實驗物理講義

中央大學物理學系

September 28, 2009

編者序

第一學期的講義內容由林致翰與林昱廷根據汪治平老師規劃的課程大綱進行編寫。其中部份機械加工的章節，GUI 介面的主要內容與函數繪圖中非線性映射的範例由林昱廷完成，其餘所有章節則由林致翰編寫完成。

機械加工的部份以助教帶領同學實機操作爲主，這份講義僅列出安全規則與一些小訣竅，比較深入的內容請同學參考指定書目的內容來學習。程式寫作課程則考量到時間上的限制與軟體是否容易取得，這門課選擇了 Scilab 來展示電腦如何幫助我們理解物理問題，筆者相信若同學們能充分掌握這份講義中的程式範例與習題，未來在學習更爲抽象的物理概念時也能透過電腦的幫助來加深你的思考深度。

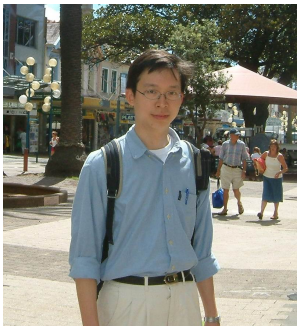
實驗課中安排了一個樹脂翻模的課程，或許這是一般物理系學生絕少會碰觸到的主題；也許這樣的技術所能提供的精度與效率並不能好好地用在實驗室上，但它代表實驗物理的一種精神：自製儀器時中不可避免的反覆檢討，逐漸將成品修改到最佳狀態的一種過程。

這本講義的另外一個特色便是習題衆多，當然預設上並不要求同學們把所有的題目都做完，以三學分的實驗課而言能完成 50% 的內容就算是及格了。設計大量題目的用意在於讓不夠過癮人有更多練習寫程式的機會，眞正的學習應該是 self-learning 才對，而非侷限在上課時台上講解的內容。

September, 2009　林致翰

授課教師

汪治平　Jyhpyng Wang
P. O. Box 23-166, Taipei, Taiwan.
jwang@ltl.iams.sinica.edu.tw



朱旭新　Hsu-Hsin Chu
R604, Department of Physics, NCU
hhchu@ncu.edu.tw

助教

林致翰　Chih-Han Lin
Dept. of Electronics Engineerig, NTU
clin@ltl.iams.sinica.edu.tw

廖孜桓　D Liau
Dept. of Electronics Engineerig, NTU
dliau@ltl.iams.sinica.edu.tw

林昱廷　Yuting Lin
Dept. of Electronics Engineerig, NTU
ytlin@ltl.iams.sinica.edu.tw

黃揚智　YC Huang
Dept. of Electronics Engineerig, NTU
ychuang@ltl.iams.sinica.edu.tw

Contents

I 課程總介 **9**

II 第一學期課程 **11**

1 科學文件寫作指引 **12**

1.1 課程大綱 12
1.2 OpenOffice 12
1.3 向量圖繪製 13
1.4 ooWriter 13
1.5 LaTeX 13
1.6 Miktex 15
1.7 OOoLaTeX 16
1.8 LaTeX 數學式 16
1.9 LaTeX 參考資料 17
1.10 預習報告 17
1.11 實習報告 17
1.11.1 學習 \array、\left 和 \right 的用法 18
1.11.2 常用字元與分式 \frac 練習 18
1.11.3 \underbrace、\mbox、\big 18

2 科學數值運算: Scilab 入門 **20**

2.1 課程大綱 20
2.2 Scilab 20
2.3 Scilab 網路學習資源 20
2.3.1 Master Scilab! 20
2.3.2 Scilab Bag Of Tricks 21
2.3.3 Gilberto E. Urroz 的 lecture 21
2.3.4 中文 Scilab 資源 21
2.4 如何寫好程式 (Scilab) 21

2.5 console 21
2.6 矩陣運算 22
2.7 運算次序 23
2.8 字串 string 23
2.9 Script 24
2.10 迴圈結構 26
2.10.1 if 26
2.10.2 case 27
2.10.3 for 27
2.10.4 while 28
2.11 資料存取 28
2.11.1 file 29
2.11.2 write 29
2.11.3 read 30
2.12 I/O 程式碼範例 30
2.12.1 寫入 txt 檔 30
2.12.2 讀取 txt 檔資料 30
2.13 實習報告 31
2.13.1 資料輸入 32
2.13.2 資料讀入 32

3 夾具，折床與帶鋸機 33
3.1 課程大綱 33
3.2 機械加工安全規定 33
3.3 預習報告 34
3.4 實習報告 34

4 塑膠翻模 35
4.0.1 課程大綱 35
4.1 Garage Kit 35
4.2 準備工具 36
4.3 製作範例 37
4.4 預習報告 37
4.5 實習報告 37

5 科學數值運算：簡易數值模擬 39
5.1 課程大綱 39
5.2 Scilab 的 function 39
5.3 deff 39

5.4 function 40
5.5 global 與 local 變數 40
5.6 迭代運算 41
5.7 河內塔 41
5.8 物理問題模擬範例 43
5.9 Runge-Kutta 法 46
5.10 聯立差分法 46
5.11 預習報告 47
5.12 實習報告 48

6 銑床的用途和操作方法 60
6.1 課程大綱 60
6.2 銑床 60
6.3 預習報告 60
6.4 實習報告 60

7 科學數值運算: 圖像處理 61
7.1 課程大綱 61
7.2 SIVP toolbox 61
7.3 imread, imwrite 61
7.4 點陣圖轉換 62
7.5 Scilab pic reading code 63
7.6 曲線套適 65
7.7 預習報告 67
7.8 實習報告 68

8 車床用途和操作方法 69
8.1 課程大綱 69
8.2 車床 69
8.3 游標卡尺測量 70
8.4 預習報告 70
8.5 實習報告 70

9 科學數值運算: GUI 介面與進階操作 71
9.1 課程大綱 71
9.2 GUI 71
9.3 建立視窗 71
9.4 uicontrol 72
9.5 互動式繪圖 74
9.6 Menus 75

9.7 執行效率 76
9.8 簡單的計時程式 77
9.9 預習報告 78
9.10 實習報告 78
9.10.1 質數表 78
9.10.2 字元統計 79
9.10.3 冰雹數列 79

10 指南車與齒輪組 81
10.1 課程大綱 81
10.2 機械加工專題 81
10.3 參考資料 82
10.4 進度報告規定 82
10.5 成果發表規定 82
10.6 預習報告 82
10.7 實習報告 82

