Gnuplot 導讀

中央研究院計算中心
ASPAC 計劃
E-mail: aspac@beta.wsl.sinica.edu.tw
技術報告： 95006

February 7, 1995
Version 1.0

ASPAC 計畫版權聲明

ASPAC（Academia Sinica PACkage）是中央研究院計算中心關於“軟體工具使用”(Software Tools) 及“問題解決”(Problem Solving) 的計畫。在這計畫下所發展之軟體及文件都屬於中央研究院計算中心所有。所有正式公開之電子形式資料（包括軟體及文件），在滿足下列軟體及文件使用權利說明下，都可免費取得及自由使用。

軟體及文件使用權利說明如下：

1. 軟體的使用權利：
   將沿用美國 FSF（Free Software Foundation）1991年6月第二版的 GNU General Public License。

2. 文件的使用權利：
   文件可以自由拷貝及引用，但不得藉以圖利。除非必要手續費的收取。

因一般的 PoorMan LaTeX中無法使用中文當成章節名稱,
所以本文中暫以英文來表示章節名稱.

Contents

1 Introduction **1**

1.1 What is gnuplot 1

1.2 Scientific Plotting and gnuplot 1

1.3 About this Touring Guide 2

2 Observing Numerical Data with gnuplot **5**

2.1 Preparing Input Data for gnuplot 5

2.2 Starting gnuplot 6

2.3 Plotting Data Files 7

2.4 Customizing Your Plot 8

2.5 Comparing the Data with Models 9

3 Producing Graphics for Your Documents **11**

3.1 Printing gnuplot Figures on Paper 11

3.1.1 Printing in UNIX 12

3.1.2 Printing in DOS 12

3.2 Placing gnuplot Figures in Your Documents 13

3.3 Editing gnuplot Figures to Fit Your Own Need 14

4 Miscellaneous **17**

4.1 Gnuplot as a Graphics Viewer of Your Program 17

4.2 Using gnuplot Anywhere 18

4.2.1 Viewing Graphics on Various Terminals 18

4.2.2 Running gnuplot on Various Computers 19

4.3 Saving and Reloading Your Work 19

4.4 Finding Information for Help 20

4.4.1 On-line Help 20

4.4.2 Useful Documents 21

4.4.3 Internet gnuplot Resource 22

Chapter 1

Introduction

1.1 What is gnuplot

gnuplot 是一個命令導向的交談式的繪圖軟體，在 1986 年由 Colin Kelley 和 Thomas Williams 發展而成[1] 。gnuplot 的功能就是把數值資料和數學函數轉換成容易觀察的平面或立體的圖形，幫助研究者找出資料背後所隱藏的意義。因此 gnuplot 並不是一般常見的美工繪圖軟體，也不是從事數學運算的程式，它最適合的是在科學研究的過程中，代替研究人員完成數據資料繪製與理論模型比對等機械化的工作，來加速研究的進行。

1.2 Scientific Plotting and gnuplot

科學研究大致上是有一定程序的，通常是先收集資料，經過一些計算

[1] 取得 gnuplot 的方法請參考 ASPAC 文件 "gnuplot 使用手冊" 的第二章，此文件可在

ftp://phi.sinica.edu.tw/pub/aspac/doc/94/94002.ps

找到。

得到較容易了解的數據，再嘗試觀察出其中的一些法則。而觀察大量數據最有效的方法，自然是透過繪圖的方式。傳統的繪圖工具是方格紙、對數紙和直尺、鉛筆等，用人工的方式描繪，其過程曠日費時且容易犯錯， gnuplot 就是為了解決這個問題而設計出來的自動化繪圖工具。它可以讓使用者很容易地讀入外部的資料檔案，立即在螢幕上看到圖形，並且選擇和修改圖形的畫法，明顯表現出資料的特性。

在初步看過資料的圖形表示後，研究者可能由此獲得一些啟發，例如懷疑資料曲線／曲面可能是某類函數所構成；或者研究者原本就推論其資料應符合某個模型，這時候必須把資料的圖形和模型所產生的圖形畫在一起做比對的工作。 gnuplot 繪製數學函數的功能就在這裡派上了用場，配合同時繪製數個曲線／曲面並標色的能力，可以很容易比對不同的資料和模型。在比對曲線時還可以加上誤差的範圍，立刻看出模型是否真正準確地預測了資料的行為。在這個過程中，研究者不斷地修正其模型或資料，用 gnuplot 繪圖加以驗證，如此重複進行，直到研究告一段落。

