

普通物理實驗講義

Scientific Writing & Programming

中央大學物理學系

August 6, 2011

第二版編者序

這份講義最初的版本在 2009 年的夏天完成，除去首章文書編輯軟體的介紹，主要的重點放在如何透過電腦程式的撰寫來幫助我們模擬一些常見的物理現象，透過繪圖的方式來呈現實驗數據並加以分析。

科學計算是現代的實驗物理工作者不可缺乏的工具，使用人工統計大量的實驗數據極困難且浪費時間，透過各式各樣的電腦軟體我們可以有效率地處理這些數據，甚至利用各類演算法來驗證，甚至預先模擬不同情況下可能發生的實驗結果。在許多大型且昂貴的實驗中，我們很難透過試誤法 (trial by error) 收斂出最佳結果，必須先透過電腦先行模擬，在龐大的參數空間過濾出幾個可能的方向再開始實驗。

即便是理論學家，也能透過相當聰明的符號式計算工具 (例如 Maple 與 Mathematica) 來簡化繁雜的符號運算，利用電腦來驗算推導過程並且可以利用數值方式將不可解析的抽象結果畫成具體的圖表來分析其中的趨勢與隱含的物理。

本書預設的讀者為大學部的物理系學生，大多數範例僅涉及微分方程，需要較多背景知識的數值模擬範例則放到附錄之中，有興趣的同學可以加以閱讀。另一方面，這份講義以大學部必修課程的參考資料為目標進行撰寫，考慮到軟體建置經費的緣故，筆者以自由軟體為媒介來實作各種在一般研究中可能遇到的數值問題與數據分析(本書提到的絕大多數軟體都有 Linux 平台的版本可執行，為了降低入門難度，仍以 Window 版本進行安裝介紹)，或許計算的速度不是最快，卻是以對初學者最友善的方式來講解。本書採用 Scilab 這個與 Matlab 系出同源的直譯式計算軟體作為講解範例(在 Linux 平台也有類似的 Octave 可用，不過圖形化介面與繪圖介面的整合度不若 Scilab 來的好)，一方面前者指令與 Matlab 這個許多實驗室使用的商用軟體極為類似，將 Scilab 上學到的概念轉換到 Matlab 上並不困難，一方面 Scilab 這類直譯式軟體對於語法的要求較為寬鬆，入門難度較低，在繪圖方面的便利度遠勝其他傳統程式語言。

本書初稿第一次用於教學時，學生最大的反應便是為何不使用 Matlab 這個 user-friendly 作為計算平台，市面上有關 Scilab 的書太少了。這是無法改變事實，自由軟體的文件支援很難贏過有大型專業團隊維護的商業軟體，在市場熟悉度尚未打開的情況下，許多優秀的自由軟體缺少足夠的中

文化說明文件來吸引更多人使用，Scilab 就是其中的一個例子。但是缺少中文解說不代表沒有出色的英文導覽可以閱讀，細心搜索，還是可以在網路上尋得不少免費且具參考價值的電子書。我個人認為 scilab 的功能足以涵蓋大學部學生可能遇到的各類數值問題與數據分析，假如有運算速度的瓶頸，它也可以與 c/c++ 和 fortran 等程式語言作連結，只是各位可能不知道或者沒有用到。

目前 Scilab 最大的不足在於圖像處理與繪圖控制器的部份，不比 matlab 直覺，且套件的限制較大。比如說如果你懂 jpg 或 bmp 的編碼格式，依然可以用 scilab 來處理 16bit 的圖檔，只是需要多花時間來讀懂 datasheet 並撰寫中介程式。這就是商用軟體為何可以賣這麼貴的原因，不是嗎？筆者由衷地希望各位在使用軟體的時候要多些自覺：這些東西並沒有想像中地那麼不值錢。許多人不喜歡自由軟體，那是因為他沒計算到正版軟體的價格因素，台灣人對於盜版太過習以為常了。考量到 C/P 值，這些免費的軟體提供了很大程度的折衷，既省了荷包，又可以作到想做的事情，只是犧牲了一些便利性。