11 簡易科學數值運算: 函數繪圖 83
11.1 課程大綱 83
11.2 基本繪圖操作 83
11.3 預習報告 83
11.3.1 Spherical harmonic function 84
11.3.2 mobius strip 84
11.3.3 wave normal in biaxial crystal 86
11.4 實習報告 88
11.4.1 對數螺線 logarithmic Spiral 88
11.4.2 Fatou 集 88
11.4.3 Julia 集 88
11.4.4 Mandelbrot 集 89
11.4.5 tips 90

A 補充教材 94
A.1 有限差分法 Finite difference method 94
A.1.1 1D Time-independent Schrödinger Equation 94
A.1.2 一維量子井結構 95
A.1.3 Scilab code 96
A.1.4 思考問題 99
A.2 四階 Runge-Kutta 法的證明 101
A.2.1 推導過程 101

A.2.2 思考問題 102

Part I

課程總介

普通物理實驗

第一學期課程:

- 程式寫作與數值模擬
- 科學文件寫作
- 基礎機械加工

第二學期課程:

- 現代電子技術
- 電子焊接加工
- 電路模擬軟體

第三學期課程:

- 實用電源電路
- 進階電子技術
- 數位邏輯電路
- 信號處理與感測元件

第四學期課程:

- 數據分析與套適
- 實驗室儀控
- 資料擷取
- 經典物理實驗專題

Part II

第一學期課程

Chapter 1

科學文件寫作指引

1.1 課程大綱

介紹本學期預定進行的大致實驗內容，分配組別，並講解實驗課進行的模式以及預習報告，實習報告的格式規範。

- 免費文書編輯軟體 OpenOffice.org 簡介
- 科學排版軟體 LaTeX 簡介
- LaTeX 數學方程式編輯
- LaTeX 與 OpenOffice 結合：oooLatex

額外補充（optional）：

- 所見即所得 LaTeX 編輯平台：GNU TeXmacs
- 中文 LaTeX 編輯範例（使用 CJK unicode 套件）

1.2 OpenOffice

OpenOffice 是一套免費且功能強大的文書編輯軟體，你可以在 `http://zh.openoffice.org/new/zh_tw/` 下載此套軟體。這堂實驗課的報告推薦大家使用 OpenOffice 作爲寫作平台，一方面在於 OpenOffice 有很好的 LaTeX 外掛模組可用，能夠排版高水準的方程式內容；另一方面 OpenOffice 的檔案可以直接輸出成 pdf 檔（ Portable Document Format）。pdf 具有字型內嵌，保有原文件格式的優點，不像一般的 MS Word 文檔常常換台電腦開啓，圖片與段落即產生錯亂（有的時候是範本設定的問題，更多情況是使用者不懂得設定格式，總以爲文字圖片隨意拉到

滿意的位置就完成排版）。因此， 實驗課的報告只接受 pdf 格式（無特效的投影片也請儘量轉換成 pdf 格式）。

pdf 的閱讀程式有（推薦後者）:

Acrobat Reader `http://get.adobe.com/tw/reader/`

Foxit Reader `http://www.foxitsoftware.com/pdf/reader/`

1.3 向量圖繪製

Inkscape `http://www.inkscape.org/`

Dia `http://projects.gnome.org/dia/`

Inkscape 是一款免費且相當好用的向量圖繪製軟體，本書絕大部分的示意圖皆以 Inkscpae 繪製，輸出成 pdf 檔再引入 LaTeX 文件中。向量圖檔除了圖片品質不受解析度影響之外，檔案所佔得記憶體空間也相當小，是高效率的圖檔儲存格式。Dia 則是流程圖繪製軟體，由 Dia 繪製成的流程圖可以 Inkscape 相容的格式匯出作進一步的修改。

1.4 ooWriter

ooWriter 功能與 M$ Word 相同，請善用樣式與格式 `[F11]`，預先將助教指定的報告格式（標題字內文字體大小）建成標準樣式，操作上會比較方便。欲排版雙欄格式可以利用 `[插入 -> 區段 -> 欄]` 。

Some Tips:

- 要換頁請利用 `[ctrl+Enter]` ，別一直按 `[Enter]`
- 修改頁面樣式: `[格式 -> 頁]`
- 加入圖片標題: 選取圖片， `[插入 -> 標籤]`

1.5 LaTeX

"It has often been said that a person does not really understand something until he teaches it to someone else. Actually a person does not really understand something until after teaching it to a computer, i.e., express it as an algorithm."

Donald Knuth, in "American Mathematical Monthly," 81

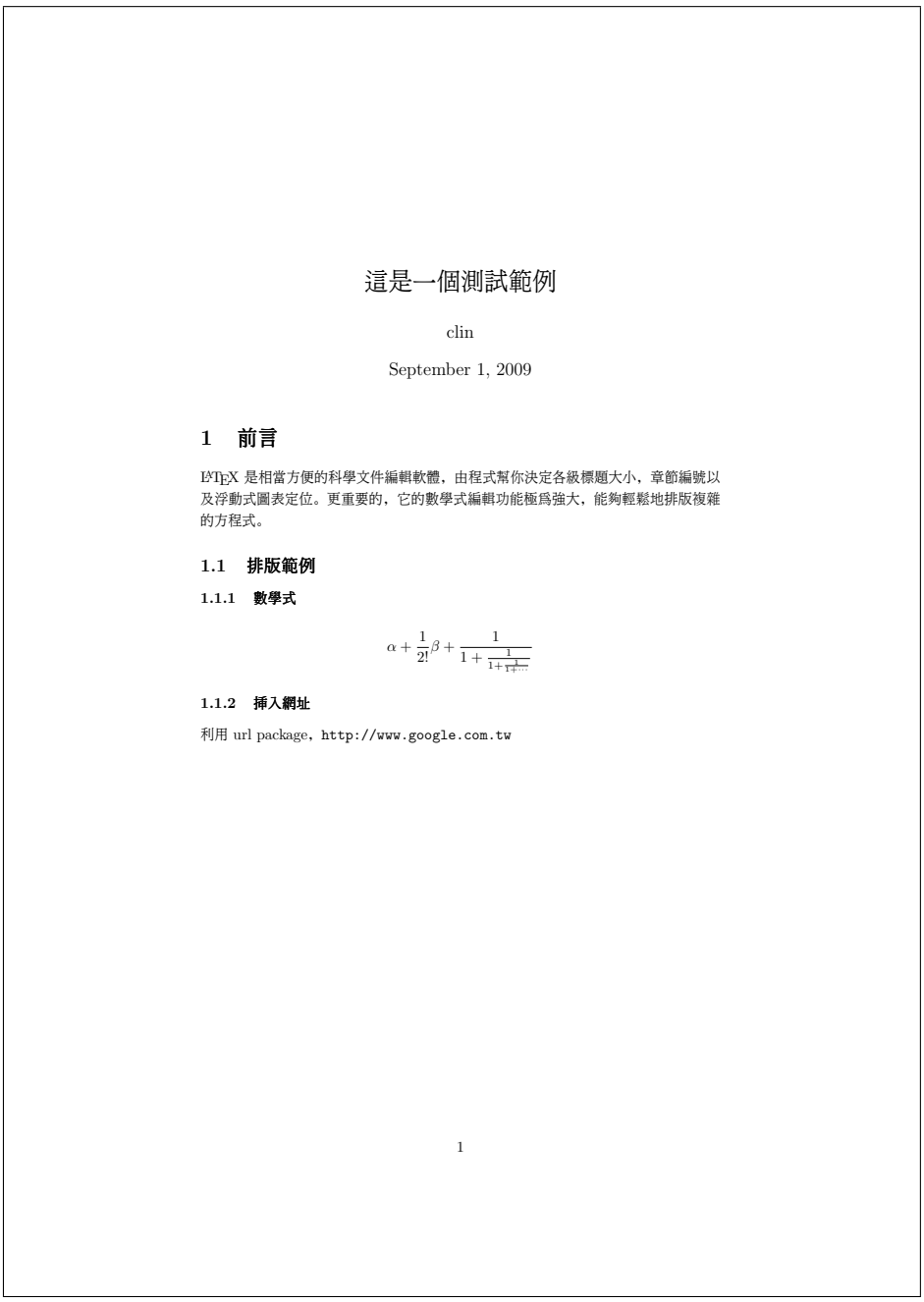
TeX 排版系統最早是由 Stanford Univ. 的教授 Don Knuth (有個正式的中文名字，高德納) 發展，當年 Knuth 開始計劃出版一套電腦科學界的巨著 "The Art Of Computer Programmnig " ，誰知道到第一册付梓時，Knuth 看到印刷廠送來排版