到了這個階段，研究者常會把這些工作的過程和結果記錄下來，他可以把 gnuplot 的圖形直接在各種印表機上印出來，輯入工作記錄之中。如果要將之撰寫成論文，可以把圖形存成所用的論文排版軟體（例如 LaTeX）所能接受的圖形格式檔案 （例如 Postscript 格式），置入論文之中。也可能他覺得 gnuplot 畫的圖還不夠好，想要自己加以修改，例如用箭頭和註解強調圖中某個特異之處，那麼可以把圖形存成一些美工軟體能夠接受的檔案格式，加以潤飾後再放入文件之中。

1.3 About this Touring Guide

這本導讀撰寫的目的，是希望讓沒有用過 gnuplot 的使用者在讀過這份簡要的導讀後，能了解到 gnuplot 在設計與使用上的精神，或者至少可以很快地知道 gnuplot 是不是他工作上所需要的工具。如果您已經懂得 gnuplot 的使用方法，或看過以後決定以 gnuplot 作為您的工作伙伴，也希望您可以從這本導讀中獲得一些想法上的觸發，讓您在使

用 gnuplot 時更能夠得心應手。

在決定這本導讀的題材時，曾經有許多的考量。它的性質有點接近常見的使用文件中的教學指引 (Tutortial Guide)，但我們不希望像教學指引一樣把不同的題材割裂開來，秀幾個範例，然後使用者還要自己去拼湊出整體的概念。我們的理想是希望能用一個接近真實情況的完整故事串起所有的題材，換句話說，我們營造一個虛擬的情境，讀者在此由實際上的需要推導出工具所應具備的功能。雖然讀者不是這個工具的製造者，但工具的觀念和動機卻是在這個過程中由讀者所引發的，也就比較能了解工具和人、工作的互動關係。由這樣的觀點應該可以保持讀者在學習過程中的主動性，不至於迷失在五花八門的功能中，避免讀了一大堆招式卻不知道如何發揮的空虛感。這也是這本小冊子決定以「導讀」 (Touring Guide) 為名的原因。

這樣的題材無可避免地會有些不盡完備的地方，最大的影響就是不可能涵蓋所有的主題，在這本導讀中只用了 gnuplot 的 2D 繪圖，如 3D 繪圖等功能並未包括在此。但這應該是值得付出的代價，因為使用工具處理事情的目的就是解決問題，原本就不講究要完全了解工具的每一個部分。如果要對 gnuplot 的功能有全盤的了解，可以另外參考 ASPAC 文件 “gnuplot 使用手冊” （取得方法見本文第 1 頁註 1）。事實上，這本導讀有多處必須參考 “gnuplot 使用手冊”，我們強烈建議您盡快拿到這本手冊參照使用。

這本導讀的內容，是以一個假想的科學研究過程中所導引出的繪圖方面的種種需要和問題為主線，說明 gnuplot 所能提供的解決方法和功能。第二章敘述在探索數值資料意義的過程中，所產生的各種繪圖上的不同需求，和 gnuplot 的繪圖環境所能提供的各種解決的方法，包括資料的輸入、圖形的重複調整與比對等。第三章敘述在研究告一段落後，如何保存和使用 gnuplot 所畫的圖，包括列印、置入文件、另加修飾等。第四章把一些和科學繪圖沒有直接關係但可以明顯改善工作環境或流程的主題收集在此，包括在 C 程式中使用 gnuplot 繪圖、在不同環境中使用 gnuplot，線上求助等。

Chapter 2

Observing Numerical Data with gnuplot

2.1 Preparing Input Data for gnuplot

gnuplot 所要讀入的資料檔案格式的要求很少，最簡單的要求是只要在檔案中有行 (column) 的結構，可用以分辨不同欄位 (field) 就行。以下就是本章的資料範例：假設某次實驗後收集得實驗資料，是一組可用以繪圖驗證的數據，這些數據存在一個檔名為 "data" 的檔案中，包括有三個欄位 – 依序是 X 值、 Y 值、和誤差值。