為了彌補 Scilab 在影像處理上的不足，本書新版中加入這套由 美國衛生研究院 (NIF) 開發並公開原始碼的軟體 ImageJ 的介紹，直觀的 GUI 介面與眾多使用者開發的外部套件讓這套軟體可以統計多種格式的圖檔中物件間距、角度等二維資訊，甚至可以批次處理連續影像的分鏡，有效率地完成物體追跡等多維度資訊。

考量到市面上 Scilab 的教材缺乏，本書往後的更新在以 Scilab 作為基礎範例的段落會追加與 matlab 對應的原始碼與提示，讓各位有機會從類似的指令查找相關的書籍來驗證所學到的概念。同時將靜態資料不容易的呈現的軟體動態操作製作成影片放至 Youtube 上作為參考資料，最終達成不需要他人講解也能透過閱讀本書與相關資料而有效地學習科學計算。

本書以 XeLateX 編譯完成，感謝中央物理系 102 級的全體學生在本書為主軸進行教學帶來的回饋與啟發，以及當時的助教群兼摯友：黃揚智、廖孜桓與林昱廷協助筆者進行如此新穎近乎瘋狂的物理教學工作。

September, 2011 林致翰

第一版編者序

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為主，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，未來在學習更為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，真正的學習應該是 self-learning 才對，而非侷限在上課時台上講解的內容。

September, 2009 林致翰

備註：原書中機械加工與翻模課程的章節在新版中已刪去另成一冊。

Contents

0.1	普通物理實驗	5
1	科學文件寫作平台	6
1.1	OpenOffice	6
1.1.1	Writer	7
1.2	\LaTeX	7
1.2.1	Miktex	8
1.2.2	TeXworks	8
1.2.3	Online \LaTeX Math editor	9
1.2.4	OOoLaTeX*	9
1.2.5	\LaTeX 參考資料	10
1.3	\LaTeX 數學式	11
1.3.1	基本操作	11
1.3.2	數式環境	12
A	補充教材	17
A.1	有限差分法 Finite difference method	17
A.1.1	1D Time-independent Schrödinger Equation	17
A.1.2	一維量子井結構	18

A.1.3	Scilab code	19
A.1.4	思考問題	23
A.2	四階 Runge-Kutta 法的證明	24
A.2.1	推導過程	24
A.2.2	思考問題	26

0.1 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：

- 簡易電源電路
- 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

Chapter 1

科學文件寫作平台

- 免費文書編輯軟體 OpenOffice.org 簡介
- 科學排版軟體 L^AT_EX 簡介
- L^AT_EX 數學方程式編輯

額外補充 (optional) :

- 所見即所得 L^AT_EX 編輯平台: GNU TeXmacs
- L^AT_EX 與 OpenOffice 結合: oooLatex
- 中文 L^AT_EX 編輯範例 (使用 CJK unicode 套件)

1.1 OpenOffice

OpenOffice 是一套免費且功能強大的文書編輯軟體，你可以從 http://zh.openoffice.org/new/zh_tw/ 下載此套軟體。之所以建議各位使用 OpenOffice 作為寫作平台，一方面在於 OpenOffice 有很好的 L^AT_EX 外掛模組可用(OooLatex)，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，不會發生如一般的 M\$ Word 文檔時常換台電腦開啓，圖片與段落便有很大機會發生錯亂的窘境 (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，總以為文字圖片隨意拉到的滿意的位置就完成排版的緣故)。除去含特效的投影片檔案，完稿形式的報告電子檔轉存成 pdf 格式流通可以最大程度讓讀者獲取你預設的版面格式而不產生誤解。

免費且好用的 pdf 的閱讀程式，有些甚至支援簡單的修改與註解功能，以下列出幾個常見且效能極佳之軟體供參考：

Acrobat Reader <http://get.adobe.com/tw/reader/>

Foxit Reader <http://www.foxitsoftware.com/pdf/reader/>

PDF-XChange Viewer <http://pdf-xchange-viewer.en.softonic.com/>

1.1.1 Writer

Openoffice 中的 writer 使用起來類似於 MS office 的 Word，編輯文件的時候請善用 **格式**→{樣式與格式(或按快捷鍵 [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，往後新增標題與內文可以減少更改字型格式的繁瑣操作。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**。
- 輸出 pdf 檔：**檔案**→**匯出成 pdf**

1.2 L^AT_EX

"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."