後慘不忍睹的樣本，差點沒氣出病來。火大之餘，Knuth 停下手邊的寫作進度，用了將近八年開發出一套基於 literate programming 概念，相當適合科學文件排版的軟體，也就是 plain TeX。也因爲 TeX 實在太好用了，越來越多的使用者加入開發以TeX 爲核心的應用套件，今天我們常用的 LaTeX 系統便是 Lamport 以 TeX 爲基礎，爲了方便使用額外加入許多套件而作成的加強版 TeX。

現在有許多出版社和科學期刊也開發自己專屬的 LaTeX 範本，利用 LaTeX 作爲完稿的排版工具；對於將來想從事學術研究的理工科同學來說，學會 LaTeX的操作就像是擁有一個強大的工具，可以不假他人之手，能以少許的時間完成具備高品質排版效果的科學文件。 LaTeX 將排版的行爲簡化成穩定且有規則的指令式排版環境，以下是一份 LaTeX文件未經編譯的原始碼:

```
\documentclass[a4paper]{article}
\usepackage{cjk}
\usepackage{url}
\linespread{1.2}
\begin{document}
\begin{CJK}{UTF8}{cwmu}
\title{這是一個測試範例}
\author{clin}
\maketitle
\section{前言}
\LaTeX ~是相當方便的科學文件編輯軟體，由程式幫你決定各級標題大小，章節編號以及浮動式圖表定位。更重要的，它的數學式編輯功能極爲強大，能夠輕鬆地排版複雜的方程式。
\section{排版範例}
\subsection{數學式}
\[ \alpha+\frac{1}{2!}\beta +\frac{1}{1+\frac{1}{1+\frac{1}{1+\cdots }}}  \]
\section{插入網址}
利用 url package ,  \url{http://www.google.com.tw}
\end{CJK}
\end{document}
```

輸出結果請參考 Fig. 1.1。 事實上，這本講義也是透過 LaTeX 編譯完成的作品。早期 LaTeX對於非英文語系的國家較不友善，需要透過一些非正式的字碼轉換 macro 來達成多國語系文件編纂的目的。近年來 LaTeX已經有相當不錯的 unicode 套件，字型與套件的安裝較以往容易許多，降低入門難度。



這是一個測試範例

clin

September 1, 2009

1 前言

LaTeX 是相當方便的科學文件編輯軟體，由程式幫你決定各級標題大小，章節編號以及浮動式圖表定位。更重要的，它的數學式編輯功能極爲強大，能夠輕鬆地排版複雜的方程式。

1.1 排版範例

1.1.1 數學式

$$\alpha + \frac{1}{2!}\beta + \frac{1}{1+\frac{1}{1+\frac{1}{1+\cdots}}}$$

1.1.2 插入網址

利用 url package，http://www.google.com.tw

1

Figure 1.1: 範例由 LaTeX輸出成 pdf 之結果

1.6 Miktex

在 WINDOWS 系統上最常使用的 TeX distribution 是 Miktex ，可以從 http://miktex.org/ 下載並進行安裝。除了 TeX 主程式以外，通常還需要配合專

屬的文字編輯器來使用才能事半功倍。由於 TEX有一定的入門難度，對於圖片的處理也不似一般的文書編輯體直觀，使用 OpenOffice 配合專用外掛來調用 LATEX 的數學式編輯功能會是讓初學者接受度較高的作法。等到熟悉 LATEX 的操作之後，想要追求更完美的排版效果時再進入純 LATEX 編輯環境便能架輕就熟，減低學習挫折感。

1.7 OOoLaTeX

LATEX 之所以能成爲物理與數學界的標準，有一半的原因來自於高效率的方程式排版，完稿效果非一般的文書編輯軟體可比。OpenOffice 有個輔助程式，名曰 OOoLaTeX ，能以簡單的方法呼叫 LATEX 編輯方程式，若有安裝指定的數學字型，還可以輸出向量格式（無失眞）的方程式內容，扣除 compile 造成的少量時間延遲（每個獨立的 equation 都要分別 compile 一次），不失爲揉合 LATEX 與一般文書編輯軟體便利性的優秀方案。

要利用 OpenOffice 輸出 LATEX 方程式區塊，你必須安裝這些程式

- OpenOffice
- OOoLaTeX (from `http://ooolatex.sourceforge.net/`)
- Miktex or other distribution
- MSYS (for OOoLaTeX , `http://www.mingw.org/wiki/msys`)
- Ghostscript (from `http://pages.cs.wisc.edu/~ghost/`)
- math fonts (from OOoLaTeX website)

