位之寬度。換言之, 在此指令環境之下, 表格欄距仍維持內設值, 各欄之寬度可以自動調整。反之, tabular* 指令環境則是固定欄位寬度, 欄位間距則自動調整以達到選定的表格寬度。

圖 10.13 之表格例子是使用 tabularx 巨集指令所排版。此例中, 另外使用了 array 巨集套件之 !{...} 指令, 故須同時引用之。第一行指令甚長, 故拆爲 3 行排出。首先, 表格寬度定爲 4.5 公分, 內含三欄資料; 2、3 欄設定居中編排, 由 LaTeX 自動計算欄寬。第一欄設定項爲 X, 指示 LaTeX 由表格寬度4.5 公分減去 2、3 欄寬度之後, 其餘用以排版第一欄。本例中, 表格左右

製造商	今年 排名	去年 排名
中國鋼鐵	1	2
宏碁電腦	2	3
南亞塑膠	3	1
建元電子	4	4

```
\usepackage{tabularx,array}
\begin{tabularx}{4.5cm}{%
  !{\vrule width 2pt}%
  X|cc!{\vrule width 2pt}}
\hline
製造商 & 今年 & 去年\\-2pt
       & 排名 & 排名\\
\hline
中國鋼鐵 & 1 & 2 \\
宏碁電腦 & 2 & 3 \\
南亞塑膠 & 3 & 1 \\
建元電子 & 4 & 4 \\
\hline \end{tabularx}
```

圖 10.13: tabularx 巨集套件

各畫了一條2pt 粗細之垂直線, 使用之指令爲!{\vrule width 2pt}。此一指令之功能上一小節已說明, 此處不再重覆。

使用 tabularx 巨集套件時, 我們也可以設定將每一欄都以 X 選項編排。譬如, 若以下列指令排版表格:

```
\begin{tabularx}{6cm}{|X|X|X|}
```

則 LaTeX 會將6公分平均分到3個欄位去。因爲設定各欄位之前後畫有垂直線, 而直線本身有寬度, 因此每一欄位之寬度將比2公分略小一些。當表格之欄位設定以 X 指令排版時, 欄位內之文字將以上一節介紹之 p{...} 指令, 也就是 \parbox[t]{...} 指令編排。如果希望欄位內之文字以 m{...} 指令格式編排, 必須在 tabularx 指令環境之前以下列指令重新定義:

```
\renewcommand{\tabularxcolumn}[1]{>m{#1}}
```

重新定義之後, X 指令欄位下之文字段落將以 m{...} 方式排版。亦即, 欄位內之文字段落中間會對齊其他欄之文字或數字。

10.4 表格標題與位置

專業排版中, 較大的圖表通常是移於在版面的上下端, 或者單獨占一整頁。LaTeX 提供 figure 與 table 指令環境; 前者用於控制圖位置, 後者用以控制

表格之位置。在 LaTeX 中, 這兩個指令環境合稱爲 float (浮動版面); 排版時, 指令環境內之文字圖表將獨立安排於版面頁之上下方, 或自成一頁。

除了控制圖表位置外, 指令環境內尙可使用 `\caption` 指令編排標題。在控制圖表位置的功能上, 這兩個指令環境並無不同, 唯一的差別是在排版標題時, `table` 指令環境內之標題會自動加上 Table 一字; `figure` 指令環境下則自動加上 Figure 英文字。本章主要介紹表格之排版, 但因爲 `figure` 與 `table` 指令環境大同小異, 故在此一併介紹。

10.4.1 浮動版面指令環境

浮動版面內通常置放排版圖或表格之指令, 但也可以是單純的文字段落。使用 `table` 指令環境控制表格位置, 指令如下:

```
\begin{table}[pos]
 ...
\caption{...}
\end{table}
```

所有排版表格之指令即置於指令環境中。本例中, 標題指令 `\caption` 置於指令環境末端, 實際上它可以置於指令環境之前端, 甚或任何位置。本例中, `table` 指令環境內用於排版表格。若是置放圖形, `table` 應改爲 `figure`。如果文稿是以兩欄式 `twocolumn` 編排, 則浮動版面須使用 * 型式, 亦即 `table*` 與 `figure*`。

顧名思義, 浮動版面是將特定文字段落或圖表另找位置排版。LaTeX 決定圖表的位置時, 首先計算圖表本身的高度, 其次要了解本頁版面剩下多少空間等等。即使圖表可以擠入版面中, 但若剩餘供排版文字的空間太小, 如此安排也不適當。LaTeX 是透過一套複雜的計算過程, 才決定浮動版面的位置。欲了解詳情, 請見 Goossens, Mittelbach, and Samarin (1994), 第 6 章。在計算圖表之位置時, LaTeX 會參考一些控制值。如果你認爲 LaTeX 所選擇之圖表位置不適當, 可試改用圖 10.14 之設定值。這些修正值是許多使用者長時間測試之後推薦使用的, 設定值須於全文設定區定義。

如果對於 LaTeX 的選擇不滿意, 可在 `table` 與 `figure` 指令環境中的位置選項 *pos* 另行設定。位置選項變數如下:

```
\renewcommand{\textfraction}{0.15}
\renewcommand{\topfraction}{0.85}
\renewcommand{\bottomfraction}{0.65}
\renewcommand{\floatpagefraction}{0.60}
```

圖 10.14: 浮動版面位置設定值

h (here): 圖表置於現址,

t (top): 圖表置於本頁上端,

b (bottom): 圖表置於本頁下端,

p (page): 圖表自成一頁。

以上 4 個位置選項變數可以進一步組合。如果不加選項, 內定值爲 [tbp]。簡單來說, 這指示 LaTeX 盡量設法將表格置於版面上端; 若不成, 則置於版面下端; 若再不成, 則讓表格自成一頁。

下指令時, 位置選項變數之順序無關緊要, LaTeX 永遠依照 h-t-b-p 之順序尋找適當位置。例如, 不管選項是 [bh] 或 [hb], 尋找位置之順序都是 h-b。如果選項中僅有一個變數, 如 [t], [b], 或者 [h], 排版時可能出現問題; 因爲可供選擇的彈性太小, LaTeX 有可能將圖表移至文稿最後一頁。因此, 選項越多, LaTeX 越能妥善處理。一般而言, 不加任何選項已能獲得良好結果。如果要 LaTeX 首先考慮將圖表置於行文中, 可試用 [htbp] 或 [htp] 等。有關於浮動圖表指令環境之使用, 請參見 Reckdahl (1997) 之說明。

如果你設計的版面特別, 可在下位置選項變數時加入 !, 例如:

```
\begin{table}[!ht]
```

此項設定要求將表格盡可能地置於行文指令處, 若不成則置於版面上方。

10.4.2 圖表標題

浮動版面指令環境之內可以使用 \caption{...} 排版標題。在 table 指令環境中, 標題之前會自動加上 Table 英文字並編入號碼, 如 “Table 3:”, 其後才是表格標題。若是 figure 指令環境, 標題前之字樣爲 “Figure 9:”。

標題指令可置於圖表上方或下方, 標題與圖表之間距有內設值。LaTeX 假設標題是置於圖表下方, 排版時標題上方會自動拉大10pt 間距, 下方則無額外空白。如果要把標題排版於表格上方, 則標題之下應留一些空白, 上

方空白反而是不必要的。若內設之間距不適用, 也可重新設定。若標題是置於圖表下方, 在全文設定區加入下列指令即可重新設定間距爲 15pt:

```
\abovecaptionskip=15pt
\belowcaptionskip=0pt
```

如果圖與表採不同方式排版標題, 圖標題置於下方, 表標題置於上方, 則以上之設定並不適用。解決方法是定義一 `\topcaption` 指令排版表標題:

```
\newcommand{\topcaption}{%
\setlength{\abovecaptionskip}{0pt}%
\setlength{\belowcaptionskip}{10pt}%
\caption}
```

定義指令後, 圖標題仍以 `\caption` 指令編排, 表標題則改用 `\topcaption` 指令排版。

如上所述, `table` 與 `figure` 指令環境唯一的差別是在於標題指令所加上之字樣不同。若排版英文稿, LaTeX 自動加入之 Table 與 Figure 字樣並無問題。但在中文稿中, 較理想之表格標題是以「表 3:」起頭; 圖標題以「圖 9:」起頭。修改辦法很簡單, 只須在全文設定區加入以下兩道指令即可:

```
\renewcommand{\tablename}{\m11 表}
\renewcommand{\figurename}{\m11 圖}
```

以上指令中, `\tablename` 原定義爲 Table 一字, 現由指令改爲中文字「表」。如果要選用 12 點之圓體字, 則 `\m11` 應改爲 `\r12`。

使用 `\caption` 指令排版標題時, 圖表會自動編上號碼。在 `book` 文件類別下, 若本章爲第 10 章, 圖表之號碼將以 10.1, 10.2 之形式出現。若要改變圖表編號之字體與格式, 最簡單的方法是使用 caption2 巨集套件, 作者爲 Harald Sommerfeldt。欲使用 caption2 巨集套件, 須在全文設定區加入下列指令:

```
\usepackage[option]{caption2}
```

大部分的設定是經由加入選項爲之。

圖 10.15 列出 caption2 巨集套件之選項, 其下爲輸入之指令。選項可分爲四類, 第一類是控制標題排版方式。若標題文字超過一行, 加入第一類選

項即可控制排版方式。譬如, normal 格式是一般排版文章段落的方式, 亦即, 右邊將切齊; 而且第一行並不內縮。center 選項之格式是標題各行皆居中, centerlast 則是最後一行居中, 前面各行以 normal 方式編排。如果圖表標題不超過一行長度, 標題將居中編排, 不依以上之設定格式。若單行之標題也要與多行標題同樣方式編排, 選項中須加入 nooneline。

第 2、3 類之指令是控制編號字體及大小。所謂「編號字體」是指標題文字前諸如 "Table 10.3:" 字串。字體選項中若選用 sf, 則上述字串將排版爲 "Table 10.3:"。若再加入 large 選項, 字體將加大一些。若標題中文字是以粗黑體或圓體排版, 則加入 sf 選項可使標題編號數字之字體與標題中文字較爲一致。

本例子之標題選用 normal 選項, 因此標題文字左右兩邊將切齊。不過, 本例之標題文字雖然占 2 行, 事實上僅有第一行是眞正的標題, 第 2 行是表格說明。爲了排版說明文字, 指令中使用了換行指令。\caption 指令內雖然可使用換行指令, 不過, 某些 LaTeX 指令不能直接用於浮動版面指令環境內, 換行指令就屬於這一類。若要使用, 其前須加上 \protect。故本例之換行指令變成 \protect\\[2pt]。

輸入之指令中有 \setcaptionwidth, 其作用是設定標題之寬度。本例子設定標題寬度爲行長的 0.77。其他設定指令請參考 Reckdahl (1977)。

10.5 引述表格

圖 10.15 例子中, \caption 指令之後緊接著一行 \label{tabcap} 指令, 其中 tabcap 是使用者自行定義的標誌 (marker), 目的是方便引用圖表。請注意, \label 指令須緊接在 \caption 指令之後。

定義了標誌之後, 文稿內任何地方都可使用 \ref{tabcap} 指令引用圖表; 或者以 \pageref{tabcap} 引用圖表出現之頁碼。舉例言之, 行文中欲排版下列句子:

> ...如第 206 頁之圖 10.15 所示 ...

輸入指令如下:

```
如第\Z\pageref{tabcap}\Z 頁之圖\Z\ref{tabcap}\Z 所示 ...
```

標題排版格式	normal, center, flushleft, flushright, centerlast, hang, indent
編號字體大小	scriptsize, footnotesize, small, normalsize, large, Large
選用編號字體	up, it, sl, sc, md, bf, rm, sf, tt
單行標題	oneline, nooneline

圖 10.15: caption2 巨集套件

此巨集套件可控制標題排版格式、字體與編號。

```
\usepackage[oneline,normal]{caption2}
\begin{table}
\centering
\setcaptionwidth{.77\textwidth}
\begin{tabular}{lm{.55\textwidth}}
\toprule
標題排版格式  & ...
...
單行標題   & |oneline|, |nooneline|\\
\bottomrule
\end{tabular}
\caption{\textsf{caption2} 巨集套件\protect\\[2pt]
{\small \m10
 此巨集套件可控制標題排版格式、字體與編號。}}
\label{tabcap}
\end{table}
```

LaTeX 會自動計算表編號與頁碼, 代入句子中。請注意, 排版時須執行 latex 兩次, 才能計算出正確的頁碼與圖表編號。

以上所輸入的文字中, 標誌指令前後所加上之 \Z 是 cwTeX 控制間距之指令, 目的是調整中文字與阿拉伯數字之間距。若不加上 \Z, 仍可得到結果, 但間距會嫌太小。

10.6 表格排版細節調整

以上所介紹之指令與巨集套件已能處理絕大部分之表格排版。不過, 有一些細節還是不容易處理。譬如, 表格中之數字中若有小數點或逗號, 上下列數字之小數點或逗號應對齊。圖 10.2 曾說明如何使用 ~ 指令作人爲調整; 表 10.6 則說明以 @{.} 指令排版上下對齊之小數點。但是, 前一方法並不精確; 後一方法則稍嫌麻煩。

若文稿中的表格有許多小數點數字, 欲讓小數點上下對齊, 可使用 dcolumn 巨集套件。另外, 有些表格第 2 欄有兩項資料, 而第 1 欄僅有一項資料。排版時, 第 1 欄資料須水平對準第 2 欄上下兩列資料中間。欲排版此種表格, 可使用 multirow 巨集套件。如果表格下方要加上說明、註解, 或者引述資料來源, 最簡單的方法可能是使用 threeparttable 巨集套件。本節介紹一些巨集套件以解決這些林林總總的問題。

10.6.1 表格數字上下對齊

表格中的數字若有小數點或逗點等, 爲求美觀起見, 這些符號位置應上下對齊。表 10.6 曾說明如何使用 @{.} 設定指令來排版上下對齊的小數點, 但是其方法稍嫌複雜。本小節所介紹的 dcolumn 巨集套件提供比較精確的控制方法。

在 dcolumn 指令環境內, 可使用下列指令設定欲上下對齊之小數點:

```
D{鍵入符號}{輸出符號}{小數點位數}
```

所謂「鍵入符號」是指使用者所輸入之符號,「輸出符號」則指 LaTeX 所排版出的符號。例如, 我們可設定成鍵入逗號, 但排版出小數點。指令中之「小數點位數」用以設定所排版數字之小數點位數。

國民所得支用面

項目	金額	比率
民間消費	$13,665	46.7%
國內投資	5,066	17.3
政府消費	4,229	14.5

```
\usepackage{dcolumn,booktabs}
\newcolumntype{d}[1]{D{,}{,}{#1}}
\newcolumntype{.}[1]{D{.}{.}{#1}}
\begin{tabular}{cd{3}.{3}}
\multicolumn{3}{c}{國民所得支用面}\\
\toprule
項目  &\multicolumn{1}{c}{金額}&
 \multicolumn{1}{c}{比率}\\
\midrule
民間消費 & \$13,665 & 46.7\rlap{\%}\\
國內投資 &    5,066 & 17.3\\
政府消費 &    4,229 & 14.5\\
\bottomrule
\end{tabular}
```

圖 10.16: dcolumn 巨集套件

爲了方便起見, 我們可以進一步使用 \newcolumntype 指令自行定義較簡化的設定指令。例如, 如果鍵入與輸出符號都是小數點符號, 我們可以定義一設定指令 .{...} 如下:

```
\newcolumntype{.}[1]{D{.}{.}{#1}}
```

以上指令所定義之設定指令, 輸入與輸出符號皆爲英文句點, 使用指令時小數點位須填入爲指令選項。

圖 10.16 提供一個應用例子。在此例子中, 除了 .{...} 指令外, 我們還定義另一個設定指令 d{...}, 其鍵入與輸出符號都是逗號。使用 d{...} 與 .{...} 設定指令時, 括號中應填入數字之小數位數。如果各行數字的小數點位數不相同, 則須填入最大的位數。如果括號中填入 -1 , 則表格中之數字小數點的位數不管幾位都可以。但是, 排版時「小數點」將排於該欄位的正中央。

圖 10.16 的表格是以底下指令排版:

```
\begin{tabular}{cd{3}.{3}}
```

表格計有 3 欄; 第 1 欄之文字居中排版, 2、3 欄內都是數字。其中, 第 2 欄三個數字都是整數, 中間夾有逗號。爲了使逗號上下對齊, 第 2 欄以 d{3} 來設定, 其中的 3 指示「小數點」之後有3位數。從這個例子可以看出來, 所謂的

年期	產量
1991−1992	934,345
1992−1993	1, 134,045

```
\usepackage{dcolumn}
\newcolumntype{a}[1]{D{-}{-}{#1}}
\newcolumntype{b}[1]{D{.}{,}{#1}}
\begin{tabular}{|a{4}|b{3}|}
\hline
\multicolumn{1}{|c|}{年期} &
\multicolumn{1}{c|}{產量}\\ \hline
1991-1992 &   934.345 \\
1992-1993 & 1,134.045 \\ \hline
\end{tabular}
```

圖 10.17: 表格數字上下對齊

「小數點」, 其實可以是任意的指定符號。第 3 欄的數字中整數有兩位, 小數有一位, 但是第一個數字之後有百分比符號。本例以 .{3} 設定小數點應上下對齊, 讓小數部分留出一個空白。因爲百分比符號並非數字, 如果直接鍵入, 排版時數字將無法對齊小數點。本例中, 我們使用 TEX 的指令 \rlap 來排版百分比符號。其功能是在 46.7 之後排出百分比符號, 然後再退回到數字 7 之右緣。如此一來, LATEX 在對齊小數點時, 不會考慮到百分比符號的存在。與 \rlap 對應的指令是 \llap; 前者處理右邊之文字符號, 後者處理左邊的文字符號。

有人可能會覺得奇怪, 爲何在定義設定指令時要區別鍵入與輸出符號? 圖 10.17 的例子說明此種區別之用途。此表格中有兩欄數字, 其中, 第 2 欄上下兩個數字位數不同, 上面爲 934,345, 底下爲 1,134,045。我們定義兩個設定指令 a{...} 與 b{...}。在 a{...} 設定指令中, 鍵入與輸出符號同樣都是一短線 (連字號)。但是在 b{...} 設定指令中, 鍵入符號定義爲小數點, 輸出符號則爲逗號。

如果將鍵入與輸出符號都定義爲逗號, 排版時上面數字之逗號將對齊底下數字左邊的逗號, 結果並不正確。爲解決此一困難, 我們利用 b{...} 設定指令, 但鍵入之數字須改爲 934.345 與 1,134.045。排版之後, 小數點會改爲逗號, 並且上下對齊而得到正確的結果。

10.6.2 橫列文字對齊

前面所介紹的 \multicolumn 指令是用於排版橫跨兩欄以上的標題。有時候, 表格中某欄位之數字可能占用數列空間。譬如, 圖 10.18 例子中, 左欄之

投資	政府部門 民間企業
儲蓄	家庭部門

```
\usepackage{multirow,booktabs}
\begin{tabular}{ll}
\toprule
\multirow{2}{1cm}{\k11 投資} &
     政府部門\\
   & 民間企業 \\
\midrule
{\k11 儲蓄} & 家庭部門\\
\bottomrule
\end{tabular}
```

圖 10.18: multirow 巨集套件

「投資」對應右欄之「政府部門」與「民間企業」。欲排版此種表格, 可使用 multirow 巨集套件, 作者爲 Terry Leichter 與 Piet van Oostrum。

此一巨集套件提供 `\multirow` 指令, 內容如下:

`\multirow{`*nrows*`}{`*width*`}[`*fixup*`]{`*text*`}`

第一選項 *nrows* 設定占用多少欄位。本例中,「投資」兩字占用兩橫列。第二選項設定欄位寬度, 本例中爲 1 公分。如果欄位寬度要由 LaTeX 計算決定, 則 `{1cm}` 應以 `*` 替代。第 3 選項 [*fixup*] 可有可無, 其功能是調整垂直位置。本例中未加入此選項, 故不調整。若改加上 `[2pt]`, 則「投資」二字將上移 2pt; 若填入負值, 則文字將往下移。最後一個選項是排版文字。在第一選項之後還可加入一控制垂直間距之選項, 詳情請見巨集套件內附之說明。

圖 10.19 是一個較複雜的例子, 但指令並無不同。爲了避免重複輸入指令之麻煩, 我們在全文設定區定義一簡單巨集指令 `\mr`, 其中設定排版文字將跨占兩橫列空間; 欄位寬度由 LaTeX 自動計算。定義 `\mr` 指令之後, 排版指令即大幅簡化, 較不易發生錯誤。

10.6.3 表格註解

圖 10.10 曾說明如何以 `\parbox` 指令排版表格下方之說明或註解。此一方法須自行設定寬度, 使用上並不方便。欲在表格下方加上註解, 較方便的方法是使用 Donald Arseneau 所寫的 threeparttable 巨集套件。此巨集套件會自動計算表格寬度, 減少來回調整的麻煩。

例	語幹	未然形 -う	連用形 -ない -た	終止形	連體形 -こ	假定形 -ば
寒い 暑い	寒 暑	かろ	く かっ	い	い	けれ

```
\usepackage{multirow,booktabs}
\newcommand{\mr}[1]{\multirow{2}*{#1}}
...
\begin{tabular}{ccccccc}
\toprule
 & & 未然形 & 連用形 & 終止形 & 連體形 & 假定形\\[1pt]
\mr{例} & \mr{語幹}  & \mr{-う} & -ない  &    &
  \mr{-こ} & \mr{-ば} \\
  & & & -た & & & \\
\midrule
寒い & 寒 & \mr{かろ} &く & \mr{い} & \mr{い} & \mr{けれ} \\
暑い & 暑 &           & かっ &  &  & \\
\bottomrule
\end{tabular}
```

圖 10.19: 對齊橫列文字 — multirow 巨集套件

巨集套件 threeparttable 提供一同名的指令環境; 此指令環境可視爲是 table 與 figure 浮動版面指令環境功能之延伸。浮動版面指令環境可將大型圖表排版於版面適當位置,指令環境內尙可使用 \caption 指令排版標題; threeparttable 指令環境亦有類似功能。圖 10.20 例子之內容與圖 10.16 相同, 但新增加了表格註解與說明。指令環境內以 \caption 指令排版標題; 以 tabular 指令環境排版表格。

排版表格註解時,先以 \tnote 指令標示符號,表格下方則以 tablenotes 指令環境排版註解。請注意,表格之註解不會自行編號,使用者須自行輸入編號或符號。此一例子中, 事實上僅有一個註解, 以 * 號表示。表格下方使用 \item[*] 指令排版註解。本例尙使用 \item[] 指令排版一段說明文字。\item 指令後緊接兩個中括號表示不加入任何標示符號。不過,如此一來, 說明文字仍會往右移動特定距離。爲了使說明文字段落的第一行往左突出,我們在其前端加上 \hspace*{-10pt} 指令。

表 1: 國民所得支用面

項目	金額	比率
民間消費	$13,665	46.7%
國內投資*	5,066	17.3
政府消費	4,229	14.5

說明: 金額爲新台幣億元。資料來源:《中華民國國民所得統計》。

* 國內投資包括政府投資與民間投資。

```
\usepackage{threeparttable}
\begin{threeparttable}
\caption{國民所得支用面}
\begin{tabular}[c]{cd{3}.{3}}
...
民間消費 & \$13,665 & ...
國內投資\tnote{*} & 5,066 ...
政府消費 &    4,229 & 14.5
\bottomrule
\end{tabular}
\begin{tablenotes}
\item[] \hspace*{-10pt}說明:
金額為新台幣億元。...
\item[*] 國內投資 ...
\end{tablenotes}
\end{threeparttable}
```

圖 10.20: 使用 threeparttable 排版表格註解

10.6.4 表格內加入括弧或斜線

有些表格須加入大括弧或斜線, 本節所介紹之巨集指令 \bpara 可以方便地處理此種情況。此一巨集指令取自 Goossens, Rahtz, and Mittelbach (1997, 頁 49), 指令之說明請見原書。簡單言之, 我們利用了 graphicx 巨集套件所提供之旋轉功能, 將選定之符號加於表格內。巨集指令 \bpara 須於全文設定區定義, 並須引用 graphicx 巨集套件。

巨集指令之內容如下:

```
\usepackage{graphicx}
\newcommand{\bpara}[4]{ % #1 x; #2 y; #3 angle; #4 height
\begin{picture}(0,0)%
 \setlength{\unitlength}{1pt}%
 \put(#1,#2){\rotatebox{#3}{\raisebox{0mm}[0mm][0mm]{%
 \makebox[0mm]{$\left.\rule{0mm}{#4pt}\right\}$}}}}%
\end{picture}}
```

\bpara 指令計有 4 個選項, 分別設定大括號之置放位置、大小與旋轉角度:

1. 橫向移動距離,
2. 縱向移動距離,

表 1: 行政區域劃分變遷

1905	1915	1920
台北廳	台北廳	台北州
基隆廳		
深坑廳	宜蘭廳	
宜蘭廳		
桃園廳	桃園廳	
新竹廳		

```
\caption{行政區域劃分變遷}
\setlength{\extrarowheight}{2pt}
\begin{tabular}{l@{\hspace{4pt}}%
  l@{\hspace{4pt}}l}
\toprule
 ~1905&  ~1915& ~1920\\
\midrule
台北廳 &        & \\
基隆廳 \bpara{0}{2}{0}{17} &
       台北廳   & \\
深坑廳 &        & \\
\bpara{34}{0}{0}{17} & 宜蘭廳
  \bpara{0}{0}{0}{35}  & 台北州\\
宜蘭廳 &        & \\
桃園廳 \bpara{0}{0}{0}{17} &
       桃園廳   & \\
新竹廳 &        & \\
\bottomrule
\end{tabular}
```

圖 10.21: 表格內之大括號

3. 旋轉角度,
4. 括號大小。

移動距離與括號大小之單位皆爲 pt; 移動距離之值可正可負, 若填入負值, 表示反方向移動。

圖 10.21 的例子中, 我們在表格中加入 4 個右大括號, 其長度視涵蓋內容而有所不同。爲了節省空間, 例子內不再重覆巨集指令之定義。第一欄有三個大括號, 第一個大括號置於「基隆廳」三個字右邊, 上移 2pt, 長度爲 17pt。中間大括號長度爲 17pt, 但該欄位內並無任何文字, 經過幾次嘗試, 我們發現若右移 34pt, 大括號可以上下對齊, 故指令爲 `\bpara{34}{0}{0}{17}`。其餘兩個大括號之指令也是反覆調整後決定的。

此一表格共三個欄位, 皆設爲靠左編排。如果我們改爲靠右編排, 則加入 `\bpara` 指令後, 該欄位內容長度加大, 各欄位之文字無法上下對齊。解決的方法是使用 `\rlap` 指令, 請參見前面例子之說明。

除大括號之外, 類似的指令可以用來排版箭頭、方括號等等。前面 `\bpara` 指令之定義中, 倒數第 2 行末端有 `\}` 指令, 其功能就是設定排版大括號。若

x \ y	y_1	y_2
1	34	55
2	25	45

```
\begin{tabular}{l@{}r|rr}
  & $y$ & $y_1$ & $y_2$ \\[-9pt]
$x$ \bline{12}{-4}{55}{28} &&&\\
\hline
1 & & 34 & 55\\
2 & & 25 & 45
\end{tabular}
```

圖 10.22: 表格內斜線

將此改爲], 則同樣的指令將排版出方括號。同理, 若改以 \rangle, 結果爲一右三角形括號。事實上, 我們可以使用表 9.13 (頁 173) 所介紹之任何一個界限符號 (delimiter)。譬如, 若想要排版箭號, 只要將 \} 改爲 \uparrow 即可。定義更改之後, 使用下列指令:

```
\bpara{0}{0}{-45}{20}
```

將排版出一長度 20pt, 箭頭朝右上方之箭號。若第三個選項填爲 45, 箭頭將變成朝右下方。

利用同樣的原理, 畫出任意角度與長度之斜線也很容易。將 \bpara 巨集指令之定義複製一份, 改名爲 \bline, 並將倒數第 2 行改爲:

```
\makebox[0mm]{\rule{0.4pt}{#4pt}}}}}
```

直線之粗細可自行選擇, 本例中定義直線粗細爲 0.4pt。利用以上定義, 即可在表格中加入斜線。圖 10.22 利用 \bline 指令在表格內加入一直線, 長度爲 28pt, 角度爲 55。

此一例子的目的是說明表格內斜線之畫法, 實際上, 加入斜線後, 表格並不見得更清楚, 也不見得美觀。這印証前面所講的, 花費力氣排版複雜版面之前, 應先想想是否有更簡單美觀的設計。

10.7 彩色表格

排版的目的是精簡地傳達大量資訊, 因此表格上應避免花花綠綠的色彩。不過, 表格內若有大量數字, 則適當地加上灰階橫條的確有助於閱讀。另外, 適當地方加上彩色也有醒目效果。欲在表格中加入灰階/色彩, 可使用 David Carlisle 之 colortbl 巨集套件。

通　訊　錄

姓名	電話	研究室
毛正之	529	E302
古芸安	526	E109
宋名涵	531	E304
李眞如	528	E205

```
\usepackage{colortbl}
\definecolor{light}{gray}{.85}
\definecolor{title}{gray}{.30}
\arrayrulecolor{light}
\begin{center} \bb12
\textcolor{title}{\csp{2cm}{通訊錄}}
\begin{tabular}{lll}
\hline \m11
姓名 & 電話 & 研究室 \\
\hline
毛正之 & 529 & E302 \\
\rowcolorgray{.85}
古芸安 & 526 & E109 \\
宋名涵 & 531 & E304 \\
\rowcolorgray{.85}
李眞如 & 528 & E205 \\
\hline \end{tabular}
\end{center}
```

圖 10.23: 彩色表格例 1

圖 10.23 說明如何在表格中加上灰階橫條紋, 使用的指令是 \rowcolor。此一指令可將特定橫列加上彩色或灰階, 指令格式如下:

\rowcolor[*color model*]{*color*}[*left overhang*][*right overhang*]

所謂 *color model* 是 color 巨集套件選定色彩的方式, 請見第 12 章之說明。欲將某一橫列加上灰階, \rowcolor 指令須加在該列最前方。本例中, 我們使用灰階 gray 模式, 0.85 則用以選定灰階之深淺度。灰階度越接近零, 顏色越深; 越接近 1, 灰階越淺。第 3 選項 *left overhang* 設定彩色或灰階凸出最左一欄左邊之尺寸; 相對的, *right overhang* 則設定凸出最右欄右邊之尺寸。若不加設定, 巨集套件會自動設定讓色彩或灰階與表格左右邊緣切齊。

除了加上灰階橫條紋之外, 本例中之細橫線也是灰階線條, 指令爲:

\arrayrulecolor{light}

其中, light 是事先以 \definecolor 指令所定義之灰階色, 粗淺度爲 0.85。請注意, \arrayrulecolor 指令後面之表格橫線與直線全部改變爲灰階。如果表格線條要回復純黑色, 首先須定義一純黑色 (亦即, 灰階度等於 0), 然後

再以 \arrayrulecolor 指令回復黑色。

除了 light 之外, 本例尙定義 title 爲 0.30 之灰階, 用於排版表格標題「通訊錄」三個字。任何文字要加上灰階或色彩, 可使用 \textcolor 指令, 詳細說明請見第 12 章。本例中, 除了將文字加上灰階之外, 同時也使用 \csp 指令將三個中文字之間隔拉大, 標題長度訂爲 2 公分。此一指令須在全文設定區定義, 方法請見圖 10.8 之例子。

對應於 \rowcolor 指令, \columncolor 指令可將表格特定欄加上色彩/灰階, 指令如下:

\columncolor[*color model*]{*color*}[*left overhang*][*right overhang*]

各選項之意義與 \rowcolor 相同。此一指令之使用須利用 array 巨集套件所提供之 >{...} 指令之功能。圖 10.24 將表格第 1 欄加上黃色背景, 其方法是在 tabular 指令環境中設定各欄位排版方式時使用下列指令:

```
\begin{tabular}{>{\columncolor{yellow}}cccc}
```

上述指令中, yellow 是 color 巨集套件內定之顏色, 而 colortbl 巨集套件選擇顏色的方法與 color 巨集套件相同。其他顏色之選定方法, 請見第 12 章之說明, 或 color 巨集套件之說明檔。本書是以黑白列印, 因此黃色之背景是以灰階出現。

圖 10.24 之標題行另外以 \rowcolor 指令加上灰階背景。由排版結果可以看出來, 當橫列與直欄同時加上色彩或灰階時, 橫列之色彩指令優先。本例表格中之小方塊是以 $\Box$ 指令排版, 前面加上 \large 指令, 稍微放大。因爲小方塊重覆出現, 我們在全文設定區定義巨集指令 \B, 代表此一小方塊。利用巨集指令, 一方面減少重覆輸入的麻煩, 另一方面也避免輸入時出現錯誤。請注意, 使用 $\Box$ 指令時, 須引用 latexsym 巨集套件。

如果要將整個表格加上彩色背景, 可使用 \colorbox, 此一指令事實上是 color 巨集套件所提供。圖 10.25 例子沿用上面所定義之 light 灰階色, 使用 \colorbox 指令將表格背景加上色彩。加上灰階或彩色背景時, 表格四周要留出多大空間可以自行設定。例如, 若在表格指令之前加入下列一行指令:

```
\fboxsep=10pt
```

項目	滿意	普通	不佳
準時上課	□	□	□
準備充分	□	□	□
講解清楚	□	□	□

```
\usepackage{colortbl,latexsym}
\newcommand{\B}{\large$\Box$}
\begin{tabular}{>{
  \columncolor{yellow}}cccc}
\rowcolorgray{.85}
\hline
項目 & 滿意 & 普通 & 不佳
\hline
準時上課 & \B & \B & \B\\
準備充分 & \B & \B & \B\\
講解清楚 & \B & \B & \B\\
\hline \end{tabular}
```

圖 10.24: 彩色表格例 2

項目	滿意	普通	不佳
準時	□	□	□
準備	□	□	□
講解	□	□	□

```
\usepackage{colortbl,latexsym}
\newcommand{\B}{\large$\Box$}
\colorbox{light}{
\begin{tabular}{cccc}\k11
項目 & 滿意 & 普通 & 不佳\\
\hline \m11
準時 & \B & \B & \B \\
準備 & \B & \B & \B \\
講解 & \B & \B & \B \\
\hline \end{tabular}}
```

圖 10.25: 圖表加上灰階背景

表格四周即留出 10pt 之空間。\colorbox 指令不僅可以用於將表格加上彩色/灰階背景, 也可施用於圖形或一般文字; 請見第 12 章之例子。

常見的表格設計方法是以較淺顏色做爲表格背景, 再以較深的顏色排版文字標題。表格標題欲以彩色/灰階排出, 可使用 \textcolor 指令, 這也是 color 巨集套件之指令; 請見第 12 章之說明。

10.8 超大型表格

超大型表格是指表格寬度大於版面寬度, 或者表格高度超過版面高度。如果超出之尺寸不大, 最簡單的解決辦法是以較小字體排版表格。若縮小字體猶不能解決問題, 可考慮將特別寬的表格轉 90 度橫排, 特別長的表格則拆爲兩三頁之接續表格。以下分別介紹表格橫排與排版超長表格的方法。

10.8.1 旋轉大型表格

欲旋轉大型表格至少有兩種方法, 第一種方法是使用 graphicx 巨集套件之 \rotatebox 指令; 第二種是使用 lscape 巨集套件。後一個巨集套件將於下一小節 (頁 220) 介紹, 此處先介紹 \rotatebox, 指令格式如下:

\rotatebox{*angle*}{*material*}

第一選項 *angle* 爲旋轉角度, 第二組大括號內爲旋轉之文字或圖表。除了選定旋轉角度外, 我們尙可設定旋轉之基準點, 請參見 12.1 節之說明。

使用此道指令須注意一點, 全部之文字與指令須置於單一段落內。亦即, 欲旋轉之表格內不得有空行或 \par 指令在內。若表格內必須留出空行, 解決辦法是將全部表格指令置於 minipage 指令環境內。圖 10.26 是旋轉大型表格的一個例子, 旋轉之角度爲 90 度。排版此一表格使用了 graphicx, booktabs, 與 threeparttable 巨集套件。因爲空間不足, 例子中並未列出引用巨集套件之指令。

10.8.2 超長表格

另一種類型的超大表格是其高度超過版面高度。前面曾說明, 若以 tabbing 指令環境排版表格, 其長度可以超過版面一頁; 但以 tabular 指令環境所排版之表格則不行。欲使用類似 tabbing 指令環境之指令排版超長表格, 可使用 longtable 巨集套件, 作者爲 David Carlisle。

巨集套件 longtable 提供 longtable 指令環境, 其指令格式與 tabular 很類似。最簡單的情況如下, 如果你原先是以 tabular 指令環境排版表格, 卻發現表格高度超過一頁, 此時僅須於全文設定區引用 longtable 巨集套件, 並將 tabular 指令替換爲 longtable, 原來之表格即自動拆爲兩部分, 分別排版於兩頁中。須注意的是, 若使用 longtable 指令環境, 排版須執行 latex 兩三次, 甚至更多次, 才能得到正確結果。

表 10.2 爲排版超長表格的一個例子。(爲調整排版空間, 長表格置於本章最後面。) 在 longtable 指令環境內也可以使用 \caption 指令排版標題; 表格之編號將接續原來 table 指令環境之編號。必要時, 我們也可以使用 caption2 巨集套件排版標題。LaTeX 的 table 指令環境是將表格置於版面適當地方, longtable 巨集套件所排版之表格長度超過一頁, 自然沒有浮動版面之「能力」。

淡水與基隆港出口貨物量

	1899	1900	1901	1902	1903	1904	1905	1906	1907	1908
淡水	11,752	10,937	10,325	12,904	11,633	9,050	8,320	6,429	2,058	755
基隆	n.d.	n.d.	n.d.	939	2,832	4,070	5,836	6,625	10,441	11,597

* 基隆台北間舊鐵道線 (在來線) 於1899年度開始營運; 1900年改良線路通車, 內含6哩「在來線」; 同年舊線撤去。資料來源: 臺灣總督府鐵道部 (1910), 下卷, 頁 187-8, 218。

```
\rotatebox{90}{
\begin{minipage}{1.2\textwidth}
\begin{threeparttable}
\centering
淡水與基隆港出口貨物量\\
\begin{tabular}{lrrrrrrrrrr}   \toprule
& 1899 & 1900 & 1901 & 1902 & 1903 ... \\   \midrule
淡水 & 11,752 & 10,937 & 10,325 & 12,904 ... \\
基隆 & n.d. & n.d. & n.d. & 939 ... \\
\bottomrule \end{tabular}
\begin{tablenotes}
\item[*] 基隆台北間舊鐵道線（在來線）... 下卷，頁187-8，218。
\end{tablenotes} \end{threeparttable}
\end{minipage}}
```

圖 10.26: 旋轉大型表格

文稿內使用 longtable 指令時, 表格將從該處開始排版。如果希望從下一頁頂端開始排版, 則指令環境之前須自行加上 \newpage 指令。此外, 表格每一頁之行數是 LaTeX 自行計算。若希望某一頁之表格要特別短一些, 可以在拆頁處加上 \pagebreak 指令。

由本例子可以看出來, 表格欄位的設定方式, 橫線的畫法, 行距之控制等, 都和 tabular 指令環境相同, 不過, longtable 特別提供指令以處理上下兩頁表格之接續問題。本例中, \endhead 指令的功能是用以排版欄位標題, 如「出口物價指數」、「進口物價指數」等等。\endhead 指令以上到另一個控制指令 \endfirsthead 之間的所有文字將出現爲表格各頁上方之欄位標題, 其中包含有「承接上頁」四個字。不過, 表格第一頁不須有「承接上頁」, 因此, 第一頁之欄位標題須特別以 \endfirsthead 設定, 重點是去除「承接上頁」四個字。

同樣的, \endfoot 與 \endlastfoot 指令是在表格每一頁下方添加說明資訊。本例中, \endfoot 指令用於在每一頁下方排版出「續接下頁」四個字; \endlastfoot 則用於排版表格末頁底下之文字。本例中, \endlastfoot 與 \endfoot 之間空白, 因此表格末頁底下即無任何添加資訊。

在 longtable 指令環境內可以使用 \footnote 排版註解, 不過註解文字會出現在該頁下方, 而不是表格末端。如果要在表格最末端加上說明或註解, 可利用 \endlastfoot 指令。本例中, 若在 \endfoot 與 \endlastfoot 兩行之間加入以下指令:

```
\multicolumn{6}{0.8\textwidth}{說明: ... }
```

則說明文字將排版於表格最後一頁下端。此一方法的缺點是排版說明文字段落的寬度須自行選定。

如果表格又長又寬, 我們還可以使用 lscape 巨集套件將 longtable 所排版之長表格旋轉 90 度。前面曾介紹如何使用 \rotatebox 指令旋轉文字圖表, 該指令可將整頁版面內一小部分的文字圖表旋轉任意角度。相對而言, lscape 巨集套件是將整頁版面旋轉 90 度, 但頁眉/頁足不動, 作者亦爲 David Carlisle。

巨集套件 lscape 提供 landscape 指令環境, 使用方法很簡單, 僅須將所欲轉置之文字圖表置於指令環境即可。例如, 要將表 10.2 旋轉 90 度, 僅須將原排版指令納入 landscape 指令環內即可:

```
\usepacakge{longtable,lscape}
\begin{landscape}
 \begin{longtable}{@{}lrrrrr@{}}
  ...
 \end{longtable}
 \end{landscape}
```

最後, 本章 10.2.3 節曾介紹 booktabs 巨集套件。不幸的是, 原始版本的 booktabs 之指令無法與 longtable 巨集套件一起使用。不過, 原作者 Simon Fear 特別提供一小段修改程式, 我們已加入原巨集套件內。利用修改過的巨集套件, 如果要在 longtable 表格內畫上橫線, 指令爲 \LTtoprule, 畫底橫線之指令爲 \LTbottomrule, 畫中間細橫線之指令爲 \LTmidrule。

```
\usepackage{longtable}
\fontsize{10.95}{13pt}\selectfont
\extrarowheight=1pt
\begin{longtable}{@{}lrrrrr@{}}
\caption{台灣長期物價指數\label{longtable}}\\[-2pt]
\hline
     &     & 台銀躉售 &     & 出口     & 進口\\[-1pt]
年期 & PPI & 物價指數 & CPI & 物價指數 & 物價指數\\
\hline
\endfirsthead
\multicolumn{6}{l}{\k11 承接上頁}\\[2pt]
\hline
     &     & 台銀躉售 &     & 出口     & 進口\\[-1pt]
年期 & PPI & 物價指數 & CPI & 物價指數 & 物價指數 \\
\hline
\endhead
\hline
\multicolumn{6}{r}{\k11 續接下頁}
\endfoot
\endlastfoot
 1896 & --~ & --~ & --~ &  60.31&  59.16\\
 ...
 1945 & 490.09 &  2392 &  --~ &  --~ &  --~ \\
 1946 &    --~ & 21344 &  --~ &  --~ &  --~ \\
\hline
\end{longtable}
```

表 10.2: 台灣長期物價指數

年期	PPI	台銀躉售物價指數	CPI	出口物價指數	進口物價指數
1896	–	–	–	60.31	59.16
1897	–	–	–	66.88	55.66
1898	–	–	–	76.87	57.84
1899	–	–	–	77.60	62.27
1900	–	–	–	77.47	74.60
1901	–	–	–	78.00	77.21
1902	69.12	–	–	90.92	73.67
1903	73.16	–	65.44	82.91	76.66
1904	60.58	–	68.88	85.77	79.81
1905	71.58	–	70.82	83.72	85.30
1906	72.67	–	71.45	84.74	89.26
1907	89.70	–	73.57	94.34	95.76
1908	77.96	–	79.08	98.62	87.83
1909	77.48	–	83.50	101.98	90.28
1910	88.97	–	86.18	93.64	95.25
1911	95.91	–	95.42	99.77	99.11
1912	110.36	–	104.17	113.43	99.57
1913	112.89	–	101.80	114.43	102.57
1914	100.00	–	100.00	100.00	100.00
1915	98.51	–	92.11	93.62	108.25
1916	108.93	–	97.53	98.05	144.69
1917	128.88	–	117.63	121.56	186.08
1918	166.89	–	146.43	151.58	220.07
1919	237.38	223	179.50	196.52	232.13
1920	253.80	264	158.14	233.51	251.96
1921	152.09	207	140.42	172.11	599.30
1922	127.23	203	131.00	166.07	545.90
1923	145.34	198	126.37	154.21	541.73
1924	157.65	198	135.49	162.67	568.81
1925	183.69	202	144.15	161.50	587.40
1926	147.39	188	140.65	159.70	506.30
1927	136.47	178	131.34	151.32	480.98
1928	133.25	180	133.47	150.32	478.92

續接下頁

承接上頁

年期	PPI	台銀躉售物價指數	CPI	出口物價指數	進口物價指數
1929	129.67	165	133.80	147.70	466.63
1930	110.49	151	119.03	129.18	388.95
1931	85.17	134	105.52	109.49	557.60
1932	94.46	140	100.79	70.16	326.53
1933	110.95	152	103.56	75.57	376.90
1934	114.82	151	106.21	82.94	375.77
1935	120.80	156	115.79	86.96	405.45
1936	130.44	166	124.34	96.49	385.65
1937	135.98	187	131.49	97.84	432.91
1938	138.26	216	165.05	125.06	468.77
1939	157.92	240	–	–	–
1940	173.08	270	–	–	–
1941	189.23	262	–	–	–
1942	202.10	296	–	–	–
1943	210.70	315	–	–	–
1944	258.66	354	–	–	–
1945	490.09	2392	–	–	–
1946	–	21344	–	–	–

11 引用外製圖形

圖形是表達作者思想的有效工具, 一般文稿或學術論文中經常使用圖形。TEX 的主要功能是文字排版, 它雖然可以繪製簡單的線條圖形, 但複雜圖形就無能爲力。同樣的, LATEX 雖然新增一些畫圖指令, 但仍然難以處理複雜圖形。不過, 仍然有一些巨集套件如 PiCTEX, LATEXCAD 等, 直接使用 LATEX 的指令繪製圖形。

雖然 TEX 的主要功能是處理文字, 但是 Knuth 當初設計時留下一個與其他繪圖系統溝通的管道。TEX 有一個指令稱爲 `\special`, 透過此一指令, 我們可以將其他軟體繪製之圖形引入版面中。TEX 並不瞭解其他繪圖軟體之指令, 因此引入外製圖形時, TEX 只是在版面上讓出一點空間, 不作任何排版動作。排版之後, 圖形在顯示器上出現或印出於白紙上必須藉助預覽/列印軟體。

市面上有各式各樣的繪圖系統, 不同的系統可能使用不同的圖形語言。因爲繪圖系統衆多, 可以想見任何一套預覽/列印軟體不可能處理所有規格之圖形。本章主要介紹 PostScript 語言之圖形, 並說明如何將此種圖形引入文稿中。此一圖形語言能力甚強, 使用很廣。事實上, TEX 排版系統與 PostScript 圖形語言之結合可以說是將排版品質推到一個新的高峰。

11.1 圖形檔案規格

簡單言之, 要畫出一個黑白或彩色圖形有兩種方法, 第一種是直接描點; 第二種是先畫出外框, 再把框內塗色。因此, 圖形檔規格可以區分爲兩大類: 「描點式」(bit-mapped) 與「描邊式」(vector-based)。「描點式」圖形是一個黑點一個黑點地在白紙上畫出圖形來。Windows 內之 imaging 軟體所畫成的就是描點圖形。掃描器 (scanner) 所產生的也是描點式圖形。相反的, 「描邊式」圖形則是描出圖形的外框, 特定區域再塗上顏色。幾個有名的繪圖軟體, 如 Corel Draw, Visio, Adobe Illustrator 等, 所產生的都是描邊圖形。

第 3 章 (頁 24) 所舉例之描點字形就是一個描點圖形; 反之, 描邊字形則是描邊圖形。

就使用彈性與排版品質而言, 描邊圖形是最好的選擇。不過, 描邊圖形本身也有許多種規格, 其中 PostScript 是最廣泛使用的一種。PostScript 圖形語言雖然主要是用於處理描邊圖形, 但它也有處理描點圖形的能力。舉例言之, 我們用掃描器產生之圖形是描點格式。但透過工具軟體可以將它轉換爲 PostScript 格式。轉換格式的好處是圖形在放大縮小時, 較不會失眞。本章主要介紹如何在文稿中引入 PostScript 圖形。不過, 我們也將簡單說明如何處理描點圖形。有關於引用外製圖形的種種細節, 請參考 Reckdahl (1997)。此一檔案對於引用外製圖形的各種問題有詳細說明, 本章許多的介紹與說明即參考該檔案。

描邊圖形檔案在列印時可以放大、縮小、拉長或壓扁, 而無損列印品質。而且, 在不同精密度的列表機上印出, 我們都可以得到同樣大小的圖形。反之, 描點圖形在不同密度的列表機上列印, 其大小會隨之而變。其次, 圖形若放大或縮小, 品質會變差。

列印品質和印表機種類有關。目前的雷射印表機每吋可印 300 點或 600 點; 噴墨式列表機之列印密度有 300 點 (如 HP 機種) 也有 360 點 (如 Canon 或 Epson 機種)。列印密度越高, 表示它所列印的每一圓點比較小。說得更具體一點, 600 點之雷射印表機印出來的一個圓點, 其直徑只有 300 點印表機的一半。高解析度印表機的圓點較小, 因此能夠較精細地描繪圖形。反過來說, 這也表示要描繪一固定大小之圖形, 必須使用較多的圓點。

想像一個描點方塊, 橫寬由 900 個黑白點, 垂直由 600 個黑白點構成。此一圖形若由 300dpi (每吋 300 點) 雷射印表機印出, 水平寬度爲 3 吋, 垂直高度爲 2 吋。但是, 同樣圖形如果由 600dpi 印表機印出, 圖形變小, 寬度縮小爲 1.5 吋, 高度變成 1 吋。因此, 如果使用兩種不同密度的印表機, 低密度印表機用於列印校對用初稿, 高密度印表機用於印出最後完稿, 則同一幅描點圖形就必須準備兩份檔案。否則初稿上圖形的大小將與最終完稿之大小不同。這是使用描點圖形不方便之處。

描邊圖形可以避免上述的缺點。不過, 描邊圖形的規格有很多種, 我們主要介紹的是 PostScript 圖形規格; 這是美國 Adobe 公司所發展, 因其能力特強, 在 1980 年代迅速普及。目前, 從 300dpi 的雷射印表機至 2540dpi 的相紙輸出機 (phototypesetter), 幾乎都能使用此一圖形語言。所謂印表機

能使用 PostScript 語言, 意思是說當電腦把一 PostScript 圖形檔傳送給印表機時, 印表機有能力將它轉換圖形元素, 在紙上印出。

市面的印表機一般是以品牌區分, 但更重要的差別其實是其所使用的圖形語言。HP 相容型印表機使用的圖形語言稱爲 PCL; Epson 或 Canon 噴墨印表機則使用另一種圖形語言。以上這些印表機若加上適當的硬體配備, 就具有列印 PostScript 圖形的能力。具有列印 PostScript 圖形能力之印表機即稱爲 PostScript 印表機, 或簡稱爲 PS 印表機。PS 印表機雖然功能較強, 但價錢較貴。

幸運的是, 網路上有一免費軟體 Ghostscript, 可以讓一般的雷射印表機或噴墨印表機具備 PostScript 能力。Ghostscript 軟體是由 L. Peter Deutsch 所發展, 它已經是專業排版者不可或缺的工具。假設我們有一 PostScript 圖形, 但印表機是普通的 HP 或 Epson 印表機; 則透過 Ghostscript 軟體可將此圖形轉換爲 HP 或 Epson 印表機之圖形語言再印出。除了列印之外, Ghostscript 也可以將 PostScript 圖形顯示於電腦螢幕上, 這對於排版工作非常方便。一般使用時, 我們通常是透過 GSview 軟體執行 Ghostscript 程式。GSview 是由 Russell Lang 所發展。

11.2 引用外製圖形

PostScript 與 TEX 的功能很類似; 只不過前者用於畫圖形, 後者用於排版文字。如果熟悉 PostScript 圖形語言, 我們也可以自行下指令畫出圖形來。但是, 如果圖形複雜, 直接下指令並不容易。一般的畫圖軟體, 如 Corel Draw, Illustrator 等, 主要的功能就是提供一使用者介面, 讓我們可以輕鬆地在視窗內以滑鼠畫出圖形。當我們把圖形儲存 (或輸出) 爲 PostScript 格式時, 繪圖軟體即產生一 PostScript 檔案。

從排版的角度來看, 我們並不是要畫一整頁的圖形, 而是要將 PostScript 圖形引入版面特定位置。爲達到此一目的, 我們須使用一特別的格式, 稱爲 Encapsulated PostScript, 通常簡稱爲 EPS。相對而言, 一般的 PostScript 圖形則簡稱爲 PS。EPS 與一般的 PostScript 檔案其實很類似, 主要的差別是前者檔案內須儲存有標示圖形大小的座標, 底下將進一步說明。

使用外製圖形的最大困難不在於如何下指令控制, 而是在於產生正確的 EPS 檔案。本節首先說明引用外製圖形的指令, 下一節再介紹幾種繪製 EPS 圖形的方法。

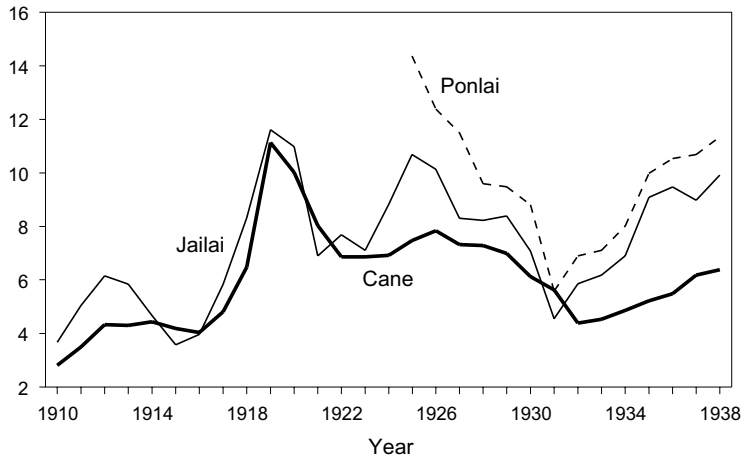


圖 11.1: 台灣的米價與蔗價

11.2.1 引用 PostScript 圖形

LaTeX 提供 graphicx 巨集套件以方便引用外製圖形。此一巨集套件除了可用以引入 EPS 圖形外, 還可以引入 pcx, TIFF 等描點圖形。不過, 底下之介紹主要是針對 EPS 圖形。

若圖形檔名爲 rice.eps, 引用圖形之指令如下:

```
\usepackage{graphicx}
\begin{figure}
 \begin{center}
 \includegraphics{rice.eps}
 \caption{\m11 台灣的米價與蔗價}
 \end{center}
\end{figure}
...
```

EPS 圖形是以 \includegraphics 指令引入, 這是 graphicx 巨集套件所提供之指令。指令環境 center 使圖形居中編排。指令環境 figure 具有浮動版面之功能, 讓圖形置放於版面適當位置; \caption 指令則用於排版標題。

本例之圖形是以原尺寸引入, 實際引用圖形時, 我們通常須將原圖放大或縮小。欲調整圖形大小, 可使用下列指令選項:

`height`	設定圖形高度,
`totalheight`	設定圖形全高,
`width`	設定圖形的寬度,
`angle`	設定圖形旋轉（反時鐘方向）角度,
`scale`	設定圖形放大（或縮小）之倍數。

想像把每一個圖形包含在一長方形中, 左下角稱爲參考點 (reference point)。由參考點起, 可以算出圖形的高度 (height), 寬度 (width) 與深度 (depth)。若圖形不旋轉, 深度爲零, 而全高 (totalheight) 等於其高度。若圖形旋轉 −45 度, 參考點仍爲原來位置, 但深度將大於零, 高度也改變了; 此時全高即爲高度與深度之和。

若要將圖形寬度拉大爲與行寬相同, 指令如下:

```
\includegraphics[width=\textwidth]{rice.eps}
```

若圖形寬度要訂爲行寬的 80%, 指令爲:

```
\includegraphics[width=0.8\textwidth]{rice.eps}
```

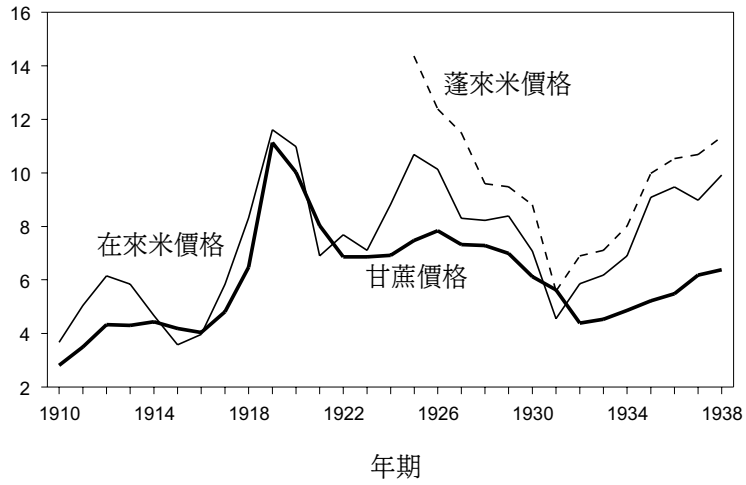
如果要將圖形縮小爲原尺寸的 80%, 須加上 `[scale=0.8]` 選項。在此選項下, 圖形橫寬與垂直高度將同比例變動。若要作不同比例的變動, 須同時選擇高度與寬度, 譬如:

```
\includegraphics[width=10cm,height=7cm]{rice.eps}
```

以上指令設定圖形高度爲 7 公分, 寬度爲 10 公分。除了以上之外, graphicx 巨集套件還有許多選項, 請參考其說明檔。

11.2.2 圖形中加入中文或數式

圖 11.1 的三條線是以英文標示其意義, 但在中文稿中, 圖形說明最好是使用中文。有時候, 我們須在圖中加上數學式, 如 $z = \sqrt{x^2 + y^2}$。在中文 Windows 的繪圖軟體中, 我們通常可以直接鍵入中文標題或說明文字。但是當圖形儲存成描點格式的 PCX 或 BMP 檔案時, 中文字不一定能正確儲存。如果把圖形儲存成 PostScript 格式, 若繪圖軟體品質不佳, 中文字部分也可能出現問題。同樣的問題也會出現在數學式或數學符號上。




```
\usepackage{psfrag,graphicx}
\begin{figure}
 \psfrag{Year}[t][b]{年期}
 \psfrag{Ponlai}{蓬來米價格}
 \psfrag{Jailai}[r][r]{在來米價格}
 \psfrag{Cane}{甘蔗價格}
\includegraphics[width=\textwidth]{rice.eps}
\end{figure}
```

圖 11.2: 圖形內之中文說明

解決以上問題的一個好方法是使用 psfrag 巨集套件。此一巨集套件原由 Craig Barratt 所發展; 3.0 版開始則由 Michael C. Grant 與 David Carlisle 共同發展。巨集套件 psfrag 的原理很簡單, 用軟體繪製圖形時, 我們先加入英文標示, 如圖 11.1 所示。引入圖形時, 我們再下指令將英文標示字元轉換爲中文或數學式。這個方法只能用於 PostScript 描邊圖形, 描點圖形無法使用這個方法。

使用 psfrag 巨集套件之過程可以簡單描述如下: 以軟體繪製圖形, 於選定位置加入英文或數字標示; 將圖形輸出爲 EPS 格式之後再以 \psfrag 指令以新字元串替代原圖形之字元串。以圖 11.1 爲例, 原圖形內包含: Ponlai, Jailai, Cane, Year 及橫軸與縱軸上之阿拉伯數字。經過轉換之後, 結果變成圖 11.2。底下進一步說明指令細節。

使用 psfrag 巨集套件, 首先須於全文設定區引入。文稿內要將 Ponlai 英文字替代成爲「蓬來米」三個中文字, 指令爲:

```
\psfrag{Ponlai}{蓬來米}
```

其中, 替代之中文字體與大小可以任意選擇。另外, 爲了讓替代字元置於適當位置, 尙可加上調整位置之指令。想像原來圖中之 Ponlai 英文字與替代字元「蓬來米」三個中文字各有一長方形外框。每一外框可以英文字母 t, b, l, r 之組合設定個參考點: t 代表外框上方中央點, b 代表下方中央點, l 代表左方中央點, r 右方中央點。以上 4 個字母可以進一步組合, 譬如, tl 代表長方形外框左上角, bl 爲左下角參考點, b 代表下方中間點; 中心點則以 [] 表示。

若要把替代字元之下方中心點置於原字元之上方中心點, 指令爲:

```
\psfrag{Ponlai}[b][t]{蓬來米}
```

換言之, 第一個方括號標示替代字元外框之參考位置, 第二個方括號標示原始字元之外框參考位置。如果不標示位置, 則兩串字元之參考位置將自動設爲 [bl], 亦即兩串字元外框之左下角將對準。

除了設定位置之外, 替代字元還可以放大、縮小或旋轉。旋轉指令標示是指替代字元外框之參考點沿逆時針方向旋轉之角度。繼續以上的例子, 若字體不放大, 亦即放大倍數等於 1, 而字元要旋轉 90 度, 指令爲:

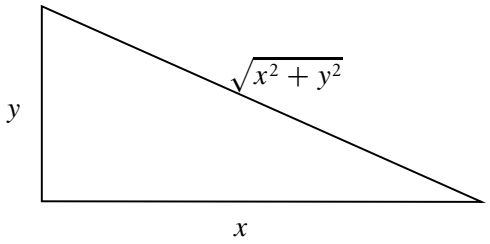
```
\psfrag{Ponlai}[b][t][1][90]{蓬來米}
```

其中, 第 3 對方括號中之數字表示放大倍數, 第 4 對方括號中之數字設定沿逆時針方向旋轉之角度。

歸結以上的說明, \psfrag 指令之格式如下:

\psfrag{*ps*}[*pn*][*PSpn*][*scale*][*rot*]{*text*}

其中, *ps* 表示原來圖形中之字元, *PSpn* 標示原字元外框之參考點, *pn* 標示替代字元之參考位置, *scale* 標示其放大倍數, *rot* 表示旋轉之角度, *text* 代表替代字串。圖 11.2 以 \includegraphics{rice.eps} 引入圖形; 在此指令之前, 替代字元則以 \psfrag 指令引入。以第一行 \psfrag 指令爲例, 替代字元「年期」外框上方中心點對齊原字元 Year 外框下方中心點。




```
\usepackage{psfrag,graphicx}
\begin{figure}
 \psfrag{x}{$x$}
 \psfrag{y}{$y$}
 \psfrag{z}{$\sqrt{x^2+y^2}$}
 \includegraphics[width=.6\textwidth]{triangle.eps}
\end{figure}
```

圖 11.3: 圖形中之數學式

圖 11.3 是另外一個例子。我們先以繪圖軟體畫出三角形, 底線以 x 標示, 垂直線標示爲 y, 斜線標示爲 z。繪圖完畢存檔之後, 輸出爲 EPS 格式, 取名爲 `triangle.eps`。輸出 EPS 檔應注意之細節, 請參考下一節之說明。特別重要的一點: 輸出時**原圖形中之文字必須存為字元**, 不能存爲描邊圖形。若圖形內之文字以圖形格式儲存, EPS 檔案中並無文字串, 即不可能作任何文字替代。

本例引用圖形之指令中加入 `[width=.6\textwidth]` 選項, 將圖形寬度調整爲行寬的 60%。EPS 圖形的優點之一是圖形放大或縮小之後列印品質仍然很好。若是描點圖形, 放大或縮小之後, 列印品質通常不及原圖。

使用 psfrag 巨集套件有一些應注意的問題, 列舉說明如下:

1. 部分軟體輸出 EPS 檔案時會將字元串拆爲兩行。譬如, Corel Draw 軟體有時會將一文字串拆成兩三行。因此, Year 可能拆成 Ye 與 ar 兩段; 或者 Yea 與 r 兩段。此時, 字元替代即無法成功。如果不能確定是否正確, 可用文字輸入軟體檢查 EPS 檔案, 看能否搜尋到原文字串。EPS 檔案是以普通文字儲存, 因此可以用文字編輯軟體修改。萬一眞的拆成兩行或三行, 怎麼辦呢? 只好回到原軟體中, 將字串縮短, 期望下次轉換出正確結果。另外, Visio 繪圖軟體 (4.01 版) 輸出 EPS

圖形檔時, 各字元串後面會自動加上一控制指令: \r。引用圖形檔之前, 須先以文字編輯軟體將此控制指令去掉。

2. 由 Excel 或 Quattro Pro 所轉換的資料圖中可能含有百分比符號 %。此符號在 LaTeX 中是註銷指令, 出現在其後的文字將被忽略。因此, 被替代之字串中不可用 %。
3. 下 \psfrag 指令時, 須確定指令中之字串與 EPS 檔案內之字串完全相同, 包括大小寫、空白等。
4. 替代之文字並不限於是幾個文字, 也可以是整段文字。此時, 整段文字應置於 \parbox 或 minipage 指令環境中。
5. 有些軟體雖然可以在圖形中加上英文或數字, 但存檔時是以外框圖形儲存, 而非字元形式。統計軟體 Gauss 就是一個例子。此時我們可以將畫好的圖形先引入畫圖軟體如 Corel Draw 或 Visio 中, 加入適當文字之後, 重新輸出爲 EPS 檔案。

11.2.3 引用 PostScript 文稿檔案

本書第 2 章舉 6 個例子說明 LaTeX 排版功能, 每一個例子之排版結果直接引入版面中以供參考。LaTeX 之排版結果可以轉換爲 PostScript 檔案, 也可以轉換爲 EPS 格式。若轉換爲後一種格式, 即可引入另一篇文稿中。本小節簡單說明處理方法。

欲將排版結果轉換爲 EPS 格式, 須注意 EPS 檔案僅能包含一頁版面。此外, 轉換時應設定使用描邊字形, 否則最後的結果品質不佳。假設原文稿檔名爲 example.ctx, 其中第 2 頁欲轉換爲 EPS 格式。LaTeX 排版完成之後, 進一步執行下列指令:

```
c:\xtemp>dvips -E -p2 -l2 example.dvi
```

選項 -E 指示轉換爲單頁之 EPS 格式; -p2 -l2 指示轉換第 2 頁。

使用 -E 選項時, Dvips 程式會嘗試計算文稿版面之大小, 並直接記錄於圖形檔案之 BoundingBox 指令內。但是, 在某些情況下計算結果可能有誤。欲正確計算圖形版面之大小, 可用下列方法: 開啓 GSview 軟體, 讀入 example.ps 圖形檔, 啓動 “File|PS to EPS” 功能, 確定視窗內之 “Automatically calculate Bounding Box” 已勾選, 按一下 Yes 鍵之後, 鍵入新檔名, 可產生一新的 EPS 檔案, 其內含正確的 Bounding Box 資訊。

以上說明產生 EPS 檔案的方法, 不幸的是引用此圖形仍然有一個問題。將此一 EPS 檔案引入文稿, 排版之後以 DVIPS 轉換爲 .ps 檔案, 以 GSview 預覽/列印時會出現錯誤。原因是目前版本的 DVIPS (5.86版) 無法正確處理某些中文字型。但以 YAP/Windvi 預覽/列印則無問題。如果你購買了完整版本的 Acrobat 軟體, 一個解決的方法如下: 先利用 Acrobat Distiller 軟體將 EPS 檔案轉換爲 PDF 格式; 再開啓 Adobe Acrobat 軟體, 讀入 PDF 檔案, 將之轉換出 DVIPS 可以正確處理之 EPS 檔案。

11.3 繪製 EPS 圖形

產生 EPS 圖形檔案的方法之一是以繪圖軟體繪製。在 Windows 系統下, 許多軟體都可以繪製圖形。譬如, Excel 試算表可由輸入資料自動畫出資料圖形; Corel Draw 或 Visio 軟體能繪製各式各樣圖形。此外, 專業統計軟體如 Gauss, 或者數學運算軟體如 Mathematica 等, 也都能畫出各種圖形。某些繪圖軟體本身即提供直接產生 EPS 檔案的工具。例如以 Corel Draw 畫圖之後, 可選擇將結果輸出爲 EPS 檔案; 但是, Excel 軟體並無此功能。

若軟體本身提供轉換工具, 原則上應直接使用該項工具。不過, 某些軟體雖然繪圖能力不錯, 但轉換工具之功能卻有問題。此時, 你可以將圖形剪貼到性能較佳的繪圖軟體中, 再由該軟體輸出。如果繪圖軟體本身並無轉換功能, 剪貼之後再輸出爲 EPS 也是可能的途徑。譬如, Excel 軟體畫出之資料可以先剪貼到 Corel Draw 再輸出。不過, 並不是所有人都有 Corel Draw 軟體。此時, 還有一個辦法可以將圖形軟換爲 EPS 檔案, 那就是利用 PostScript 印表機驅動程式將圖形「列印」爲 EPS 檔案。以下將先介紹此一方法, 之後再介紹以繪圖軟體輸出 EPS 之方法。

產生 EPS 檔案的方法之一是透過 PostScript 印表機驅動程式。底下兩小節分別說明安裝驅動程式及產生 EPS 檔案的方法與細節。

11.3.1 安裝 PostScript 印表機驅動程式

Windows 作業系統提供 PostScript 印表機驅動程式。但是, Adobe 公司所提供之驅動程式功能最強, 產生之 EPS 檔案最正確。若你要使用印表機驅動程式產生 EPS 檔案, 應安裝此驅動程式。

光碟片 \miktex\adobe 檔案夾下有 aps426eng.exe 檔案, 直接執行即開始安裝。經過幾個選項畫面之後, 會出現下列之選項畫面:

```
Install PostScript Printer from PPD
```

此時, 須點選右欄之 Default PostScript Printer 才能繼續。接下來, 幾個畫面之後會出現 Add Printer 選項畫面; 此畫面中之:

```
Would you like to print the last page?
```

應選擇 No。

安裝程式複製檔案之後, 即進入 "內容 Default PostScript Printer" 畫面, 點選第 5 項 PostScript, 並將 PostScript output format 改爲 *Encapsulated PostScript (EPS)*。點選此項時, 安裝程式出現提醒訊息: 圖形應「列印」至檔案而非印表機; 按「確定」之後, 即完成安裝。

安裝完成之後, 進入 Windows 的「我的電腦」, 點選「印表機」, 除了原已安裝之印表機之外, 會出現另一項 Default PostScript Printer; 列印 EPS 檔案時應選用此印表機。

11.3.2 以 PostScript 驅動程式產生 EPS 圖形檔

利用 PostScript 列印驅動程式產生 EPS 圖形檔的方法很簡單, 以下以 Excel 軟體爲例說明之。畫 Excel 資料圖時最好把圖形置於獨立的一頁中; 英文/數字請選用 Windows 原有之英文字型如 Arial, Times New Roman 等。如果圖形內有中文字, 驅動程式會把它們轉成描點字型, 列印品質較差。因此, 中文最好是以上一節所述之 psfrag 巨集套件處理。