『常言道：學而能教謂之通。事實上，我認為能教授電腦做出一樣的事情才意味著真正的瞭解，這樣的過程也就是演算法的實作。』

Donald Knuth, in "American Mathematical Monthly," 81

TeX 排版系統最早是由 Stanford 大學的教授 Don Knuth (有個正式的中文名字，高德納) 發展。七零年代末，Knuth 著手計劃出版一套電腦科學界的巨著 "The Art Of Computer Programmig

" (此書曾被 "American Scientist" 《科學人》選為與 Dirac 所著之量子力學, Einstein 的相對論並列的二十世紀最佳學術專著), 誰知道該系列第二冊付梓時, Knuth 便被印刷廠送刷樣本中慘不忍睹的數學式弄得火冒三丈。氣憤之餘, Knuth 乾脆停下手邊的寫作進度, 用了近八年的時光開發出一套基於 literate programming 概念, 專門用於科學文件排版的系統, 也就是 plain T_EX, 並包含一套名為 Metafont 的向量字體描述語言。T_EX 使用上的便利導致越來越多的使用者加入開發以 T_EX 為核心的應用套件, 今日我們常用的 L^AT_EX 系統便是 Lamport 以 T_EX 為基礎, 額外加入許多套件而釋出的加強版 T_EX。

許多出版社和科學期刊也開發自己專屬的 L^AT_EX 範本 (比如說美國物理學會 APS 的 Rev-T_EX4), 利用 L^AT_EX 作為完稿的排版工具, 對於將來想從事學術研究的理工科同學而言學會 L^AT_EX 的操作就像是擁有一個強大的工具, 可以不假他人之手, 以相對少的時間完成高品質排版效果且滿足格式要求的科學文件。L^AT_EX 將排版的行為簡化成穩定且有規則的指令式排版環境。事實上, 這本講義也是透過 L^AT_EX 編譯完成的作品。早期 L^AT_EX 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 L^AT_EX 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。

1.2.1 Miktex

在 WINDOWS 系統上最常使用的 T_EX distribution 是 Miktex, 可以從 <http://miktex.org/> 下載並進行安裝 (目前已更新至 2.9 版, 需注意的是新的版本釋出以後, 舊版本的套件庫更新會在一段時間後停止支援)。除了 T_EX 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 T_EX 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 L^AT_EX 的數學式編輯功能會是讓初學者接受度較高的作法, 或者利用一些線上 L^AT_EX 數學式編輯網頁輸出方程式圖檔轉貼至 office 上。等到熟悉 L^AT_EX 的操作之後, 想要追求更完美的排版效果時再進入純 L^AT_EX 編輯環境便能架輕就熟, 減低學習挫折感。

1.2.2 TeXworks

新版的 MikTeX 中已內建一個優秀的文字編輯器 TeXworks, 其圖形化介面已經嵌入數種 L^AT_EX 編譯器的快捷鍵, 不需要額外的設定便能編譯寫好的 L^AT_EX 文檔。

1.2.3 Online L^AT_EX Math editor

可輸出各種圖片(png, pdf, svg, emf) 格式的 L^AT_EX online editor

<http://www.codecogs.com/latex/eqneditor.php>

對於只想使用 L^AT_EX 編輯數學式，卻不想安裝 L^AT_EX 系統的使用者而言，利用網際網路上其他人架設好的伺服器來輸出數學式圖片十分方便。若想將方程式圖檔匯入向量編輯軟體如 Inkscape, CorelDraw 等等可選擇下載成 svg 檔，若是希望將數學式匯入 MS powerpoint 作成的投影片或 office 相關的文書軟體則可以轉成 emf 檔。