安裝完 OOoLaTeX 後，我們會發現 OpenOffice 工具列左端多出 OOoLaTeX 的 button；要先在 Config 中設定 Latex，Ghostscript 以及 MinSYS (MSYS) 的執行檔位址才能插入方程式。在前述軟體使用預設目錄安裝的情況下應該要填入:

- Latex: `C:\ProgramFiles\MiKTeX2.7\miktex\bin\`
- Ghostscript: `C:\ProgramFiles\gs\gs8.60\bin\`
- MinSYS: `\C:\msys\1.0\bin\`

1.8 LATEX 數學式

若你還沒安裝 OOoLaTeX ，可以在一些提供線上 LATEXequation compile 服務的網頁輸入指令來練習 LATEX 方程式編輯。

`http://www.codecogs.com/components/equationeditor/equationeditor.php`

在 LaTeX 中以反斜線（\）作爲指令提示前綴字元，大括號（{}）則代表指令作用區塊。詳細的說明請參閱吳聰敏，吳聰慧教授的 cwTeX 使用手册的第九章。

OOoLaTeX 的 Equation 模式相當於 LaTeX 的隨文數式與方程式環境，若要插入正體英文說明敍請使用 指令。

一般而言，在輸入LaTeX 方程式的大概都是邊查表邊繕打特殊符號或是排版指令，我們也可以利用專門針對LaTeX 設計的文字編輯器來加速整個撰寫的過程，利用軟體整合好的特殊符號指令表與即時指令查詢功能省去查書的時間。針對這個部份，推薦適用 Texmaker 與 LED editor 這兩款專爲LaTeX 設計的文字編輯器。

Texmaker `http://www.xm1math.net/texmaker/`

LED editor `http://www.latexeditor.org/`

1.9 LaTeX 參考資料

吳聰敏，吳聰慧教授的 cwTeX 使用手册

`http://homepage.ntu.edu.tw/~ntut019/cwtex/cwtex.html`

雖然本書主要是介紹吳教授撰寫之中文 TeX 轉換外掛（以外加轉換字集方式讓原本非 unicode 的 TeX 系統可輸出 Big5 文件，事實上在支援 unicode 的 LaTeX 出現之後，亞洲語系的使用者不再需要這類繁瑣的轉換程式，可以直接編寫中文 TeX 文件，比方說這份講義就是使用 CJK package ），但其中的排版概念也相容於原本的 ，書中第九章講解數學式繕寫的部份對初學者而言相當親切，廣而足的內容也足以應付勝任方程式輸入的 FAQ，是很出色的參考資料。

美國數學協會（AMS）的 Short Math Guide for LaTeX

`ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf`

符號速查表

`http://amath.colorado.edu/documentation/LaTeX/Symbols.pdf`

colorado's LaTeX online tutorials

`http://amath.colorado.edu/documentation/LaTeX/tutorial/`

1.10 預習報告

因爲是第一堂課，不需要繳交預習報告。

1.11 實習報告

請藉由下述範例熟悉 LaTeX 指令。

1.11.1 學習 \array、\left 和 \right 的用法

$$F_{ij}=\left(\begin{array}{cccc}0&B^3&-B^2&cE_1\\ -B^3&0&B^1&cE_2\\ B^2&-B^1&0&cE_3\\ -cE_1&-cE_2&-cE_3&0\end{array}\right)$$

對應的 LaTeX code:

```
F_{ij}=\left(\begin{array}{cccc}0&B^3&-B^2&cE_1\\
-B^3&0&B^1&cE_2\\
B^2&-B^1&0&cE_3\\
-cE_1&-cE_2&-cE_3&0\end{array}\right)
```

1.11.2 常用字元與分式 \frac 練習

$$\nabla^2\Phi=\frac{1}{r^2}\left[\frac{\partial}{\partial r}\left(r^2\frac{\partial\Phi}{\partial r}\right)+\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial\Phi}{\partial\theta}\right)+\frac{1}{\sin^2\theta}\frac{\partial^2\Phi}{\partial\lambda^2}\right]$$

對應的 LaTeX code:

```
\nabla ^2\Phi=\frac{1}{r^2}\left[\frac{\partial}{\partial r}
\left(r^2\frac{\partial\Phi}{\partial r}\right)+
\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}
\left(\sin\theta\frac{\partial\Phi}{\partial\theta}\right)
+\frac{1}{\sin^2\theta}\frac{\partial^2\Phi}{\partial\lambda^2}\right]
```

1.11.3 \underbrace、\mbox、\big

以底括號指令 `\underbrace` 插入正體說明文字（`\mbox`），以手動字體放大指令 `\big` 或 `\Big` 手動調整左右括號的大小。

$$\rho_{ng}^{(1)}=\frac{1}{i\hbar}\int_{t_0}^{t}dt_1A_{ng}(t-t_1)H'_{ng}(t_1)\rho_{gg}^{(0)}(t_1)$$

對應的 LaTeX code:

```
\rho^{(1)}_{ng}=\frac{1}{i\hbar}\int^t_{t_0}
dt_1A_{ng}(t-t_1)H'_{ng}(t_1)\rho_{gg}^{(0)}(t_1)
```

$$H'_{ng}(t_1) = -\frac{1}{2}\mu_{ng}\underbrace{\big[E(t_1)e^{-i\omega t_1}}_{\text{resonant}} + \underbrace{E^{*}(t_1)e^{+i\omega t_1}\big]}_{\text{antiresonant}}$$

對應的 LaTeX code:

```
H'_{ng}(t_1)=-\frac{1}{2}\mu_{ng}
\underbrace{\big[E(t_1)e^{-i\omega t_1}}_{\mbox{resonant}}+
\underbrace{E^{\ast}(t_1)e^{+i\omega t_1}\big]}_{\mbox{antiresonant}}
```

參考上述三題範例，請自行選取一則方程式利用LaTeX 打出來。

Chapter 2

科學數值運算：Scilab 入門

2.1 課程大綱

2.2 Scilab

Scilab 是法國 INRIA 與 ENPC 自 90 年代發展起的 open source 科學數值運算軟體，與頗富盛名商業軟體 Matlab 系出同源，兩者的語法極爲相似。以 Scialb 作爲這門實驗課的平台一方面是鼓勵大家使用不必花錢，功能強大不遜於商業軟體的開放源代碼自由軟體（商業軟體的 license 動輒數萬元，以教育爲目的初級課程也並不需要太過專門的指令集與加強功能），一方面日後同學需要轉換至 Matlab 平台（相當實用的簡易計算工具，如果你是學理論的或是非工程的學科恐怕得學其他的高階語言）也不會有疏離感。