資料圖畫好之後, 先點選圖形區域, 再將圖形印出。在接下來的「列印」畫面上, 將「印表機」改爲 "Default PostScript Printer", 並勾選「輸出至檔案」; 按「確定」之後, 填入自選的檔名, 再按「確定」即產生 PostScript 檔案。一般 EPS 檔案是以 eps 爲延伸檔名, 但我們建議先以 ps 爲延伸檔名。EPS 圖形檔與一般 PostScript 檔案不同的地方在於前者記錄有標示圖形大小的座標。不過, 由驅動程式所轉換出來的檔案, 其座標數字通常大於圖形實際大小。引入 TEX 文稿之後, 圖形四周將出現多餘的空白。

簡單的解決辦法如下: 以 GSview 軟體開啓 PostScript 檔案, 以滑鼠點選 File, 其下有 "PS to EPS" 選項; 點選此一選項即可轉換出正確的 EPS

檔案。有時候, 你會發現 GSview 所讀取之圖形超過視窗所顯示的紙面範圍。此時, 須回到 Excel 軟體中將圖形尺寸縮小一些, 重新列印出 EPS 檔案。

11.3.3 使用繪圖軟體繪製 EPS 圖形

不同之軟體各有其繪製圖形的方法。一般的科學軟體, 如 Mathematica 或 Gauss 等, 都可以直接產生 PostScript 檔案。若使用 Corel Draw 或 Visio 繪圖軟體, 我們直接在版面上拉出線條、上色彩、加文字。若使用 Windows 之 Excel 或 Lotus 123 試算表, 我們先輸入數字, 再由軟體畫出資料圖 (chart)。這種種方法所產生之圖形, 都可以轉換爲 PostScript 檔案。底下將以上述軟體爲例, 說明轉換 PostScript 檔時須注意之細節。

以下所介紹之軟體大部分是商業繪圖軟體。商業軟體經常更新, 各版本之功能不一定相容。而且, 大部分軟體是國外公司所發展出版, 因此在台灣可能同時流通中英文版本。一套軟體的不同版本性能可能有異; 或者性能相同, 但執行方式已改變。以下所介紹描述的主要是我們所熟悉使用者。如果你使用不同版本, 甚或完全不同之軟體, 你應該能從底下的說明中摸索出繪製 EPS 圖形的方法。

首先, 我們說明繪圖時應注意之事項。大部分繪圖軟體都是由美加各國所開發, 其設計主要是以英文使用者爲對象。這些軟體經過中文化處理之後, 我們可以在圖形中鍵入中文字, 但是大部分軟體對於中英文之處理並不相同。簡單來說, 大部分軟體都把中文字以描點圖形方式處理。相反的, 如果你選用 Windows 內附之 `Arial` 或 `Times New Roman` 等英文字體, 則圖形內之英文/數字大都以描邊字形處理。兩種處理方法的主要差別是在於列印品質的高低不同。如果繪製之圖形內並無中文, 則英文/數字應選用 Windows 英文字體。

處理中文的主要考慮是品質。目前大部分軟體皆能直接鍵入中文字於圖形中, 輸出之 EPS 檔案內也含有 (描點) 中文字。但是, 最後之列印品質並不理想。使用 TEX 排版的主要理由是追求品質。因此, 底下所介紹的方法是以列印品質最佳爲出發點。基本的技巧是透過 psfrag 巨集套件將中文字代入圖形中。

繪圖軟體: Corel Draw

在 Windows 下, 我們可以選用之英文字型很多。若無特別偏好, 英文/數字

應選擇較常用之字體, 如 Times 或 Arial; 原則上勿使用中文字型檔內之英文/數字。

畫好圖形, 存檔之後, 即可輸出 (Export) EPS 格式之檔案。以 Corel Draw 7.0 英文版爲例, 選定所欲輸出之圖形, 點選 "File|Export"; 「存檔類型」應選擇 Encapsulated PostScript (EPS)。下一個畫面是 EPS Export; 其中, Export text as 選項應選擇 Text, 並勾選 Include fonts。EPS Export 畫面左邊第 2 欄位是 Image header, 其下之 Include header 請留爲空白, 勿勾選。若勾選此項, 所產生之 EPS 圖形檔案前端會加入一段代表圖形之文字串。實際引用圖形時, 可能會出現問題。

Export text as 選項下若選擇 Curves, 則圖形輸出爲 EPS 時, 圖形內之文字將轉換爲一般之 PostScript 指令。反之, 若選擇第二次選項 Text, 圖形中之文字將直接存爲字元, 不作特別處理。就 TEX 的排版應用而言, 存爲字元圖形檔較小, 而且能夠以 psfrag 巨集套件作文字替換。

以上之說明是以 Corel Draw 英文 7.0 版爲例; 9.0 英文版也是以類似方法處理。若使用其他版本, 作法可能稍有不同。而且, 即使是同一版本號碼, 中文版與英文版也可能不同, 必須仔細測試才能確定。

繪圖軟體: Visio

繪圖軟體 Visio (英文版 4.0), 轉換檔案之基本原理與 Corel Draw 相同, 但選項名稱稍有不同。在圖形繪製完畢開始輸出時, 檔案格式應選擇 Encapsulated PostScript File。接下來的畫面有 Profiles, Color Translation 及 Line Cap Mode 三選項。首先, Profiles 應選用 Standard Options-EPS; 其次, Color Translation 應選用 normal; Line Cap Mode 可以選用 Device。畫面下方另有 4 個選項: 從 Background Rectangle 到 Include Preview, 應全部留爲空白。

試算表軟體: Excel

Excel 所畫之資料圖可以利用 PostScript 印表機輸出爲 EPS 檔案, 也可以利用 Windows 之複製與貼下 (Copy and Paste) 功能, 將圖形貼到繪圖軟體 Corel Draw 或 Visio 中, 經過修改再輸出。自 Corel Draw 貼下時, 須選用 Paste special 項下之 Picture (Metafile)。相關之細節請見上文之說明。

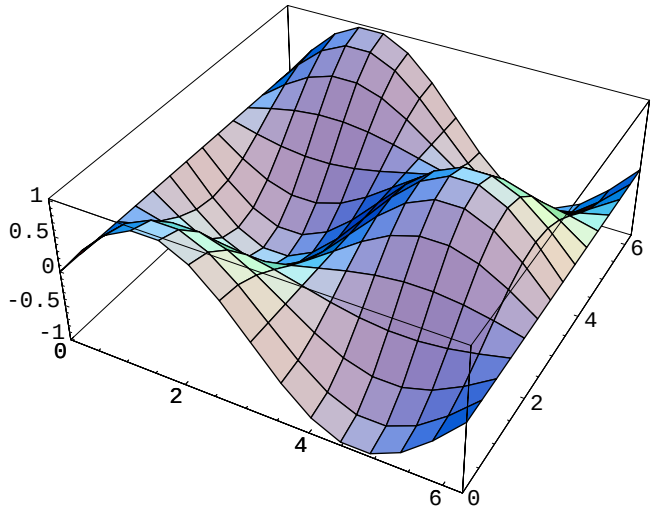


圖 11.4: 引用 Mathematica 軟體所產生的 EPS 圖形

Excel 圖形內若有中文字, EPS 圖形內仍會出現。但中文字是以點陣方式 (bitmap) 儲存。圖形放大縮小時, 中文字之列印品質較差。若要求較佳的列印結果, 請使用 psfrag 巨集套件之方法。

計算軟體: Mathematica

Mathematica 是一個很有名的計算/繪圖軟體, 應用甚廣。底下以 3.0 版爲例, 說明如何產生 EPS 圖形檔案。在 Mathematica 內執行下列指令:

```
Plot3D[Sin[x]Cos[y],{x,0,2π},{y,0,2π}]
```

螢幕上即畫出一彩色立體圖形。在圖形附近點選滑鼠左鍵, 將圖形框入於一長方形點線內。在圖框內按下滑鼠右鍵, 選取 Save Selection As 中之 EPS, 並鍵入圖形檔名, 硬碟內即產生 EPS 圖形檔。

以上所產生之圖形檔會把背景白紙包含在內, 直接引入 TEX 檔案中, 圖形本身會變的很小。解決辦法如下: 開啓 GSview 軟體, 讀入圖形檔, 啓動 "File|PS to EPS" 功能, 確定視窗內之 "Automatically calculate Bounding Box" 已勾選 (內定值), 按一下 Yes 鍵之後, 鍵入新檔名, 可產生一新的 EPS 檔案, 內含正確的 Bounding Box 座標。圖 11.4是排版後之結果。

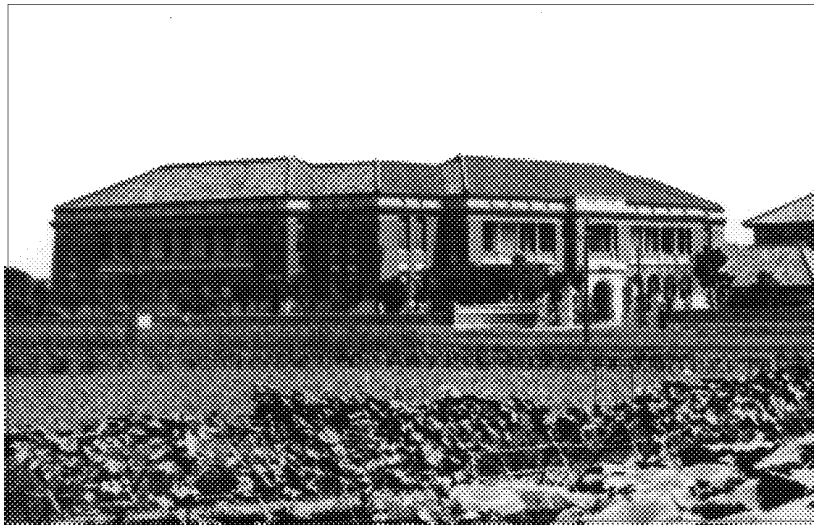


圖 11.5: 台北高等商業學校 (今日台大社會科學/法學院)

統計軟體: Gauss

Gauss 是有名的統計計量軟體, 亦可繪製 PostScript 圖形。以 MSDOS 系統下之 3.2.12 版爲例, 將圖形檔轉換爲 EPS 格式之步驟如下: 當圖形已出現於顯示器上, 請按朝上之箭頭鍵 (代表輸出圖形), 再按 c 鍵 (代表檔案轉換), 最後再按 e 鍵, 即可轉出 EPS 檔案。依原始設定, EPS 檔名將存爲 pqgcvt.out。如果要自動存爲自己選擇之檔名, 請自硬碟中找出 pqgrun.cfg 設定檔案, 將 "cvt_filename=" 設定爲自取之檔名。

畫圖時, 格點請勿設太細, 否則轉換出來之 EPS 檔案會很大, 預視或列印時速度會變慢。Gauss 爲英文軟體, 無法直接鍵入中文字若要在圖中加入中文或數式, 請使用 psfrag 巨集套件。

11.4 引用描點圖形

描點圖形格式衆多, 最常用的包括 tiff, pcx, bmp 等等。例如, 掃描器所產生之圖形通常儲存爲 tiff 格式; Windows 系統內 imaging 軟體所繪出之圖形可儲存爲 pcx 或 bmp 格式。描點圖形有固定解析度。例如, 300dpi 掃描器通常產生 300dpi 之描點圖形。因爲解析度固定, 使用上並不方便。同一描點圖形在 600dpi 印表機印出時, 圖形大小僅爲 300dpi 印表機印出的 1/4。

描點圖形要引入文稿中, 較方便的方法還是先把圖形轉換爲 EPS 格式。依上一節的說明, 利用 PostScript 印表機驅動程式可在 Windows 繪圖軟體內把圖形「列印」爲 PostScript 檔案; 其次利用 GSview 軟體再一次轉換爲 EPS 檔案, 以得到正確的圖形尺寸。圖 11.5 之例子原先是以掃描器掃瞄一張老照片, 取得 school.tif 描點圖形, 再依上述方法轉換爲 school.eps, 最後再引入文稿內。引用圖形之指令如下:

```
\usepackage{graphicx}
\renewcommand{\figurename}{\m11 圖}
\begin{figure}
\centering
\includegraphics[width=\textwidth]{school.eps}
\caption{台北高等商業學校（今日台大社會科學/法學院)}
\end{figure}
```

11.5 圖形位置與標題

上一節圖 11.5 的例子使用 \caption 指令排版圖標題。此一指令僅能用於 table 與 figure 指令環境內; 在 LaTeX 中這兩者合稱爲「浮動版面」指令環境。上一章 10.4 節曾說明如何利用 table 指令環境以控制表格之置放位置, 並排版標題。基本上, figure 指令環境之功能與 table 指令環境相同, 差別只在於標題之名稱不同。在 figure 指令環境內, 使用 \caption 指令排版時, 圖編號之前會自動加上 “Figure” 名稱。若排版中文稿件, 我們可以將 \figurename 改定義爲「圖」或其他中文字, 如圖 11.5 所示。

有關於 figure 指令環境之用法, 請見 10.4 節, 詳細說明請參考 Reckdahl (1997)。

12 圖形與彩色

上一章說明引用外製圖形的方法。巨集套件 graphicx 除了可引進外製圖形外, 還可旋轉文字、圖表等。另外, color 巨集套件可以將圖形、文字上色彩, 或加上灰階 (grayscale) 之背景方塊。若圖形不複雜, 我們也可以使用 picture 指令環境中之指令直接繪製圖形於文稿中。

LaTeX 的繪圖功能有限, 大部分的繪圖功能必須透過預覽/列印軟體來執行。簡單來說, 文稿中若引用外製圖形, 排版時 LaTeX 只在適當地方留出圖形大小的空白。等到列印或預視時, 圖形才由預覽/列印軟體引進版面上。本章將介紹與圖形處理有關的一些巨集套件。若使用這些巨集套件, 通常排版文稿須先以 Dvips 程式轉換爲 PostScript 檔案, 再以 GSview 預覽/列印。若直接以 YAP/Windvi 預覽/列印, 有時候無法得到正確的結果。

12.1 graphicx 巨集套件

LaTeX 提供 graphicx 及 graphics 兩套圖形巨集指令。兩套巨集套件之功能相近, 但前者使用較方便, 故本章僅介紹前者之指令。

巨集套件 graphicx 之指令可用以引入外製圖形, 旋轉或縮放文字圖表。如果是以 Dvips 程式將排版結果轉換爲 PostScript 檔案, 則使用 graphicx 巨集套件時, 應加入選項:

```
\usepackage[dvips]{graphicx}
```

另外一個辦法是建立 graphics.cfg 檔案, 置於 \texmf\tex\latex\config 檔案夾內; 檔案只須包含一行指令:

```
\ExecuteOptions{dvips}
```

若建有此檔案, 引用巨集套件時可以省略 [dvips] 選項。MiKTeX 與 fpTeX 系統都已自動建立設定檔案, 使用者不須再費心。

巨集套件 graphicx 的第一項功能是引入外製圖形, 上一章對此已有詳細的說明。除此之外, graphicx 之指令可用來旋轉或放大縮小文字、圖表。以下分別介紹之。

12.1.1 旋轉文字圖表

有時候, 我們須將版面上部分圖文旋轉某一角度。譬如, 若表格橫寬太大, 須旋轉 90 度才能排入版面, 此時可使用 `\rotatebox` 指令, 其格式如下:

`\rotatebox[`*options*`]{`*angle*`}{`*material*`}`

其中, *options* 表選項, *angle* 表旋轉角度, *material* 表欲旋轉之文字或圖表。若文字段落超過一行, 應將文字段落置於 `\parbox` 或 `minipage` 指令環境中。應注意的是, 全部文字應置於一個段落內。換言之, 欲旋轉之文字或表格中, 不得空一行或加上 `\par` 指令。但是換行 `\\` 指令可以使用。

旋轉角度是以逆時鐘方向計算; 旋轉時以基準點 (reference point) 爲軸心。若爲單一文字, 基準點是基線與左外框之交點; 若爲圖表, 基準點爲左下角。若要以特定點爲旋轉軸心, 我們可以經由選項 `origin=` 來控制。選項設定可以選擇 `c`, `l`, `r`, `t`, `b`, `B` 分別代表文字圖表的中心點, 左, 右, 上, 下, 基線。因此, 若以圖形中心點爲旋轉軸心, 選項指令爲 `origin=c`。上述之設定項還可加以組合, 例如 `lb` 代表旋轉軸心爲左下角。若有必要, 我們還可以直接選擇軸心之座標。細節請見 graphicx 巨集套件之說明檔, 或 Goossens, Rahtz, and Mittelbach (1997)。

圖 12.1 之例子說明不同旋轉軸心點之效果。例子中, 我們首先定義排版 cwTEX 標誌符號之指令。爲了方便比較, 每一行排出三個標誌符號; 前後兩個不作旋轉, 中間的符號則旋轉15度。但依旋轉軸心不同, 排版結果也有明顯差異。第 1 行指令選擇以圖形中心點旋轉; 第 2 行未設定選項, 因此以基準點爲軸心; 第 3 行直接標示軸心點位置距離基點 (`20pt,10pt`)。

圖 12.2 是另一個旋轉文字圖表的例子。例子上半部分先排版一簡單表格, 其中使用 `picture` 指令環境之 `\line` 與 `\put` 指令畫一斜線。有關於 `picture` 指令環境之畫圖指令的使用方法, 請見下一節之說明。例子下半部分說明如何將表格旋轉 90 度。旋轉圖表之指令爲 `\rotatebox`, 但首先須引入 graphicx 巨集套件; 旋轉軸心設定爲原表格之中心點。

cwTEX cwTEX cwTEX

cwTEX cwTEX cwTEX

cwTEX cwTEX cwTEX

```
\usepackage{graphicx}
\def\cw{{\tt cw}\TeX{}}
\cw \rotatebox[origin=c]%
     {15}{\cw} \cw\\
\cw \rotatebox{15}{\cw} \cw\\
\cw \rotatebox[x=20pt,y=10pt]%
    {15}{\cw} \cw
```

圖 12.1: 旋轉文字圖表例 1

x \ y	y_1	y_2
1	34	55
2	25	45

```
\begin{tabular}{l|rr}
~~~~y & & \\[-10pt]
x \put(15,-4){\line(-3,2){20}}
   & $y_1$ & $y_2$ \\
\hline
~1 &  34   &  55   \\
~2 &  25   &  45
\end{tabular}
```

x \ y	1	2
y_1	34	25
y_2	55	45

```
\usepackage{graphicx}
\rotatebox[origin=c]{90}{
\begin{tabular}{l|rr}
...
\end{tabular}}
```

圖 12.2: 旋轉文字圖表例 2

12.1.2 放大或縮小文字圖表

cwTEX 安裝程式設定排版時使用描邊字型, 此一設定的優點是文字可以任意放大縮小, 品質不受影響。就圖形部分而言, 若引入之圖形爲 EPS 檔案, \includegraphics 指令之選項也可以任意放大縮小而無損品質。因此, 一般而言我們並不需要另一套放大/縮小之指令。

不過, 如果是要作不同比例之放大, 則使用 graphicx 巨集套件所提供之指令還是較方便。欲放大與縮小文字圖表, 可使用下列指令:

```
\scalebox{scalefact}{material}
\resizebox*{h-scale}{v-vscale}{material}
\reflectbox{material}
```

第一道指令 \scalebox 是用於將文字圖形比例放大爲選定之倍數, *scalefact*

數字選項設定放大倍數。放大或縮小時, 水平與垂直方向是同一比例。若水平與垂直方向要以不同比例放大或縮小, 則原設定倍數爲水平放大倍數; 其後須加上垂直放大倍數選項。例如, `\scalebox{2}[1.5]{`*text*`}` 表水平倍數爲 2, 垂直倍數爲 1.5。請注意, 垂直倍數是置於方括號內, 請見以下例子:

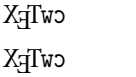
```
\usepackage{graphicx}
\scalebox{2}{\cw}\\
\scalebox{1.2}[1.4]{\cw}
```

若要放大或縮小爲選定之尺寸, 應使用 `\resizebox*` 或 `\resizebox` 指令。例如, `\resizebox*{5cm}{4cm}{...}` 可將文字圖表轉變成 5 公分寬, 4 公分高。若使用無 * 號指令格式, 則計算放大倍數時, 只考慮原文字圖表之高度 (height), 深度 (depth) 不列入考慮。一般而言, 使用加上 * 之指令格式較方便。

若要將文字圖表之寬度拉大至特定大小, 如 5 公分, 而高度希望同比例放大, 可以使用下列指令: `\resizebox*{5cm}{!}`。請參見底下的例子。

```
\resizebox*{2cm}{!}{\cw}
```

`\scalebox` 指令之水平倍數若爲負值, 結果爲鏡面反映。因此, `\reflectbox` 指令之效果與 `\scalebox{-1}[1]` 完全相同。



```
\reflectbox{\cw} \\
\scalebox{-1}[1]{\cw}
```

12.2 picture 指令環境

要在文稿中直接繪製簡單線條圖形, 可使用 picture 指令環境, 或者透過巨集套件間接引用 PostScript 或 METAFONT 之繪圖指令。如果只是簡單的線條圖形, picture 指令通常就能完成使命。使用 picture 指令的好處是圖形指令直接下於文稿中, 不須另存圖形檔。本節將簡單介紹 picture 指令環境之功能。

圖形占有一定之空白, 繪製圖形時首先須在版面上留下特定大小之空白。一般圖形所占用之空白爲長方形, 其大小以座標點表示。例如, 寬度

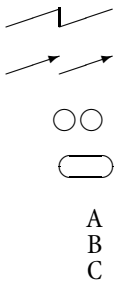
100、高度 60 之長方形, 可定爲左下角座標爲 (0,0), 右上角爲 (100,60)。排版時, 長方形之左下角將排於基線上。座標的單位長度可以 \unitlength 指令任意設定, 例如:

```
\setlength{\unitlength}{mm}
```

即選定公厘 (mm) 爲長度單位。如此, 上例之長方形寬度爲 10 公分, 高度爲 6 公分。長方形之左下角位置可以進一步設定, 例如以下指令所定義之長方形其左下角之位置距離原基準點 (20mm,-10mm):

```
\setlength{\unitlength}{mm}
\begin{picture}(100,60)(20,-10)
 ...
\end{picture}
```

在 picture 指令環境下, 畫直線之指令爲 \line。譬如 \line(2,3){10} 表示畫一直線, 起始點爲 (0,0), 方向爲 (2,3); 長度爲 10。進一步利用 \put 指令, 即可將此直線移至版面上選定的位置。畫箭號之指令爲 \vector, 圓圈之指令爲 \circle, 橢圓之指令爲 \oval。底下列出幾個簡單的例子。



```
\line(3,1){20}\line(0,1){7}\line(3,1){20}
\vector(3,1){20}\vector(3,1){20}
\circle{8} \circle{8}
\put(-10,0){\oval(20,8)}
\put(-10,-25){\shortstack{A\\ B\\ C}}
```

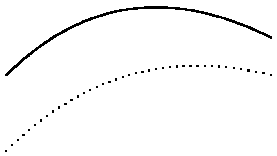
我們將 picture 指令環境內之畫圖指令簡單歸納如下:

- \put(x,y){...}: 將繪製之線條、方塊文字等置於 (x,y) 座標處。
- \multiput(x,y)(a,b){n}{...}: 將繪製之圖形文字或線條等重覆排版 n 次, 起始位置爲 (x,y) 座標, 下一個位置爲 (x+a,y+b), 等等。譬如, \multiput(0,0)(10,1){3}{$\bullet$} 指令產生: • • •

- \line(x,y){*length*}: 繪製直線, 起始點爲 (0,0), 角度方向爲 (x,y), 長度爲 length。參見上面例子之第一行。特別注意的是, 設定角度方向之座標有下列限制: (1) 座標值必須是正負整數, 不可使用小數; (2) 數值只能介於 -6 與 6 之間; (3) x,y 座標值不能有公約數。譬如, (3,6) 不能使用, 因爲兩數可以 3 整除, 但 (1,2) 則無問題。
- \vector(x,y){*length*}: 類似 \line 指令, 但前端加上箭頭。
- \circle{x}: 以基準點爲圓心, x 爲半徑畫圓。
- \oval{x,y}: 畫橢圓, 寬度爲 x, 高度爲 y。橢圓可以想像爲是一長方形, 大小爲 (x,y); 其四角改爲弧形之後即變爲橢圓。
- \qbezier[n](x1,y1)(x2,y2)(x3,y3): 畫 bezier 曲線。選項 [n] 若省略, 結果爲實曲線; 若加入, 爲虛曲線。下面將舉一例子說明。
- \shortstack: 將文字圖表垂直疊在一起, 各行以斷行指令隔開。
- \thicklines, \thinlines: 設定指令環境內線條之粗細。後者爲內設值。此外, 我們尙可直接設定粗細。例如, \linethickness{2pt} 指令即設定線條粗細爲 2pt。
- \dashbox{w}(x,y)[*pos*]{...}: 將圖表加上點折線之方形外框。其中, {w} 選項設定每一點折線之長度。(x,y) 設定長方形外框之尺寸, [*pos*] 設定文字圖表位於框內之位置。[*pos*] 可選用 t, b, l, r, 及 s。前四選項分別代表文字圖表在框內靠上, 靠下方, 靠左, 靠右。以上之選項可進一步組合, 譬如 [tl] 表靠左上角。s 選項則表示圖表在水平方向將選定之外框填滿, 垂直方向則居中。此外, \framebox 與 \makebox 等指令也可以在 picture 指令環境內使用。

以上所介紹之指令大部分用於畫直線或圓形, 但我們可以使用 \qbezier 指令畫曲線圖, 圖 12.3 是一個簡單的例子。此項指令須設定 3 個座標, 第 1 個座標爲曲線起點, 第 2 個座標爲線條前進之方向, 第 3 個座標爲經過中間轉折之後, 線條所抵達的終點。第二道指令之後加上 [40] 之選項, 整條曲線變成由 40 個細點所構成的曲線。

以上所介紹的 picture 指令雖然功能簡單, 但用於繪製簡單之線條圖形卻是綽綽有餘。不過, 繪圖時須設定每一線條之位置、長度等等, 使用上不是很方便。有一些巨集套件以上述指令爲基礎, 但讓使用者方便繪製圖形。譬如, Peter Vanroose 之 trees 巨集套件用於繪製樹狀圖甚爲方便。



```
\setlength{\unitlength}{1mm}
\begin{picture}(40,30)
\qbezier(5,15)(20,30)(40,20)
\qbezier[40](5,5)(20,20)(40,15)
\end{picture}
```

圖 12.3: 曲線圖

另外一些巨集套件則是延伸 picture 指令環境之功能。譬如, Sunil Podar 之 epic, 或者 Conrad Kwok 之 eepic 巨集套件都屬於此類。以上之巨集套件可以自網路上取得, 其中內附有說明檔。使用者亦可參考 Goossens, Mittelbach, and Samarin (1994) 與 Goossens, Rahtz, and Mittelbach (1997) 之比較說明。

12.3 彩色圖文

欲將文字或圖形加上色彩, 最簡單的方法是使用 LaTeX 所提供之 color 巨集套件。本節所謂的「色彩」, 包括不同層次之灰階 (grayscale)。使用 color 巨集套件時, 必須配合以特定之列印與視覽程式。最簡單的方法是以 DVIPS 程式產生 PostScript 檔案, 再以 GSview 預覽/列印。如果以 YAP 或 Windvi 有些顏色可能無法正確顯示或列印。

色彩巨集套件最常見之應用有兩類, 一是將文字著色, 一是在文字或圖形之後加上灰階或顏色背景。color 巨集套件提供一些事先定義之顏色, 但也可以自行定義所要的顏色。使用色彩指令, 首先在全文設定區引入 color 巨集套件:

```
\usepackage[dvips]{color}
```

其中, [dvips] 選項的目的是設定使用 DVIPS 程式轉換排版結果。MiKTeX 與 fpTeX 都已自動設定, 故此選項可以不加。

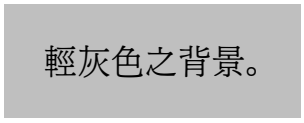
欲將文字塗上色彩, 指令如下:

```
\textcolor{red}{\bb11 請注意}, 三分鐘之後 ...
```

排版之後, 以 GSview 軟體預視, 「**請注意**」三個字將爲紅色之粗黑體字。可

以直接使用之色彩選項爲 black, white, red, green, blue, yellow, cyan 與 magenta。若要自行調色, 請參考 color 巨集套件之說明, 或者 Goossens, Rahtz, and Mittelbach (1997)。

欲將文字或圖形加上灰階背景, 可以使用 \colorbox 指令。使用灰階色之前, 首先須定義灰階色。灰階色深淺是以一介於 0 與 1 之間的數字代表, 指數爲 1 表示全白, 指數 0 表示全黑。底下的指令定義 slight 爲 0.75 度之灰階色, 再以之創造之灰階方塊:




```
\usepackage{color}
\fboxsep=15pt
\definecolor{slight}{gray}{0.75}
\colorbox{slight}{\m14 輕灰色之背景。}
```

第 2 行指令定義灰階之深淺, gray 是 color 巨集套件之指令, slight 則是自取之名稱。另外, 背景外框與文字之距離可由 \fboxsep 指令控制, 本例設定爲 15pt。

若要將整段文字加上灰階背景, 可將整段文字排於 minipage 指令環境內, 再以顏色指令色上色。例如:

```
\colorbox{light}{\begin{minipage}
  ...
\end{minipage}}
```

如果要將整頁版面填上顏色, 可使用 \pagecolor 指令。

底下例子以 \textcolor 與 \colorbox 兩項指令創造出黑底白字之效果。其中, \colorbox 指令先創造出一接近黑色之方塊, 再以 \textcolor 指令在其中「挖出」白字。指令第 4 行之 white 代表純白色, 這是 color 巨集套件所提供之現成顏色, 故不須再自行定義。


```
\usepackage{color}
\fboxsep=15pt
\definecolor{dark}{gray}{0.4}
\colorbox{dark}{\textcolor{white}%
{\bb20 請特別注意 {\LARGE !}}}
```

本例也使用 \fboxsep=15pt 指令控制方格邊緣與其內文字之間距。

12.4 contour 巨集套件

較晚近的繪圖軟體大都提供創造立體陰影 (dropping shadows) 效果之指令。要在 LaTeX 文稿中產生類似的效果, 可使用 contour 巨集套件, 作者爲 Harald Harders。此一巨集套件的能力當然不能和一般的繪圖軟體相比, 但在某些應用下, 使用相當方便。

欲使用此巨集套件, 首先須定義一彩色或灰階色。使用上一節所定義之 slight 灰階色, 則以下指令:

```
\usepackage{contour}
\contour{slight}{\bb30 台大經濟系}
```

排版結果爲:

台大經濟系

如果將 slight 改爲其他深淺度之灰階色, 排版效果會不一樣, 有興趣者請自行嘗試。

加上立體陰影之文字也可以置於灰階背景上, 例如:

```
\colorbox{dark}{\contour{white}{\bb30 台大經濟系}}
```

排版結果爲:

台大經濟系

巨集套件 contour 所產生的陰影效果尙可自行調整。有興趣者, 請參閱其說明檔。

12.5 wrapfig 巨集套件

引用外製圖形時, 通常是使用 figure 指令環境將圖表置於版面上方或下方。但有時候, 我們希望把圖形或表格置於正文方格內, 或者版面靠左邊, 或者靠右邊; 甚至是置於正方塊內, 四周以文字包圍。此時可以使用 Donald Arseneau 所寫的 wrapfig 巨集套件。表面上看來, wrapfig 之功能類似 figure 指令環境, 但指令與實際功能並不相同。

舉例言之, 版面旁邊的 EPS 圖形是以 wrapfig 指令環境引入, 使用之指令如下:

```
\begin{wrapfigure}[3]{r}[1cm]{3.4cm}
\includegraphics[width=3cm]{cat.eps}
\end{wrapfigure}
\mbox{}
舉例言之，版面 ...
```

以上指令是下於上段與本段文字之間, 其下立即接上本段內文。請注意, 在指令環境之後我們須特別加上 \mbox{} 指令, 否則排版時會出現錯誤。(其原因不明, 但可能是下接之文字爲中文字有關。)

wrapfig 指令環境第 1 選項 [3] 設定圖形高度占 3 行文字。此選項若不加上, 巨集套件會自動計算。第 2 選項 {r} 指示將圖形置於版面右邊。若選項改爲 {l} 則圖形將置於左邊。第3 選項 [1cm] 指示將圖形凸出於版面邊緣外 1 公分。若不加入此項設定, 內設值爲 [0cm]。第 4 選項 {3.4cm} 指示圖形所占寬度爲 3.4 分。實際引入圖形時, 寬度設定爲 3 公分, 讓左右兩邊各有一些空白。巨集指令會自動計算圖形本身所占之高度, 不過我們也可以自行設定其高度; 設定方法請見巨集套件內附之說明。

利用 wrapfig 巨集套件, 我們可以將章節起頭第一個字特別放大。譬如, 本節第一個「引」字就是以下列指令排版:

```
\intextsep=0pt
\begin{wrapfigure}{l}{20pt}
\textcolor{heavy}{\mbox{\mu30 引}}
\end{wrapfigure}
\noindent 用外製圖形時, ...
```

在洋文書中, 此種作法很常見, 稱之爲 drop caps。字體放大之後, 筆劃變粗。若以全黑印出, 太過醒目, 因此我們使用 \textcolor 指令設定以較淺的灰階排版。

請注意, wrapfigure 指令環境之後須緊接著輸入段落文字, 否則會出現錯誤。上述指令第一行爲 \intextsep=0pt, 其功能是設定圖形與周圍文字之間距。巨集套件 wrapfig 應用上有一些限制, 其使用說明直接置於 wrapfig.sty 檔案內, 請自行參考。

12.6 PSTricks 巨集套件

上一節說明如何在文稿中直接繪製線條圖形。但是, LaTeX 的繪圖指令功能不強, 較複雜的圖形即無能爲力。第11章曾說明如何引用外製圖形。LaTeX 引用 PS 檔案時, 實際上是引入一連串的 PostScript 指令。因此, 必要時我們也可以在文稿中直接使用 PostScript 指令。要在文稿中使用 PostScript 畫圖指令, Timothy van Zandt 所寫的 PSTricks 巨集套件可能是功能最強、指令最完整者。此一巨集套件內含完整的指令說明檔案。Goossens, Rahtz, and Mittelbach (1997) 對此巨集套件有詳細的說明, 並與其他繪圖巨集套件作比較, 很值得參考。本節之說明主要參考該書。

PSTricks 功能強大, 指令很多, 我們無法在此詳細說明。底下只以幾個例子展示其功能, 欲了解這些例子所使用的指令, 請參考巨集套件所附之說明檔。PSTricks 套件指令直接使用 PS 指令, 因此我們必須使用 PS 印表機, 或者 GSview 軟體才能列印或預覽排版結果。

PSTricks 早在1993年就發展出來, 它和後來發展的 LaTeX 有部分地方不完全匹配。如果你要同時使用 graphicx, color 與 PSTricks 巨集套件, 你還必須引入 David Carlisle 所寫的 pstcol 巨集套件, 而且各巨集套件應依以上順序引入。譬如, 若欲引用畫樹狀圖之巨集套件 pst-tree, 全文設定區所下之指令爲:

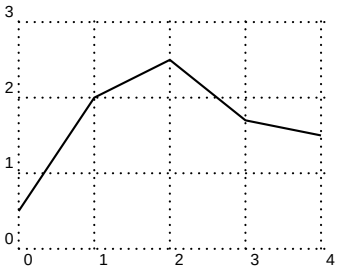
```
\usepackage{graphicx,color,pstcol,pst-tree}
```

若引入順序不對, 排版時出現錯誤。

整套 PSTricks 巨集套件包含幾個較小的巨集套件, 各有不同功能。底下以幾個例子展示其功能。第一個例子是曲線圖。畫曲線圖須使用 pstcol 巨集套件, 因此我們先於全文設定區引用 pstcol 巨集套件。此外, 我們還定義巨集指令 \showgrid, 其功能是畫出點格線。巨集指令之定義如下:

```
\usepackage{pstcol}
\newpsobject{showgrid}{psgrid}%
{subgriddiv=1,griddots=10,gridlabels=6pt}
```

因爲定義式稍長, 我們以 % 指令將之拆爲兩行。經過以上定義之後, 畫圖指令之第1行界定長方形的大小; 第2行指令則以 \psline 指令之座標點直接描出折線。




```
\usepackage{pstcol}
\usepsobject{showgrid}{...}
\begin{pspicture}(0,0)(4,3)\showgrid
\psline(0,.5)(1,2)(2,2.5)(3,1.7)(4,1.5)
\end{pspicture}
```

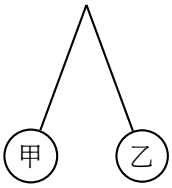
第二個例子是將文字加入 PS 指令所繪成的橢圓中。指令中設定橢圓是以紅線畫出。我們還可以進一步使用指令讓 cwTEX 標誌符號以另一種顏色排版出來。若使用 GSview 軟體預視, 即可看到彩色的圖案; 使用彩色印表機即可印出紅色橢圓。以單色印表機列印時, 紅線自動轉換爲灰階線條。


```
\usepackage{pstcol,pst-grad}
\psovalbox[linecolor=red]
 {\Large{\tt cw}\TeX}
```

PostScript 處理文字的功能特別強。12.1 節曾介紹如何以 graphicx 巨集套件指令將文字圖形放大、縮小、旋轉等。PSTricks 巨集套件同樣有這些功能。而且, 後者的某些功能是前者無法做到的。最後一個例子先畫出一條灰階曲線, 再讓文字與數式沿著此一條線的形狀排列出來。

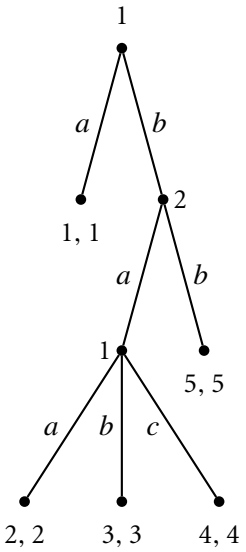

```
\usepackage{pstcol,pst-text}
\begin{pspicture}(-4,-3)(3,0)
\psset{linecolor=lightgray}
\pstextpath
{\pscurve(-4,-2)(-3,-1)(-1,-2)(0,-1)}
 {數式
  $\sum^\infty_0 U(c_t, x_t)$
  與曲線 \ldots}
\end{pspicture}
```

最後一個例子說明如何繪製樹狀圖 (trees)。許多學科領域都可能須繪製樹狀圖, 但在 LATEX 文稿內直接繪製樹狀圖並不容易。PSTricks 繪製樹狀圖之能力甚例, 本例是一個最簡單的例子。繪製本圖形, 除了 pstcol 巨集套件之外, 尚需 pst-node 及 pst-tree 巨集套件。實際應用時, 樹狀圖可以一層一層地接續下去; 每一分叉處可以排版文字、粗黑點、圓圈、方塊等等。




```
\usepackage{pstcol,pst-node,pst-tree}
\pstree{\Tp}{
\Tcircle{甲}
\Tcircle{乙}}
```

以上是一個簡單的流程圖, PSTricks 有能力排版各式各樣複雜的流程圖, 底下是另一個流程圖例子。




```
\usepackage{pst-node,pst-tree}
\psset{labelsep=2pt,tnpos=a,radius=2pt}
\pstree{\TC*~{1}}
{\TC*~{}~[tnpos=b]{$1,1$}\tlput{$a$}
 \pstree{\TC*~{}~[tnpos=r]{2}\trput{$b$}}
 {\pstree{\TC*~{}~[tnpos=l]{1}\tlput{$a$}}
  {\TC*~{}~[tnpos=b]{$2,2$}\tlput{$a$}
   \TC*~{}~[tnpos=b]{$3,3$}\tlput{$b$}
   \TC*~{}~[tnpos=b]{$4,4$}\tlput{$c$}}
   \TC*~{}~[tnpos=b]{$5,5$}\trput{$b$}}}
```

13 PostScript 字體與軟體工具

中國於第 11 世紀發明活字排版技術, 德國人 Johannes Gutenberg 則於 15 世紀發明類似的技術。從 15 世紀至 20 世紀中葉, 活字排版的技術基本上並無改變。傳統排版工廠裡, 排版工人自字盤上取出鉛字, 一個一個併在一起, 組合成一頁一頁的版面。字模沾上油墨, 壓印於白紙上, 即印出書稿。因此, 排版品質的高低決定於兩項因素:一是字體本身是否優美, 二是版面設計是否適當。

1980 年代初期, 電腦排版開始發展; 排版軟體取代了傳統的排版工人, 電腦字體則取代了傳統的鉛字。從這個角度來看, TEX 排版系統分兩部分, 第一部分是 TEX 軟體本身, 其功能是版面安排; 第二部分是各式各樣的英數字體。傳統的鉛字是將字體鑄於鉛塊上, 電腦字體則是將文字的形狀記錄於程式裡。換言之, 每一個單字的形狀, 不管是中、英文, 都是以電腦程式描繪其形狀。排版之後, 列印程式解讀每一個字型程式, 變成黑白或彩色細點, 驅動印表機印出結果。在現代科技下, 每一個字形都是一組程式。

但是, 同樣一個英文字母卻有不同的程式寫法。譬如, Knuth 所使用之造字軟體稱爲 METAFONT, 其所創造出來的字體稱爲 METAFONT 字體。另一套有名的字型規格是 Adobe 公司所發展的 PostScript 字體, 又稱爲 Type1 字體。雖然 TEX 一開始使用 METAFONT 字體, 但以目前的發展狀況來看, PostScript 字體的使用彈性最大, 字型選擇也最多。本章將簡單說明 LATEX 如何使用 PostScript 字型排版, 並介紹一些與使用字體有關的工具程式。

13.1 METAFONT 與 PostScript 字體

Knuth 設計了八十多種字體, 其中有些是一般的英數字母, 其餘是用於排版數學式的符號。大部分的字型檔案是以 `cm` 起頭, 代表 Computer Modern, 以下將簡稱爲 CM 字體。譬如, 12 點之羅馬字體 Computer Modern Roman 檔名爲 `cmr12.mf`; 10 點之仿打字機字體 Computer Modern Typewriter 檔名

爲 cmtt10.mf。METAFONT 軟體是以數學式精確地描繪每一個字母的外框, 譬如, cmr12.mf 檔案內即包含約一百組程式, 每一組程式描繪一個字母。

排版時, TEX 僅需要每一個字母之寬度、高度等資訊, 並不須知道字母實際的黑白點構成。但是, 排版完成後, 預覽/列印軟體就需要每一個字母之描點字型檔案, 才能顯示或列印。描點字型檔可由描邊字型檔創造出來。METAFONT 描邊字型檔所產生之描點字型檔之延伸檔名爲 .pk, 譬如 cmr12 描點字型檔名爲 cmr12.pk。在 MiKTEX 系統下, 開啓 YAP 預覽/列印軟體時, 顯示器有時會出現 "YAP is creating cmtt10.pk..." 訊息, 即表示正由 cmtt10.mf 產生描點字型檔。fpTEX 之 Windvi 軟體也有類似的動作。

以描點字型預覽/列印, 好處是速度快, 缺點是不同印表機需不同的描點字型。舉例言之, 如果你有一部 HP1100 印表機, 精密度爲 600dpi。爲了加快列印速度, 平常可能以 300dpi 列印初稿, 等到最後一稿再以 600dpi 列印。如此一來, 硬碟須同時儲存兩種精密度之描點字型檔。如果排版文稿最後印成書籍, 以 1200dpi 列印, 則電腦還須產生 1200dpi 描點字型檔。

雖然 PostScript 字體也是描邊字型檔, 但它卻容許直接以描邊字型檔列印, 不須先產生一套描點字型檔。具體言之, 排版完成之後, 先以 DVIPS 將 DVI 檔案轉換成 PostScript 格式, 其內儲存文稿所使用的每一個英文字母或中文單字之描邊字型程式。以 GSview 預覽/列印時, 軟體即時在記憶體內產生描點字型以供使用。因此, 預覽/列印速度會稍慢, 但好處是硬碟內不須儲存一大堆描點字型檔; 而且, 同一個 PostScript 檔案可在任何精密度之顯示器或印表機上顯示/列印。

1990 年代初期, Microsoft 公司發展 Windows 作業系統時, 同時發展一套字型程式規格, 稱爲 True Type。因爲 Windows 系統之普及, True Type 字體也非常普及。目前中文 Windows 系統上使用之字體, 絕大部分都是 True Type 規格。Windows 文書軟體也是直接以描邊字型檔列印。譬如, Word 軟體列印文稿時, 也是先在記憶體內將 True Type 字體轉換成描點字型, 再進一步處理印出。但是, True Type 字體與 TEX 之結合使用尚未十分成熟, 因此本章之介紹將以 PostScript 字體爲主。

13.1.1 PostScript 格式之 CM 字體

許多個人電腦的應用繪圖軟體, 譬如, Corel Draw 與 Illustrator 都附有多種 PostScript 字型檔。Ghostscript 4.0 版開始, 附有幾套各字體設計公司捐贈

的 PostScript 字體。其中, 德國 URW++ 公司所捐贈的一整套字體, 品質相當不錯。因此, 有一些 PostScript 字體是可以免費使用的。但是, 如果你要排版的是數學文稿, 則問題是在於能用於排版數學式之 PostScript 字體很少。有些人排版一般文字時使用 PostScript 字體, 但數學式卻被迫使用 METAFONT 數學字體。但是, 每一種字體有其風格, 如果你使用 PostScript 之 Times 字體排版正文, 數學式卻使用 METAFONT 字體, 版面看來並不協調, 我們在稍後會以實際例子說明。

因爲一般商用之 PostScript 字體無法提供足夠的數學符號, 有人即設法將 METAFONT 之數學符號字體轉換爲 PostScript 格式。目前, CTAN 上至少有兩套免費使用之 PostScript 格式 CM 字體, 其中一套是由美國數學學會提供。利用此 PostScript 字型檔, 若以 CM 字體排版文稿, 預覽/列印時可使用描點或描邊字型檔。譬如, 以 YAP 或 Windvi 預覽/列印時, 使用描點字型。反之, 排版結果若先以 DVIPS 轉換爲 PostScript 檔, 再以 GSview 預視/列印, 則可直接使用描邊字型。

本章的討論將會介紹一些有名的英文字體, 如 Garamond, Palatino 等。有意進一步探究的讀者, 請參考 Bringhurst (1996)。

13.1.2 中文 PostScript 字體

cwTEX 系統所使用的中文字體都是 PostScript 格式。一般的英文 PostScript 字體組, 如 Times, 包含: 中體字、斜體字、粗體字、與粗斜體字共計 4 個檔案。cwTEX 中文字體並無類似的字體組合; 但是, PostScript 字體可以作傾斜或水平放大/縮小之變形。因此, 由一種中文字體可以產生多種變化字體出來。譬如, 原始之12pt 粗黑體字, 指令爲 \bb12; 我們可由此產生三種變化字體: 斜粗黑字體, 指令爲 \bbs12; 狹長粗黑字體, 指令爲 \bbe12; 以及兩種變形同時出現, 指令爲 \bbes12。同理, 若是 12pt 明體斜體字, 指令爲 \ms12。中文變形字體之例子請見圖 13.1。

中文狹長字體或斜體字的主要用途是在排版標題或者簡短的傳單海報上。如果使用狹長字體排版文稿段落, 你會發現英文標點符號後之空白顯得太大; 英文單字之間距也顯得太大。欲縮小英文字距與標點符號後之空白, 可於段落之前使用下列指令:

```
\fontdimen2\font=0.9\fontdimen2\font
```

\m14 -- 明體正體字型

\ms14 -- *明體斜體字型*

\bb14 -- **粗黑體正體字型**

\bbe14 -- **粗黑體狹長字型**

\bbes14 -- ***粗黑體狹長斜體字型***

圖 13.1: 中文斜體與狹長字型

其中, \fontdimen2 爲 TEX 控制英文字距之指令。以上指令將字距縮小爲原值的90%。

13.1.3 重新設計中文變形字

中文斜體字或狹長字都是由正體字體變形而來。安裝 cwTEX 時, 安裝程式會自動設定; 其中, 斜體字之傾斜度爲16.7 度, 狹長字則是將原始字體水平縮小 91%。如果你想要改變傾斜度或水平縮小比例, 最簡單的方法是重跑一次 addfont.bat。該檔案置於光碟 \windows\miktex\cwtex 檔案夾內, 請將檔案複製於 c:\xtemp, 以文字編輯軟體開啓, 依其中之指令進行。以下簡單說明設定的原理。

首先, 我們說明字型規格檔的概念。排版時, TEX 需要知道每一個字母或中文字之寬度與高度。每一個 TEX 字體都有一個字型規格 (font metrics) 檔案, 延伸檔名爲 .tfm。譬如, 英文 cmr12 字型規格檔名爲 cmr12.tfm; 中文字體 m2 之字型規格檔名爲 m2.tfm。PostScript 字體也有其字型規格檔, 以 .afm 爲延伸檔名。譬如, m2 字型規格檔名爲 m2.afm。TEX 之字型規格檔與 PostScript 字型規格檔內容並不同; 不過, 由後者可以創造出前者。

欲使用水平縮小 (放大) 或斜體字體, 我們須創造 TEX 系統之 .tfm 字型規格檔案。在 MiKTEX 或 fpTEX中, 中文描邊字型皆置於 \texmf\fonts\type1\cwtex 檔案夾內。以明體常用字爲例, PostScript 描邊字型檔案名稱爲 m0.pfb 至 m26.pfb, 共計 27 個; 而字型規格檔之延伸檔名爲 .afm, 置於 \texmf\fonts\afm\cwtex 檔案夾內。

TEX 排版所需之 .tfm 字型規格檔案可以由 PostScript 之 .afm 字型規格檔案轉換出來, 轉換方法是利用 DVIPS 所附之工具程式: afm2tfm。以

m2 字型檔爲例, 欲產生斜體字 ms2 之 ms2.tfm 字型規格檔, 請先進入存放 PostScript 字型規格檔之檔案夾內, 執行:

```
c:\texmf\fonts\afm\cwtex>afm2tfm m2 -s 0.167 ms2
```

將由 m2.afm 產生 ms2.tfm, 其中, -s 0.167 設定傾斜度爲 0.167。此一角度是普遍使用之值, 但我們可以任意選擇, 甚至負值也可以使用。

依同樣方法產生全部的 27 個字型規格檔案, 再將所有的 .tfm 檔案移至 \texmf\fonts\tfm\cwtex 檔案夾內。檔案移入之後, 須更新檔案資料庫; MiKTEX 使用者請於 DOS 模式內執行:

```
c:>initexmf -u
```

若安裝 fpTEX, 請執行:

```
c:>mktexlsr
```

以上設定已足以讓 LATEX 作排版工作, 但要順利執行 DVIPS, 還須進一步設定。在 \texmf\dvips\cwtex 檔案夾內可找到 cwtex.map 字型對應檔, 以文字編輯軟體叫出, 其內容如下:

```
m0 CWTEX-M0 <cwtex.enc <m0.pfb
m1 CWTEX-M1 <cwtex.enc <m1.pfb
m2 CWTEX-M2 <cwtex.enc <m2.pfb
...
```

第一行指令說明 m0 字體之全名爲 CWTEX-M0, 每一個單字在檔案內之排列是依照 cwtex.enc 檔案所指定之順序; 列印時應自硬碟中取用 m0.pfb 字型檔。同理, 第 2-3 行分別定義 m1 與 m2 等字體。

實際列印時, 假設 DVIPS 發現文稿內使用了 m1 字體中第 115 個單字, 它即自硬碟內尋找 m1.pfb 字型檔案, 依據 cwtex.enc 所排定之順序, 找出第 115 個單字之描邊程式, 將之轉換爲一般之 PostScript 格式; 下一階段再由 GSview 印出。

爲了方便區分, cwtex.map 檔案僅包含原形字體之對應指令; 變形字之對應設定則置於 cwtex1.map 檔案內。明體斜體字形事實上是由正體字轉換而來, 因此 cwtex1.map 檔案也須設定 27 行定義。再以前三個字體爲例, 指令如下:

```
ms0 CWTEX-M0 ".167 SlantFont" <cwtex.enc <m0.pfb
ms1 CWTEX-M1 ".167 SlantFont" <cwtex.enc <m1.pfb
ms2 CWTEX-M2 ".167 SlantFont" <cwtex.enc <m2.pfb
...
```

第一行指令之意義爲: 斜體字 ms0 之全名爲 CWTEX-M0, 列印時應自硬碟中取用 m0.pfb 字型檔, 但字體應傾斜 0.167 度。文稿中若使用 12pt 之斜明體字 ms2, DVIPS 程式將取用 m2.pfb 字型檔, 並作指定之變形。

如圖 13.1 所示, 除了斜體字之外, 我們也可以將字體水平放大或縮小。若粗黑體字要水平縮小 91%, 首先須製造 .tfm 字型規格檔。以 bb3 字體爲例, 指令爲:

```
c:\texmf\fonts\afm\cwtex>afm2tfm bb3 -e 0.91 bbe3
```

其中 -e 0.91 表示字體將水平縮小 91%; 此比率數字可以任意選定。若數值小於 1, 變形字體將較原字體爲瘦; 若參數值大於 1, 變形字體將較胖, 其次, cwtex1.map 檔案內亦須添加 27 行定義, 指令如下:

```
bbe0 CWTEX-M0 "0.91 ExtendFont" <cwtex.enc <bb0.pfb
bbe1 CWTEX-M1 "0.91 ExtendFont" <cwtex.enc <bb1.pfb
bbe2 CWTEX-M2 "0.91 ExtendFont" <cwtex.enc <bb2.pfb
...
```

其中, "0.91 ExtendFont" 表示字體應水平縮小 91%。

經過以上設定, \ms20 可用以排出 20 點之明體斜體字, \bbe20 則排出 20 點水平縮小 91% 之粗黑體。若是水平縮小與傾斜同時作用, 則須一起加入上兩項設定, 請參見 cwtex1.map 檔案。該檔案置於 \texmf\dvips\cwtex 檔案夾內。有關於 afm2tfm 之使用方法, 請見 13.4.2 節進一步的說明。

13.2 英文 PostScript 字體

大部分的 PostScript 字體都是商業軟體, 但也有少部分可以自網路免費下載。Adobe 公司的 Times 字體是一商業軟體, 但該公司的 Acrobat Reader 可免費下載, 其中即附有此套字體。因此, 電腦安裝 Acrobat Reader 時, 即同時安裝此一字體。Times 字體族計有中體、斜體、粗體、斜粗體等 4 種, 分爲 4 個字型檔案。取得了 4 個 PostScript 字體之後, 我們如何使用它們來排版呢? 這個問題看來簡單, 其實其中牽涉許多問題。

首先, TEX 排版時需讀取字型規格檔 .tfm 之字體資訊, 但 PostScript 字體僅提供 .afm 字型規格檔。但此一問題不難解決。上一節曾介紹 afm2tfm 程式, 可用來將 .afm 檔案轉換爲 .tfm 檔案。其次, TEX 所使用之 METAFONT 字體, 一個字型檔含128個字母; 但一般 PostScript 字型檔約含有220個字母。除了字母個數不同之外, 字母排列順序也不完全相同。換言之, 兩種字型檔字元排序 (encoding) 方式不同。如果直接取用 PostScript 字型檔內之字母, 有可能把英磅符號 £ 排版成貨幣符號 $。

解決以上問題的辦法是透過巨集套件。例如, 若硬碟中已安裝 Times 描邊字型檔案, 則利用 LATEX 所提供的 times 巨集套件, 即可使用此一字體。輸入文稿時, 在全文設定區加入下列一行指令:

```
\usepackage{times}
```

文稿之正文即全部改用 Times 字體排版。以上指令引入 times 巨集套件, 它是 PSNFSS 的一部分, 作者爲 Sebastian Rahtz。

PostScript 字體數以千計, 但常用的並不多。而且, 不同字體公司可能提供同一檔名之字體, 但卻各有獨特的設計。譬如, 大部分字體公司都提供 Garamond 字體, 但每一家各有其獨特設計的風格。此外, 新字體每年不斷出現。因此, 要對每一種字體都提供巨集套件顯然有困難。PSNFSS 所提供的巨集套件主要是針對使用率較高的字體。譬如, 一般 PostScript 印表機內附有 Adobe 公司的35種字體, 這可能是最普遍流行的 PostScript 字體。PSNFSS 巨集套件主要即支援這些字體。

顧名思義, PSNFSS 是架構在新版 LATEX 的 NFSS 字體選用法之上。每一種字體依其特徵可以作底下的分類: 字體族 (font family), 字體型 (font shape), 字體序列 (font series) 等。譬如, 在 Times 的字體族之下, 我們可以選用中體字序列, 也可以用粗體字序列; 可以選用正體字形, 也可以用意大利斜體字形。萬一我們所購置之 PostScript 字型檔不足, 則當文稿中選用硬碟中所沒有的字形時, PSNFSS 會自動選用替代字形。

上文提及, 有些字體設計公司捐贈了幾套 PostScript 字體免費提供使用。其中, URW++ 公司所捐贈之字體即涵蓋了 Adobe 的35種標準字體。此外, Adobe 公司所捐贈之 Utopia 字體及 Bitstream 公司所捐贈之 Charter 字體, 品質也相當好。以下簡單介紹如何使用英文 PostScript 字體。

13.2.1 charter 與 utopia 巨集套件

安裝 cwTEX 系統時, 硬碟內會自動安裝三套使用 PostScript 字體之巨集套件, 其中 charter 與 utopia 僅含一般英數文字, mathptmx 則尚含有數學符號。欲使用 chapter 字體時, 僅須在全文設定區引用巨集套件即可:

```
\usepackage{charter}
```

排版時, 正文的羅馬字族即自動以 Charter 字體替代。同理, 若指令行內之 charter 改爲 utopia, 則選用 Utopia 字體。底下是兩種字體的樣式:

Charter: ABCDEFGHIJKL 13579 mnopqrstuvwxyz
Utopia: ABCDEFGHIJKL 13579 mnopqrstuvwxyz

Charter 字體的設計風格是筆劃較粗,因此即使是在較低密度的印表機上列印, 效果也很好。

13.2.2 mathptmx 巨集套件

上一小節所介紹的 charter 巨集套件僅取代正文羅馬字體, typewriter, sans serif, 與數學字體仍使用原來的 METAFONT 字體。但是每一套字體各有其獨特風格, 因此字體之搭配須仔細選擇。有些人選用 Times 字體排版正文, 但數學式仍使用 TEX 原來之數學字體, 其結果並不見得理想。

PSNFSS 提供 mathptm 巨集套件, 其中使用 Times 字體排版正體字, 並使用 Symbol 字體排版數學符號。不過, 有些特別的數學符號, 如開根號或積分符號, 仍取自 TEX 之數學符號。cwTEX 安裝程式會自動安裝 mathptmx 巨集套件, 此爲 mathptm 之更新版本; 正文之 PostScript 描邊字型則取自 Adobe 公司免費下載之 Acrobat Reader 3.0 版。使用 mathptmx 巨集套件僅須在全文設定區加入下列指令即可:

```
\usepackage{mathptmx}
```

若文稿內並無數式, 以上指令仍將正文改以 Times 字體排版。

圖 13.2 比較三種正文與數式字體。最上一列是 TEX 標準的 METAFONT 字體; 中間一列是 mathptmx 所排版之結果; 除了正文字體不同外, 數學符號也改變了。底下一列正文字體使用 Adobe Garamond, 數學符號則使用

Computer Modern math fonts:

$$\phi(t) = \frac{1}{\sqrt{2\pi}} \int_0^t e^{-x^2/2} dx$$

$$a_0 + \left(\sum_{t=0}^{\infty} \frac{y_t}{1+r} - \sum_{t=0}^{\infty} \frac{c_t}{1+r} \right) = 0$$

ABCDEFGHIJKL 1234567890 mnopqrstuvwxyz

---

mathptmx package:

$$\phi(t) = \frac{1}{\sqrt{2\pi}} \int_0^t e^{-x^2/2} dx$$

$$a_0 + \left(\sum_{t=0}^{\infty} \frac{y_t}{1+r} - \sum_{t=0}^{\infty} \frac{c_t}{1+r} \right) = 0$$

ABCDEFGHIJKL 1234567890 mnopqrstuvwxyz

Adobe Garamond plus Mathtime math fonts:

$$\phi(t) = \frac{1}{\sqrt{2\pi}} \int_0^t e^{-x^2/2} dx$$

$$a_0 + \left(\sum_{t=0}^{\infty} \frac{y_t}{1+r} - \sum_{t=0}^{\infty} \frac{c_t}{1+r} \right) = 0$$

ABCDEFGHIJKL 1234567890 mnopqrstuvwxyz

圖 13.2: 三種正文與數式字體之比較

Mathtime 字體。這兩套字體都是商業軟體, 後者是由 Y&Y 公司發展, 相關資訊請見網站:

`http://www.yandy.org`

13.3 創造英文 PostScript 字體巨集套件

字體是排版品質的關鍵。即使是中文稿, 同一中文字體搭配不同的英數字體即可產生完全不同的效果。大部分的英文繪圖軟體都附有 PostScript 字體, 譬如, 英文版 Corel Draw 即附有數百種字體, Free Hand 軟體亦附有多種字體。

取得字型檔後, 我們須透過巨集套件才能使用這些字體。Ghostscript 附有德國 URW++ 公司所捐贈的數十套字型檔案, 下一小節將以其中的 Palatino 及 Helvetica 兩套字體爲例, 說明創造 PostScript 巨集套件的方法。

13.3.1 fontinst 巨集套件

創造 PostScript 字體巨集套件最好的方法是使用 Alan Jeffrey 與 Rowland McDonnell 所寫的 fontinst 巨集套件。一套英文 PostScript 字體通常包含四個 `.pfb` 字型檔案, 分別對應正體、斜體、粗體與粗斜體字。另外, 我們還需要字型規格檔案 `.afm`。如果軟體公司並未提供字型規格檔, 我們可以從 CTAN 網站上搜尋。萬一連網站上都找不到, 我們還可使用 Ghostscript 所提供之工具程式 pf2afm; 或者利用字體設計軟體如 Fontographer 產生。

若電腦內已安裝 Ghostscript 6.0版, 則 URW++ 所捐贈之字型檔案應置於 `c:\aladdin\gs6.0\fonts` 檔案夾內。底下先說明如何創造 Palatino 字體之巨集套件。此一字體族含有 4 個描邊字型檔, 檔名如下:

`p0520031.pfb` (正體字)

`p0520231.pfb` (斜體字)

`p0520041.pfb` (粗體字)

`p0520241.pfb` (粗斜體字)

GSview 6.0版已附有字型規格檔案, 但其他繪圖軟體所附字型不見得含有字型規格檔; 故底下仍簡單說明如何自行產生字型規格檔。

硬碟內先創造 `c:\texmf\fonts\type1\palatino` 檔案夾, 將以上四個字型檔案移入其中。以文字編輯軟體開啓 `c:\aladdin\gs6.0\lib` 檔案夾內之 `pf2afm.bat` 檔案, 將最後一行指令更改如下:

```
c:\aladdin\gs6.0\bin>gswin32c -q -sDEVICE=nullpage -- pf2afm.ps %1
```

爲了方便起見, 將此批次檔與上述四個字型檔案都複製於 c:\xtemp 檔案夾內執行下列指令:

```
c:\xtemp>pf2afm p052003l
```

硬碟內將產生 p052003l.afm。依同法, 產生其他三個字型規格檔案。

取得字型規格檔案後, 下一步驟是須更改檔名。爲了避免混淆, 數以千計的 PostScript 字體必須有一套命名方法。目前通用的重新命名方法是由 Karl Berry 所設計者。在此命名之下, Palatino 字體名字簡化爲 pl 兩字; Helvetica 字體則簡化爲 hv。許多字體設計公司各有其 Palatino 字體, 因此字體名稱上還須標示公司名稱以資區別。德國 URW++ 軟體公司是以字母 u 代表, 因此以上兩種字體將分別改名爲 upl 與 uhv。有關於各式各樣字體之重新命名方法, 請見 \texmf\fontname 檔案夾內之說明檔。

依以上原則, Palatino 4 個字型規格檔案應分別改名如下:

```
p052003l.afm  ==>  uplr8a.afm
p052023l.afm  ==>  uplri8a.afm
p052004l.afm  ==>  uplb8a.afm
p052024l.afm  ==>  uplbi8a.afm
```

新檔名末端之 8a 是用於標示字元排序 (encoding) 的方式。檔名中之 r 爲 regular 簡寫, 代表正體; i 爲 italic 簡寫, 代表斜體, b 爲 bold 簡寫, 代表粗體; b_i 則代表粗斜體。

接下來, 我們須利用 fontinst 巨集套件進一步處理。進入 DOS 模式, 執行下列指令:

```
c:\xtemp>tex fontinst.sty
```

執行以上指令之後, 程式將讀取檔案夾內之 4 個字型規格檔案, 並由之創造 TEX 排版時所需之各種檔案。當顯示器上出現 * 號, 應鍵入:

```
\latinfamily{upl}{}\bye
```

程式執行之後, 硬碟中將出現三組檔案, 延伸檔名分別爲 .pl, .vpl, 與 .fd; 前兩組檔案須進一步轉換。

欲處理這兩組字型規格檔案需要 pltotf.exe 與 vptovf.exe 兩個程式。如果你使用 MiKTEX 我們建議改用 emTEX 所提供者, 原因是執行較方便。(如

果你使用 fpTEX, 可直接執行。) 將 c:\emtex\bin 檔案夾內這兩個程式檔案複製至 c:\xtemp 內, 執行下列兩道指令:

```
c:\xtemp>for %f in (*.pl) do pltotf %f
c:\xtemp>for %f in (*.vpl) do vptovf %f
```

以上指令將創造出十幾個 .tfm 與 .vf 檔案。將所有的 .tfm 檔案複製於 \texmf\fonts\tfm\palatino 檔案夾內。請注意, ...\tfm\palatino 次檔案夾原本並不存在, 必須先創造之。同樣的, 先創造 ...\vf\palatino 次檔案夾, 再將所有的 .vf 檔案複製於其內。

在 \xtemp 檔案夾內另可找到 4 個 .fd 檔案, 其中 OT1upl.fd, T1upl.fd 與 TS1upd.fd 三個字體驅動檔案, 以文字編輯軟體開啓 OT1upl.fd, 其內容略加整理之後有下列之指令:

```
...
\DeclareFontFamily{OT1}{upl}{}
\DeclareFontShape{OT1}{upl}{b}{n}{ <-> uplr7t}{}
\DeclareFontShape{OT1}{upl}{b}{sc}{ <-> uplrc7t}{}
...
```

每一行指令設定某種字體的使用方法。譬如, 第 2 行指令即指定以 uplr7t 字型檔排版 Palatino 粗體字, 並使用 OT1 之字元排序方式。每一種字體有其特定之設計尺寸。若一篇文稿中混用幾種字體, 即使同樣使用 12 點字體排版, 不同字體之間可能會大小不一。因此, 字體大小須略作調整。

經過幾次實驗, 我們發現 upl 須略微縮小才能與中文字體搭配。因此, OT1upl.fd 檔案內容須修改如下:

```
...
\DeclareFontFamily{OT1}{upl}{}
\DeclareFontShape{OT1}{upl}{b}{n}{ <-> [.94] uplb7t}{}
\DeclareFontShape{OT1}{upl}{b}{sc}{ <-> [.94] uplbc7t}{}
...
```

各行內所加上之 [.94] 之作用是將指定字體縮小爲原尺寸的 94%。修改完畢之後, 先創造 \texmf\tex\latex\palatino 檔案夾, 再將 4 個 .fd 檔案移入其中。T1upl.fd 與 TS1upd.fd 請以同樣方法處理。Palatino 字體是屬於 serif 字體, 我們另選取一 sans serif 字體搭配使用。

在 URW++ 所提供的字體中, Helvetica 是一個不錯的選擇。依同樣的命名規則, 此字型檔名應以 uhv 開頭, 因此 4 個字型規格檔應重新命名如下:

```
n019003l.afm  ==>  uhvr8a.afm
n019023l.afm  ==>  uhvri8a.afm
n019004l.afm  ==>  uhvb8a.afm
n019024l.afm  ==>  uhvbi8a.afm
```

倣照處理 Palatino 字型檔的方法, 對於 Helvetica 字型檔同樣處理一遍。如此, 硬碟內已存有使用兩種 PostScript 字體所需之檔案。最後, 以文字編輯軟體寫出 uplhv.sty 檔案, 內容如下:

```
%% This is file `uplhv.sty',
\renewcommand{\rmdefault}{upl}
\renewcommand{\sfdefault}{uhv}
\endinput
%% End of file `uplhv.sty'.
```

第 1 行指令設定 upl 字體作爲正體字, 第 2 行指令設定 uhv 字體作爲 sans serif 字體。此檔案應移入 \texmf\tex\latex\palatino 檔案夾內。

以上所製作的檔案可供 LaTeX 排版之用, 但爲了讓預覽/列印軟體能找到所需的字型檔案, 尙須製作一字型對應檔 (font mapping file)。表 13.1 爲字體對應檔案之內容, 檔案取名爲 uplhv.map。請特別注意, 因爲版面寬度容納不下全部文字, 我們將其中字元串 TeXBase1Encoding ReEncodeFont 暫以字母 x 替代。實際製作檔案時, 字母 x 應替代回原始字串。

字型對應檔案內每一行文字分五個段落, 第一段爲 LaTeX 排版時所認定之字體名稱。以第一行爲例, 檔名爲 uplr8r, 代表 Palatino 正體字。第二段爲 URWPalladioL-Roma, 此字體名取自 .pfb 字型檔內之 /FontName 指令行, 這是 PostScript 字體之正式名稱。第三與第四段落設定字型檔內各字母之排序方法, 最後一段爲硬碟內描邊字型檔之名稱。由表 13.1 可以看出, 每一字體族須作 6 行設定。前 4 行分別爲正體字、斜體字、粗體字, 與粗斜體字; 第 5 行爲數學斜體字, 第 6 行爲數學粗斜體字。

檔案 uplhv.map 製作完成之後, 應創造 \texmf\dvips\uplhv 檔案夾, 將字體對應檔移入其內。最後, 以文字編輯軟體開啓 \texmf\dvips\config 檔案夾內之 config.ps 檔案, 其中可找到下列設定行:

表 13.1: uplhv 巨集套件之字型對應檔案

```
uplr8r  URWPalladioL-Roma "x" <8r.enc <p052003l.pfb
uplri8r URWPalladioL-Ital "x" <8r.enc <p052023l.pfb
uplb8r  URWPalladioL-Bold "x" <8r.enc <p052004l.pfb
uplbi8r URWPalladioL-BoldItal "x" <8r.enc <p052024l.pfb
uplro8r URWPalladioL-Roma ".167 SlantFont x" <8r.enc <p052003l.pfb
uplbo8r URWPalladioL-Bold ".167 SlantFont x" <8r.enc <p052024l.pfb
uhvr8r  NimbusSanL-Regu "x" <8r.enc <n019003l.pfb
uhvri8r NimbusSanL-ReguItal "x" <8r.enc <n019023l.pfb
uhvb8r  NimbusSanL-Bold "x" <8r.enc <n019004l.pfb
uhvbi8r NimbusSanL-BoldItal "x" <8r.enc <n019024l.pfb
uhvro8r NimbusSanL-Regu ".167 SlantFont x" <8r.enc <n019003l.pfb
uhvbo8r NimbusSanL-Bold ".167 SlantFont x" <8r.enc <n019024l.pfb
```

說明: 表中之 x 應以 TeXBase1Encoding ReEncodeFont 字串替代。

```
p +cwtex.map
p +utopia.map
...
```

請加入:

```
p +uplhv.map
```

所有設定即大功告成。設定完成之後, 須更新檔案資料庫。MiKTEX 使用者請執行:

```
c:\xtemp>initexmf -u
```

fpTEX 使用者請執行:

```
c:\xtemp>mktexlsr
```

最後, 以上的處理過程會在 c:\xtemp 內留下許多輔助檔案, 爲節省空間可將之全數刪除。

字體對應檔之運作原理可簡單說明如下: 文稿若使用 uplhv 巨集套件排版, 正文將取用 Palatino 正體字。文稿排版完成轉換爲 PostScript 格式時, DVIPS 程式發現文稿內使用 uplr8r 正體字, 開始在硬碟內尋找描邊字型檔。DVIPS 首先檢查 config.ps 檔案內每一個字體對應檔案, 當檢查到

uplhv.map 時, 發現檔案內含有 uplr8r 字體之對應設定, 即循此自硬碟內讀取 p0520031.pfb 描邊字型檔, 並進行轉換工作。

13.3.2 選用字體

巨集套件創造出來後, 排版時只須在全文設定區引用巨集指令:

```
\usepackage{uplhv}
```

LaTeX 即以 Palatino 作爲正文字體, 以 Helvetica 作爲 sans serif 字體。以上指令會改變整篇文稿之字體。如果只是要改變文稿中某一段落之字體, 可選用下列指令:

```
{\fontfamily{upl}\selectfont text}
```

大括號內之 *text* 將改用 Palatino 字體。以上之 \fontfamily 指令用以選用字體族。除此之外, 我們也可以改變字體系列, 指令爲 \fontseries。同理, \fontshape (字形) 指令用以選擇正體、斜體; \fontsize 指令則用以選擇字體大小與行距。

字形指令 \fontshape 可使用 4 個選項: n, it, sl, 與 sc; 其中, n 爲正常形 (normal), it 爲斜體形, sl 代表數學斜體字, sc 代表 small capital 字體。字體系列指令 \fontseries 同時標示重量 (weight) 與寬度 (width) 兩項特徵。其中, 重量是指筆畫粗細, 譬如, 粗體字爲 b, 標準寬度爲 m。字體寬度由極小之 uc (ultracondensed) 到極大之 ux (ultraexpanded); 標準或正常寬度爲 m。如果選用標準寬度, m 指令可省略, 譬如, 標準寬度之中體字指令爲 \fontseries{m}; 又如, 標準寬度之粗體字指令爲 \fontseries{b}; 相對的, 狹長 (condensed) 中體字指令爲: \fontseries{mc}, 其中, m 代表中體字, c 代表 condensed。

根據以上說明, 以下指令:

```
\fontfamily{bch}\fontseries{b}\selectfont
```

選擇 Charter 之粗黑體, 其中 bch 後兩個字母代表 Charter 字體, 字母 b 則代表出版字型檔之 Bitstream 公司。如果在 \selectfont 指令之前又加入 \fontsize{15}{20pt}, 則字體大小將變爲 15pt, 行距爲 20pt。若再加入 \fontshape{it} 指令, 即指示斜體字。再舉一個例子, 若輸入以下指令:

```
{\fontfamily{bch}\fontseries{b}\fontshape{sc}\selectfont Dvips}
```

排版結果爲: **DVIPS**。指令中選用 Charter 字體族, 粗體系列, 標準寬度, 及 small capital 形狀。

上一節曾介紹 charter 巨集套件, 其功能是設定羅馬字體選用 Charter 字體; 但 sans serif 仍使用 LaTeX 原始字體。如果要把 charter 巨集套件內之 sans serif 字體也改用 Helvetica 字體, 只要將 `charter.sty` 檔案依前面 `uplhv.sty` 作類似修改即可。

13.3.3 使用 True Type 字體

每一套 Windows 作業系統是都附有一些英文 True Type 字體, 例如 Times New Roman, Arial 等。目前已有人嘗試要讓 LaTeX 使用 True Type 字體排版, 但迄今爲止尚不十分成熟。

目前的發展中, pdfLaTeX 的作法是直接取用 True Type 字型檔。另外一種作法則是先將 True Type 字型檔轉換爲 PostScript 字型檔, 再利用 fontinst 產生所需之巨集套件。網路上有幾套免費下載程式, 可將 True Type 字型檔轉換成 PostScript 字型檔, 其中之一名爲 ttf2pfb; MiKTeX 與 fpTeX 系統內都含有此一軟體。有興趣嘗試者, 可至 Free Type 網站取的進一步資訊:

`http://www.freetype.org`

13.4 Dvips 與 psutils 工具程式

欲引入 PostScript 圖形或使用 PostScript 字型, 我們必須將排版結果轉換爲 PostScript 格式。目前使用最廣的轉換驅動程式可能是 Dvips, 此爲 Tom Rokicki 所創作。底下兩小節分別介紹 Dvips 與附屬工具程式 afm2tfm 之功能。

13.4.1 Dvips 程式

Dvips 程式的主要功能是將 TeX 排版結果轉換爲 PostScript 格式。若使用 cwTeX 之設定, 在 WinEdt 視窗內按下功能鍵 [F11], 即執行 Dvips 將排版文稿全文轉換爲 PostScript 格式。不過, 有時候我們僅換某幾頁, 或者須轉換爲特別格式, 則執行程式時須加入選項。

DVIPS 程式之指令格式如下:

```
c:\>dvips -option file.dvi
```

其中, `file.dvi` 爲 TEX 排版結果, 延伸檔名 `.dvi` 可省略; *-option* 爲程式選項, 底下列出 DVIPS 較常用之選項。

- `-p` 列印範圍之首頁。譬如, 選項 `-p 3` 即選擇自第 3 頁開始列印。
- `-l` 列印範圍之末頁, 譬如, `dvips -l 8` 即選擇列印至第 8 頁。綜合以上兩選項, `-p 3 -l 8` 即指示列印的 3-8 頁。
- `-pp` 另一種選擇列印範圍的方法。譬如, `-pp 3,5,7:10` 選擇列印第 3, 5 兩頁, 及 7-10 頁。單獨之頁碼以逗號分開, 連續之頁碼以冒號區隔首末頁。
- `-n` 設定總列印頁數。譬如, `-p 3 -n 20` 表示自第 3 頁開始, 共列印 20 頁, 至第 22 頁爲止。
- `-o` 設定列印檔案名稱。不加設定時, 若文稿原名爲 `file.dvi`, 列印結果爲 `file.ps`。
- `-O` 調整列印在紙面的位置。此選項須設定 `X,Y` 兩項座標數字。例如, 若列印區域要自原先設定位置右移 1 公分, 下降 0.5 公分, 應加入之選項爲 `-O 1cm, 0.5cm`。(我們也可以直接在 GSview 軟體內調整列印位置。)
- `-E` 列印爲 Encapsulated PostScript 格式。加上此選項後, 所產生之檔案內將包含有 `%%BoundingBox` 指令。

除了以上所列之外, DVIPS 還提供許多選項, 請參考其說明檔。

13.4.2 afm2tfm 程式

Tom Rokicki 另外寫了一個工具程式 afm2tfm, 其功能是將 PostScript 字體之 `.afm` 字型規格檔案轉換爲 TEX 排版時所需之 `.tfm` 字型規格檔。前面介紹中文變形字體時曾說明其應用方法。此一程式之指令格式如下:

```
c:\>afm2tfm inputfile -option outputfile
```

inputfile 是指 PostScript 字型規格檔, 例如, ptmr8r.afm 即爲 Times 字型規格檔名。輸入時, 附屬檔名可省略; outputfile 爲轉換結果之檔名。若省略輸出檔名, 程式將自動取名爲 ptmr8r.tfm。

常用之選項有下列兩個:

-e 字體水平放大或縮小。例如 -e 0.91 設定將字體內每一個字之寬度壓縮爲 91%。

-s 字體傾斜。例如, -s 0.167 設定將字體向右傾斜 16.7 度。若數字爲負值, 字體將向左傾斜。

其他選項尙可控制字元排序 (encoding) 方法, 請見說明檔。

13.4.3 psutils 工具程式

排版是以一頁爲基本單位, 但列印時有時候需要把版面順序重新調整。譬如說, 爲方便列印於全開的紙面上, 排版者須把 9 頁合併於一個大版面中; 台灣排版業者稱此爲「落大盤」。Angus Duggar 寫了一套 PostScript 工具程式, 主要功能是重新組合版面, 本節簡單介紹此套程式之功能。

工具程式 psutils 包含好幾個程式, 一般人最常用的程式是 psnup, 其功能是集合兩頁 (或多頁) 列印於一頁版面上。另一個較常用的程式是 psbook, 其功能也是調整版面順序; 新順序方便作書本裝訂。

● 數頁合併於一頁: psnup 程式

psnup 程式的應用是將兩頁版面列印於一頁上, 以節省用紙, 指令如下:

```
c:\xtemp>psnup -2 infile.ps outfile.ps
```

其中, -2 表示每兩頁併爲一頁, infile.ps 爲原檔名, outfile.ps 爲新檔案名稱。若要 4 頁合併於一頁中, 選項 -2 應改爲 -4。

此程式可加入許多選項以控制列印結果, 擇其要者簡介如下:

```
psnup -n -ppaper -Ppaper -sscale -mmargin infile outfile
```

第一選項 n 項代入數字, 表示要將 n 頁併入於一頁中。第 2 選項 -p 供選用輸出版面之紙張大小, 可填入 a4, letter, b5 等等。第 3 選項 -P 之功能與第

2選項相同, 但指的是原文稿之紙張尺寸。若不加入紙張尺寸選項, 程式假設皆爲 A4 紙張。第4選項 -s 設定原版面縮小之比例。例如, -s0.75 指示版面縮小爲原尺寸之 75%。最後一個選項 -m 設定輸出版面四周所留出之空白。例如, -m1cm 設定紙面四周各留下一公分空白。

● 調整版面順序: psbook 程式

若某一篇文稿共計8頁, 列印時將依1-8頁之順序。現假設我們利用上面介紹之 psnup 程式每兩頁合併爲一頁, 則新的第一頁內有1-2之頁碼; 第2頁上有3-4之頁碼等。列印之後, 將此4頁疊在一起, 從中間對折可變成一本小册子。但我們會發現此小册子之頁碼順序不正確。譬如, 小册子封面上出現的是第2頁而不是第1頁。

psbook 程式之功能即是將頁碼順序重新調整, 使小册子之頁碼順序得以正確出現。舉例言之, 若文稿原檔名爲 file.ctx, 排版後得 file.ps。先執行 psbook 調整順序:

```
c:\xtemp>psbook file.ps file1.ps
```

再執行 psnup 將8頁合併爲4頁:

```
c:\xtemp>psnup -2 file1.ps newfile1.ps
```

將 newfile1.ps 列印出來, 4頁疊在一起, 中間對折, 即可得到一正確頁碼順序之小册子。

14 巨集指令

巨集指令 (macros) 就是把用來完成特定工作的一連串指令集合在一起, 並給予一個名稱。文稿內若使用此巨集指令, 排版時 LaTeX 將依序執行其中的指令。定義巨集指令的用途之一是節省輸入長串指令的時間, 避免輸入錯誤。此外, 排版長篇文稿時, 前後的版面格式必須一致; 譬如, 各章節標題須使用同一字體。欲解決此一問題, 最好是利用巨集指令。

事實上, LaTeX 本身就是由 TeX 原始指令所定義出來的一套龐大的巨集指令。本書所介紹的各種巨集套件也是各種巨集指令組合而成。欲深入了解巨集指令的概念, 請參考 Knuth (1990) 或 Kopka and Daly (1995)。本章僅介紹最簡單的巨集指令的概念與用法, 最後一節說明排版本書所使用之巨集指令, 以供參考。

14.1 定義巨集指令

巨集指令的功能之一是用以節約輸入長串指令之時間。舉例言之, 若文稿內經常使用 `\medskip` 指令, 我們可以在全文設定區作下列定義:

```
\newcommand{\ms}{\medskip}
```

LaTeX 使用 `\newcommand` 以定義巨集指令。第一個大括號內爲巨集指令名稱。取名時, 請注意勿與現有之巨集同名。巨集指令名稱以反斜線起頭, 但名字只能使用大小寫英文字母, 不能使用數字。

因爲 LaTeX 本身是由 TeX 指令組合而成, 因此, 我們也可以直接使用 TeX 指令定義巨集指令。若使用 TeX 指令, 上述之定義可以簡化爲:

```
\def\ms{\medskip}
```

TeX 之定義指令爲 `\def`, 巨集指令名稱則直接附加於其後, 前後不須加上大括號。經過以上定義, `\ms` 指令即等於是 `\medskip`。

另外一個例子, 論文裡經常須列舉他人之研究文獻。以下之巨集指令 \laref 設定一個排版參考文獻的格式:

```
\newcommand{\laref}{\par\noindent\hangindent=\parindent}
```

此一巨集指令之意義如下: 定義一開頭以 \par 指令結束前一段落; 接下來以 \noindent 指令設定下一行開頭不內縮; 最後的 \hangindent 指令設定每一項文獻的第 2 行開始內縮 \parindent 距離。

經過以上定義之後, 排版時可將 \laref 指令置於每一項文獻前端, 每一文獻的第一行將由版面左緣開始編排, 第二行以後每一行都會內縮一點距離。如果兩項文獻之間距還要加大一些, 可在以上定義中 \par 指令之後加入 \smallskip。

再舉一個巨集指令之應用例子, 本書經常使用的 cwTEX 標誌符號是由下列巨集指令所定義:

```
\newcommand{\cw}{{\tt cw}\TeX}
```

定義一開始首先選用打字機字體排版 cw 兩個字母, 其後再加上 \TeX 指令。如此, 文稿內輸入 \cw{} 即可排出 cwTEX。

與 \newcommand 指令類似的是 \renewcommand。前者用於定義新的巨集指令, 後者則用於修改原已存在之巨集指令的內容。譬如, LATEX 原定義 \abstractname 指令作爲英文摘要之標題: **Abstract**。但我們可使用下列指令重新定義爲中文標題:

```
\renewcommand{\abstractname}{摘要}
```

巨集指令可由使用者自行加入變數。譬如, 使用 LATEX 之迷你指令環境時, 我們須設定迷你版面的寬度。若經常使用此項指令, 且版面寬度大小不一, 我們可以定義下列兩道巨集指令:

```
\newcommand{\bmp}[1]{\begin{minipage}{#1\textwidth}}
\newcommand{\emp}{\end{minipage}}
```

第二道巨集指令很簡單, 以 \emp 指令替代較長的 \end{minipage}。第一道巨集指令 \bmp 定義迷你版面之開端, 其中之 [1] 設定表示使用者須自行加

入一數字。執行時, 此一數字即代入指令後面的 #1。定義巨集指令之後, 若輸入 \bmp{0.4}, 其作用與 \begin{minipage}{0.4\textwidth} 完全相同。

巨集指令可添加之變數不限於一項。若有兩項變數, 則巨集指令名稱之後變成 [2], 定義內容分別以 #1 與 #2 代表這兩項變數。第 9 章 (頁 167) 曾說明排版迴歸方程式之巨集指令 \tb。原指令是以 \def 定義; 若改用 LaTeX 之 \newcommand 指令定義, 應爲:

```
\newcommand{\tb}[2]{\mathop{#1\vphantom{\sum}}%
  \limits_{\displaystyle #2}}
```

LaTeX 可以定義更複雜的巨集指令。譬如, 巨集指令可以加入選項; 或者在符合特定條件下才執行某項動作。詳細說明請見 Kopka and Daly (1995), 頁 189-208。

14.1.1 設定字級之巨集指令

排版文稿時經常須變更字級與行距。LaTeX 提供相對字級指令, 如 \small, \large 等等。但我們也可以直接選用特定字級。利用第 5 章所介紹的「新式字體選用法」, 若要選用 14.4pt 英文字體並把行距拉大成 20pt, 指令爲:

```
\fontsize{14.4}{20pt}\selectfont
```

如果字體再變回 12pt, 行距縮小爲 18pt, 則必須再下指令

```
\fontsize{12}{18pt}\selectfont
```

以上之指令稍嫌複雜, 爲了簡化輸入動作, 我們可以將指令定義於一巨集指令內, 請見圖 14.1 的例子。

此一巨集指令是以 TeX 之 \def 定義。巨集指令名字不能使用數字, 因此我們將字級以變數方式填入。常用之字級包括 10pt, 10.95pt, 12pt 等, 因此, 巨集指令即以此等字級爲主。以 12pt 字體爲例, 我們希望輸入指令簡化爲 \sz12。若定義巨集時只設一項變數, 輸入指令時數字 12 須置於括號內, 變成 \sz{12}。爲了進一步簡化, 定義巨集時設定兩項變數, 字級之十位數爲第一項變數, 個位數爲第二項變數。如此, 巨集指令可以簡化爲 \sz12; 其中, 數字 12 之 1 爲第一項變數, 2 爲第二項變數。

```
\newcount\fs
\def\sz#1#2{\fs=#1#2
 \ifnum\fs=10\fontsize{10}{12.5pt plus.4pt minus .2pt}\selectfont
 \else\ifnum\fs=11\fontsize{10.95}{16.5pt plus.3pt minus.2pt}\selectfont
 \else\ifnum\fs=12\fontsize{12}{18pt plus.4pt minus .3pt}\selectfont
 \else\ifnum\fs=14\fontsize{14.4}{20pt plus.4pt minus .3pt}\selectfont
 \else\ifnum\fs=17\fontsize{17.28}{22pt plus.5pt minus .4pt}\selectfont
 \else\ifnum\fs=20\fontsize{20.73}{28pt plus.5pt minus .4pt}\selectfont
 \else\ifnum\fs=25\fontsize{24.88}{33pt plus.6pt minus .4pt}\selectfont
 \else\ifnum\fs=09\fontsize{9}{11pt plus.4pt minus .3pt}\selectfont
 \else\ifnum\fs=08\fontsize{8}{9.5pt plus.4pt minus .3pt}\selectfont
 \else\ifnum\fs=07\fontsize{7}{9pt plus.4pt minus .3pt}\selectfont
 \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi}
```

圖 14.1: 設定字級與行距之巨集指令

定義巨集時, 我們同時也改變行距。例如, 10pt 字體之行距爲 12.5pt; 伸縮彈性爲正 0.4pt 與負 0.3pt。11pt 字體之行距爲 16.5pt 加減伸縮彈性; 12pt 字體之行距爲 18pt, 等等。若覺得以上的行距不妥, 可以代入自己喜歡的數字。圖 14.1 亦定義 \sz09, \sz08 巨集指令。定義巨集指令之後, 選用字體級數就變得很簡單。譬如, 選用 14.4pt 之英文字只須鍵入 \sz14 指令即可。選用 17.28pt 字體, 則鍵入 \sz17; 12pt字體, 則鍵入 \sz12; 餘此類推。如果是選用 8pt 之字體, 則須鍵入 \sz08。請注意, 輸入時不要忘了十位數之 0, 否則將出現錯誤。

若每一篇文稿都要輸入圖 14.1 之巨集指令定義, 顯然不方便。我們可以使用更簡單的辦法: 將巨集指令儲存成一檔案。事實上, 圖 14.1 之巨集指令已儲存爲 mymacro.tex, 安裝於 TEX 系統內。因此, 我們只須在全文設定區輸入:

```
\input{mymacro}
```

文稿內即可使用 \sz12 等指令改變字級。

排版時, LATEX 若遇到 \input{...} 指令, 它會先讀取並處理該檔案之內容。因此, \input 指令之效果等於是將 mymacro.tex 之內容全部輸入於指令所在的位置。換句話說, \input 指令的目的也是在簡化輸入: 把原本要放在文稿中內一大堆指令另置一處, 以供隨時取用。巨集指令 mymacro.tex 置於 c:\texmf\tex\latex\package 檔案夾內, 其內容僅含定義字級/行距

之指令; 但我們也可以將其他常用之巨集指令置於其中。

14.1.2 巨集指令與中文

上一小節所定義之巨集內不含中文字。事實上, 一般的巨集指令內也可輸入中文字, 但使用上須注意兩件事:

- 巨集檔案須以 `.ctx` 爲延伸檔名;
- 巨集檔案須置於 cwtex 程式可找到之檔案夾內。

舉例言之, 若巨集檔案之主檔名爲 `mymacro`, cwtex 轉換中文時, 若發現文稿內有 `\include{mymacro}` 或 `\input{mymacro}` 指令, 它會在目前的檔案夾內尋找 `mymacro.ctx` 檔案。若檔案存在, 即將之轉換爲 `mymacro.tex`。

接下來 latex 進行排版工作時, 讀取的檔案是並非原始的 `mymacro.ctx`, 而是中文碼已經轉換過的 `mymacro.tex`。因此, 排版工作不是出現任何問題。但如果 latex 直接讀取 `mymacro.ctx`, 因爲無法辨認中文碼之意義, 執行時即出現問題。以上的步驟中, 請特別注意: cwtex 程式僅在目前檔案夾 (current directory) 內尋找以 `.ctx` 爲延伸檔名之檔案。巨集檔案若取名爲 `mymacro.tex`, cwtex 找不到檔案, 也不作任何轉換動作。如果任取其他延伸檔名, cwtex 也不會去尋找檔案。

綜合以上所述, 自行創造之巨集檔案若僅含英文, 延伸檔名可取爲 `.tex`, 並且可置放於 LaTeX 可以搜尋到的任何檔案夾內。反之, 若巨集檔案內含有中文, 請記得一定要以 `.ctx` 爲延伸檔名, 並且要與文稿檔案置於同一檔案夾內。

14.1.3 依條件處理之巨集指令

排版書籍時, 單雙頁面可能須作不同的處理。舉例來說, 某些圖表的寬度超過正文一行的長度, 一個解決的辦法如下: 若圖表是在雙數頁 (左頁), 可將圖表稍左移; 若圖表是位於單數頁 (右頁), 則左邊切齊文字版面左沿, 右邊則稍凸出一些。欲以巨集指令處理此一問題, 我們需有一巨集指令能判斷本頁是單數頁還是雙數頁, 並能依此作進一步的處理。David Carlisle 所寫的 ifthen 巨集套件可用來處理此一問題。事實上, 此一巨集套件除了判明頁碼之外, 也可以應用在其他方面。

巨集套件 ifthen 提供指令 \ifthenelse, 底下是一個簡單例子:

```
\usepackage{ifthen}
\newcommand{\chk}[2]{\label{#1}
\ifthenelse{\isodd{\pageref{#1}}}{\noindent\ignorespaces}%
   {\noindent\hspace*{-#2\textwidth}\ignorespaces}}
```

在全文設定區引入巨集套件之後, 接下來定義一巨集指令 \chk。此巨集指令使用了 \ifthenelse 指令, 並且用了兩個參數。第一個參數是使用者填入之標誌 (label), 第 2 個參數設定左移之距離。利用此一指令, 文稿內若輸入下列一行指令:

```
\chk{example}{0.1}{This is a test}
```

排版時, LaTeX 首先在下指令處加入標誌 example, 接下來利用 \ifthenelse 指令判斷此標誌是位於單數頁或雙數頁。若在單數頁, 則取消行首內縮 (indent) 之動作; 反之, 若位於雙數頁, 則以 \hspace*{-0.1\textwidth} 指令左移一點距離。不管是單數頁或雙數頁, 最後都會排版 “This is a test”。

14.2 定義指令環境

除了巨集指令之外, 我們也可以定義或修改指令環境。底下僅舉兩個例子簡單說明。第一個例子定義指令環境 nm, 其功能與 LaTeX 之 enumerate 指令環境類似, 唯一不同的地方是加入 \itemsep=-2pt 設定, 讓條列項之間距縮小 2pt。

```
\newenvironment{nm}{\begin{enumerate}\itemsep=-2pt}%
  {\end{enumerate}}
```

由此例子可知, 以 \newenvironment 定義新指令環境時, 第一對大括號內置放指令環境名稱; 接下來的兩對大括號, 前一對大括號內含指令環境之定義指令; 後一對則含結束之指令。

指令環境可包含參數, 底下是一個簡單的例子。

```
\newenvironment{mymp}[2][10mm]{\par\noindent\hspace*{#1}%
  \begin{minipage}{#2\textwidth}}{\end{minipage}}
```

首先, 指令環境名稱爲 mymp, 其後之 [2] 表示此指令環境帶有第二項參數。但是, 第一項參數是選擇性輸入 (optional)。如果使用此指令環境時僅輸入一項參數值, 則第一項參數將自動代入定義中之 10mm。舉例言之, 經過以上之定義, 若文稿中輸入下列指令:

```
\begin{mymp}{0.8} ... \end{mymp}
```

則鍵入文字內容將排版於迷你版面內, 其寬度爲正文行寬的 80%。排版時, LaTeX 先結束上一段落, 從版面左沿右移 10mm, 再開始排版迷你版面。反之, 若鍵入之指令爲:

```
\begin{mymp}[4mm]{0.8} ... \end{mymp}
```

則迷你版面離正文版面左沿僅 4mm, 而非內定值之 10mm。

14.3 更改特定標題為中文

TeX 原本是以排版英文稿件爲主, 但預留許多空間可以排版其他語文。LaTeX 的設計也留有許多彈性, 以方便排版其他語文。譬如, 以 \caption 指令排版表格標題時, LaTeX 會自動加上 “Table” 一字並編上號碼。事實上, “Table” 是以 \tablename 定義。若以列指令重新定義,

```
\renewcommand{\tablename}{\m12 表}
```

標題字即可改爲中文。

爲方便參考, 表 14.1 列出所有之特定標題, 及對應之英文標題。表中指令, \enclname, \ccname, \headtoname, \headpagename 等是用於 letter 文件類別; \seename 與 \seealsoname 則用於索引。

14.4 計數器

LaTeX 定義許多的計數器 (counter), 用以記錄章節、註解、方程式之編號。一般的情況下, 計數器內的數字會自動加減; 但必要時也可以重新設定。譬如, 排版書籍時, 若本頁爲本章結束, 下一頁要留爲空白; 則排版再一頁之前, 頁碼計數器須先加 1。此外, 我們也可以利用指令變更排版計數器數字之字體。

表 14.1: 特定標題之指令與內定值

指令	英文標題
\abstractname	Abstract
\appendixname	Appendix
\bibname	Bibliography
\ccname	cc
\chaptername	Chapter
\contentsname	Contents
\enclname	encl
\figurename	Figure
\headpagename	Page
\headtoname	To (letter)
\indexname	Index
\listfigurename	List of Figure
\listtablename	List of Table
\partname	Part
\prefacename	Preface
\seename	see
\alsoseename	see also
\tablename	Table

LaTeX 所定義的計數器計有下列:

```
part          paragraph       figure        enumi
chapter       subparagraph    table         enumii
section       page            footnote      enumiii
subsection    equation        mpfootnote    enumvi
subsubsection
```

以上計數器的意義大多很清楚, 唯一須解釋的是最右欄的 4 項計數器。其中, enumi 計數器記錄條列指令環境第一層項目數; enumii 記錄第二層, 餘此類推。另外, mpfootnote 是用以記錄迷你版面指令環境內之註解數目。

欲重新設定某計數器之值, 須使用 \setcounter 指令, 譬如:

```
\setcounter{footnote}{20}
```

可將註解計數器定爲 20。當 LaTeX 碰到下一個 \footnote 指令時, 計數器將增加 1, 因此註解編號變成 21。如果書籍是分章編排, 則利用同樣方法可以透過 chapter 計數器設定每章編號。

14.5 排版本書之巨集指令

本節列出排版本書所使用之巨集指令, 並略作說明。事實上, 本節所介紹之巨集指令都已在前面各章介紹過; 彙總於此, 作爲參考。首先, 我們說明 \include 與 \includeonly 指令之用途。

本書正文計有 19 章, 一開始時是各章分開排版。到了接近完成階段, 才創造一主檔案, 取名爲 cxbook.ctx。各章之檔名爲 cx0.ctx, cx1.ctx 等, 是以 \include 與 \includeonly 指令引入主檔案內; 如圖 14.2 所示。請特別注意, 各子檔案須以 .ctx 爲延伸檔名; 但是, 延伸檔名卻不可鍵入, 因爲 cwtex 只搜尋以 .ctx 爲延伸檔名之檔案。

本例中, 全文設定區內以 \includeonly 指令引入各章, 排版時, 各章皆納入處理。現假設在文稿仍在修正階段, 而我們僅引入 cx0.ctx 與 cx5.ctx 兩章。並假設在前一輪排版整本書時, 已產生了頁碼、圖表目次、數學式編碼等, 則重新排版 cx0.ctx 與 cx5.ctx 時, LaTeX 將擷取原先產生之頁碼與圖表目次資訊。因此, 如果修改後之 cx5.ctx 增加了頁碼, 則全書之頁碼將不正確。欲得正確結果, 須把輔助檔案全部刪除, 重新排版全書。

類似的情況也會出現在中文稿件中。譬如, 前一輪排版時產生了全部書稿目錄, 內含許多中文字體指令。若接下來之排版動作僅 \include 兩個檔案: cx0.ctx 與 cx5.ctx, 則 cwtex 所產生之中文字體指令可能不包含原書稿所用之全部字體指令, 接下來執行 latex 時會出現某些中文字體沒有定義的訊息。遇有此種情形出現, 須在主檔案內 \include 所有的子檔案即可解決問題。

編排整本書時, 必然會使用較多的字體。在 TeX 系統裡, 每一字型檔案最多含256個字; 而且每一文稿只能使用256種字型檔。中文因爲字數衆多, 編排長篇文稿或書籍時, 很容易超過此限制。遇有此種情況, MiKTeX 即無法處理; 此時建議改用 emTeX 之 htex386 排版。此一程式允許一篇文稿中可使用 750 餘種字型檔案。改用的方法很簡單, cwTeX 設定 Ctrl-[F10] 功能鍵執行 emTeX 之 htex386。或者, 我們也可使用 omega 程式集中之

lambda 程式擴充 latex 對於字型檔數目之限制, 允許使用較多的字型檔。轉換爲 PostScript 檔案時, 則可使用 odvips 程式。

回到排版本書之例子, 圖 14.2。主檔案一開始首先以 \input 指令引入巨集指令 cxmacro.tex, 其內爲各項設定與巨集指令, 下文將進一步介紹其內容。主檔案內使用 \frontmatter 指令以設定排版內頁封面、版權頁、目錄等之格式。開始排版正文第 1 章之前, 我們加入 \mainmatter 指令以變更格式; 參考文獻與索引則排版於 \backmatter 之後。其中, 參考文獻之內容鍵入於 cx20.ctx, 索引則是以 \printindex 指令排版。因爲正文各章之標題設計與序、目錄、索引等不同, 因此序、索引章之前還加入一些設定指令。請見 7.6.6 節之說明。

前面 14.1.2 節 (頁279) 曾說明巨集指令內含有中文時之處理原則。本例中, cxmacro.tex 內未含中文; 所有含中文之巨集指令皆集中於主檔案前端。這些巨集指令的主要目的是設定中文標題; 譬如, 本例子前端的兩行指令重新定義圖表之中文標題。

因爲所有的巨集指令已彙總於 cxmacro.tex 檔案或者主檔案前端, 因此每一章檔案之開頭變得很簡單。例如, 本章爲第 14 章, 檔名爲 cx14.ctx, 檔案開頭如下:

```
% file: cx14.ctx
\chkodd

\chapter[巨集指令]{巨集指令}
\noindent
巨集指令 (macros) 就是把用來完成 ...
```

指令第 1 行以註銷指令 % 記錄檔名, 第 2 行指令 \chkodd 之功能是檢查單雙頁, 並作必要之跳頁。指令內容請見下文說明。全書章節標題之中文字體是以 \ctxfdef 字體指令設定, 置於主檔案內。因此, 每一章之 \chapter 指令內不須加上字體設定。

接下來, 我們簡單說明 cxmacro.tex 巨集指令之內容。檔案第一行爲文件類別指令, 接下來以 \usepackage 指令引入所使用之巨集套件。爲了節省篇幅, 僅列出部分巨集指令。

```
% master file of cxTeX manual
\input{cxmacro}
\renewcommand{\figurename}{\m11 圖}
\renewcommand{\tablename}{\m11 表}
...
\ctxfdef{\chapter}[\f11]{\f20}
\ctxfdef{\section}[\f11]{\f13}
\ctxfdef{\subsection}[\f11]{\f12}
\ctxfdef{\subsubsection}[\f11]{\k11}
\ctxfdef{\footnote}{\m10}
\ctxfdef{\caption}{\m11}

\includeonly{cx0,cx1,cx2,cx3,cx4,cx5,cx6,cx7,cx8,cx9,%
  cx10,cx11,cx12,cx13,cx14,cx15,cx16,cx17,cx18,cx19}

\begin{document}
\frontmatter
\include{cx0}

\mainmatter
\include{cx1}
\include{cx2}
...
\include{cx19}

\backmatter
\include{cx20}

\printindex
\end{document}
```

圖 14.2: 排版本書之主檔案

```
\documentclass[11pt,twoside,openany]{book}
\usepackage{latexsym,tabularx,equation,multicol}
\usepackage[nops,rm,small]{titlesec}
```

文稿內經常使用某些標誌名詞, 如 cwTEX, PostScript, GSview 等。爲了簡化指令輸入, 並避免錯誤, 我們以巨集指令定義如下:

```
\newcommand{\cw}{\texttt{cw}\kern-.6pt\TeX}
\newcommand{\ps}{PostScript}
\newcommand{\gv}{\textsf{GSview}}
\newcommand{\chkodd}{\clearpage\ifodd\count0 \else%
  \thispagestyle{empty} \mbox{}\clearpage \fi}
```

第 4-5 行定義 \chkodd, 指令內容說明請參考 7.3.2 節。請同時參考本章 14.1.3 節 (頁 279) 所介紹之 \chk 指令。

章標題是以 titlesec 巨集套件設計, 指令如下:

```
\newcommand{\chfont}{\fontfamily{ppc}\fontseries{m}%
   \fontsize{30}{25pt}\selectfont}
\titleformat{\chapter}[hang]
  {\fboxsep=0pt
  \vspace*{-2.9cm}\hspace*{-1.7\parindent}
  \colorbox{slight}{\mbox{\rule{\textwidth}{0pt}%
   \rule{0pt}{1mm}}}\\[1cm]
  \chfont}{\textcolor{heavy}{\rule{3mm}{0pt}\thechapter}}
  {4mm}{}[]
```

首先, 我們定義 \chfont, 選用 Adobe Poetica 字體。設計章標題之指令細節, 請見 7.6 節。

本書頁眉/頁足詞之設計使用 fancyhdr 巨集套件, 設定指令請參見 7.8 節之說明。最後, 本書正文是以 11 點字體排版, 標準行距爲 16.5pt。

15 有用的工具

TEX 主要是用來排版書籍與論文, 但是它也可以用來幫助處理日常文書事務。譬如, 它可以用來排版信函與大宗郵件; 也可以用來排版投影片、習題解答等等。本章介紹一些常用的排版工具, 除了信函、投影片之外, 我們也將說明製作書籍索引的方法。

15.1 信函

LATEX 有一套現成的巨集套件可排版信函。排版短文時, 我們引用 article 文件類別; 若要排版信函, 我們可以直接引用信函文件類別 letter; 也可以使用自行設計具個人風格之巨集指令。若是利用 letter 文件類別, 排版信函之指令如下:

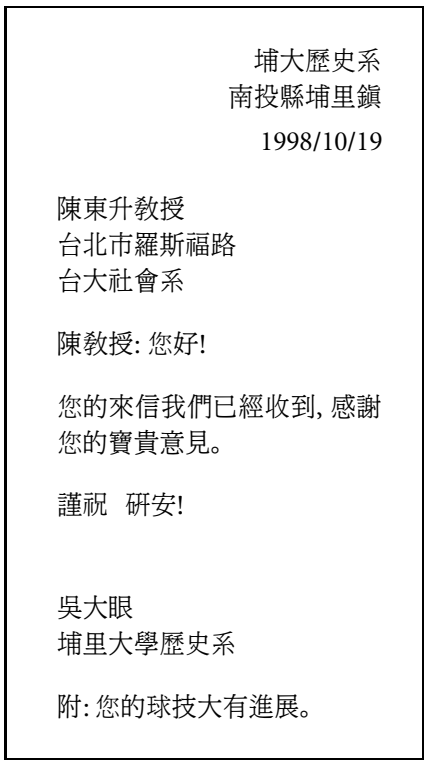
```
\documentclass[12pt]{letter}
\begin{document}
\begin{letter}{...}
...
\end{letter}
\end{document}
```

上例中, [12pt] 選項是指定使用 12pt 之英文字體。若不加此選項, 則以內定之 10pt 字體編排。

LATEX 的信函有特定的日期、地址、信頭、與信尾結語格式, 但這些都可以調整。例如, 信函內自動加上當天的日期, 其形式如 “October 19,1999”。因爲月份是以英文表示, 用於排版中文信函並不適合。底下將說明修改的方法。

15.1.1 排版信函指令

爲了說明方便起見, 我們將信函分成前端、正文、結尾三部分。前端包括



埔大歷史系
南投縣埔里鎮
1998/10/19

陳東升教授
台北市羅斯福路
台大社會系

陳教授: 您好!

您的來信我們已經收到, 感謝
您的寶貴意見。

謹祝　研安!

吳大眼
埔里大學歷史系

附: 您的球技大有進展。