資料檔 "data"

```
-4.000000  25.071400 5.535340
-3.416670  26.609600 5.186600
-2.833330  32.507702 4.563950
-2.250000  42.865200 5.719100
-1.666670  68.307297 4.863300
-1.083330 113.533997 4.761980
-0.500000 161.464996 4.625440
 0.083333 181.237000 5.792480
 ...
```

2.2 Starting gnuplot

啟動 gnuplot 相當簡單，例如在 UNIX 系統上，只要下 gnuplot 命令，便會出現 gnuplot 的啟始畫面：

```
%
% gnuplot


 G N U P L O T
 unix version 3.5
 patchlevel 3.50.1.17, 27 Aug 93
 last modified Fri Aug 27 05:21:33 GMT 1993

 Copyright(C) 1986 - 1993   Colin Kelley, Thomas Williams

 Send comments and requests for help to info-gnuplot@dartmouth.edu
 Send bugs, suggestions and mods to bug-gnuplot@dartmouth.edu

Terminal type set to 'x11'
gnuplot> _
```

因為是在一部有 X 繪圖能力的工作站上執行，所以在這個螢幕畫面上顯示的終端機形態為 x11 (X 終端機) [1]。除此之外，在其他作業系統上，例如 DOS, VAX/VMS 等，也與此情形類似。

若要結束 gnuplot 的工作，可用 quit 命令：

```
gnuplot> quit

% _
```

[1] 有關終端機繪圖的設定，請參考 4.2.1 節。

2.3 Plotting Data Files

基本上， gnuplot 的繪圖命令只有二個 — plot 和 splot ，分別用來繪製二維和三維的圖形。我們可以立刻用 plot 命令描出 "data" 檔中的資料點：

```
gnuplot> plot "data"
```

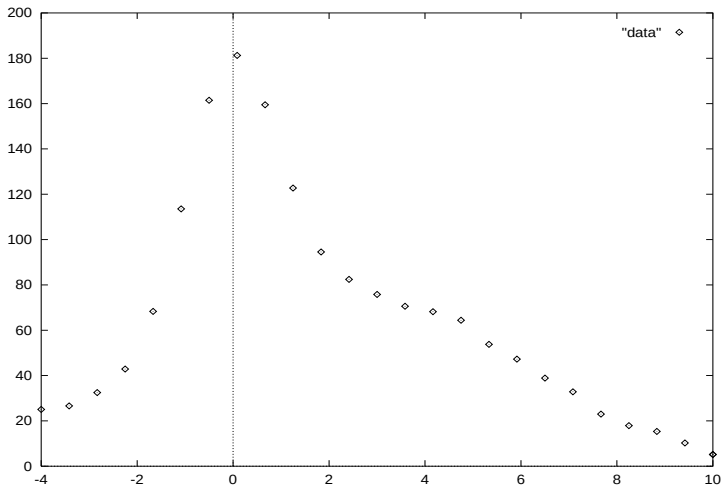

Figure 2.1: Plotting Points

由於上面命令中並未指定讀入資料欄位的方法， gnuplot 預設以第一行 (column) 為橫軸座標值，第二行為縱軸座標值來描點。

科學用程式有時為了方便和易讀，而同時將多行資料存在同一個檔案中，並加上註解讓其他人能知道這些資料的意義， gnuplot 會自動跳過以 # 開頭的每一行註解，並且可以用類似 C 語言中 scanf() 函數所用的格式字串 (format string) 來分辨一列中的不同欄位，和自行指定橫軸、縱軸、誤差範圍等資料所用的欄位。詳細的命令格式可參考 ASPAC 文件 "gnuplot 使用手冊" 的第 5 章第 3 節（取得方法見本文第 1 頁註 1）。

2.4 Customizing Your Plot

plot 預設只作描點的工作，但描點可能不足以讓人看出資料的意義。用下面的命令可以改變 plot 命令的繪圖參數，把圖上各點連接起來，觀察整個曲線的行為：

```
gnuplot> plot "data" with linespoints
```