如果學好程式語言，那麼這門實驗課是完全不夠的，正常的教法該要從 C/C++，Java 或 Fortran 之類的語言教起，但等你熟悉資料結構，學會如何運用運算子來處理簡單的物理問題並繪製圖形時，這個學期早就結束了。無奈之餘，只能從 Scilab 這種入門難度低，有配套 GUI 繪圖部件的直譯式語言學起。這麼做的好處是就算一面查語法，一面改範例來寫模擬都不容易出錯，壞處是即興寫作的特性會讓人養成不好的程式寫作習慣。這門實驗課也許是同學們第一堂與程式語言有關的課，折衷下只能採取非正統的教學方式來增加同學的成就感，眞正有志於程式寫作的人應該要選修其他的專業課程來補足基礎才對。

你可以從 Scilab 的官方網頁下載最新版本的 Scilab: `http://www.scilab.org/`

2.3 Scilab 網路學習資源

2.3.1 Master Scilab!

`http://home.hit.no/~finnh/scilab_scicos/`

2.3.2 Scilab Bag Of Tricks

http://kiwi.emse.fr/SCILAB/sci-bot/book1.htm

2.3.3 Gilberto E. Urroz 的 lecture

http://www.neng.usu.edu/cee/faculty/gurro/Scilab.html

2.3.4 中文 Scilab 資源

非到萬不得已，請不要依賴中文手册。並不是所有的軟體都有人幫你寫好中文說明。 http://science.openfoundry.org/ade/scilab/

2.4 如何寫好程式 (Scilab)

1. 官方的語法說明是最好的參考書
2. 把 Scilab 本身提供的 demostrations 執行過一遍，對照著程式碼
3. 與別人討論
4. 參考 Knuth 所著的 TAOCP （ The Art of Computer Programming ）

2.5 console

安裝好 Scilab，點擊捷徑後會看到白底黑字的命令列視窗。輸入算式後鍵入 [Enter] 可以執行運算，要清除 commond history 請按 [F2]。

```
--> 1+2*3/(5-2) [Enter]
ans = 3.
--> ans*2 [Enter]
ans = 6
```

要指定變數也很簡單

```
--> a=3 [Enter]
a = 3.
--> a [Enter]
a = 3.
--> a^2-1 [Enter]
ans = 8.
--> exp(a+3*%i) [Enter]
ans = - 19.884531+2.844711 i
```

Scilab 指令爲關鍵字後加上一組圓括號（parents），括號內爲函數的輸入值，上面的例子 `sin()` 即 Scilab 內建的數學函數。`%i` 爲系統內建的常數，也就是虛數單位，前綴字元 `%` 與關鍵字結合即呼叫該函數值，如圓周率 π 即 `%pi`，自然常數 e 即 `%e`。常用的數學函數請參考 Help 中 Elementary Function 章節。要查詢特定函數（或指令）的用法可以使用 `help <function name>`。

2.6 矩陣運算

Scilab 的一大特色便是矩陣運算，以下介紹陣列與矩陣的宣告與運算。

```
--> a1= [1 0 1 0 0 1] [Enter]
a1 = 1.  0.  1.  0.  0.  1.
--> a1(3) [Enter]
ans = 1.
--> a2 = [1,2;3,4] [Enter]
a2 = 1.  2.
     3.  4.

--> a3 = 1:4 [Enter]
a3 = 1.  2.  3.  4.
--> a4 = 0:0.2:1 [Enter]
a4 = 0.  0.2  0.4  0.6  0.8  1.0
--> a5 = [1:3;2:4] [Enter]
a5 = 1.  2.  3.
     2.  3.  4.

--> row1 = a5(2,:)
row1 = 2.  3.  4.
--> column3 = a5(:,3)
column =  3.
          4.

-->a2*a5
ans = 5.   8.   11.
     11.   18.   25.

-->a2^2
ans = 7.    10.
     15.    22.
```

```
-->a.^2
ans = 1.  4.
      9.  36.
```

+ - * / 加上前綴字元 .(dot) 形成的 .+ .- .* ./ 代表矩陣元素的個別運算。: (colon) 是 Scilab 中相當常用的一個運算子，請特別注意它的用法。宣告矩陣元素時可以用，(comma) 區隔，或者直接加上一格空白。要換到下一列的時候則加上 ; (semicolon) 標記。

2.7 運算次序

```
--> 2^2*3^2 [Enter]
ans = 36.
--> a=[1 2;3 4]; b=[1 2;1 2]; [Enter]
--> a^2*b^2 [Enter]
ans = 51.   102.
      111.   222.

--> a.^2.*b.^2 [Enter]
ans = 5.   20.
      25.   100.

--> (a.^2).*(b.^2) [Enter]
ans = 1.   16.
      9.   24.
```

倒數第二條敘述可能會被我們誤解爲與最末條敘述等義，事實上 Scilab 會把第二個句點判別爲指數 2 之小數點: 在不確定運算優先序的情況下，還是不厭其煩地使用括號來輔助比較保險。一個敘述若是以 ; (semicolon) 作結尾，計算結果不會輸出在螢幕上: 若以，(comma) 作結尾則相反。

2.8 字串 string

Scilab 可以處理字串類型的資料，宣告時以單引號或是雙引號標示，要將數值類型的變數轉換成字串可以使用 string 指令，反之想將字串類型的傳換成數值可以使用 eval 指令。字串當然也可以宣告成矩陣的形式，但是字串與數值變數不允許混用。

```
--> a=[ 'what is your name?' , 'how old are you ?' ] [Enter]
```

```
a = !what is your name?  how old are you ?  !
-->name = input( a(1),'s' ) [Enter]
what is your name?
[type] clin
name = clin

--> age = input( a(2) ) [Enter]
how old are you ?
[type] 21
age=21.

printf('You are %s. Next year you will be %d. ', name, age+1) [Enter]
You are clin. Next year you will be 22.
```

`printf` 即 Scilab 模擬 C language 列印輸出的函數，用法與 C language 的相同，範例中的 `<%+specifer>` 可以看成不同資料型態的變數，`%s`，`%d` 依序對應到後頭以 comma 分開的變數 `name` 與 `age-1`；不同的 specifer 代表不同的 data type，比方說 s 代表字串，d 代表十進位制的數值變數，詳細的用法請自行查找 Scilab 的說明。