```
\documentclass[11pt]{letter}
\m11
\address{埔大歷史系\\
    南投縣埔里鎮}
\signature{吳大眼\\
    埔里大學歷史系}
\renewcommand{\today}%
   {\number\year/\number%
   \month/\number\day}
\begin{document}
\begin{letter}{陳東升教授\\
台北市羅斯福路\\
台大社會系}

\opening{陳教授: 您好!}
您的來信我們已經收到,
感謝您的寶貴意見。

\closing{謹祝　研安!}
\ps 附: 您的球技大有進展。
\end{letter}
\end{document}
```

圖 15.1: 排版信函

發信人地址、受信人姓名、地址與發信日期, 結尾包括發信人姓名、職稱、附言等等。圖 15.1 之例子說明排版指令與結果, 檔名爲 `letter1.ctx`, 置於 `c:\texmf\cwtex\examples` 檔案夾內, 請試自行排版。

信函正文是以 `\begin{letter}{...}` 起頭, 大括號內輸入受信人姓名、地址等。如果受信人姓名及地址要留爲空白, 請鍵入 `\begin{letter}{\ }`, 否則排版時會出現錯誤。接著, 以 `\opening{...}` 指令排版受信人姓名及致敬詞句; 其下即輸入信函正文。最後以 `\closing{...}` 指令加入結尾祝福言辭。

同一檔案中可以排版數封信函。每封信函都以 `\begin{letter}` 開頭, 以 `\end{letter}` 結尾。若在全文設定區以 `\address` 指令宣告發信人之地址, 此地址會出現在每一封信函前端。本例中, 全文設定區內有三道指令, `\address` 指令是用以排版發信人地址。地址若有兩行以上, 則以 `\\` 換行指

令隔開。其次, \signature 指令用以排版信函結尾處之發信人姓名及職稱。若有兩行以上的內容, 也須以換行指令 \\ 隔開。排版之後, 發信人地址之下會自動排出當天日期。但日期是以英文格式出現, 並不適用於中文信函。要改變日期格式, 最簡單的方法是在全文設定區重新定義。本例重新定義 \today 指令, 排版結果日期格式將變成 “1998/10/19”。若把指令中的 “/” 右斜線符號改成 “.”, 排版結果將變成 “1998.10.19”。

信函正文可以使用一般 LaTeX 指令編排。信函結尾處除了發信人姓名、職稱之外, 可以用 \cc{...} 指令列出其他受信人的姓名。另外, \ps 指令則用以排版附言。如果隨函附有其他文件或物品, 我們可以用 \encl{...} 加以說明。若使用 \encl{...} 指令, 排版時 LaTeX 自動加上 “encl:” 符號; 使用 \cc{...} 指令, 則自動加入 “cc:” 符號。但是, \ps 指令並不加上任何符號, 因此用來排版中文附言很方便。以上之英文標題都可以重新定義, 請見 14.3 節之說明。

15.1.2 設計個人信頭標誌

如果常有信件來往, 我們可以設計一信頭標誌置於信函第一頁上端。排版信函時, 我們使用的是 letter 文件類別。此一巨集套件之設計原就允許加入個人信頭標誌。一旦設計好信頭標誌, 要引入信函內並不困難。

若 TeX 系統是安裝於硬碟 c:, 則在 c:\texmf\tex\latex\base 檔案夾下可找到 letter.cls 檔案。欲在信函上加上個人信頭標誌, 我們應另寫一個巨集套件, 假設取名爲 myletter.cls。此新巨集套件之前端應該先引用 letter 文件類別, 其後再定義個人信頭標誌。如此, 排版時仍可使用原有之信函指令。圖 15.2 是 myletter.cls 檔案的一個簡單例子。前 5 行指令定義文件類別檔名爲 myletter.cls, 並引用 letter 文件類別。爲了易於區分起見, 第 6 行留爲空行。第 7-13 行爲個人信頭標誌之定義。爲了方便設計及修改, 第 7 行先以 \newsavebox 設定使用文字方格, 取名爲 \ltrhead; 第 8-13 行以 \sbox 指令將信頭標誌指令置於文字方格中,

若只是要設計簡單的英文信頭標誌, 指令很簡單。本例中, 信頭左方是粗體字 **ABC Company**, 右方是地址與電話, 底下再畫出一條直線。我們使用兩個 \parbox 指令, 第一個容納整個信頭標誌, 第二個用於排版地址與電話。第一個迷你版面之寬度爲 .9\textwidth, 此爲信頭標誌之寬度。公司名稱選用較大之粗黑字體排版; 地址與電話則占用版面寬度的 30%。因爲


```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{myletter}
\DeclareOption*{\passOptionsToClass{\CurrentOption}{letter}}
\ProcessOptions
\LoadClass[12pt]{letter}

\newsavebox{\ltrhead}
\sbox{\ltrhead}{\parbox{.9\textwidth}
{{\Large\bf ABC Company}\hfill
  \parbox[b]{.3\textwidth}{\small
   123 xyz Street, Taipei\\[-2pt]
   (02)2987-6543}\\
\rule{.9\textwidth}{.6pt}}}
\renewcommand{\ps@firstpage}
 {\setlength{\headheight}{2cm}\setlength{\headsep}{.8cm}%
 \renewcommand{\@oddhead}{\usebox{\ltrhead}}}
 \renewcommand{\@evenhead}{}
\pagestyle{headings}
```

圖 15.2: 個人信函標誌

\parbox 指令加入 [b] 選項, 因此地址與電話之迷你版面下沿將對準公司名稱之基線。\hfill 指令將公司名稱與地址迷你版面推向版面的兩邊。

信頭標誌通常是出現在信函的第一頁, 我們須在指令中宣告標誌本身的高度。指令第 14-16 行的目的是重新定義信函首頁之信頭標誌, 我們利用 \headheight 指令定義信頭標誌之高度爲 2cm, 以 \headsep 指令定義標誌與信函內文之間距的 0.8cm。信函第 2 頁 (偶數頁) 開始不再列印信頭標誌, 故將 \@evenhead 定爲空白。若信頭之外, 信函下端也有特別之標誌, 我們也可以依類似方法設計。請見考 Kopka and Daly (1995) 附錄 A 之說明。

檔案 myletter.cls 製作完成之後, 即爲一新的文件類別。將此檔移入 \localtexmf\tex\latex 或其下之檔案夾內; 並更新檔案資料庫, 請排版程式可以找到此檔案。安裝 MiKTEX 者, 請執行下列指令:

```
c:\>initexmf -u
```

安裝 fpTEX 者, 請執行指令:

```
c:\>mktexlsr
```

排版信函時, 第一行指令應爲:

```
\documentclass{myletter}
```

信函內容之輸入方法則仿照圖 15.1 之例子。請特別注意, 個人信頭標誌內通常已含有發信人地址, 因此信函內容不能再使用 \address 指令。若加入此指令, 排版後信函首頁將不會出現個人信頭標誌。

信函排版之後, 首頁上方即出現自行設計之信頭標誌。若信函長度超過一頁, 第 2 頁開始, 個人標誌不再出現, 每頁上方僅列出頁碼、收信人姓名、日期等。c:\texmf\cwtex\examples 檔案夾內存有 letter2.ctx 例子, 有興趣自行設計信函標誌者, 請試排版。欲了解圖 15.2 各指令之意義, 請參考 Kopka and Daly (1995) 附錄 A 之說明。

15.1.3 設計中文信頭標誌

如果我們的目的只是製作一個簡單的英文信頭標誌, 則仿照圖 15.2 之指令應已足夠。若信頭含有中文字, 則設計方法相同, 但處理上較爲複雜。假設要我們設計一個如圖 15.3 所示之信頭標誌。因爲版面較複雜, 我們先以一 LaTeX 檔案編排此標誌。圖 15.3 下方爲指令內容。此檔案取名 ntueltr.ctx 置於 cwTEX 例子檔案夾內, 有興趣者請自行取出排版。

我們將信頭標誌設計爲一文字方格。實際上, 此例使用兩個文字方格指令, 第一個方格爲 \headlogo 用以存放標誌左上方的三道斜線, 第二個文字方格爲 \ltrhead, 存放整個信頭標誌。整個信頭標誌首先引入斜線標誌 \headlogo, 其右爲「台大經濟系」, 底下爲地址。右方之姓名、職稱、電話號碼等則利用迷你版面指令編排。標誌底下加上一條橫線, 長度略大一些。

信頭標誌設計完成之後, 我們必須把它引入 ntueltr.cls 中。如前例所示, 若標誌中只有英文字, 則文字方格可以直接引入。但此例中含有中文字, 因此我們須進一步處理; 主要原因是**中文字不能置於巨集指令內**。圖 15.3 所示之檔案名爲 ntueltr.ctx, 請將之複製於 c:\xtemp 檔案夾內。首先, 執行 cwtex 將中文字轉換成 TEX 格式:


```
\documentclass{article}
\begin{document}
\newsavebox{\headlogo}
\newsavebox{\ltrhead}
\sbox{\headlogo}{\parbox[b]{8mm}{
 \fontfamily{cmss}\fontshape{sl}%
  \fontsize{17.28}{20}\selectfont III}}
\sbox{\ltrhead}{\parbox{.9\textwidth}{
  \parbox[b]{.6\textwidth}{\small
  \usebox{\headlogo}
  {\bb17 臺大經濟系}\\[6pt]
  {\m10 台北市徐州路21號}}
 \parbox[b]{.25\textwidth}{\m10
 張長谷\\ 台大經濟系教授\\
 Tel: (02)2351-9641}
 \hspace*{-.03\textwidth}\rule{.95\textwidth}{.6pt}}}
\usebox{\ltrhead}
\end{document}
```

圖 15.3: 中文個人信頭標誌

```
c:\xtemp>cwtex -i ntueltr
```

執行時, 請加入 -i 選項, 讓中文字體之定義指令直接置於 ntueltr.tex 檔案前端。此一檔案須進一步整理才能引用。

以文字編輯軟體叫出 ntueltr.tex, 檔案前半部分爲中文字體定義指令, 後半部分爲中文信頭設計指令。檔案前半部分之指令又分爲兩部分, 一爲中文字距指令: \z, \Z, \zZ; 其次是中文字體指令。以明體字爲例, 中文字型檔案之檔名爲 \MaQ, \MbQ, ... 等等。其中, MaQ 爲 10 點明體字之第 0 個字型檔。若爲粗黑體字, 檔名將爲 \cBaQ。每一個中文字在其所屬字型檔案中所占位置, 其順序是以 \cH 指令指定。譬如, 若某中文字位於第 96 個位置,

指令爲 \cH96。修改的第一步驟是將 \documentclass 等 4 道指令前端加上註銷指令或者直接刪除, 參見圖 15.4。

如果將 ntueltr.tex 直接移入 ntueltr.cls, 排版信函時將會出現錯誤訊息。理由如下: 信函正文中也使用中文字體指令。因此, 信函正文之中文字經轉換爲 TEX 格式後, 其所使用之字體指令與 ntueltr.tex 中所定義之指令相同; 排版信函時會出現錯誤訊息。解決此一困難的簡單辦法是稍爲更動 ntueltr.tex 中巨集指令之名稱。利用文字編輯軟體將檔案中全部的 \z, \Z, \zZ 等, 改變成 \xz, \xZ, \xzZ。將 \cH 更改爲 \xcH。同樣的, 將字體指令 \MaQ, 改成 \xMaQ; \McQ, 更改成 \xMcQ 等。

經過以上的修改, ntueltr.tex 之內容將如圖 15.4 所示。爲了方便閱讀, 較長的指令已利用 % 指令拆爲兩行。最後, 再把此檔案之內容仿照圖 15.3 檔案之內容, 製作 ntueltr.cls 文件類別檔案。請注意, 檔案第 2 行內之指定檔名須由 myletter 改爲 ntueltr。設計中文信頭標誌至此即告完成。以上過程中三個檔案: ntueltr.ctx, ntueltr.tex 與 ntueltr.cls 都可以在 cwTEX 例子檔案夾內找到, 請自行參考。檔案夾內另有 letter3.ctx 測試檔, 可排出 ntueltr.cls 之標誌。如果你自行設計信頭標誌, 檔案完成之後, 建議移入

```
c:\localtexmf\tex\latex\mytex
```

檔案夾內, 若檔案夾不存在, 請先建立。之後, 請更新檔案資料系統。MiKTEX 使用者, 請執行:

```
c:\xtemp>initexmf -u
```

fpTEX 使用者, 請執行:

```
c:\xtemp>mktexlsr
```

若信函長度超過一頁, 第二頁開始版面上方將會排出受信人姓名、頁碼、及發信日期。若希望頁眉空白, 信函底端也空白, 只要將以下三行指令加入 ntueltr.cls 檔案倒數第一行之前即可。

```
\renewcommand{\ps@headings}
  {\renewcommand{\@oddhead}{}}
\renewcommand{\@oddfoot}{}
```

```
\def\xz{\hskip 0.0pt plus0.2pt minus0.1pt}
\def\xZ{\hskip 1.2pt plus0.4pt minus0.2pt}
\def\xzZ{\hskip 3.6pt plus1.2pt minus0.8pt}
\def\xcH{\char}
\font\xMaQ=m0 scaled 1000
\font\xcBcXsy=bb2 scaled 1728
\font\xcBaXsy=bb0 scaled 1728
\font\xcBbXsy=bb1 scaled 1728
\font\xcBhXsy=bb7 scaled 1728
\font\xMbQ=m1 scaled 1000
\font\xMfQ=m5 scaled 1000
\font\xMcQ=m2 scaled 1000
\font\xMiQ=m8 scaled 1000
\font\xMhQ=m7 scaled 1000
% \documentclass{article}
% \begin{document}
\newsavebox{\headlogo}
\newsavebox{\ltrhead}
\sbox{\headlogo}{\parbox[b]{8mm}{
 \fontfamily{cmss}\fontshape{sl}%
  \fontsize{17.28}{20}\selectfont III}}
\sbox{\ltrhead}{\parbox{.9\textwidth}{
 \parbox[b]{.6\textwidth}{\small
  \usebox{\headlogo}
  {{\xcBcXsy\xcH67}\xz{\xcBaXsy\xcH215}\xz{\xcBcXsy\xcH37}%
    \xz{\xcBbXsy\xcH200}\xz{\xcBhXsy\xcH205}}\\[6pt]
  {{\xMaQ\xcH171}\xz{\xMaQ\xcH148}\xz{\xMbQ\xcH13}\xz{\xMfQ\xcH15}%
    \xz{\xMbQ\xcH8}\xz{\xMcQ\xcH152}\xZ21\xZ{\xMcQ\xcH85}}}
 \parbox[b]{.25\textwidth}{%\m10
  {\xMbQ\xcH34}\xz{\xMcQ\xcH197}\xz{\xMiQ\xcH235}\\ {\xMaQ\xcH171}%
   \xz{\xMaQ\xcH215}\xz{\xMcQ\xcH37}\xz{\xMbQ\xcH200}%
   \xz{\xMhQ\xcH205}\xz{\xMbQ\xcH96}\xz{\xMfQ\xcH164}\\
 Tel: (02)2351-9641}
 \hspace*{-.03\textwidth}\rule{.95\textwidth}{.6pt}}}
% \usebox{\ltrhead}
% \end{document}
```

圖 15.4: 設計個人信頭標誌

15.1.4 大宗信函

本書第一版曾介紹 merge 巨集套件以處理大宗信函。此巨集套件的功能較簡單, 好處是使用容易。若欲編排較複雜的大宗信函, 可使用 Mike Piff 所寫的 textmerg。

排版大宗信函須須準備兩份檔案, 一爲信函內容, 一爲地址檔案。信函正文之格式類似一般的 `letter` 文件類別, 但其中不輸入收信人地址與收信人尊稱。收信人地址須另輸入成一單獨檔案, 其中存放所有受信人之姓名、地址、稱呼或其他相關資訊。

圖 15.5 例子中, 左邊爲信函內容, 右邊爲地址檔內容。此例中, 每一封信將自地址檔案中取用 4 項資訊: 收信人頭銜、地址、姓氏; 最後一項資訊內容爲空白。我們特別定義一空白資訊, 目的是讓地址檔案內所排列之資訊易於區分。此例中, 地址檔取名爲 `address.ctx`, 其中有三位收信人。假設信函檔名爲 `bulkmail.ctx`, 我們在其內以 `\Fields{...}` 指令定義地址檔案內 4 項資訊之指令名稱, 分別爲: `\Title` (收信人頭銜), `\Add` (地址), `\Surname` (姓氏), 與 `\en` (空白)。最後的 `\en` 代表空白資訊。以上之指令名稱可任取。本例中, 地址名稱並不長, 因此全部輸入爲一行。若地址較長, 我們可以將之拆爲兩部分, 譬如, 門牌與街名以 `\street` 表示, 城市以 `\city` 表示。

信函檔案內首先引入 textmerg 巨集套件。其次, 爲方便處理地址檔案, 我們加入 `\include{address}` 與 `\includeonly{}` 兩道指令。 本例中, 信函與地址皆爲中文。LaTeX 編排信函時, 須自地址檔中讀取收信人相關資訊。因此, 開始編排之前, 我們須先使用 cwtex 轉換信函正文與地址檔案內之中文字。若 `bulkmail.ctx` 與 `address.ctx` 皆置於同一檔案夾內, 而且前一檔案內有 `\include{address}` 指令, 則 cwtex 在轉換 `bulkmail` 檔案時, 會自動找出 `address.ctx`, 同時轉換檔案內之中文字。**請注意**, 地址檔案之延伸檔名必須是 `.ctx`。

以 `\include` 指令轉換地址檔時, 請在全文設定區加入 `\includeonly{}` 指令, 此可避免 LaTeX 把地址檔內容也編排於信函內。本例中, `address.ctx` 地址檔最前面有 3 行說明文字及一行中文字體指令 `\m12`。經轉換中文之後, 前面 4 行都是以 % 指令起頭。轉換爲 `.tex` 格式之後, 以 latex 排版 `bulkmail` 之前, **請將這4行刪除**, 因爲 textmerg 巨集套件無法處理這幾行。

```
\documentclass{myletter}
\usepackage{textmerg}
\m12
\signature{吳聰敏\\
  台大經濟系}
\includeonly{}
\begin{document}
\include{address}
\Fields{\Title\Add\Surname\en}
\Merge{address.tex}{%
\begin{letter}{\Surname\Title\\ \Add}
\opening{\Surname 主任: ~ 您好!}
\fontsize{12}{20pt}\selectfont
陳中文先生擬申請貴校教職,
本人很高興推荐之。...
敬請\Surname 主任惠予考慮。
耑此, ~ 順祝
\closing{研安!}
\end{letter}}
\end{document}
```

```
% example of add file
% file: address.ctx
%
\m12
主任
台中商專\\ 台中市三民路
簡

主任
糖業研究所\\ 台南市生產路
蕭

主任
高雄工專\\ 高雄市建工路
趙
```

圖 15.5: 大宗郵件 — textmerg 巨集套件

大宗信函與地址檔案之編排控制是以 `\Merge{address.tex}{...}` 指令爲之。其中, 第一圈大括號內之 `address.tex` 是經過轉換、而且已刪除前端說明文字之地址檔案。第二圈大括號內則爲 `letter` 指令環境之內容。textmerg 巨集套件可以和上一小節所製作之個人信頭標誌一起使用。事實上, 圖 15.5的例子即引用了自行設計的信頭。

以上說明以 `\include` 指令轉換地址檔案的方法。若不使用此方法, 也可以自行轉換, 方法如下。地址檔案輸入完成之後, 若檔名爲 `address.ctx`, 且置於 `c:\xtemp` 檔案夾內, 首先執行以下指令:

```
c:\xtemp>cwtex -- address
```

請注意要加上 `--` 選項, 其目的是避免將定義中文字體指令之 `cinput.tex` 檔案引入 `address.tex` 內。轉換地址檔案之後, 必須緊接著轉換信函檔案 `bulkmail.ctx`, 中間不可處理其他文稿:

```
c:\xtemp>cwtex -+ bulkmail
```

請特別注意加入選項 -+, 其目的是將轉換 address.ctx 與 bulkmail.ctx 兩項檔案後所產生之 cinput.tex 合併。轉換完畢, 若 address.tex 檔案前端仍有以 % 開頭之說明指令, 請將之刪除。

15.2 固定格式標籤

大宗郵件的地址通常先列印於自粘式標籤上, 再貼上郵件。欲排版固定格式之標籤, 可使用 labels 巨集套件, 這是由 Sebastian Rahtz, Leonor Barroca 與 Grant Gustafson 所合力創作。不管是大宗郵件之地址、唱片或錄音帶標籤之標示等, 都可以用此巨集套件編排; 標籤格式很容易自行設定。

市面上可以買到各種品牌之自粘標籤, 格式形形色色。以美國 Avery 牌子爲例, 產品編號 5260 之標籤在一張 letter size 紙張上印有 3 欄 10 行共計 30 張之空白標籤。產品編號 5360 則是 3 欄 7 行, 共計 21 張空白標籤。列印標籤時, 紙張大小尺寸須控制精準, 否則內容可能印到標籤之外。其他品牌之標籤可能會標示與某一種 Avery 格式是相容的。

台灣同時通行 letter size 與 A4 兩種規格之紙張尺寸。如果你使用 letter size 紙張, 請在文件類別指令中加上 letterpaper 選項, 若使用 A4 紙張, 請加入 a4paper 選項。選用 a4paper 時, labels 巨集套件自動設定 3 欄/7 行格式。若選用 letterpaper, 內設值爲 3 欄/8 行格式; 但此內設值可自行更改。

如圖 15.6 所示, labels 巨集套件提供 labels 指令環境, 紙張與標籤大小之控制指令須置於全文設定區。本例中, 標籤爲 3 欄 7 行, 計 21 張。每一張標籤內文字排版位置可自行控制, 譬如, \LeftBorder 指令用以調整標籤內左方之空白。標籤之內容輸入於 labels 指令環境內, 格式很簡單。如圖 15.6 所示, 若某一張標籤內容有三行文字, 直接鍵入即可; 標籤之間以空行分隔。本例爲了簡化起見, 僅輸入兩項。

標籤紙面上下方之空白可以用 \TopBorder, \BottomBorder 兩項指令調整。本例中, 分別設定爲 9mm 與 2mm。以 A4 紙張爲例, \paperheight 等於 29.7 公分。由此高度減去 \Topborder, 再減去 \BottomBorder 之後, 除以 \LabelRows 即可算出每一小標籤之高度。實際排版時, 須來回測試幾次, 先試列印於普通紙上。得到正確結果之後, 最後再列印於自粘標籤紙上。

以上例子中, 標籤文字直接輸入 TEX 檔案內, 但我們也可以將標籤內容全部輸入於單獨檔案中, 再引入排版文稿。輸入標籤內容時, 各單項之間以

```
\documentclass[12pt,a4paper]{article}
\usepackage{labels}
\LabelCols=3%      Number of columns of labels per page
\LabelRows=7%      Number of rows of labels per page
\LeftBorder=8mm%   Space added to left border of each label
\RightBorder=8mm%  Space added to right border of each label
\TopBorder=9mm%    Space to leave at top of sheet
\BottomBorder=2mm% Space to leave at bottom of sheet
\begin{document} %  End of preamble
\fontsize{12}{13.5pt}\selectfont\m12
\begin{labels}
台大經濟系
台北市徐州路21號
(0)2351-5468

蕭耀基
糖業研究所
台南生產路54號
\end{labels}
\end{document}
```

圖 15.6: 固定格式標籤

空白行分隔。多行之空白視同一行空白處理。標籤內容中若還有註銷符號 %, 該符號後面文字即不加處理。若標籤內容存放於 names.dat 檔案內, 排版指令如下:

```
\usepackage{labels}
\begin{document}
\labelfile{names.dat}
\end{document}
```

請注意, \labelfile 指令並不須置於 labels 指令環境內。

不過, 如果 names.dat 內含有中文的話, 以上檔案 LaTeX 排版時會出現問題, 理由很簡單: 中文字必須先轉換爲對應之 TeX 指令, 否則無法處理。解決的方法很簡單, 先把地址檔案內之中文轉換爲 TeX 指令, 再進一步處理。雖然問題不難解決, 實際操作起來卻有點麻煩。

最簡單的處理辦法是利用 \include 與 \includeonly 指令。步驟如下: 首先將 names.dat 改名爲 names.ctx, 再將主檔案指令更改如圖 15.7。本例

```
\usepackage{labels}
\includeonly{}
\begin{document}
\include{names}
\labelsfile{name.tex}
\end{documnet}
```

圖 15.7: 標籤

中, \include 指令僅能出現於 document 指令環境內, 其功能視全文設定區 \includeonly 指令之內容而定。若後一指令之內容空白 (如本例所示), 則 \include 指令等於是 \clearpage, 沒有其他作用。換言之, LaTeX 將結束本頁之排版, 將遺留之浮動版面圖表全部排出, 準備排版下一頁。如果本頁目前並無任何內容, 則 \include 指令並無排版作用。

除了排版收信人地址外, labels 巨集套件也可以用來排版發信人地址。在西式信封上, 信函左上方填寫送信人地址。經常寫信的人可以用自粘標籤排出一整頁姓名/地址, 寫完信後, 撕下一標籤粘上信封即可寄出。圖 15.8 是排版送信人地址的例子, 此例子適用於 letter size 紙張之 3 欄 10 行之標籤格式, 也就是 Avery 品牌產品編號 5260 之格式。因爲一頁紙面上將排出 30 張標籤, 而且每一標籤之內容完全相同, 故全文設定區以下列指令設定重複列印 30 份標籤:

```
\numberoflabels=30
```

標籤內容是以 \addresslabel 指令排版, 每行末端須自行加上換行指令 \\。請注意, \addresslabel 僅須鍵入一次, 排版時程式會自動複製於 30 份標籤之內。如果要列印兩整張的標籤, 上述指令中之 30 須改爲 60。

15.3 投影片

論文報告或成果展示經常須使用投影片。LaTeX 有幾套排版投影片的巨集套件, 其中 seminar 文件類別的功能甚強, 使用也方便, 作者是 Timothy van Zandt。如果投影片內容複雜, 此巨集套件可與同一作者所寫的 PSTricks 配合使用。對一般的使用者而言, 本巨集套件所提供的簡單指令應該就可以滿足需求問題。

```
\documentclass[11pt,letterpaper]{article}
\usepackage{labels}
\LabelRows=10
\numberoflabels=30
\voffset=1.5cm
\LeftBorder=1.2cm
\TopBorder=1.3cm
\BottomBorder=1.2cm
\begin{document}
\fontsize{11}{12pt plus.6pt minus.4pt}\selectfont\m11
\addresslabel{台大經濟系\\
台北市徐州路21號\\
(O)02-2351-5468\\
(F)02-2351-1826}
\end{document}
```

圖 15.8: 重覆內容之標籤

seminar 屬於文件類別, 須直接以 \documentclass 引用。每一張投影片置於 slide 指令環境內, 若要排版 5 張投影片, 須使用 5 個指令環境。若無其他設定, 投影片將採橫式列印, 這是所謂的 labelscape 模式。slide 指令環境內可選用不同字體, 也可放大縮小。不過, 列印時字體大小是由內部控制。基本上, 整張投影片會自動放大到填滿整張紙爲止。因爲字體大小會自動縮放, 因此最好是選用描邊字型, 透過 DVIPS 轉換爲 PostScript 檔案, 再以 GSview 預覽/列印。

```
\documentclass{seminar}
\begin{document}
\begin{slide}
  [first slide]
\end{slide}
\end{document}
```

圖 15.9 例子內加入 a4 與 portrait 選項, 前者表示選用 A4 紙張, 後者表示選擇垂直列印。除了加入 portrait 選項之外, 排版時尙須使用 slide* 指令環境。有時候, 一組投影片中大部分皆採垂直編排, 但其中一張須排爲橫式。解決的辦法是將該投影片旋轉 90 度, 指令爲 \rotatebox{90}。此一指令所涵蓋之內容 LaTeX 將視同爲單一字元處理, 因此, 必要時全部文字須

```
\documentclass[a4,portrait]{seminar}
\usepackage{graphicx}
\begin{document}
\begin{slide*}
  [first slide]
\end{slide*}
\begin{slide*}
\rotatebox{90}{
\begin{minipage}{\textwidth}
\centering
  [second slide]
\end{minipage}}
\end{slide*}
\end{document}
```

圖 15.9: 投影片

置於 minipage 指令環境內。請注意, 迷你版面指令環境內不能出現空行, 或使用 \par 指令。此外, \centering 指令是用於調整投影片內容之位置。

在圖 15.9 例子中, 所排版投影片文字內容四周會自動加上外框。如果不要外框, 請在全文設定區加入下列一行指令:

```
\slideframe{none}
```

或者, 我們也可以使用同一指令選擇或自行定義外框之式樣, 詳細作法請參考說明檔。

每一張投影片之內容是列印於投影片上, 不過, 投影片使用者須要有一份列印於白紙上之內容。若在全文設定區加入下列指令:

```
\twoup
```

則整份投影片之內容將重新以 2up 形式排版。換言之, 每兩頁將並列, 列印於一頁之版面上。雙面並列之格式一方面節省紙張, 另一方面讓講者更容易掌握全部投影片之內容, 可說是一舉兩得。

15.4 習題與解答

學校的教師經常出習題/考題給學生, 考試之後則提供解答。準備習題或考題時, 最好同時也備妥答案。Mike Piff 所寫的 answers 巨集套件即作此用

```
\usepackage{answers}
\usepackage[sf,small]{titlesec}
\Newassociation{sol}{Solution}{ans}
\begin{document}
\Opensolutionfile{ans}[ans1]
\section{\r14 習題}
\fontsize{12}{16pt plus1pt minus.6pt}\selectfont\m12
\begin{enumerate}
\item First exercise.
 \begin{sol}
   First solution.
 \end{sol}
\item Second exercise.
 \begin{sol}
   Second solution.
 \end{sol}
\end{enumerate}
\Closesolutionfile{ans}
\newpage
\section{\r14 習題解答}
\input{ans1}
\end{document}
```

圖 15.10: 習題與解答: 例 1

途。 從老師的角度來說, 習題/考題之解答最好是直接輸入題目之下, 排版時, 解答則另成一頁, 排版於習題後面。如果你是排版一本書, 習題可能散佈於書內各處, 解答則集中於書末。巨集套件 answers 的基本功能是將解答集中存入一檔案, 再於文稿末端, 或任何自行指定之地點引入。

圖 15.10 例子中, 第一行引入 answers 巨集套件; 第 2 行所引入之 titlsec 巨集套件與習題解答並無關係, 目的只是讓節標題字體與大小更適配。第 3 行指令如下:

```
\Newassociation{sol}{Solution}{ans}
```

第一選項創造一指令環境 sol, 文稿內之習題解答即輸入於此指令環境內。本例子有兩個題目, 解答即輸入於 sol 指令環境內。相對而言, Solution 也是一個新創造之指令環境, 其目的是用於設計習題頁每一習題之排版格式。如果接受內定之排版格式, 我們不須直接引用此一指令環境。

圖 15.10 指令 \Newassociation 第 3 選項爲 ans, 這是 answers 自動創造之檔案, 檔名爲 ans.tex, 其內即存放習題解答內容。請注意, 以上三個選項名稱可自行選定。譬如, 若第一選項名稱改爲 answer, 則文稿內輸入習題解答時, 須置於 answer 指令環境內。

以上指令雖然選定檔案名稱 ans.tex, 實際排版時須下一指令要求電腦開啓此檔案, 以便開始接受輸入之解答。圖 15.10 第 5 行指令即作此用途。

```
\Opensolutionfile{ans}[ans1]
```

此行指令末端加上 [ans1] 選項, 目的是把檔案名稱進一步改爲 ans1.tex。若不加此選項, 檔名即爲內設之 ans.tex。檔名爲何進一步改變? 理由是在較複雜的文稿中, 可能有必要開啓不同名稱的檔案以存放不同內容之解答。譬如, 若是教科書單數題答案欲附於書後, 雙數題答案打算排版於教師手册內, 則開始兩個不同名稱的檔案即可分別儲存單雙數習題解答。另外一種應用是假設一本書內有 12 章, 每一章之解答可以存入各自之檔案內以利分別處理。相關之細節請參考巨集套件內附說明檔。

如圖 15.10 倒數第 5 行所示, 解答輸入完成之後, 須加入下列指令:

```
\Closesolutionfile{ans}
```

以確認各習題已儲存於檔案內。倒數第 4 行 \newpage 讓版面前進到下一頁。下一行指令是以 \section 指令排版習題解答之標題; 實際上我們可以使用任何指令作任何設計。接下來以 \input{ans1} 指令將前面已儲存之習題解答引入文稿內; 最後一行指令結束文稿內容。

排版時, 解答將依原先題目之格式出現。圖 15.10 的例子是以 enumerate 指令環境排版習題, 解答也以同一格式出現。依 answers 巨集套件之設定, 解答編號將以粗黑體字排版。不過, 使用者可以重新設定。譬如, 在全文設定區加入下列一行指令:

```
\renewcommand{\solutionstyle}[1]{\fbox{#1}}
```

解答編號將以正體字排版, 並加上一四方形, 如 1. 。巨集套件 answers 的原始設定就是使用 \textbf{#1}, 因此編號數字以粗體字排版。

圖 15.11 是第 2 個習題與解答的例子。本例中, 習題是排版於 ex 指令環境內, 此指令環境則是由 \newtheorem 所設定, 請參見第 9 章之說明。開啓

```
\documentclass[12pt]{book}
\usepackage{answers}
\Newassociation{sol}{Solution}{ans}
\newtheorem{ex}{Exercise}
\begin{document}
\Opensolutionfile{ans}[ans-ch\thechapter]
\section{Problems}
\begin{ex}
  First exercise ...
   \begin{sol}
     Solution to first exercise.
   \end{sol}
\end{ex}
\begin{ex}
  Second exercise ...
   \begin{sol}
     Solution to second exercise.
  \end{sol}
\end{ex}
\Closesolutionfile{ans}
\section{Solutions to Chapter \thechapter}
\input{ans-ch\thechapter}
\end{document}
```

圖 15.11: 習題與解答: 例2

解答檔案之指令爲:

```
\Opensolutionfile{ans}[ans-ch\thechapter]
```

其中, \thechapter 是章編號之計數器。若本章爲第15章, 則上列指令將開啓檔案 ans-ch15.tex。相對應的, 倒數第2行以 \input 指令引入本章之習題解答。因此, 若一本書有16章, 則上述之作法即可開啓16個習題檔案。

15.5 索引

除了小說、戲劇等文學作品之外, 大部分的書籍都需要索引。清楚完整的索引是好書的必要條件。編製索引的工作非常繁瑣。以往電腦尚未應用於排版時, 中文書很少有編製索引。LaTeX 提供一些輔助工具, 使編輯索引的工作簡化了許多。

15.5.1 標註索引名詞

編輯索引的第一步驟是標註欲編入索引之名詞, 使用之指令爲 \index。譬如, 在一篇討論資訊產品的書稿中, 若「顯示器」一詞要編入索引中, 則文稿中此一名詞出現之處, 其後應鍵入 \index{顯示器}。此一指令完全不影響正文文稿內容之編排, 其功能只是標註此一名詞應列入在索引中, 使 LaTeX 可以找出頁碼。

以上指令讓「顯示器」一詞在編排完畢之後單獨成爲索引中之一項。如果要把此一名詞排列在「週邊設備」索引項下, 則我們應鍵入之指令爲:

```
\index{週邊設備!顯示器}
```

如果「週邊設備」一詞本身又是「電腦產品」底下的次項, 則鍵入之指令再加一層, 變成:

```
\index{電腦產品!週邊設備!顯示器}
```

排版完畢之後, 索引版面將如下例所示:

電視機, 102
電腦產品, 25
  主機版, 37, 45
  週邊設備, 72
    滑鼠, 36, 63, 64
    顯示器, 35, 63
電熱器, 135, 138

有時候, 索引項目本身須加上排版指令。譬如, 本書之索引中有 \beta 項目。輸入索引項時, 若直接鍵入 \index{\verb+\beta+}, 索引排序時, 此項目將排列於 \verb 處; 而正確的位置應該是在 \baselineskip 之後。遇有此種情形, 我們可以在索引項目之前加入一排序用文詞, 眞正的索引項則置於其後, 兩者之間以 @ 符號隔開。因此, 以上的索引項應輸入爲:

```
\index{beta@\verb+\beta+}
```

索引名詞排序時, 將取用 @ 字元之前的 beta 排序。

有些索引名詞是出現在連續的幾頁中, 欲標誌此類索引項, 可在索引名詞連續出現的開頭處使用下列指令:

```
\index{dvips@\textsf{dvips}|(}
```

本例中, 連續出現的索引名詞爲 dvips, 指令 |(標示開端。連續索引結束處則加上下列指令:

```
\index{dvips@\textsf{dvips}|)}
```

指令 |) 標示結尾。若開端是在第 12 頁, 結尾是在第 15 頁, 索引將以下列形式出現:

dvips, 12-15

由上可知, \index 指令內有一些符號具有特定用途。具體言之, 索引項目內若含有 !, @, 與 | 三個符號, 輸入在 \index{...} 指令內時, 其前面須加上 " 符號, 變成 "!, "@, 與 "|。進一步的細節請見 Goossens, Mittelbach, and Samarin (1994), 第 12 章。

15.5.2 排版索引的步驟

索引是列出各名詞的頁碼所在,因此我們應等到全書排版完成,頁碼不再更動之後, 才開始編排索引。索引的編排必須經過下列幾個步驟:

1. 首先, 確定全篇書稿中那些名詞或文字要編入索引中, 並以 \index 指令標註出來。標註方法, 請見下一小節之說明。
2. 全文設定區須加上 \usepackage{makeidx}, 以引入 makeidx 巨集套件。同時, 指令之後須加上 \makeindex 一行指令。最後, 文稿末端 \end{document} 的前面須加上 \printindex 指令。 若不加上此行指令, LaTeX 仍然會進行索引編輯的工作, 但不會將索引列印出來。
3. 若文稿檔名爲 cxbook.ctx, 以 cwTeX 轉換中文, 並且執行 latex 兩次之後, 硬碟中會出現輔助檔案 cxbook.idx。此檔案內含索引項目及其頁碼, 但尚未排序。每一索引項都置於 \indexentry 指令之後; 其內之中文字爲 TeX 指令格式。

4a. 若爲純英文檔案, 執行指令即可將各索引項正確排序:

```
c:\xtemp>makeindex cxbook
```

排序之後, 硬碟內產生 cxbook.ind 檔案。接下來, 再執行 latex 一次, 文稿末端即自動排入索引項。

4b. 若是中文稿件, 處理過程較複雜。爲了避免錯誤, 可使用 cwidx.bat 批次檔案, 執行指令如下:

```
c:\xtemp>cwidx cxbook
```

請注意, 勿鍵入延伸檔名, 否則結果不正確。執行以上指令後, 硬碟內將產生 cxbook.ind, 最後, 再執行 latex 一次, 文稿末端即得索引。

以上簡單說明產生索引的過程, 實際上排版時有些細節須特別處理, 以下進一步說明之。首先, 我們解釋 cwidx.bat 之內容。此批次檔計有 5 行指令, 主要功能是在產生正確中文排序之索引檔案:

```
tex2xtc %1.idx
cwmkidx %1.xtc
copy %1.ind cwtemp.ind
cwtex -- -+ cwtemp.ind
copy cwtemp.tex %1.ind
```

爲了讓中文索引正確排序, 第 1 行指令先執行 tex2xtc 將 cxbook.idx 之 TEX 指令形式之中文字指令轉回普通中文字, 檔名自動取爲 cxbook.xtc。

第 2 行指令執行 cwmkidx 將檔案內之中英文索引項排序, 檔名自動取爲 cxbook.ind。cwmkidx 程式之功能與 makeindex 程式類似, 但後者在處理某些中文字時會出現錯誤, 原因是 makeidx 在處理索引時, 將某些字元視爲特殊字; 而這幾個字元恰好是中文字所使用的內碼。因此, 若某些中文字的內碼恰是這幾個特殊字元, 排序時即會出現問題。

cwTEX 所提供的 cwmkidx 程式能避免以上這個問題。不過, 中文字的排序是直接利用中文內碼之順序, 因此排序結果並不會百分之百正確, 原因如下: 首先, 中文 Big-5 內碼分常用字與非常用字兩部分。排序時, 非常用字即使其筆劃較少, 也會被排在常用字的後面。另外一種情況是使用者自行造

字。自行造字之內碼通常是排於常用字與非常用字的後面, 排序時即使是筆劃少也會排在最後面。因此, 中文字排序之後有時候須略加調整才能得到正確的結果。欲調整中文排序, 直接以文字編輯軟體修改 cxbook.ind 檔案即可。

除了排序之外, cwmkidx 或 makeindex 程式還會檢查文稿輸入之索引格式是否正確。cwmkidx 程式會將執行結果記錄於 cwindexg.ilg 檔案內, 其內容如下例所示:

```
Scanning input file ...  (38 entries accepted, 0 rejected).
Sorting entries....done (193 comparisons).
Generating output file ... (87 lines written, 0 warnings).
Output written in cwindexg.ind.
Transcript written in cwindexg.ilg.
```

本例中, 第 1 行之 0 rejected 訊息顯示原始索引檔案並無問題, 第 3 行之 0 warnings 訊息則說明排序後之索引檔也正確無誤。如果原先標誌索引名詞時發生錯誤, 訊息將記錄於 cwindexg.ilg 檔案內, 說明錯誤出現在那一行。底下的簡單例子說明錯誤是位於 cwindexg.tmp 檔案的第 30 行:

```
!! Input index error (file = cwindexg.tmp, line = 30)
```

事實上, cwindexg.tmp 是一暫存檔, 其來源是 cxbook.xtc 檔案。因此, 我們應至 cxbook.xtc 檔案的第 30 行更正錯誤。除了標誌索引項目可能發生錯誤之外, 若索引名詞內容太過複雜, tex2xtc 程式將 cxbook.idx 轉換爲 cxbook.xtc 時也可能產生錯誤。

更正所有錯誤之後, 即可作下一階段的處理。cwmkidx 程式雖然產生正確排序的索引檔案, 但檔案內之中文字並非 TEX 指令格式。因此, 排版之前須先轉換索引檔案內之中文字。cwidx.bat 程式後三行指令即透過一暫存檔案作中文轉換之動作, 最後之檔案仍然命名爲 cxbook.ind。

以上的步驟可以在文稿尾端產生排版索引, 但其格式是 LATEX 標準的兩欄格式。若版面寬夠大, 我們也可以三欄排版。此外, 在中文書籍的索引中, 我們通常在索引版面上加入筆劃數之小標題以方便查詢。中文索引項目排序之後, 相同筆劃之索引項會集中在一起。依慣例, 在名詞筆劃數目爲10之索引項之前, 通常會加上「**十劃**」之小標題; 筆劃數爲 11 之索引項之前, 則加上「**十一劃**」小標題。加上小標題的方法很簡單, 以下是一個簡單的例子。

在 cxbook.ind 中, 每一索引項之前都有 \item, \subitem 或 \indexspace 等指令。此爲 LaTeX 所自動加上者, 前兩項指令控制索引之排版形式, 後一項指令控制間距。找到正確筆劃數之索引項目, 在其之前加入下列指令:

```
\item ...
\par\medskip
\noindent{\bb12 十劃}
\par\medskip
\item ...
```

排版後, 小標題將以粗黑字體居左編排, 上下各空出 \medskip 之空白。

排版之後, 索引首頁會加上 **Index** 英文字爲標題。如果是中文稿, 而希望以 14pt 之粗黑體「索引」作爲標題, 可在全文設定區重新定義如下:

```
\renewcommand{\indexname}{\bb14 索引}
```

15.5.3 編輯索引的輔助套件

索引編輯完成之後將直接排版於書末, 但在編輯過程中須核對每一頁版面上有那一些名詞。LaTeX 提供 showidx 巨集套件以利核對每一頁之索引項, 作者爲 Leslie Lamport。

使用 showidx 巨集套件的方法很簡單, 直接在全文設定區加上下列一行指令即可:

```
\usepackage{showidx}
```

若是索引項有中文, 爲了讓中文正確出現, 請特別注意下列兩項:

- 使用 showidx 巨集套件時, 請勿同時使用 \makeindex 指令。
- 巨集檔案 showidx.sty (v1.0k) 內第 26 行的 \@sanitize 指令須暫時以 % 註銷。

巨集套件 showidx 的功能是將每一索引項直接列於該頁邊欄上, 以利核對。一旦核對完成, 即可去掉此巨集套件, 直接以 makeidx 巨集套件將索引編排於書末。巨集套件 showidx 有時候無法處理較特別的索引名詞, 譬如, 若索引項內容爲: \index{par@\verb+\par+}, 排版時會出現錯誤。若以 \url 指令替代 \verb, 通常可以得到正確的結果。(在某些複雜的索引項上,

```
\documentclass[12pt]{book}
\usepackage[cam]{crop}
 \paperheight=22.5cm
 \paperwidth=15.5cm
 \setlength{\textheight}{17.5cm}
 \setlength{\textwidth}{11cm}
 \evensidemargin=0.5cm
 \oddsidemargin=-1.0cm
```

圖 15.12: 截角記號

showidx 巨集套件仍可能產生錯誤。但是, makeidx 巨集套件之處理結果應該是正確的。)

15.6 截角標記

排版書籍時, 我們可在每一頁版面上下左右四個角落加上截角記號 (crop-marks), 書籍印刷之後, 裝訂廠即可依此記號裁剪紙張。本書第一版曾介紹 Philip Taylor 所寫之 cropmarks.tex 巨集套件。此巨集套件原用於 TEX 系統, 在 LATEX 系統中可使用 crop 巨集套件, 作者爲 Melchior Franz。

截角記號的排版很簡單, 圖 15.12 一個例子。引用巨集套件時, 我們加上 cam 選項, 此選項選擇截角記號之樣式爲圓圈內加上十字符號, 這是排版界的標準記號。若改用 cross 選項, 截角記號將變成十字型。

每一本書之設計尺寸不同, 本例子之尺寸爲 22.5×15.5 公分; 文字版面則爲 17.5×11 公分。因爲採用 book 文件類別, 左右兩頁將對稱於書脊編排。單雙頁文字版面之位置須以 \evensidemargin 與 \oddsidemargin 調整。排版時, 此一巨集套件會自動在右上角加上檔名、排版日期、頁碼等資訊。我們對此稍作調整, 將此等資訊移於版面左下角, 字體選用打字機字體。

15.7 排版德文或其他國家文字

除了英文之外, LATEX 也可以排版德文、法文等歐洲國家文字。網路上可以找到幾套專門用來排版某國文字之巨集套件, 譬如, german.sty 可用以排版德文, french.sty 可用來排版法文。不過, 由這些巨集套件所衍生的 babel 巨集套件是使用最方便, 彈性最大的系統, 作者是 Johannes Braams。

babel 系統可處理二十多種歐洲國家文字,對每一個國家的文字都提供下列功能:

- 將英文標題字替換爲該國文字之標題,
- 提供指令以供輸入特別字母,
- 提供正確的音節斷音 (hyphenation)。

除此之外, babel 系統允許在同一篇文稿內排版兩種以上的文字。舉一個例子來說, 如果某篇文稿內同時有英文與德文, 排版時在全文設定區輸入下列指令:

```
\usepackage[english,german]{babel}
```

方括號內選定排版之文字, 最後出現的 german 表示文稿內容以德文爲主。

文稿一開始排版幾節德文段落之後, 若接下來出現英文, 應在英文段落之前鍵入下列一行指令:

```
\language=1
```

則英文段落內之音節斷字即能正確處理。再接下來若遇有德文, 則使用

```
\language=0
```

指令即可回到德文模式。有關於 babel 系統的使用細節, 請參考巨集檔案內附之說明檔, 或者參考 Goossens, Mittelbach, and Samarin (1994), 9.2 節之說明。

16 網路出版

對許多人來說, 上網取用資訊已經是生活中不可或缺的一項活動。因此, 愈來愈多的人想要學習如何將資訊發表在網路上。網路出版的概念是在 TEX 系統之後才出現的, TEX 系統當然不可能包含網路出版的功能。不過, 不少 TEX 使用者希望能將排版結果送上網路發表。譬如說, 學術論文若能置於網路上流通, 對於作者與讀者都可省下許多郵件往來的成本。同樣的, 越來越多的老師將授課的講義、習題或者學期成績, 直接置於網路上, 由學生自行讀取。本章的目的是從 LATEX 系統的角度說明如何作網路出版 (web publishing)。

網路排版最常使用的語言是 HTML (Hypertext Markup Language)。我們進入網站後, 呈現在顯示器上的圖形或文字, 大部份是利用 HTML 語言所建構出來的。不過, 也有不少網頁資訊是使用 PDF (Portable Document Format) 格式。HTML 與 PDF 各有其相對長處與弱點, 本章並不是要介紹這兩種語言。我們的目的是介紹幾套有用的工具程式與巨集套件, 透過這些工具, 即可將 TEX 排版結果轉換成 PDF 或 HTML, 送上網路發表。以下的說明假設使用者對於 HTML 語言已有基本的了解。

16.1 HTML 與 PDF 的比較

介紹轉換工具程式之前, 我們首先簡單比較 PDF 與 HTML 之特點。PDF 檔案格式是由 Adobe 公司所發展, 事實上它是 PostScript 語言的簡化與延伸。所謂簡化, 是指 PDF 並無 PostScript 某些程式語言功能。但反過來說, Adobe 公司在 PDF 中加入一些網路排版的功能, 例如填表 (form) 功能, 因此它是 PostScript 語言的延伸。

PDF 檔案格式通常用於排版專業文稿。相對而言, HTML 則適用於版面不大複雜的文稿。舉例言之, 若文稿中有複雜的數學式, HTML 語言難以應付; PDF 則輕而易舉。雖然 PDF 有相當強的功能可以作複雜的網路排

版, 但這也表示自行下指令編排並不容易。幸運的是, TEX 使用者不須再學習此一語言。我們可以使用現成的工具程式將 TEX 的排版結果直接轉換成 PDF。透過這些工具, PDF 可以說得來全不費工夫。

對於 TEX 的使用者而言, PDF 雖然容易轉換, 但是它的主要問題之一是檔案太大。特別是中文稿件中若使用許多字體種類, 一份 30 頁的稿件, 檔案可能高達 600K 大小。如果文稿內容不大複雜, 而且網路傳輸速度是重要考慮, 則 HTML 語言是較佳的選擇。HTML 語言的概念與 TEX 語言相當類似。從使用者的角度來看, 我們甚至可以說 HTML 是一套簡化的 TEX 排版系統。欲排版 HTML 文稿, 我們使用任意的文字編輯軟體輸入文字與排版指令。送上網站之後, 利用瀏覽器 (browser) 即可觀看排版結果, 必要時可以列印出來。因此, 瀏覽器就是 HTML 系統的預視與列印軟體。

就一般性質的排版功能而言, HTML 語言比 TEX 簡單多了。不過, 就網路排版而言, HTML 有一些功能是 TEX 所沒有的。譬如, HTML 可以排版填表 (form); 當出現在畫面上時, 使用者填入相關資料之後, 程式可以進一步處理。網路購物時, 我們須填入姓名、地址、信用卡帳號, 這就是利用 HTML 的填表功能。

學習 HTML 語言並不困難, 但如果爲了偶而一次的網路排版而花時間去學習, 也是不小的負擔。所幸的是, 網路上也有幾套免費下載的工具程式, 可以直接將 TEX 排版結果轉換爲 HTML。本章將介紹其中功能特別強的 LATEX2HTML 程式。

綜合以上所述, PDF 與 HTML 是網路排版的兩套語言。透過以下所介紹的工具程式, 使用者可以將排版結果直接轉換爲其中之一。因此, 剩下來的問題是要選擇哪一種格式作爲網路排版? 這個問題就必須由你自行決定了! 底下首先介紹轉換爲 PDF 之工具程式。

16.2 轉換為 PDF 格式

將 TEX 排版結果轉換爲 PDF 有幾個可行的路徑。不過, 轉換的結果品質有好有壞。使用 PDF 格式的主要考慮是排版品質; 其中一個重要關鍵則是使用之字型規格。如果要求最佳品質, 不管你用的是那一種工具軟體轉換, 都須設定使用描邊字型。若使用描點字型, 則在顯示器上或是列印結果, 字跡都會模模糊糊的。

在介紹轉換工具之前, 我們先簡單說明如何讀取網路上的 PDF 檔案。目前, 市場占有率最高兩套瀏覽器是網景 (Netscape) 公司的 Navigator 與微軟 (Microsoft) 公司的 Internet Explorer。這兩套軟體可以讀取 HTML 檔案, 但無法直接讀取 PDF。欲讀取 PDF 檔案, 須使用 Acrobat Reader 或 GSview。前者是 Adobe 公司的免費使用軟體, 後者是 Ghostscript 在 Windows 系統上的介面軟體, 作者爲 Russell Lang。Acrobat Reader 的優點是它所呈現的 PDF 畫面品質較精細; 另外, 它與網路瀏覽器有較佳的結合。若在較新版的瀏覽器下安裝 Acorbat Reader, 以滑鼠點選網路上的 PDF 檔案時, 瀏覽器會自動執行 Acrobat Reader 軟體, 在瀏覽器的視窗內顯示檔案內容。反之, 若使用 GSview, 我們須先將 PDF 檔案下載、儲存於硬碟中, 才能預覽、列印。不過, GSview 的優點是它有多方面的功能。譬如, 除了 PDF 外, 它也可以預覽 PostScript 檔案, 甚至也可以將 PostScript 檔案轉換爲 PDF。

以下依序介紹使用 Acrobat Distiller, pdfLaTeX, dvipdfm 與 GSview 軟體工具轉換 PDF 的方法。如果你僅是要將排版結果送上網站供他人下載, 則這些工具程式大抵而言已能夠滿足需求。但是, 如果你希望 PDF 文稿也具有網路文件之特性, 例如串接文稿 (hypertext) 或者填表 (forms), 最簡單的方法是在 LaTeX 文稿中引用 hyperref 巨集套件。此巨集套件可以和上述的工具程式配合使用, 我們將在最後作簡單介紹。

16.2.1 Acrobat Distiller 程式

Acrobat Reader 是 Adobe 公司的產品, 可自網路免費下載。Adobe 是商業軟體公司, 它的生存靠的是出售軟體賺取利潤。既然如此, 爲何它會免費提供 PDF 預視軟體呢? 原因很簡單, 它出售 Acrobat 軟體套件, 其中最重要的就是製作 PDF 之工具軟體: Distiller。此一工具軟體的主要功能是將 PostScript 檔案轉換爲 PDF。因此, 如果 TeX 排版結果已經由 Dvips 轉換爲 PostScript 格式, 只要再執行 Distiller 一次, 即可得到 PDF, 非常方便。

比起底下兩小節所介紹的 dvipdfm 與 pdfLaTeX, Distiller 有兩項特點。第一, 它所轉換出來的檔案較小; 第二, 原先 PostScript 檔案內所引用之外製圖形, 也可以成功轉換。不過, 如前所述, 決定 PDF 文件之品質的關鍵因素是在於 PostScript 檔案所使用的字型規格。以 Dvips 產生 PostScript 檔案時, 如果使用的是描點字型, 則轉換後之 PDF 文件不管是預視或列印, 品質都不理想。欲求良好品質, PostScript 檔案須設定使用描邊字型。

cwTEX 網站上的排版例子是以由描邊字型之 PostScript 檔案轉換而成 PDF。根據使用者的反應, 以某些版本之瀏覽器預視或列印這些例子時, 會出現缺字甚或整篇文稿空白的錯誤情況。Adobe 公司網站曾討論此一問題, 並建議幾個解決方法。解決方法之一是由使用者將 PDF 檔案下載至硬碟內, 再以 Acrobat Reader 或 GSview 預覽/列印。下載 PDF 檔案的方法是以滑鼠右鍵點選檔案, 再依指示進行。

解決方法之二是請作者對 PDF 檔案加工處理。在 Acrobat 軟體套件中有 Adobe Acrobat。此一軟體基本功能與 Acrobat Reader 類似, 但它進一步可以對 PDF 檔案作最適化 (optimization) 處理。先啓動 Adobe Acrobat, 讀取硬碟內之 PDF 檔案, 再重新儲存 (`Save As`) 即可。但儲存時, 須點選 `Optimize` 選項。其他的解決方法, 請參照 Adobe 公司提供的資訊。

16.2.2 Ghostscript 程式

除了 Distiller 程式外, Ghostscript 也提供工具程式可將 PostScript 檔案轉換爲 PDF。若利用6.0版 Ghostscript 作轉換, PDF 檔案之品質甚佳。而且, 文稿內所引用之 EPS 圖形檔案也能正確轉換出來。

在 WinEdt 視窗上方設定有一 ps2pdf 圖像 (見圖 3.3, 頁 23), 文稿排版完成, 轉換爲 PostScript 格式後, 點選此一圖像, 即可開啓 Ghostscript 工具程式, 將排版結果轉換 PDF 檔案。

16.2.3 pdfLaTeX 程式

以上所介紹的是如何將 PostScript 檔案轉換爲 PDF, pdfLaTeX 程式則更進一步, 它可以將 TEX 或 LaTeX 文稿直接排版爲 PDF。此程式延伸 TEX 系統的功能, 作者是 Sebastian Rahtz 與 Hàn Thế Thành。

相對於 Distiller 而言, pdfLaTeX 所轉換出來的 PDF 檔案較大。另外, 如果文稿內引用了外製圖形, 這些圖形必須是 PDF 或 JPEG 格式; PostScript 圖形不能直接使用。雖然有這些限制, 如果你只是偶而轉換 PDF 檔案, pdfLaTeX 程式不失爲是一個好的選擇。事實上, 它有一項特點是其他工具程式所沒有的: 可以使用 True Type 字型排版。詳情請見該軟體之說明檔。

16.2.4 dvipdfm 程式

dvipdfm 作者是 Mark A. Wicks, 此工具程式可以把 LaTeX 排版結果之 DVI

檔案轉換成爲 PDF 檔案。如果文稿檔名 test.ctx, 排版之後檔名爲 test.dvi。若依前一小節之步驟, 必須先以 DVIPS 轉換爲 test.ps, 再以 Distiller 轉換爲 test.pdf。若使用 dvipdfm, 則下列指令:

```
c:\xtemp>dvipdfm test.dvi
```

即直接轉換爲 test.pdf。

根據以上說明, dvipdfm 之轉換途徑更直接了當, 而且此一軟體是免費下載, 那麼我們何必花錢購買 Acrobat 軟體工具? 理由之一, dvipdfm 所轉換成之檔案比 Distiller 轉換結果更大。對於下載 PDF 檔案的使用者而言, 檔案當然是越小越好。理由之二, dvipdfm 軟體處理外製圖形的能力遠不如 Distiller。目前, dvipdfm 僅能處理 PDF 與 JPEG 圖形規格, 而且 PDF 圖形檔案內不能含有文字。理由之三, PostScript 有許多工具程式可供進一步「加工」之用。譬如, psnup 工具程式可以將 PostScript 檔之兩頁合併成一頁, 而轉換後之檔案仍可由 Distiller 轉換爲 PDF。若採用 dvipdfm 途徑, 目前尚無類似之工具軟體可用。

16.2.5 hyperref 巨集套件

LaTeX 系統提供 \label, \ref, 與 \pageref 指令, 使文稿具有引述 (cross-referencing) 之功能。網路排版的 HTML 語言在此一方面有更強的功能。譬如, 我們可以由文稿某處跳到另一處; 可以由甲文稿跳到乙文稿; 甚至可以由甲文稿跳至某一網站, 或者發電子信函給某人。爲方便說明, 我們稱此爲串接連結 (hypertext links)。

如果希望轉換後之 PDF 文件具有網路文件之串接連結功能, 方法之一是使用 Sebastian Rahtz 所寫的 hyperref 巨集套件。首先, 在全文設定區加上下列指令:

```
\usepackage[dvips]{hyperref}
```

方括號之 dvips 選項表示接下來要以 DVIPS 與 Distiller 程式轉換爲 PDF。若要使用dvipdfm, 則選項應改爲 dvipdfm; 若要使用 pdfLaTeX 轉換, 則選項應爲 pdftex。

舉例言之, 若文稿中之表格標題以 \caption 指令排版, 而且以 \label 指令設定徵引標的; 以 \ref 指令設定徵引功能。排版之後, 假設此一表格

爲文稿中第 3 表格, 則文稿中徵引處自動排版爲「表 3」。以上是 LaTeX 本身即具備的功能。引用 hyperref 巨集套件之後, 原 LaTeX 之引述指令在 PDF 檔案中將具有串接連結能力。具體言之, PDF 檔案之表格標題與檔案內之「表 3」處將顯示爲特別顏色。以 Acrobat Reader 預覽時, 若以滑鼠點選文稿內之「表 3」, 畫面即跳至表 3 處。以上的例子說明, 若使用 hyperref 巨集套件, PDF 檔案內之徵引功能可以說得來全不費工夫。

除了圖表之外, 章節目次、註解、數式編號也會出現徵引功能。如果進一步希望文稿有 HTML 語言特有之串接連結功能, 須使用 hyperref 特別提供之指令。譬如, 由文稿某處要連結特定網址, 可使用 `\href{`*url*`}{`*text*`}` 其中 *url* 爲特定網址, *text* 爲文稿內之文字串。若文稿內之甲文字串欲連結至乙文字串, 可以使用 `\hyperlink` 與 `\hypertarget` 指令。詳細說明請參考巨集套件之說明檔。

以 Acrobat Reader 預覽 PDF 檔案時, 畫面左邊另有一欄, 存放 PDF 檔案之章節目錄, 這就是所謂的 bookmarks (書籤)。PDF 檔案之 bookmarks 功能類似普通文稿的章節目錄。若文稿分 5 節, 以滑鼠點選 bookmarks 內第 3 節, 畫面立即顯示第 3 節開頭之內容。對於長篇文稿而言, bookmarks 有其方便之處。hyperref 巨集套件處理 LaTeX 文稿時會將各章節之標題匯總爲 bookmarks。但若是中文文稿, bookmarks 顯示的卻是一些奇怪的指令。事實上, 這的確是章節標題, 不過卻是中文標題經 cwTeX 轉換後之指令。如此的內容對於使用者並無任何幫助。

據我們所知, 要在中文 PDF 文稿上製作出正確的書籤, 唯一的辦法是以 Adobe Acrobat 軟體直接編輯轉換出來的 PDF 檔案。

16.3 LaTeX2HTML 程式

網路排版第二條途徑是 HTML。事實上, HTML 應該是網路排版的主要語言。若版面不太複雜, 只要對 HTML 語言有簡單的了解, 自行以文字編輯軟體寫出網頁並不困難。不過, 若文稿含有數式或圖形, 網頁的製作就困難一些。主要的原因是目前 HTML 語言處理數式的能力不強。

網路上有好幾套工具程式可以轉換 LaTeX 文稿變成 HTML, 不同程式處理數式的方法也不同。本節所要介紹的是 LaTeX2HTML, 此一程式是由多人合作完成, 原始作者爲 Nikos Drakos。此一工具是由 perl 語言寫成, 實際執

行時須用到 DVIPS, Ghostscript 及其他工具程式。這是一套複雜的系統, 但也可預期其功能甚強。LaTeX2HTML 主要特點是它將所有的數式都轉換爲描點圖形, 而且所有的轉換動作都自動處理, 不勞使用者費心。對於中文稿件而言, 文章的中文標題可以設定爲以描點圖形排版, 內容文字則直接輸出, 不作轉換。

16.3.1 LaTeX2HTML 安裝方法

LaTeX2HTML 原先是在 Unix 系統上發展, MSDOS 的版本是由 D. Taupin 所整理 (版本爲 98.2), 適用於 Windows 系統下之 DOS 視窗, 或純 DOS 環境。本小節之說明假設電腦的作業系統是 Win95/98, 安裝之後要在 DOS 視窗下執行。

除了 LaTeX 與 Ghostscript 等程式外, LaTeX2HTML 執行時須使用 perl 程式。安裝 LaTeX2HTML 之過程較複雜, 請仔細依照底下步驟進行。安裝之前, 請確定電腦中之 Ghostscript, MiKTeX/fpTeX 與 cwTeX 已能順利執行。自光碟 `\latex2html` 檔案夾執行 `l2h982.exe`, 將檔案解壓縮至 `c:\xtemp` 檔案夾。請注意, 以下所說明之安裝程序假設檔案是解壓於 `c:\xtemp` 內, 若解壓於其他檔案夾, 請先修改安裝程式。

安裝過程計分4大步驟, 以下依序說明。

- 第1步驟是安裝 perl 與相關工具程式。進入 Windows 之 DOS 模式, 進入 `c:\xtemp` 檔案夾內, 執行下列指令:

  ```
  c:\xtemp>setupdj
  ```

  解壓縮之後, 以文字編輯軟體叫出 `c:\c` 檔案夾內之 `djgpp.env` 檔, 將檔案第10行之 `+LFN=n` 改爲 `+LFN=y`。此項設定讓我們可以使用長檔名。其次, 回到硬碟 `c:` 根目錄, 在 `autoexec.bat` 檔案的 path 指令中加入 `c:\c\bin;`。請重新開機, 使上述設定生效。
- 第2步驟是安裝 LaTeX2HTML。回到 `c:\xtemp` 檔案夾, 執行下列指令:

  ```
  c:\xtemp>setupl2h
  ```

  解壓縮之後, 硬碟中出現 `c:\l2h` 檔案夾。請進入 `c:\l2h\latex2html` 檔案夾內, 以文字編輯軟體叫出 `latex2html.config` 檔案, 第25行

以 $LATEX= 開頭之指令設定 LaTeX 執行檔位置; 第26行設定 DVIPS 之位置。如果 MiKTeX 系統是安裝於 c:, 則這兩行應更改爲:

```
$LATEX = "c:/texmf/miktex/bin/latex.exe"
$DVIPS = "c:/texmf/miktex/bin/dvips.exe"
```

請注意, 檔案夾是以正斜線標示, 而非反斜線。第 30 行與第 31 行分別設定 $TEX 與 $INILATEX, 請同樣修改。如果你使用 fpTeX, 以上兩行應改爲:

```
$LATEX = "c:/bin/win32/latex.exe"
$DVIPS = "c:/bin/win32/dvips.exe"
```

如果你的 TeX 系統是安裝於 d: 硬碟, 以上的 c: 應改爲 d:。

- 第 3 步驟是調整設定。在 \l2h\latex2html 檔案夾內執行下列指令:

  ```
  c:\l2h\latex2html>inst-dj
  ```

  執行時畫面上出現許多訊息。大部分之訊息只須按下 [Enter] 即可。若你是使用 fpTeX, 畫面可能出現: "I can't find 'latexdum.tex'" 之訊息, 此時請鍵入 c:/l2h/latex2html/latexdum.tex。接下來, 畫面仍會出現找不到 article.cls 之訊息, 請直接鍵入 x 以結束此一部分之測試動作。以上的訊息並不影響安裝結果。

  再接下來的訊息中, 有一個問題是詢問要使用 GIF 圖形格式或 PNG。請按下 g 鍵, 選用 GIF 格式。

- 最後調整。LaTeX2HTML 程式是透過 Ghostscript 程式轉換圖形; 原程式之設定假設電腦中使用的是 Ghostscript 3.33版。最後的調整是重新設定使用 Ghostscript 6.0版。在 c:\l2h\latex2html 檔案夾內找出 localdos.pm 檔案, 以文字編輯軟體開啓, 第 13 行的 $GS_LIB 開頭之設定行應改爲:

  ```
  $GS_LIB = '.;c:/aladdin/gs6.0/lib;c:/aladdin/gs6.0/fonts';
  ```

  其次, 第 22 行以 $GS 開頭之設定, 應修改爲:

```
$GS = 'c:/aladdin/gs6.0/bin/gswin32c.exe';
```

同一檔案夾內另有 l2h-conf 檔案, 其第 109 行爲

```
# $DVIPS = "dvips -E";
```

請將行首之 # 符號去掉。此一設定可加快轉換速度, 並避免一些不必要的錯誤。

最後, 在 c:\l2h\latex2html\cwtex 檔案夾內可找到 cwtex.sty 檔案, 請移入 \localtexmf\tex\latex\package 檔案夾內。檔案夾若原本不存在, 請自行建立。另外, 在 c:\l2h\latex2html\texinputs 檔案夾內有一些 LaTeX2HTML 所特別提供之巨集套件。爲了讓 LaTeX 排版時能找到這些檔案, 請將這些檔案移入同一檔案夾內。

複製完成之後, 請更新檔案資料庫。MiKTeX 使用者, 請執行:

```
c:\>initexmf -u
```

fpTeX 使用者, 請執行:

```
c:\>mktexlsr
```

安裝完成之後, 請試排版下一小節之測試檔案。若結果無誤, c:\xtemp 內之檔案即可刪除。

16.3.2 使用 LaTeX2HTML

LaTeX2HTML 系統甚爲複雜, 但從使用者的角度來看, 排版一般性文稿並不困難。完整的說明檔 manual.ps 置於 \l2h\latex2html\docs 檔案夾內, 請自行印出參考。

說明使用方法之前, 我們簡單說明此一系統之設計理念。HTML 語言可以處理簡單的數學式, 但太複雜的數式就無能爲力。不過, HTML 處理圖形的能力不錯, 因此一個可能的途徑是將所有的數式轉爲圖形。LaTeX2HTML 轉換數式爲圖形的方法如下: 先將文稿排版一次, 找出其中的數式或引用之外製圖形; 並將這些數式與圖形利用 Dvips 轉換爲一個一個的 PostScript 檔

案。接下來, 利用 Ghostscript 程式將檔案轉換爲 GIF 格式。最後, 再把原文稿轉換爲 HTML 格式, 並將所有數式或圖形以 GIF 圖形檔替代。以上的轉換步驟全部自動處理, 不勞使用者費心。

那麼 LaTeX2HTML 如何處理中文呢? 中文可以用兩種方式處理: 第一是把中文視爲圖形; 第二種方式是中文不作任何加工, 直接轉入 HTML 檔案中。要把中文視同圖形處理, 方法很簡單, 僅須在 LaTeX 文稿中讓中文以數式模式出現即可。譬如, cwTeX 網頁是以 LaTeX2HTML 製作, 其中一項標題是「總體經濟學」。在原 LaTeX 文稿中, 此一標題是以下列方式輸入:

```
$\mbox{\bb15 總體經濟學}$
```

LaTeX2HTML 認定此爲數學式, 即將這5個中文字轉換爲一個圖形檔。此例中, 5個中文字視爲一圖形, 故轉換結果僅得一圖形檔案。要把5個中文字轉換爲5個檔案也可以, 細節請參考說明檔。以上是以中文標題爲例, 英文標題當然也可以使用同樣方式處理。

中文的第二種處理方式是不作任何加工, 直接移入 HTML 文稿內。直接移轉中文看似簡單, 不過 LaTeX2HTML 系統設計是以英文爲主要對象; 文稿中若有中文字, 有些中文字處理之後會變得無法辨識。爲了正確處理中文, cwTeX 提供一巨集套件 cwtex.sty 及一對應之 perl 程式 cwtex.perl。任何中文稿若要轉換爲 HTML, 請在全文設定區加入下列指令:

```
\usepackage{cwtex}
```

在作普通文字排版時, 此一巨集套件並無任何作用。但是, 在作 HTML 轉換時, LaTeX2HTML 會自動引入 cwtex.perl 程式, 讓中文字得以正確處理。

事實上, 以上處理 cwtex 巨集套件的方法是 LaTeX2HTML 的標準程序: 若 LaTeX 文稿引用了任何巨集套件, 系統中必須有一對應的 perl 檔案, 否則不能正確處理。LaTeX2HTML 說明檔內列出它目前所支援的巨集套件, 未來支援巨集套件會越來越多。不過, 這也表示如果你的文稿要轉換爲 HTML, 文稿內最好僅使用標準的 LaTeX 指令。欲使用額外的巨集套件, 請先確定它是在 LaTeX2HTML 的支援範圍之內。

另一個相關的問題是字體指令。HTML 語言僅提供簡單的字體指令, 如正體、斜體、放大、縮小等; 但無法使用特別的字體, 如 Garamond。[1] 就英文

[1] 如果要使用特別字體作網路排版, 應使用 PDF 格式, 而非 HTML。

字部分而言, LaTeX2HTML 無法處理 \fontsize 指令, 因此也無法處理類似 \sz12 之巨集指令。就中文字而言, 特定的標題或句子若要轉換爲圖形檔, 應加入中文字體指令; 其餘要直接移入 HTML 文稿內之中文則不應使用中文字體指令。請參考以下例子之說明。

爲說明轉換中文的方法, \l2h\latex2html\cwtex 檔案夾內有五個測試檔。第一個檔案 test-h1.tex 爲純英文檔。此測試文稿內含一圖形, 檔名爲 dmark.eps, 請先創造 \c:\tex\html 檔案夾, 將兩個檔案移入其內。欲轉換爲 HTML 檔案, 先以 latex 編排 test-h1 兩次; 其次, 執行:

```
c:\xtemp>l2h test-h1
```

l2h.bat 爲一批次檔, 置於 c:\c\bin 檔案夾內。批次檔中設定一些選項, 若這些選項不合你的需求, 請自行修改之。

若執行成功, c:\xtemp 檔案夾下將出現 \test-h1 次檔案夾, HTML 即置於其中。以瀏覽器開啓 index.htm 或者 test-h1.htm 檔案, 即可看到結果。檔案夾內可發現約二十來個 GIF 檔案, 這些都是由文稿內之數式與引用之外製圖形檔轉換而來。

實際轉換之前爲何須先執行 latex 兩次? 原因是文稿中若使用了串接連結指令, 則 LaTeX2HTML 轉換時須讀取輔助檔案 test-h1.aux 內之資訊。先執行 latex 兩次, 其目的即在產生輔助檔案。因此, 若文稿並未使用 LaTeX 的引述指令, 如 \ref 或 \label; 也未使用 LaTeX2HTML 的串接連結指令, 如 \htmlref, 則文稿可直接轉換, 不需先前的步驟。換言之, 某一文稿若執行 latex 時硬碟中不產生 .aux 輔助檔案, 文稿可直接執行 l2h.bat 程式轉換。反之, 若文稿中使用了串接連結指令, 但轉換之前忘掉先產生輔助檔案, 則產生之檔案就喪失了串接連結功能。

以上第一個例子之檔名爲 test-h1.tex, 但轉換之後會產生兩個 HTML 檔案, 分別爲 test-h1.htm 與 index.htm。這兩個檔案內容相同。在 HTML 語言中, 首頁 (main page) 須取名爲 index.html, 這是 LaTeX2HTML 產生兩個檔案的原因。如果你是要設計網頁之首頁, 原始 LaTeX 檔案應直接取名爲 index.tex 或 index.ctx。

第 2 個例子 test-h2.ctx 爲一中文檔, 該檔案前端簡單說明執行步驟。此檔案之內容單純, 無數學式或圖形, 但文稿中使用引述指令 (\htmlref 與 \label) 指令。因此我們首先將此檔案依一般 LaTeX 文稿處理: 先執行 cw-

tex, 再執行 latex 兩次。排版完畢, 硬碟中出現 test-h2.aux 輔助檔案, 這是下一階段所需要的。

接下來, 須再執行 cwtex 一次, 但指令中須加入 -i 選項:

```
c:\xtemp>cwtex -i test-h2
```

此選項之作用爲何? 如果以文字編輯軟體叫出 test-h2.ctx 檔案, 第 17 行有一指令 \ctxfoff, 這是 cwTEX 控制中文字體轉換之指令。執行 cwtex 時, 若不加入 -i 選項, cwTEX 僅在該指令前方加上 %, 因此對於以下之排版步驟毫無影響。 相反的, 執行時若加入 -i 選項, 則該行指令以下之中文不會轉換爲 TEX 控制碼。換言之, 所有中文字原樣錄於 test-h2.tex 檔案內。最後一個步驟執行 l2h.bat 時, 這些中文字即直接轉入 HTML 檔案內。

綜合以上所述, 欲轉換爲 HTML, 共須執行 5 道指令。若經常作轉換, 我們可以寫一簡單批次語言, 一方面節省時間, 同時也避免錯誤。我們所用的批次檔案如下:

```
deltree %1
cwtex %1
latex %1
latex %1
cwtex -i %1
l2h %1
```

假設批次語言取名爲 new.bat, 則執行指令如下:

```
c:\xtemp>new test-h2
```

第一行指令的目的是將舊有檔案夾刪除。

上面已經說明, LATEX2HTML 系統是針對英文設計。爲了讓它正確處理中文, 我們須在中文稿的全文設定區加入巨集套件:

```
\usepackage{cwtex}
```

文稿中加上此巨集套件時, LATEX2HTML 處理時將自硬碟中尋找 cwtex.perl 檔案, 其中包含有正確處理中文之指令。基本上, 所有的中文稿都須使用此巨集套件, 否則中文字無法正確處理。

第 3 個例子 test-h3.ctx 內容與上一例子相同, 但標題改爲圖形檔, 故可選用較大之特定字體。因爲標題內有中文字, 因此 \ctxfoff 指令置於中文標題之後。(爲什麼?) 標題之圖形檔將自動命名爲 img1.gif。排版完成, 上載 HTML 時, 請記得將圖形檔也上載。若文稿中有兩個圖形檔, 第 2 個圖形檔將命名爲 img2.gif。

現假設文稿內稍後另有一行標題, 也是要以圖形方式處理, 則在第二個標題之前須加入 \ctxfon, 啓動中文轉換功能, 轉換第二行標題。但是, 請記得在中文標題的下一行加入 \ctxfoff 指令關閉之。否則, 以下所有的中文字會一個一個全部轉換成圖形。

第 4 個例子延續上述例子, 但文稿內有一表格; 表格標題是以 \caption 指令排版, 文稿內另以 \ref 指令徵引表格。排版時, 此表格自動編爲「表 1」。爲產生引述功能, LaTeX 排版時會產生 test-h4.aux 輔助檔案, 其中與表格有關的一行內存有表格標題。不過, 標題內之中文字已轉換爲 TeX 指令。LaTeX2HTML 若直接讀取此一行指令, 因無法辨識指令內容, 會產生錯誤訊息。故在執行最後的轉換動作之前, 先執行下列兩行指令:

```
c:\xtemp>tex2xtc test-h4.aux
c:\xtemp>copy test-h4.xtc test-h4.aux
```

第一行指令是利用 tex2xtc 程式將 test-h4.aux 檔案內之 TeX 指令轉換回中文, 轉換後之檔名自動取爲 test-h4.xtc。第二行指令再將此檔案覆蓋原有之 test-h4.aux。全部之執行步驟請見 test-h4.ctx 前端說明。

第 5 個例子爲 test-h5.ctx 檔案, 來源是製作 cwTeX 網頁之檔案。我們將之簡化, 原首頁之照片以一張貓的圖畫替代。執行成功之後, 首頁有兩個串接連結點, 一是連往「總體經濟學」, 以英文字母 m 表示; 另一連往「cwTeX 排版系統」, 以英文字母 c 表示。實際製作 cwTeX 網頁時, 我們曾對轉換出來的 test-h5.htm 檔案稍作編輯, 讓連結點改變成是文字標題本身。了解 HTML 語言的使用者, 應不難作類似修正。

以上的例子使用了一些 LaTeX2HTML 程式特別提供的指令, 如 \htmlref, \label, \htmladdnormallink 等, 這些指令可讓文稿具有串接連結功能, 可從本文稿連結至其他檔案、網站、或傳送電子郵件等。詳細使用方法, 請見 LaTeX2HTML 說明檔案。

17 造字

中文排版的問題是: 中文字永遠有不足的問題。萬一用到字型檔沒有的中文字, 使用者只好自行造字。造字時, 可以造描邊字 (outline font), 也可以造描點字。造描邊字需要有特別之軟體。描點字則只需使用 emTEX 所附工具程式。第 17.1 節簡單介紹 cwTEX 中文字之排序; 17.2 節簡單說明造新字的方法; 17.3 節說明如何使用新造字。

17.1 中文字之排序

日常生活中經常使用之中文字大約在三千到四千之間。目前個人電腦系統所提供的每一套中文字型, 字數都是 13,053 字 。整套字分常用字與次常用字兩部分, 前者含 5,401 字。

cwTEX 的基本中文字型 (明體、粗黑體、楷體、圓體與仿宋體) 造滿13,053字, 其餘少部分字型檔都僅含 6,844 字。我們稱此 6,844 字爲 cwTEX 常用字。常用字分佈於0-26 個字型檔中, 每一字型檔最多可置放 256 個字。以明體字爲例, m0 字型檔內之中文字從第 44 排至第 255; m21 字型檔內從 0 排至 238; 其餘字型檔都是從 0 排至 255。cwTEX 系統之次常用字有6,209 個, 放置於字型檔 27-51 之間。其中, 第 27-50 字型檔皆存放 256 字, 第 51 字型檔的 0-64 存放 65 個次常用字; 第 87-255 則存放日文平假名與片假名。若不計入日文字, 常用字與次常用字合計爲 13,053 字。

如果你的文章中用了某一個字不在上述的範圍之內, 執行 cwTEX 時會出現警告訊息。此時, 你的文章仍然可以編排, 但是在排版的版面上該字將成爲空白。缺字可能有兩種情形: 第一種是該字可輸入文稿中, 但無法預視或列印。這表示此字屬於 cwTEX 次常用字範圍。第二種情況是連在中文系統下都無法輸入此字, 表示此字爲13,503 字以外。

在第一種情況下, 我們只要依下一節的說明造完中文字, 問題即可解決。第二種情況較麻煩, 我們還須對所缺之中文字設定中文內碼, 並讓中文

系統可以鍵入此字。MSDOS 系統下的倚天中文或者中文 Windows 系統各有其造字的方法。此一部分請自行參照相關手册處理。爲了讓 cwTEX 能夠處理新造字, 新造字之內碼必須在下列範圍內: FA40-FEFE, 8E40-A0FE 及 8140-8DFE; 而新造字必須置放於第 52-57 字型檔中。譬如, 若所缺字爲明體, 則一開始之新造字應存於 m52 字型檔內。

若不想在中文系統下也造一對應新字, 一個偷懶的解決辦法如下: 從中文系統的次常用字中選擇一個幾乎不可能用到的中文字。輸入時以此字替代我們所要造的新字。造好新字之後, 將此字之內碼對應到選用替代字上, 即可預視或列印正確的文稿。請見 17.3 節之說明。

17.2 造新字

cwTEX 系統之中文字是 Type 1 描邊字型規格, 預視或列印時再經由 ps2pk 程式轉爲描點字型檔。使用者可直造描邊字, 也可以造描點字; 但前者需要有適當的造字軟體。若只是偶而缺一個字, 直接造描點字是比較簡單的。

17.2.1 造描點字

在說明如何造描點字之前, 我們先說明 LATEX 字體尺寸規格。第5章說明每一個中文字或英文字體都有不同的放大倍數。以 HP 300dpi 雷射印表機使用之 10pt 明體字爲例, 明體點陣中文字型檔爲 m0.pk 到 m26.pk, 置於硬碟:

```
c:\localtexmf\fonts\pk\laserjet\public\cm\300dpi
```

檔案夾之下。若爲 12pt 明體描點字型, 置放之檔案夾爲 ...\cm\360dpi。

造字之前, 須先確定所缺中文字之點數爲何。同一個字若缺乏 10pt 與 12pt 點數, 我們須造兩個字。反之, 若缺乏的是 12pt 字, 但造完之後置於不正確的檔案夾內, 結果還是無法使用。

造描點字最簡單的方法是利用 emTEX 的 pkedit 程式修改現有的字形檔。如果要造 12pt 之明體字, 且假設新造字要置於 m52.pk 檔中, 則第一個步驟是從現有的字型檔案複製 m52.pk 檔及 m52.tfm。譬如, 常用字「涂」是以「水」爲部首。但也有姓氏是寫爲「凃」, 以兩點爲部首, 右半邊則完全相同。由「涂」字造此新字很容易。首先, 我們須找出「涂」之字型檔。在中文系統下輸入 test.ctx 檔, 其中僅含一中文字「涂」。執行 cwtex 轉換之

後, 硬碟內產生 cinput.tex 與 test.tex 兩檔案。前者之內容指示字型檔爲 m22; 後一檔案之內容指示「涂」字位於字型檔第 59 個字元位置。

造字的第一步是將 \texmf\fonts\tfm\cwtex 檔案夾內之 m22.tfm 檔案複製爲 m52.tfm。接著, 在 \localtexmf\fonts\...\360dpi 檔案夾內, 複製 m22.pk 描點字型檔爲 m52.pk。

利用 pkedit 造新字, 指令爲:

```
c:\localtexmf\ ...\360dpi>pkedit m52.pk
```

進入造字畫面之後, 首先選擇所要修改或造新字的字碼。最簡單的方法是按 C 鍵, 再按 A 鍵。字鍵 C 代表 character, 字鍵 A 代表欲修改在 A 字碼位置之中文字形。按照 ASCII 字碼之編列, 字母 A 之字碼位置爲 65。因所造新字置於第 59 字碼位置, 則由字母 A 字碼按 6 次 PgDn 鍵, 即可進到字碼位置。(顯示器的左上角有字碼位置之標示。)

開始修改, 請按 E 鍵 (代表 edit)。造字或修改的方法很簡單, 在游標位置按 Ins 鍵, 則加一黑點, 按 Del 鍵則去除黑點。如果你裝有滑鼠, 左邊按鍵之功能爲加點, 右邊按鍵爲除點。你會發現游標只能在某一範圍內移動, 此一範圍事實上就界定字體的大小。造完字或修改完畢後, 請按 [Enter] 鍵離開。我們可以用 [PgDn] 或 [PgUp] 進到另一個字碼位置修改或造字。最後, 按 X 鍵儲存並離開。使用新造字的方法, 請見 17.3 節。

17.2.2 造描邊字

如果有適當的字型設計軟體, 如 Fontographer, 則可考慮造 Type 1 描邊字。造好一描邊字型之後, 可由工具程式產生任可尺寸之描點字型檔。使用上較爲方便。造字時應注意新造字之大小須與原有字體相當。因此, 理想的方法是以現有字型檔爲基礎開始造字。其中細節, 請見各軟體之使用手册。

造完字之後, 字型軟體可以產生 Type 1 字型檔案, 其中包含各式之字型規格檔案。假設字型檔名爲 m52, 我們僅留存 m52.pfb 與 m52.afm 兩檔案, 其餘可刪除。使用 afm2tfm 程式由 m52.afm 產生 m52.tfm, 即大功告成。最後, 請將 TEX 字型規格檔 m52.tfm 移入 \texmf\fonts\tfm\cwtex 檔案夾; Type 1 描邊字型檔 m52.pfb 移入 \texmf\fonts\type1\cwtex 檔案夾; Type 1 字型規格檔 m52.afm 則移入 \texmf\fonts\afm\cwtex 檔案夾。

17.3 使用新字

欲使用新造字, 我們必須先建立一字元對照檔案 cwfont.usr。此一檔案必須置於 \texmf\cwtex 檔案夾中。檔案中, 每一行標示一新造字、對應之字型檔名、及字元位置。其順序爲:「新字 字型檔名 字元位置」。例如:

```
涂 52 59 % m12
廊 52 1  % b14, m14
```

上例中,「涂」爲所造的第一個新字, 置於 m52.pk 中的第 59 個字碼位置;「廊」爲第 2 個字, 置於 m52.pk 中的第 1 個字碼位置。

% 符號以後爲備忘資訊, 提醒使用者已造了那些字。譬如, 若已造了「涂」字之明體 12pt 字型檔, 應標示爲 m12。若同時造了放大一級及放大二級之明體字, 則可標示爲 m12, m14。如果你同時造了放大兩級之明體及黑體字, 則可標示爲 b14, m14, 如上例中第 2 行所示。

經過以上設定, 轉換中文字時若遇有缺字情形, 如上例中之「涂」字, 轉換程式即檢查 cwfont.usr 檔案內是否有字元對應。若發現有對應, cwTEX 即設定取用 m52 字型檔之第 59 個字元排版文稿。

若所缺之字並不在 Big-5 之 13,053 字範圍內, 我們還須在中文系統下造對應新字, 才能輸入於文稿中。但是, 中文系統之造字也很麻煩, 因此, 17.1 節提供一偷懶的解決辦法: 從次常用字選用一字作爲缺字之對應。再以上面的例字說明, 事實上,「涂」字不在 Big 5 內碼內, 因此無法直接輸入於文稿。假設我們已造好此中文字, 並置於 m52 字型檔第 59 個字碼位置。則 cwfont.usr 檔案第一行之第一個字可以一幾乎不會使用之次常用字替代, 譬如,「捈」字。輸入文稿時,「涂」之位置即鍵入「捈」; cwtex 程式轉換中文時, 遇有次常用字會先到 cwfont.usr 檔案搜尋是否有字元對應。若有對應, 所缺之字即選用 m52 字型檔之第 59 個字排版。如此, 雖然文稿所鍵入的不是正確的字, 但透過替代字元對應仍可排出正確的結果。

18 偵測錯誤

排版過程不免出現錯誤。錯誤的來源很多: 少輸入右大括號, 左大括號誤爲左圓括號, 指令鍵入錯誤, 指令的用法錯誤等等。排版時, cwTEX 與 LATEX 會試著指出錯誤來源。但是, 有些錯誤一開始難以判斷其原因及發生處。譬如, 以 \footnote 排版註解時, 若忘了鍵入右大括號, LATEX 即設法將其下的文字全部排爲註解。最後, 因爲容納註解文字之空間不足, 乃出現 TeX capacity exceeded ... 之訊息。而實際上, 眞正的原因是文稿中少了一個右大括號。

解讀錯誤訊息的能力愈強, 我們就能愈快的完成排版工作。底下 18.1 節說明 cwTEX 之錯誤與警告訊息, 18.2 節說明 TEX 與 LATEX 的錯誤與警告訊息的意義。18.3 節提示迅速找出錯誤原因的訣竅以供參考。

18.1 cwtex 訊息

cwTEX 的錯誤與警告訊息主要是在中文字體方面。執行 cwtex 轉換程式時, 若遇上字型檔中沒有中文字, cwTEX 會發出警告訊息, 說明錯誤發生於檔案的那一行, 並將該字留爲空白。

以第3章之 test.ctx 爲例, 若第5行末端之「易」輸入錯誤, 變成是字型檔案中沒有的中文字, 則轉換中文字時, 顯示器上將出現下列警告訊息:

```
! Line 5: X is not character of cwfont. Ignore.
```

其中 X 爲鍵入之中文字。

此外, 在轉換中文字時, cwtex 程式會簡單檢查文稿檔案中可能的格式錯誤。首先, 輸入指令時左右大括號必須對稱出現。因此, 文稿中的右大括號數目若少於左大括號, 轉換程式即發出下列訊息:

```
! Too many {'s.
```

反之, 若在最後一行 (第 7 行) 的 \end{document} 指令中, 遺漏了左大括號, 而使文稿中右大括號數目多於左大括號, 則訊息如下:

```
! Line 7 too many }'s.
Press Enter key to continue or x and enter key to quit ?  [9])
```

若暫時不想處理此一錯誤, 應按 [Enter] 鍵繼續以下的轉換工作。

LaTeX 文稿是以 \end{document} 結尾, 因此, cwtex 程式會檢查檔案結尾處是否有此指令。輸入時若遺漏左大括號, 最後一行變成 \enddocument}, cwTeX 會發出下列錯誤訊息:

```
! I can't find \end{document}.
```

LaTeX 編排文稿時須使用中文字體規格檔案, 附加檔名爲 .tfm。cwtex 於轉換中文時, 會同時檢查硬碟內是否有所需之中文字體規格檔, 若缺乏所需字體規格檔案, 即發出底下訊息:

```
! I can't find xx.tfm
```

其中, xx.tfm 爲所缺之中文字體規格檔案。譬如, 若所列檔名爲 b2.tfm, 表示硬碟中並無中黑字體。解決辦法如下: 安裝文稿中所使用之字型檔案, 或者文稿中不要使用此種字體。

若文稿檔名爲 test.ctx, 執行 cwtex 程式之後將產生 test.tex 輔助檔案。但是, 若原始文稿之附加檔名取爲 .tex, 例如 mydoc.tex, 中文轉換後之輔助檔案亦將取爲同名之 mydoc.tex 遇有此種情況, cwtex 會將原始檔案改名爲 mydoc.$$$; 並發出下列訊息:

```
! Don't use .tex as extension name, please rename your mydoc.tex
```

因此, 文稿內若有中文, 請勿以 .tex 爲附加檔名。

除此之外, cwtex 發出的錯誤訊息主要是提示 \verb 與 \url 指令內, 以及 verbatim 指令環境內不得使用中文字。若文稿檔名爲 test.ctx, 中文轉換完畢後, cwtex 會將執行過程記錄於 test.xlg 檔案內。必要時, 我們可以開啓此檔以了解細節。

18.2 latex 訊息

執行 latex 時所產生之訊息, 部分來自 LaTeX, 部分來自原始的 TeX 程式。不過, 對一般的使用者而言, 此一區分並不重要。重要的是, 如何理解警告與錯誤訊息, 以更正錯誤。

再以第3章的 test.ctx 爲例, 假設輸入第6行指令時, $\sqrt{\beta}$ 誤爲 $\sqrt{\beta$, 亦即少鍵入右大括號。在轉換中文時, 雖然 cwTeX 會發出如上一節所述之警告訊息, 但仍然會將全文轉換完畢。但接下來, latex 編排時, 顯示器上將出現下列的錯誤訊息:

```
! Missing } inserted.
<inserted text>
                }
l.6 ...S\cH23}, {\MaS\cH224} $\sqrt{\beta$
                                          , {\MeS\cH171}\z...
?
```

第 1 行 ! 符號之後的文字表示錯誤的原因, 或 LaTeX 所自動採取的補救方法。此例中訊息內容爲 Missing } inserted., 表示 LaTeX 肯定錯誤的原因是少鍵入一個右大括號, 因此自行將漏掉的大括號插入。

第4行開頭的訊息 l.6 說明指令錯誤發生於第6行 (line 6)。該行末端之 ... \beta$ 與下一行開頭處之 , {\MeS... 指明錯誤發生所在。拿以上兩行與原始 test.ctx 檔案之第6行比較, 我們發現原檔案之第6行的中文字在錯誤訊息中變成奇怪的符號, 如 {\MaS\cH224}, {\MeS\cH171} 等等。事實上, 這些符號正是原來中文字經過轉換之後的格式。

錯誤訊息的最後一行之 ? 號表示 latex 正等著使用者之動作。此時, 如果我們鍵入另一個 ? 號, 顯示器上即出現下列訊息:

```
Type <return> to proceed, S to scroll future error mes-
sages,
R to run without stopping, Q to run quietly,
I to insert something, E to edit your file,
1 or ... or 9 to ignore the next 1 to 9 tokens of input,
H for help, X to quit.
?
```

若鍵入 [Enter], latex 將繼續執行, 直到遇到下一個錯誤才停止。如果不再

碰到錯誤, 即排版至最後一頁。

若鍵入下列各鍵, 其效果如下:

- S latex 將繼續編排文稿至最後一頁, 中間即使再有其他錯誤也不停止。所有之排版訊息會存於 test.log 檔案中, 稍後可以文字輸入軟體閱讀之。
- R 與 S 選項類似。但 S 選項若遇有找不到檔案之情況仍會停下來等待進一步之指示; 此選項則不停。
- Q 與按 R 鍵類似。但前者會使排版訊息出現於顯示器上, 此一選項則使訊息只存於 test.log 中, 不出現在顯示器上。
- H 求助選項, latex 將試著進一步分析錯誤之原因, 並提出解決之道。
- X 要求立即停止編排, 並退出排版程式。

排版時, latex 所產生的訊息中, 最重要的是錯誤發生於那幾行。若能確認錯誤之所在, 通常我們很快就能找出問題原因。不過, 如本章開頭的 \footnote 例子所示, 有時候, latex 所指示的行數不一定是錯誤的源頭, 可能只是排版程式無法繼續執行之處。大部分的錯誤訊息內容都很清楚, 以下只舉幾個常見的訊息略加說明。

輸入 test.ctx 時, 若第 6 行之 $\sqrt{\beta}$ 誤爲 $\sqrr{\beta}$, 將出現下列錯誤訊息:

```
! Undefined control sequence.
<recently read> \sqrr

l.6 $\sqrr
          {\beta}$.
?
```

latex 不會說鍵入的指令錯誤, 只會提醒說 \sqrr 指令未曾無定義過。另外一個類似的錯誤訊息爲:

```
! LaTeX Error: Environment ... undefined.
```

此訊息指示文稿中所使用之指令環境並無定義; 最可能的原因仍然是指令環境之名字輸入錯誤。

```
! LaTeX Error: Can be used only in preamble.
```

文稿內以 \usepackage 指令引用巨集套件時, 應下於全文設定區 (preamble), 亦即在 \begin{document} 指令之前。若引用巨集套件之指令誤下於 \begin{document} 指令之後, 排版軟體即發出上列訊息。

```
! TeX capacity exceeded, sorry ...
```

依字面解釋, 此項訊息說明 TEX 程式的空間不足, 無法處理複雜的文稿。但是, 如本章一開頭的例子所示, 眞正原因通常是忘了在 \footnote{...} 指令中加上右大括號。

另一個易於出現上述錯誤訊息的情況如下。若文稿內有許多的圖表, 且每一圖表都置於浮動圖表 (figure 或 table) 指令環境內, 也容易產生此項錯誤訊息。事實上, 如果大部分圖表占一整頁空間, 我們可直接以指令控制圖表出現之位置, 不須借助於浮動圖表指令環境。另外, 一般的 TEX 版本只允許在一文稿中使用256種字體。如果中文文稿內使用太多的字型檔, 也可能出現此項錯誤訊息。解決辦法之一是將文稿拆成幾個部分, 分開編排; 或者使用 emTEX 系統之 htex386 程式排版。

```
! I can't find file `...'
Please type another input file name =
```

執行 latex 時須鍵入文稿檔名; 或者文稿內所引用之巨集檔案可能集中於另一檔案內。若文稿檔名鍵入錯誤, 或者置於某檔案夾中但 latex 無法自動找到, 此時即產生以上之訊息。解決的辦法是輸入正確檔名。萬一硬碟中無任何 .tex 檔可以輸入, 可試著輸入 null, 此爲一虛擬檔案, latex 應會繼續排版動作。

```
! Fatal format error; I'm stymied
```

此一訊息通常出現於新裝設 cwTEX 系統或更新版本之後。latex 執行時須從硬碟中讀取某些格式檔案 (format file)。若格式檔案的版本不對, 或者設定之檔案夾錯誤, 即出現上述訊息。請再確認安裝是否正確。

18.3 確認錯誤來源

偵測錯誤的第一步是確認錯誤之所在。LaTeX 文稿是以 \end{document} 結尾, 文稿任何地方出現此一訊息, 其後的文字或指令都不再處理。因爲 latex 所提供的錯誤訊息都會指出錯誤發生的那一行, 因此我們可以在該行之後兩三行處下 \end{document} 指令, 重新排版, 看看錯誤訊息是否仍與原來的相同。訊息若相同, 表示錯誤是發生於該行之前。

接著, 我們由此行往上尋找錯誤。若來回數遍仍無所獲, 可以試著在此行之前十數行處再下 \end{document} 指令。重新排版之後, 若錯誤訊息已消失, 表示錯誤是發行於兩個 \end{document} 指令當中。以此方法, 逐步縮小範圍, 即不難確認錯誤所在。

19 取用軟體

本書所介紹之軟體全部置於光碟中, 安裝方法請見第 4 章。請注意, 部分軟體爲 shareware, 你可以試用一段時間; 若欲繼續使用須付費註册。大部分軟體或巨集套件仍繼續發展中, 隨時可能更新。爲了方便取得新版本, cwTEX 設有一網頁, 存放與光碟內容相同或更新的版本, 網址爲:

```
http://ceiba.cc.ntu.edu.tw/tmwu
```

除了存放更新軟體之外, cwTEX 網站另闢有討論區, 供使用者交換心得。由此網站可連結到其他相關網站, 例如 CTAN, 或中文 TEX 之 BBS 討論站:

```
telnet: 140.112.18.32
```

國際間收集 TEX 排版系統軟體最完整的網站是 CTAN。國內至少有兩個網站複製了 CTAN 全部或部分資料:

```
ftp://ftp.ccu.edu.tw/pub1/tex
ftp://ftp1.sinica.edu.tw/pub2/tex
```

欲取用新版軟體, 可至上述網站下載。

進入 CTAN 網站之後, 根目錄通常稱爲 \tex-archive 或者 \tex, 其下區分多個檔案夾, 分門別類存放各種軟體與巨集套件。

- `systems` 收集各種作業系統之 TEX 程式。其中, MiKTEX 與 fpTEX 置於 \win32 檔案夾下。
- `support` 存放輔助軟體與工具程式, Ghostscript 軟體即置於其中。
- `macros` 存放各式巨集套件。LATEX 巨集套件置於 \latex\contrib 檔案夾內, 其下分 \supported 與 \other 兩檔案夾。
- `fonts` 收集各式字體與工具程式。

爲了節省空間, CTAN 所儲存之檔案通常是壓縮檔, 下載之後須先行解壓縮, 方能使用。

如果是巨集套件, 較簡單的格式是一個 .sty 檔案, 使用說明通常直接附於其內。巨集套件 url.sty 就是一個例子。檔案應下載於個人檔案夾:

```
c:\localtexmf\tex\latex
```

爲方便管理, 我們也可以在 \latex 之下再建立一次檔案夾, 如 \latex\mytex, 專間存放個人收集之巨集套件。複製完成之後, 須更新檔案資料庫, MiKTEX 使用者請執行:

```
c:\>initexmf -u
```

fpTEX 使用者請執行:

```
c:\>mktexlsr
```

有些巨集套件是儲存爲兩個檔案。以 labels 巨集套件爲例, CTAN 存放的是 labels.ins 與 labels.dtx。其內同時存放巨集套件程式與說明檔。請下載這兩個檔案至 c:\xtemp 檔案夾內, 執行:

```
c:\xtemp>latex labels.ins
```

硬碟內即產生巨集套件 labels.sty。接著執行:

```
c:\xtemp>latex labels.dtx
```

硬碟內即產生 labels.dvi 此爲巨集套件之使用說明。將 labels.sty 複製於適當檔案夾內, 更新檔案資料庫之後即可開始使用。

除了以上兩種型式之外, 有些較複雜的巨集套件使用特別的安裝方法, PSTricks 與 titlesec 就是兩個例子。安裝之前, 請先閱讀其說明檔。目前, cwTEX 系統並未置於 CTAN 內, 要取得新版本, 請至 cwTEX 網頁或者以 ftp 方式至下列網址下載:

```
ftp://140.112.150.253/cwTeX
ftp://ftp1.sinica.edu.tw
ftp://192.192.110.1/cwTeX
```

參考書目

排版一般的書籍文稿, 本書所介紹之指令應已足夠。所有的巨集套件都附有說明檔, 有些甚爲詳細, 有些較簡單。底下將列出一些我們認爲特別有用的說明檔案。若欲深入探究 TEX 與 LATEX 的奇妙世界, 請進一步參考下列書籍及其中所引書目。

American Mathematical Society (1996) "$\mathcal{AMS}$-LATEX Version 1.2 User's Guide".
[附於 $\mathcal{AMS}$-LATEX 巨集套件內之說明檔, 介紹詳盡, 排版一流。]

Bringhurst, Robert (1996) *The Elements of Typographic Style*, 2nd ed., Point Roberts: Hartley & Marks Publishers.
[字體設計家 Hermann Zapf 認爲「所有桌上排版者都應閱讀此書」。雖然本書內容是以英文排版爲主。不過, 觸類旁通, 書內所討論的原則對於中文排版仍有相當價值。]

Goossens, Michel, Mittelbach, Frank, and Samarin, Alexander (1994) *The LATEX Companion*, New York: Addison Wesley.
[新版 LATEX 工作小組所寫的參考手册, 介紹許多實用的巨集套件以及新版 LATEX 之功能。]

Goossens, Michel and Rahtz, Sebastian (1999) *The LATEX Web Companion*, New York: Addison Wesley.
[介紹如何將 LATEX 文件轉換爲網路文件。]

Goossens, Michel, Rahtz, Sebastian, and Mittelbach, Frank (1997) *The LATEX Graphics Companion*, New York: Addison Wesley.
[新版 LATEX 工作小組所寫之參考手册, 主要介紹圖形巨集套件, 特別是如何使用 PostScript 語言。]

Knuth, Donald E. (1999) *Digital Typography*, California: Leland Stanford Junior University.
[Knuth 細數 TEX 與 METAFONT 系統發展過程的點點滴滴, 生動有趣。]

Knuth, Donald E. (1990) *TEX book*, New York: Addison Wesley.
[TEX 的起源, 進階使用者之聖經。]

Kopka, Helmut. and Daly, Patrick W. (1995) *A Guide to* LATEX2ε, 2nd ed., New York: Addison Wesley.
[此書詳盡地介紹 LATEX 指令, 可以替代 Lamport (1994) 之使用手册。]

Lamport, Leslie (1994) *LATEX: User's Guide and Reference Manual*, 2nd ed., New York: Addison Wesley.
[LATEX 巨集套件原創者所寫之使用手册。]

Oostrum, Piet van (1996) "Page Layout in LATEX".
[fancyhdr 巨集套件說明檔, 詳細介紹排版頁眉/頁足之指令。]

Reckdahl, Keith (1997) "Using Imported Graphics in LATEX2ε", 2nd ed.
[如何引入 Postscript 圖形於 LATEX 文稿內, 詳盡而清楚。另外, 還說明如何使用浮動版面 (floats) 指令環境。可自 CTAN 下載。]

Taylor, Philip (1995) "Book Design for TEX Users"
[說明排版書籍的基本原則, 精簡扼要。分兩期載於 *GUST* 雜誌, 可自網站下載。已附於 `\texmf\cwtex\doc` 檔案夾內, 檔名爲 `Taylor-1.ps` 與 `Taylor-2.ps`。]

索引