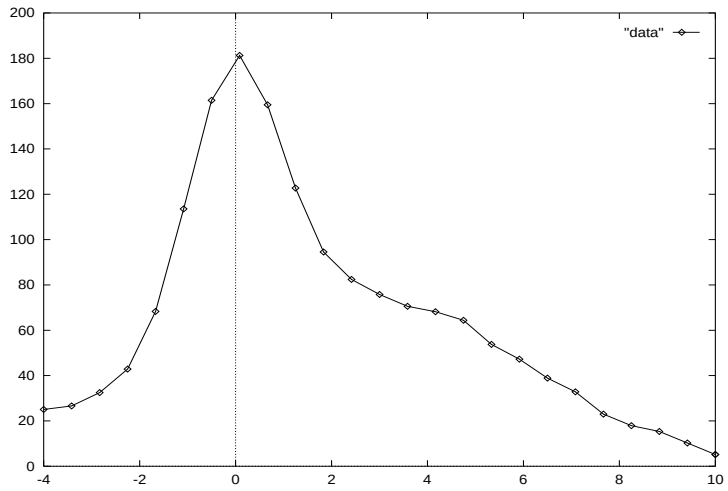

Figure 2.2: Plotting Points with Lines and Points

由此圖可以明顯看出，資料中有一個尖峰 (peak)，但在它的右側還有一個較不明顯的凸起。

針對不同意義的資料，gnuplot 提供了 9 種不同的圖樣 (styles) 供使用者選擇，包括描點、連線、長條圖、階梯圖、誤差圖等。詳細的項目和使用方法請參考 ASPAC 文件 "gnuplot 使用手冊" 的第 5 章第 3, 4 節（取得方法見本文第 1 頁註 1）。

2.5 Comparing the Data with Models

假設這條曲線應該由二個高斯函數和一個常數所組成，

$$G(x) = G_1(x) + G_2(x) + c$$

所算出最接近實驗資料的模型函數是：

$$\begin{aligned} G_1(x) &= 135.64e^{-0.5x^2} \\ G_2(x) &= 57.5824e^{-0.05556(x-3)^2} \\ c &= 13.38 \end{aligned}$$

再用 gnuplot 同時把模型函數 $G(x), G_1(x), G_2(x)$ 輸入並畫圖觀察，為了解模型函數是否準確預測了實驗結果，可用誤差圖的方式畫出實驗資料加以比對：

```
gnuplot> G1(x)=135.64*exp(-0.5*x**2)
gnuplot> G2(x)=57.5824*exp(-0.05556*(x-3)**2)
gnuplot> G(x)=G1(x)+G2(x)+13.38
gnuplot> plot "data" using 1:2:3 with errorbars, \
"data" with lines, G(x), G1(x), G2(x)
```

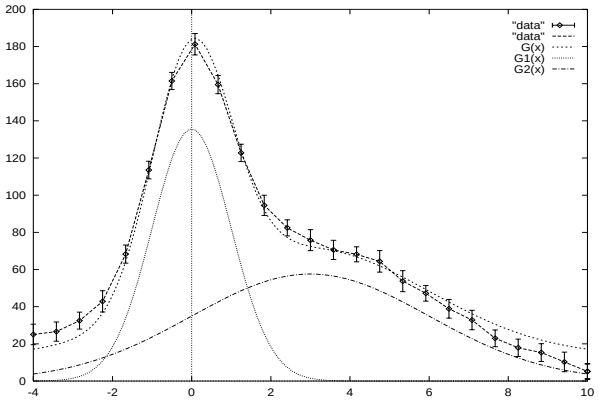

Figure 2.3: Plotting Data, Errors and the Model

可以發現模型曲線在圖形的中央部分雖然足夠準確，但在兩側部分卻出現明顯的誤差，這表示模型很可能需要修正，才能正確預測實際的狀況。假設新的模型是用一個一次項來修正舊的，

$$G(x) = G_1(x) + G_2(x) + ax + b$$

重新計算後，得到新的逼近函數：

$$\begin{aligned} G_1(x) &= 125.268e^{-0.5x^2} \\ G_2(x) &= 62.4193e^{-0.05556(x-3)^2} \\ a &= -1.41 \\ b &= 16.2 \end{aligned}$$