2.9 Script

利用指令 `scipad` 或點擊工具列上的快捷鍵可以進入 script 編寫環境，與直譯式的程式語言相近，可以撰寫內容較長的程式碼來完成比較複雜的計算，繳交報告的時候請附上相關的 .sce 檔供助教檢閱。由 scipad 工具列 `Execute -> Load in scilab` 或按下組合鍵 `[ctrl+I]` 可以執行 script 程式碼。

```
//---------script demo--------

// constant table
grvy = 9.8;      // m/s
m = 1;           // kg
time = 20;      // 10^(simu_t_order) sec
t_order = -3;      // ms
simu_t_order =-4;  //10^-1 MS
x0=0;             // meter
v0=0;            // m/s

//----------parameter----------
total_step = time*(t_order-simu_t_order);
```

```
t = 1:total_step;
t_scale = (t-1)/10^(-simu_t_order) ;

// ---------free falling---------
displacement = x0+0.5*grvy*t_scale.^2;
vlcty = v0+grvy*t_scale;

// ----------plot----------------
subplot(121)
plot(t_scale,displacement,'x');
xtitle('dispalcement of free falling','10^(-1) ms','meter');
subplot(122)
plot(t_scale,vlcty,'o');
xtitle('velocity of free falling','10^(-1) ms','meter');
```

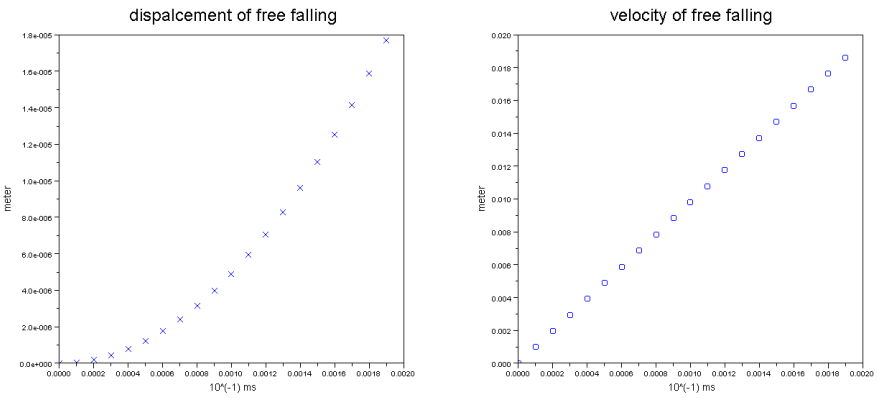


Figure 2.1: script demo 輸出圖形

在編輯大型程式的時候，有幾點要特別注意:

1. 要記得加上註解（使用連續兩個斜線（double slash）//）
2. 變數名稱要設的易懂且有意義
   - 以大寫間隔: `PleaseInputYourName`
   - 以底線區隔: `please_input_your_name`
   - 去母音縮寫: `pls_inpt_ur_nme`
3. 有意義的常數值要設成變數以方便未來可能的修正

4. 程式落入無窮迴圈想終止時在 console 視窗按 [ctrl+c] 即可

寫程式一方面是要簡化繁複的運算，建立穩定可靠、具備特殊用途的個人化計算器，一方面也要有可維護、可更新的彈性，讓我們可以透過少許修改來擴充用途。適當的命名變數與註解讓程式碼不再是艱澀難解，也不會有看不懂自己寫的程式的窘境（事實上，這是極有可能發生的慘劇）。雖然這入門的程式實驗課程中不大會碰到動輒數百行的龐大程式，但是良好的程式寫作習慣必須趁早培養：就 team work 的觀點，程式寫的再好，別人看不懂也是沒有用。因此，作爲報告一部份的程式碼若必須適當註解才行（加入報告文本詳細說明者例外）。

2.10 迴圈結構

2.10.1 if

```
if <condition> then <statement>
elseif <condition> then  <statement>
elseif <condition> then  <statement>
....
else <statement> end
```

顧名思義，if 迴圈可以根據某些條件來判斷哪些程式碼應該執行，利用 elself 與 else 可以連結依續執行的多組 if 敍述。then 語句可用 comma (，)取代。

if demo code

```
num=input('Please input an integer .');
mod=modulo(num,3);

if (mod==0) then
  printf('%d mod 3 = 0',num);
elseif (mod==1) then
  printf('%d mod 3 = 1',num);
else
  printf('%d mod 3 = 2',num);
end;

result:

Please input an integer .20 [Enter]

20 mod 3 = 2
```

2.10.2 case

```
select <keyword>
case <value1> then <statement>
case <value2> then <statement>
....
(else block)
end
```

case 迴圈就像遙控器，根據傳入的 keyword 值來決定執行哪些敘述，效果跟一連串 if 有點像，只不過判別條件比較單純。我們可以把上一個小節的 if 迴圈改寫成的形式。

case demo code

```
num=input('Please input an integer .');
mod=modulo(num,3);

select mod
  case 0 then printf('%d mod 3 = 0',num);
  case 1 then printf('%d mod 3 = 1',num);
  else printf('%d mod 3 = 2',num);
end
```

2.10.3 for

```
for <loopdefine>, <statement>, end
```

for 迴圈會根據某組預先設計好的條件重複執行迴圈內的敘述，比方說 <loopdefine> 取爲 i=1:10 時，會在 i 等於 1,2,3,...,10 的情況下依續執行迴圈內的敘述，該敘述被執行十次。如果想要反序遞減 i 值，可以使用 i=10:-1:1。

for loop demo code

```
for i=1:5,
  for j=1:5,
    printf('%d x %d = %d  ',i,j,i*j);
  end
  printf('\n');
end
```

```
result:

1 x 1 = 1  1 x 2 = 2  1 x 3 = 3  1 x 4 = 4  1 x 5 = 5
2 x 1 = 2  2 x 2 = 4  2 x 3 = 6  2 x 4 = 8  2 x 5 = 10
3 x 1 = 3  3 x 2 = 6  3 x 3 = 9  3 x 4 = 12  3 x 5 = 15
4 x 1 = 4  4 x 2 = 8  4 x 3 = 12  4 x 4 = 16  4 x 5 = 20
5 x 1 = 5  5 x 2 = 10  5 x 3 = 15  5 x 4 = 20  5 x 5 = 25
```