若想將方程式匯入僅接受點陣圖檔的程式，轉存成 300 dpi 的 png 檔會是不錯的處理方式，或者儲存成 pdf 檔，透過 pdf 閱讀程式的照相機功能存取更高解析度的方程式結圖。以 Acrobat Reader X 為例，**編輯**→**偏好設定**→**一般** 中勾取使用固定的解析度來顯示快照工具節舉的影像，調整右欄位的像素/英吋 (即 dpi)，即可使用最高 720 dpi 的解析度將 pdf 檔的內容轉換成點陣圖儲存至剪貼簿中。

1.2.4 TeXmaker

一般而言，在輸入 L^AT_EX 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 L^AT_EX 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 這兩款專為 L^AT_EX 設計的文字編輯器。

Texmaker <http://www.xmlmath.net/texmaker/>

LED editor <http://www.latexeditor.org/>

1.2.5 OOoLaTeX*

L^AT_EX 之所以令人愛不釋手，有一半的原因來自於它超凡入聖的方程式排版功能，其完稿呈現美觀之至，絕不是一般的文書編輯軟體可匹敵的。OpenOffice 有個輔助程式，名曰 OOoLaTeX，能以簡單的方法呼叫 L^AT_EX 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（無失真）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），不失為揉合 L^AT_EX 與一般文書編輯軟體便利性的優秀方案。

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

- OpenOffice
- O^OLaTeX (from <http://oo-latex.sourceforge.net/>)
- MikTeX or other distribution
- MSYS (for O^OLaTeX , <http://www.mingw.org/wiki/msys>)
- Ghostscript (from <http://pages.cs.wisc.edu/~ghost/>)
- math fonts (from O^OLaTeX website)

安裝完 O^OLaTeX 後，我們會發現 OpenOffice 工具列左端多出 O^OLaTeX 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\

1.2.6 L^AT_EX 參考資料

吳聰敏，吳聰慧教授的 cwT_EX 使用手冊

<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 T_EX 轉換外掛 (以外加轉換字集方式讓原本非 unicode 的 T_EX 系統可輸出 Big5 文件，事實上在支援 unicode 的 L^AT_EX 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 T_EX 文件，比方說這份講義的初稿就是透過 CJKutf8 package 完成，新版則加入少許指令改用用 XeL^AT_EX 編譯)，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 L^AT_EX

符號速查表

<http://amath.colorado.edu/documentation/LaTeX/Symbols.pdf>

colorado's L^AT_EX online tutorials

<http://amath.colorado.edu/documentation/LaTeX/tutorial/>

1.3 L^AT_EX 數學式

1.3.1 基本操作

L^AT_EX 中所有指令皆以反斜線 `\` (通常位於 `Backspace` 與 `Enter` 鍵之間) 開頭, 指令區分大小寫, 若需輸入參數則於該指令後加入大括號`{}`。比方說在文檔中輸入 `\LaTeX{}` 會輸出成 L^AT_EX, 要在文中排版數學符號則必須同時使用隨文數式環境: `\sqrt{\pi}` 會排版成 $\sqrt{\pi}$ 。有些指令會需要多個參數, 比如說分式指令 `\frac`, 此時搭配的大括號不只一組:

$$\text{\texttt{\$}\texttt{\frac{1}{2}}\texttt{\$}} \implies \frac{1}{2}$$

各個指令可巢狀套用, 換言之某個指令大括號內部可以再放入一組新的指令:

$$\text{\texttt{\$}\texttt{\sqrt[3]{\frac{1}{2}}}\texttt{\$}} \implies \sqrt[3]{\frac{1}{2}}$$

有些指令會在大括號加入一組方括號 `[]` 來標示進階參數設定, 通常方括號標示的參數可以不輸入(取用內定值), 比方說上式的根號指令 `\sqrt` 指令可以加入 `[<n>]` 來改變根號左上方的次數, 若不輸入任何則預設為空白, 即一般的平方根。

除了反斜線與大括號以外, L^AT_EX 還有許多有意義保留字元, 這些字元通常需要以指令的方式輸入才能正確顯現(若為 unicode 中文系統則可直接輸入全形標點):