並仿照前面的方式重新繪圖比對，得到新的模型和資料比對圖：

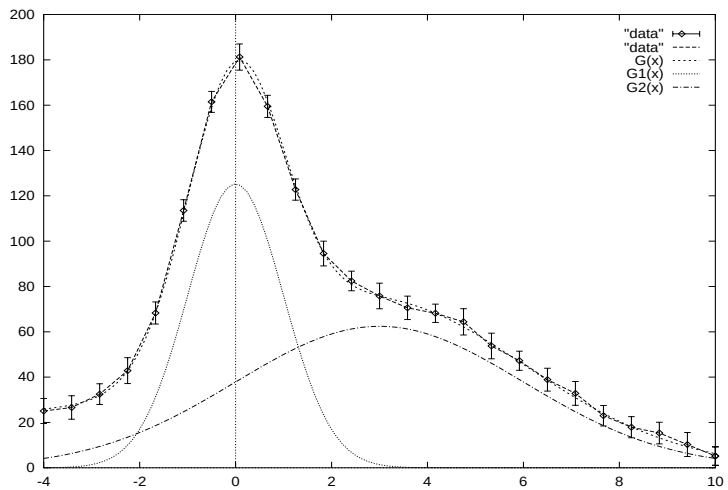

Figure 2.4: Plotting Data, Errors and the New Model

於是發現新的模型可以完全吻合所有的實驗資料，等於替這個修正的模型提出了有力的證據。在研究告一段落後，必須著手整理出研究報告或寫成論文， gnuplot 在前面所畫的圖除了幫助研究的進行，還能以高品質輸出，很方便地輯入文件之中。下一章將專門討論這個主題。

Chapter 3

Producing Graphics for Your Documents

3.1 Printing gnuplot Figures on Paper

gnuplot 可以把圖形輸出到各種印表機、繪圖機，包括 Epson, NEC, HP 等印表機和 Tektronix 繪圖機等輸出裝置，或是任何 Postscript 印表機，作為簡單的工作記錄，或用作和他人討論的材料。方法是

1. 設定輸出格式；
2. 設定輸出檔案；
3. 重繪圖形至檔案；
4. 由作業系統將檔案印出。

其中輸出格式因印表機而有所不同，列印的命令也隨作業系統而有差異，下面以 UNIX 和 DOS 二作業系統為例說明。

3.1.1 Printing in UNIX

在 UNIX 作業系統上，假設要將 gnuplot 的圖在一台名為 sparc 的 Postscript 印表機上印出，可用下面的命令達成：

```
gnuplot> plot sin(x)
gnuplot> set terminal postscript
  Terminal type set to 'postscript'
  Options are  'landscape  . . . .
gnuplot> set output 'sin.ps'
gnuplot> replot
gnuplot> quit

% lp -d sparc sin.ps
```

因為 UNIX 作業系統提供有 pipe 功能，故可將輸出直接傳給 lp 命令印在指定的印表機只要將 output 參數改變如下即可：

```
gnuplot> set output '|lp -d sparc'
```

那麼下過 replot 命令後就會直接印出圖形。

3.1.2 Printing in DOS

在 DOS 作業系統上執行 gnuplot，無法如同在 UNIX 上一樣直接印出，必須先把圖形存成檔案再行列印，例如要在 Epson 印表機印出 DOS 上的 gnuplot 圖形：

```
gnuplot> plot sin(x)
gnuplot> set terminal epson_180dpi
  Terminal type set to 'epson_180dpi'
gnuplot> set output 'sin.ep'
gnuplot> replot
gnuplot> quit

C:\gnuplot> copy sin.ep /b lpt1:
```

讀者可以發現這裡不同於前一小節之處，只有終端機格式設定和作業系統列印命令而已。

3.2 Placing gnuplot Figures in Your Documents

在論文或報告等較正式的文件中，不能像前一節那樣，把圖印在一張紙上夾入文件就好，而必須把圖形和文字合併成一份整體的文件。如今已經有許多文書排版軟體可以完成圖文整合的工作，例如寫作科學論文常用的 LaTeX。雖然這些軟體所能接受的圖形檔案格式並不統一，gnuplot 還是能將所畫的圖存成許多種常見的檔案格式，例如 postscript, LaTeXpicture, pbm 等[1]，直接置入排版軟體的文件之中。以您眼前這本導讀為例，圖 2.1 是由 gnuplot 畫出，存在 "PlotDots.ps" 檔案中，格式為 postscript，存檔方法和 3.1.1 節一樣。在本文的 LaTeX原始檔案中，是用 psfig 命令置入此圖形檔：

```
\documentstyle[psfig]{article}
...
\begin{figure}[h]
\psfig{figure=PlotDots.ps,width=4in,height=2.5in,angle=-90}
\caption{Plotting Points}
\end{figure}
...
```