2.10.4 while

```
while <condition> do <statement> (else block) end
```

`while` 效果相當於未定次數的 `if` 迴圈，其中的敍述語句會被重復執行直到 `<statement>` 的判斷式爲 false 方可跳出迴圈；加入 `else` 語句可以進一步設定結束迴圈後要執行的語句。`while` 在使用的時候要別小心條件語句的設定，否則極易陷入無窮迴圈的窘境。

while loop demo code

```
i=1;j=1;
while i<6 do
while j<6 do printf('%d x %d = %d  ',i,j,i*j); j=j+1;
else printf('\n'); end
  j=1;i=i+1;
end

result:

1 x 1 = 1  1 x 2 = 2  1 x 3 = 3  1 x 4 = 4  1 x 5 = 5
2 x 1 = 2  2 x 2 = 4  2 x 3 = 6  2 x 4 = 8  2 x 5 = 10
3 x 1 = 3  3 x 2 = 6  3 x 3 = 9  3 x 4 = 12  3 x 5 = 15
4 x 1 = 4  4 x 2 = 8  4 x 3 = 12  4 x 4 = 16  4 x 5 = 20
5 x 1 = 5  5 x 2 = 10  5 x 3 = 15  5 x 4 = 20  5 x 5 = 25
```

2.11 資料存取

Scilab 可以利用 `save` 與 `load` 存取 binary data，也可以使用 `file` 與 `write`，`read` 將檔案存取成 txt 檔。前者用法較爲簡單，不做贅述。資料的存取主要在 Scilab 的 work directory 下進行，你可以從工具列 `File -> Chage Current Directory` 或是使用 `cd`，`chdir` 指令更改工作目錄。

2.11.1 file

```
fileID = file ( <action>, file-name, <status> )
```

<action>

- 'open' 開啓檔案
- 'close' 關閉檔案
- 'rewind' 將指標切回檔案最前端

<status>

- 'new' 目錄中尚未有 file-name 這個檔案
- 'old' 目錄中已存在 file-name 這個檔案
- 'unknown' 未知的 status

2.11.2 write

```
write ( file-name, [var1, var2, ...], <format> )
```

<format> 爲 fortran format，引號內加上一組括號，例如 '(e10.5 , f8.4)' 代表第一個變數使用指數表示，總長度十位，小數點後保留五位；第二個變數代表十進位表示，小數點後留四位。若 file-name 設爲 %io(2) 則代表將資料寫入螢幕這個 device，與 disp 和 printf 有類似的效果。

```
--> format('v',20)  // 將變數輸出格式設爲十進位，有效數字 20 位
--> a=1234567890.123456 ;
--> write ( %io(2), [a a a ], '(f15.4, f15.3, f15.2)' ) [Enter]
1234567890.1235 1234567890.123  1234567890.12

--> write ( %io(2), [a a a ], '(f15.4,3x, e15.8,3x, d15.8)' ) [Enter]
1234567890.1235    0.12345679E+10     0.12345679D+10

--> write  ( %io(2), a, '(f15.5)') [Enter]
***************  // 顯示的位數大於指定的有效位數，應將 format 改爲 f16.5

-->str1 = 'alpha' ; str2 = 'beta' ;
-->write ( %io(2), [str1 str2; str1 str2] ,'(''str='',2x,a5)' ) [Enter]
str=  alpha        // 在 format 中額外加入的字串以連續兩個引號標示
str=  alpha        // 使用 write 時字串與數值不可混用
```

```
str=   beta          //  x 代表空格，2x 即空兩格
str=   beta
```

2.11.3 read

```
data1 = read ( file-name, row , column, <format> )
```

當你知道讀取的資料行數的時候，row 設爲行數，不曉得行數時，row 設爲 -1 可以確保讀取所有資料。

2.12 I/O 程式碼範例

2.12.1 寫入 txt 檔

```
time = 1:0.2:2; //製造三筆資料
data_sin = sin(time);
data_cos = cos(time);

title_str=['time','data_sin','data_cos'];
title_name = input('title name?','s'); // 讓使用者決定輸出檔名
title name? IOdemofile [Enter]

file_name = title_name+'.txt';
f1=file('open',file_name,'new'); // 開新檔案

write(f1,title_str,'(a)'); // 先寫入各欄變數標題

for i=1:size(time,2),  //依序寫入一列三筆，共十八列資料
write(f1,[time(i),data_sin(i),data_cos(i)],'(d,5x,e,e)');
end;

file('close',f1); // 完成操作以後記得關閉檔案
```

2.12.2 讀取 txt 檔資料

一旦知道資料排列格式，很容易就能利用 read 指令取出之前存入的變數。

```
//----------script file-------------
f3 = file ('open','IOdemofile.txt','unknown');
file ('rewind',f3);
title1 = read(f3,3,1,'(a)');
```

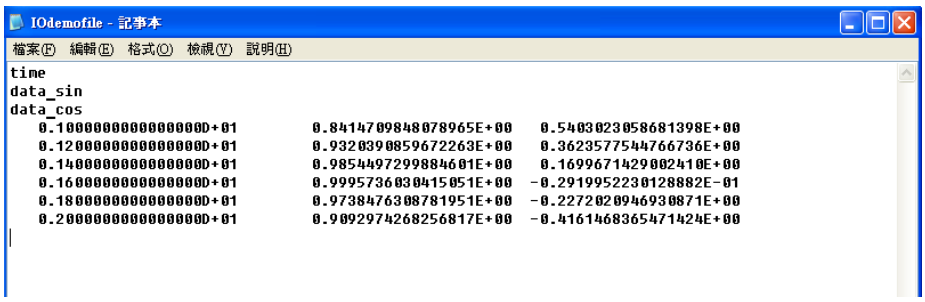


Figure 2.2: IOdemofile.txt 的內容

```
data = read(f3,6,3,'(d,5x,e,e)')
write (%io(2),title1(1),'(''column1 = '',a)');
write (%io(2),title1(2),'(''column2 = '',a)');
write (%io(2),title1(3),'(''column3 = '',a)');

for i= 1:6,
write (%io(2),[data(i,1),data(i,2),data(i,3)],'(d,e,e)');
end;

//----------screen---------------

column1=time
column2=data_sin
column3=data_cos
   0.1000000000000000D+01   0.8414709848078965E+00   0.5403023058681398E+00
   0.1200000000000000D+01   0.9320390859672263E+00   0.3623577544766736E+00
   0.1400000000000000D+01   0.9854497299884601E+00   0.1699671429002410E+00
   0.1600000000000000D+01   0.9995736030415051E+00  -0.2919952230128882E-01
   0.1800000000000000D+01   0.9738476308781951E+00  -0.2272020946930871E+00
   0.2000000000000000D+01   0.9092974268256817E+00  -0.4161468365471424E+00
```

2.13 實習報告

使用任何程式語言（除了助教講解所使用 Scilab，你也可以利用C，C++ 或 java，Python 等各類你熟悉的程式語言）做字串的輸入（從鍵盤或檔案）與輸出（到螢幕或檔案）與檔案合併。題目如下