- `\%` \longrightarrow %, 註解字元
- `\&` \longrightarrow &, 表格分欄字元
- `\#` \longrightarrow #, 巨集指令參數
- `\$` \longrightarrow \$, 隨文數式提示字元
- `\verb+~+`
 \longrightarrow ~, 加入空白

- `\verb+^+`
→ [^], 數式上標
- `_{}→_`, 數式下標

1.3.2 數式環境

基本的數學指令可以參考吳聰敏，吳聰慧教授的 `cwTEX` 使用手冊第九章。L^AT_EX 的數學式環境主要可分為以下三種：

- 隨文數式 (inline): 以一對 `$$` 符號夾住數學敘述 `$<equation>$`，可直接將數學式嵌入文句中，例如 `$$\sqrt{\pi}$$` 會顯示成 $\sqrt{\pi}$ 。很多時候隨文數式的高度超過行高而自行壓縮符號大小，例如 `$$\sqrt{1}{2}$$` 會顯示成 $\frac{1}{2}$ ，這個時候可以加入 `\displaystyle` 強迫輸出正常比例的數學式，並自動加大行高，例如 `$$\displaystyle\frac{1}{2}$$` 會顯示成 $\frac{1}{2}$ 。使用 `\displaystyle` 雖然可以正確顯示數學式字體比例，更多時候卻會破壞整體段落的美觀，因此對於簡單的分式來說我們會將它改寫成橫式 `$1/2$ 1/2`，既不破壞行高 也能正確表達式子。若數學語句太過於複雜而不能改成橫式，我們則傾向將之獨立成行。
- 無編號單行展示數式 (display): 數學式獨立成行，一般玆延需要使用 `\begin{}`

```
\begin{displaymath}
\sum_{k=0}^n \frac{1}{n} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}.
\end{displaymath}
```

$$\sum_{k=0}^n \frac{1}{n} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}.$$

上述數式環境指令可以用 `\[<equation> \]` 或一對雙錢號 `$$ <equation> $$` 的簡化版指令來取代。複雜結構的連分式也同樣可以使用 `\displaystyle` 來還原結構

```
\frac{1}{1+\frac{1}{1+\frac{1}{1+\cdots}}}.
```

$$\frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \cdots}}}.$$

`\frac{1}{1+\displaystyle\frac{1}{1+\displaystyle\frac{1}{1+\cdots}}}`.

$$\frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \cdots}}}.$$

.

- 隨文數式 (inline) 以一對 符號夾住數學敘述`$<equation>$`，可直接將數學式嵌入

Ex 1.3.3

常用字元與分式 `\frac` 練習

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

Ans:

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

L^AT_EX 可以利用 `\left` 與 `\right` 自動調節括號包含 `()`, `[]`, `{ }` 與 `⟨ ⟩` 的大小，前文出現 `\left` 則文後必然出現一個 `\right` 與之配對，否則編譯文檔時會出錯。有些時候我們希望僅顯現單邊的符號時該如何？我們可以利用 `\left.` 或 `\right.` 來配對，以其替代掉不希望出現的某邊括號，請參考以下範例：

$$\left.\frac{d^2x}{dt^2}\right|_{x=1} \Rightarrow \frac{d^2x}{dt^2}\bigg|_{x=1}$$

Ex 1.3.4

學習 `\array` 的用法。

$$F_{ij} = \begin{pmatrix} 0 & B^3 & -B^2 & cE_1 \\ -B^3 & 0 & B^1 & cE_2 \\ B^2 & -B^1 & 0 & cE_3 \\ -cE_1 & -cE_2 & -cE_3 & 0 \end{pmatrix}$$

Ans:

```

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

```

或者可以使用 `pmatrix` 的指令，必須注意的是這個指令必須先在文稿全文設定區加上 `\usepackage{amsmath}` 引入 `amsmath` 套件方能使用。將關鍵字 `pmatrix` 改為 `bmatrix`, `Bmatrix`, `vmatrix` 或 `Vmatrix` 可以改變矩陣括號形式，請讀者自行練習。

```

F_{ij}=
\begin{pmatrix}
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{pmatrix}