有關 UNIX 作業系統上的圖文整合方面，請參考 ASPAC 文件 "X 環境下圖文整合"[2]。

---

[1] 關於 gnuplot 所能輸出的圖形檔案格式，請參考 ASPAC 文件 "gnuplot 使用手冊" 6.6.2 節（取得方法見本文第 1 頁註 1）。

[2] 此文件可在

ftp://phi.sinica.edu.tw/pub/aspac/doc/94/94001.ps

找到。

3.3 Editing gnuplot Figures to Fit Your Own Need

gnuplot 可以在圖中加上一些說明文字，使圖形的意義更加清楚，例如座標軸的名稱、刻度以及圖的標題等[3]。但在作為插圖之用時，有時還必須對圖形再作修改，例如強調圖中某個部分並加以說明，針對這方面的需要， gnuplot 能把圖形存成其他美工軟體可以接受的格式，修改完成後再置入文件中。下面以圖 2.3 的資料-模型比對圖為例，先產生 tgif 格式圖形檔，經過 UNIX 上的 tgif[4] 軟體修改後，轉存為 postscript 檔，再仿造前一節的方法置入文件中：

```
gnuplot> plot "data"
gnuplot> set terminal tgif
  Terminal type set to 'tgif'
gnuplot> set output "model2.obj"
gnuplot> replot
```

用 tgif 軟體修改完成後見下頁圖。

除了 tgif 軟體外，許多其他的繪圖工具也可以對輸出圖形加工， gnuplot 支援多種圖形檔格式[5]，只要所用的繪圖工具能接受其中一種格式就可以，例如 fig(xfig), gpic(groff) 等，修飾完成後再選擇適當格式輸出。除了置入文件外，也可用來印製投影片或做其它用途。

---

[3] 請參考 ASPAC 文件 “gnuplot 使用手冊” 的第 4 章（取得方法見本文第 1 頁註 1）。

[4] 有關 tgif 的使用，請參考 ASPAC 文件 “X 環境下圖文整合” 4.4.2 節（取得的方法見本文第 13 頁註 2） 。

[5] 請參考 ASPAC 文件 “gnuplot 使用手冊” 6.6.2 節（取得方法見本文第 1 頁註 1）。

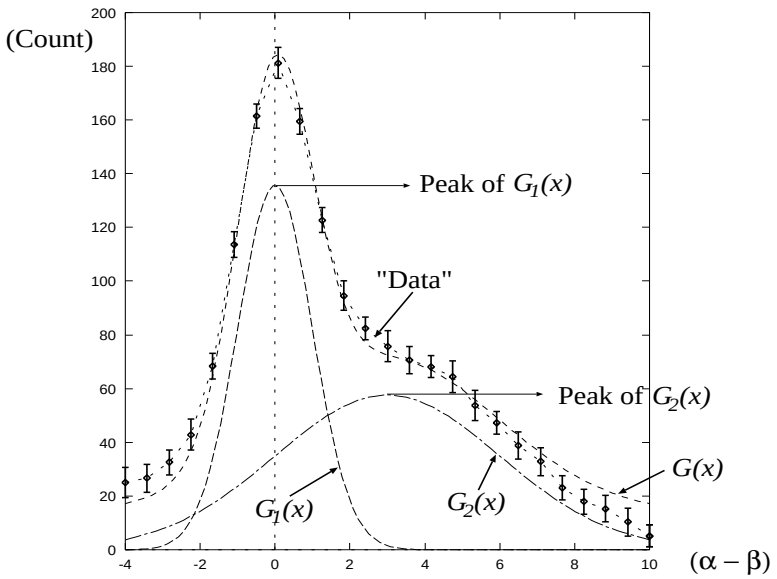

Figure 3.1: Data, Errors and the New Model Plotting, Modified by tgif

Chapter 4

Miscellaneous

4.1 Gnuplot as a Graphics Viewer of Your Program

如果使用 UNIX 作業系統， gnuplot 也可以在程式中直接擔任畫圖的工作，雖然 gnuplot 通常是以交談 (interactive) 方式來使用，但只要先決定好繪圖參數，也可以在程式中透過程序間通訊 (IPC, InterProcess Communication) 將繪圖資料和繪圖命令傳給 gnuplot ，在程式執行時就畫出圖形[1] 。