2.13.1 資料輸入

以鍵盤鍵入 10 ~ 100 個英文字，輸出成 txt 檔，使用者可以決定檔案名稱。 如果你使用 Scilab 作爲工具，可以利用 `input` 或是 GUI 函數 `x_dialog`，`x_mdialog` 來完成這個問題。

2.13.2 資料讀入

由助教給予包含特定資訊的 txt 檔（可能是利用某台儀器量取到的電壓電流值，某一行第一個數字代表電壓值，加一個空格後第二個數字代表電流值），利用程式將 txt 檔中的資料擷取成矩陣或陣列的資料格式，並繪製成圖檔輸出。

Chapter 3

夾具，折床與帶鋸機

3.1 課程大綱

- 機械加工注意事項
- 夾具、折床、帶鋸機的用途和操作方法

3.2 機械加工安全規定

- 留長髮者請將髮絲紮好或者穿戴帽套，避免捲入機件或工作物導致受傷
- 請勿穿戴手套，戒指或飾品
- 儘量著短袖上衣，上衣下擺紮入褲頭
- 操作機器時至少要有兩人在場，一人操作，一人監督
- 操作機器時請勿靠在機器或加工物上
- 機器操作時以單人爲限，多人操作容易導致誤會與意外
- 請勿與機器操作者交談，使之分心
- 機器運轉時操作者請勿離開現場
- 熟悉操作機器的緊急停止開關以應付突發事件
- 開動機器前先檢查刀具與加工物是否牢固
- 切忌觸碰運轉中或尚未停止的機件
- 操作有問題請立即詢問指導人員

- 發生任何受傷事件請立即通報並進行治療
- 機器使用完畢要做好清潔工作，除去滑軌上的碎屑

3.3 預習報告

各種合金（鋁、黃銅、不鏽鋼）的性質（彈性、硬度、熱膨脹係數、延展性、導電性）。

3.4 實習報告

使用折床和沖床，製作鋁質金屬盒，上面預留D型電源插座孔位和附氖氣燈電源開關孔位。

Chapter 4

塑膠翻模

4.0.1 課程大綱

- 塑膠翻模複製技術簡介
- RTV 矽膠模具製作
- 環氧樹脂澆鑄實習

4.1 Garage Kit

一般而言，現代大量生產之塑膠製品大多以不鏽鋼模具進行生產，雖然鋼模成本極高，卻能藉由高耐用度壓低單一複製品的製作成本。若爲小規模的複製（複製品少於百件），使用不鏽鋼模具成本過高，也無財力採購機器進行生產，取而代之的便是人工複製技術：以室溫硫化（RTV）矽膠進行高精度的物件複製。

這類技術多爲模型愛好者（高階玩家）使用，一方面可以根據個人需求複製多量市售零件以便改造，一方面可以先利用油土或者可塑性材質自行雕塑原件，再以矽膠翻製成可研削修整的塑膠複製品。在學術上的應用並不多（有一部分研究表面微結構的人會利用類似的技術來翻製特殊結構，比方說蓮葉，當然使用的材料有異），我們希望引入這門技術與機械加工互補，部份低精度，輕量化且需大量使用的機械元件可以透過塑膠化來取代，利用塑膠容易加工，便宜的特性來製作原型。

矽膠模具的優點:

- 細部複製能力強
- 可重複使用（相較於石膏模）
- 低成本（相較於鋼模）
- 模具富彈性，可複製特殊結構

4.2 準備工具

要完成基本的翻模作業，必須有下列幾樣工具與材料才行。

- 電子秤
  - 調和矽膠或是環氧樹脂原料的時候，都需要按照該產品提供的重量比來控制主劑與硬化劑的重量比才能取得最好的效果。一般 RTV 矽膠的硬化劑約爲主劑的 3% 至 5% ，太少的硬化劑無法使矽膠正常固化，加太多則形同浪費。
- 油土
  - 若要製作雙面模具，以油土舖底固定原件會相當方便。油土是可重複使用的材料，價格低廉可於一般的美術器材行購得。若是利用厚紙板或是積木製作隔板灌注矽膠，可以使用油土塡塞縫隙避免外漏。
- 離型劑
  - 製作雙面模具的時候一定會使用到。離型劑的成份爲酒精與潤滑油，可以區隔原件，矽膠與塑膠複製品間的介面。在矽膠模具的表面加太多的離型劑會破壞製品精度，太少則可能脫模不易，甚至識損害模具。凡士林也是相當優秀的離型劑。
- 製模用 RTV 矽膠
  - 可於化工行購得。根據類型，加入的硬化劑比例以及硬化時間有所不同。
- 清潔劑
  - 沾粘至桌面或地板的矽膠黏液可用有機烷清潔，或加入硬化劑待購化後除去。環氧樹脂可用酒精或丙酮，POLY 可用丙酮去除。

可供小規模量產使用的塑膠材料有:

- POLY (低成本，有高硬度，可微量固化版本，質脆)
- EPOXY 環氧樹脂 (低成本，低毒性，低黏性，低透明度，質軟)
- PU 系發泡材料
- Casting PU (高韌度，質輕，快速硬化)
- 滑石粉（ 輔助塡料 ）

考量到成本與操作的便利性，本實驗將採用環氧樹脂作爲塑膠複製品的材料。Poly 雖然便宜，但是操作時有嚴重的臭味問題；Casting PU 通常是複製人物模型選用的材料，價格高昂，操作上較不方便。使用環氧樹脂作爲填料除了有高透明度的優點外，低毒性，幾乎沒有臭味的優點更適合在實驗課講授。

進行灌模時請在通風處作業，儘量戴上專用口罩與乳膠手套以避免有機溶劑對人體的傷害。

4.3 製作範例

製作過程約五個小時，所有步驟與相關講解皆製作成紀錄片，同學們可以從 `ww.veoh.com/users/b95901149` 線上播放該影片，或者向助教索取檔案。

4.4 預習報告

從下述材料中任選三項進行資料收集並整理成報告，分析其化學組成和性質（彈性、硬度、熱膨脹係數、延展性、絕緣度、熔點）以及相關應用。各組的報告主題於前一週抽籤決定。

1. 壓克力 Polymethylmethacrylate，PMMA
2. 電木（酚醛樹脂）Bakelite
3. 聚氯乙烯 Polyvinyl chloride，PVC
4. 環氧樹脂 EPOXY
5. 鐵氟龍 Teflon, Polytetrafluoroethylene
6. 室溫硫化矽膠 RTV silicone
7. ABS 樹脂 Acrylonitrile butadiene styrene

4.5 實習報告

利用市售室溫硫化矽膠（RTV）配合油土灌製模具複製金屬物件。

Figure 4.1: EPOXY 翻製而成的透明輪胎零件