在這種情況下使用 gnuplot 畫圖，其意義不同於第 2 章的使用方式。第 2 章的情況是已經有處理完畢的資料要分析，但不清楚要用何種方式繪圖觀察，所以一邊畫圖，一邊根據圖形調整資料的表現方式，直到找出資料的意義。在程式中呼叫 gnuplot 時，必須已經清楚資料表現的方式， gnuplot 只是單純地將數值資料轉成指定的圖形，也可能設計成不必等到程式結束才繪圖，而是計算到一定階段就畫圖檢查結果。這樣設計的意義和自行撰寫程式來繪圖是一樣的，但可減少自行撰寫的時間和精力的浪費。

---

[1] 詳細的方法請參考 ASPAC 文件 “gnuplot 使用手冊” 的附錄 B （取得方法見第 1 頁註 1）。

4.2 Using gnuplot Anywhere

雖然前面所討論的材料基本上以 UNIX 工作站環境為主， gnuplot 在很多常見的電腦上也可以執行，它也支援一些常見的圖形終端機標準，使用者可以針對不同的工作環境，調整使用 gnuplot 的方式。

4.2.1 Viewing Graphics on Various Terminals

X Terminals

在本文 2.2 節的範例中，因為所用終端機具有 X 視窗繪圖能力， gnuplot 會在啟動時自動將輸出格式設為 x11 。 X 視窗可顯示最佳的圖形品質，但需要快速的網路傳輸，所以通常必須在辦公室、實驗室等有區域網路的地方才能流暢地使用。大部分工作站等級以上的電腦和專用的 X 終端機都可使用 X 視窗繪圖。

Tektronix Terminal Emulation

但如果你的電腦不在區域網路上，例如在家中只能透過數據機 (modem) 連線至辦公室的工作站，為了看到 gnuplot 的繪圖，就必須使用能模擬圖形終端機的撥接 (dial-up) 軟體，例如 MS-DOS Kermit[2] 可模擬成 Tektronix 繪圖終端機，只要在 gnuplot 中用下面命令設定 terminal 參數就可以在家中看到圖形：

```
gnuplot> set terminal 'kc_tek40xx'
```

其中的 'c' 是彩色 (color) 的意思。

---

[2] 有關 MS-DOS Kermit 模擬 Tektronix 繪圖終端機的使用，請參考 ASPAC 文件 "MS-DOS Kermit 使用手冊 " 3.1.4 節，此文件可在

ftp://phi.sinica.edu.tw/pub/aspac/doc/94/94008.ps

找到。

當然您也不一定要用 MS-DOS Kermit 才能看到圖形，其他可以模擬 Tektronix 繪圖的通訊軟體通常也行，例如 NCSA Telnet ， 但所用的 terminal 設定會有些不同，這些設定可以參考第 1 頁註 1 所提 ASPAC 文件 “gnuplot 使用手冊” 的 6.2.1 節。

4.2.2 Running gnuplot on Various Computers

gnuplot 在目前許多常見的電腦上都可執行，除了使用 UNIX 作業系統的電腦外，也可以在 VAX/VMS, MS-DOS[3] , OS/2[3], MS-Windows[3], Macintosh[4] 等電腦作業系統使用。

在前面的討論中，我們都是以 UNIX 作業系統為背景，在 UNIX 上使用 gnuplot ，可以很方便配合各種工具來完成工作。但並非所有地方都能使用工作站等級之上的電腦，如果只有較便宜的個人電腦，那麼仍然可以用個人電腦版本的 gnuplot 來作事。

4.3 Saving and Reloading Your Work

gnuplot 可以讓您把一些命令存在檔案中，再整批載入 gnuplot 執行，就如同您一一鍵入這個檔案中的每一行到 gnuplot 。例如載入一個檔案 "script.gp" ，可用 load 命令達成：

```
gnuplot> load "script.gp"
```

透過這項載入命令功能，您可以把在 gnuplot 中的工作狀態，包括繪圖環境參數用另一個命令 save 存在檔案中，重新載入此檔案就能回到和

---

[3] 獲得這些作業系統上的 gnuplot 可執行檔的方法，請參考 ASPAC 文件 “gnuplot 使用手冊” 2.1 節（取得方法見第 1 頁註 1）。

[4] Macintosh 版的 gnuplot 可執行檔可以在 imaux.mgt.ncu.edu.tw 的 /pub/mac/umich/graphics/graphicsutil/gnuplot3.5.cpt.hqx透過 Anonymous FTP 得到。

當時一樣的狀態繼續工作。例如存下工作狀態到檔案 "script.gp" 的命令是：

```
gnuplot> save "script.gp"
```

4.4 Finding Information for Help

4.4.1 On-line Help

gnuplot 提供方便的線上求助 (On-line Help) 功能，讓使用者在有困難時可以立刻查閱，它的線上求助是一個樹狀的結構，用 help 命令來找出所要的使用說明。例如僅鍵入 help 可以列出所有的項目：

```
gnuplot> help
 GNUPLOT is a command-driven interactive function plotting
 program.
 For help on any topic, type `help` followed by the name
 of the topic.
 The new GNUPLOT user should begin by reading the `introduction`
 topic (type `help introduction`) and about the `plot` command
 (type `help plot`).  Additional help can be obtained from the
 USENET newsgroup comp.graphics.gnuplot.

Help topics available:
      autoscale      binary-data    bugs           cd
      clear          comments       copyright      environment
      exit           expressions    help           introduction
      line-editing   load           pause          plot
      print          pwd            quit           replot
      reread         save           set            shell
      show           splot          startup        substitution
      userdefined

Help topic: _
```

如果在 Help topic: 之後鍵入 plot，就可以看到有關 plot 命令的說明， plot 的說明末尾還有列出更細的說明項目：

```
. . . . .
Subtopics available for plot:
      data-file      datafile       errorbars      parametric
      ranges         style          title          with

Subtopic of plot:
```

在 `Subtopic of plot:` 之後可再鍵入項目繼續查詢。

4.4.2 Useful Documents

下面是一些很有用的 gnuplot 文件：

- Gnuplot Reference Manual
  標準的 gnuplot 參考手冊，附在 gnuplot 原始程式裡，裡面詳細描述了所有命令用法，適合在使用時隨時查閱參考。

- Gnuplot User Guide
  一本簡短扼要的使用手冊（約 17 頁），介紹 gnuplot 中最常用的功能，附以簡單的範例說明，對於沒有英文閱讀上問題者，此書也是一本頗不錯的入們書籍。您可在

  ftp://picard.tamu.edu/pub/gnuplot

  中找到它的 LaTeX原始檔案。

- Gnuplot Demo
  這是一個附在 gnuplot 原始程式裡的範例集，展示了 gnuplot 的大部分功能。它並不是一份文件，而是一組放在原始程式目錄下的 "demo" 目錄內的 gnuplot 批次檔，讓使用者可以自行載入實驗 gnuplot 的各項功能。如果不想自己動手，在

  http://www.cs.dartmouth.edu/gnuplot/gnuplot_demo.html

  可以看到圖文並陳的原始命令和執行結果。

- Gnuplot FAQ
  一般常見的 gnuplot 問題集，內容多來自 Usenet 上的討論，並定期在其上更新，除此之外，也可以透過 Anonymous FTP

  ```
  ftp://rtfm.mit.edu/pub/usenet/news.answers/graphics/gnuplot-faq
  ```

  或 WWW

  ```
  http://fg20.rz.uni-karlsruhe.de/~ig25/gnuplot-faq.html
  ```

  等不同方式獲得。

4.4.3 Internet gnuplot Resource

- Usenet
  在 Usenet 上有一個專門討論 gnuplot 的討論區 (newsgroup) ：

  ```
  comp.graphics.gnuplot
  ```

  前面提到的 Gnuplot FAQ 就是收集這裡的討論，並且定期在此公佈更新的版本。

- WWW Info Service
  這是 gnuplot 的 WWW home page ：

  ```
  http://www.cs.dartmouth.edu/gnuplot_info.html
  ```

  前面所提到的文件多可以在這裡找到，而且多已做成 html 格式，使查詢更加快速方便。

  ```
  http://www.cs.dartmouth.edu/gnuplot_info.html
  ```