```

Ex 1.3.5

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

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

Ans:

\$\$

$H'_{ng}(t_1) = -\frac{1}{2}\mu_{ng}$

$\underbrace{\big[E(t_1)$

$e^{-i\omega t_1}\big]}_{\text{resonant}}$

$+ \underbrace{E^*(t_1)$

$e^{+i\omega t_1}\big]}_{\text{antiresonant}}$

\$\$

Appendices

Appendix A

補充教材

A.1 有限差分法 Finite difference method

有限差分法是一種將微分方程離散化，讓求解過程轉換為計算其係數矩陣之特徵值（eigenvalue）與特徵向量（eigenvector）。以現在電腦的計算能力而言，在取樣點不多的情況下（小於一千），這類計算通常不到一秒的時間就完成。以下將以一維定態量子問題作為有限差分法的範例。

A.1.1 1D Time-independent Schrödinger Equation

$$\left[-\frac{\hbar^2}{2m^2} \nabla^2 + V(z) \right] \phi(\mathbf{r}) = E\phi(\mathbf{r}) \quad (\text{A.1})$$

分離變數，波函數可以表達成

$$\phi(\mathbf{r}) = \frac{1}{\sqrt{A}} e^{i(k_x x + k_y y)} \phi_n(z) \quad (\text{A.2})$$

代入 Schrödinger 方程式 eq. (A.1)

$$\left[\frac{\hbar^2}{2m_{\parallel}} (k_x^2 + k_y^2) + \frac{\hbar^2}{2m_z} \frac{\partial^2}{\partial z^2} + V(z) \right] \phi_n(z) = E_T \phi_n(z) \quad (\text{A.3})$$

將 m_z 當成常數，設 $a = -2 \hbar^2 / m_z$ ， $E_n = E_T - \hbar^2 (k_x^2 + k_y^2) / 2m_{\parallel}$ 可以得到

$$\left[a \frac{\partial^2}{\partial z^2} + V(z) \right] \phi_n(z) = E_n \phi_n(z) \quad (\text{A.4})$$

利用差分 $\partial^2 \phi_i / \partial z^2 = (\phi_{i+1} - 2\phi_i + \phi_{i-1}) / (\Delta z)^2$ 將上述為微分方程化成一系列的代數式，

$$\begin{aligned} \frac{a(-2\phi_1 + \phi_2)}{(\Delta z)^2} + V_1 \phi_1 &= E \phi_1 \\ \frac{a(\phi_1 - 2\phi_2 + \phi_1)}{(\Delta z)^2} + V_2 \phi_2 &= E \phi_2 \\ &\dots \dots \dots \\ \frac{a(\phi_{N-2} - 2\phi_{N-1} + \phi_N)}{(\Delta z)^2} + V_{N-1} \phi_N &= E \phi_{N-1} \\ \frac{a(1\phi_{N-1} - 2\phi_N)}{(\Delta z)^2} + V_N \phi_N &= E \phi_N \end{aligned} \quad (\text{A.5})$$

取 $a' = a / (\Delta z)^2$ ，將上式改寫成矩陣的形式：

$$\begin{pmatrix} -2a' + V_1 & a' & 0 & \dots & 0 & 0 \\ a' & -2a' + V_2 & a' & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & a' & -2a' + V_{N-1} & a' \\ 0 & 0 & \dots & 0 & a' & -2a' + V_N \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{N-1} \\ \phi_N \end{pmatrix} = E \begin{pmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{N-1} \\ \phi_N \end{pmatrix}$$

利用 Scilab 的 `spec` 指令可以很快地解出此矩陣的特徵值與特徵向量，換言之系統的能階與波函數的在 z 軸上的變化也可以求出。我們可以設定任意形狀（越平滑利用有限差分求出的解越好）的位能項 $v(z)$ 來模擬該情況下波函數的行為。利用電腦計算不規則位能分佈下的波函數比起求出解析解來得容易許多。

A.1.2 一維量子井結構

我們取 GaAs 材料中 10 nm 的方形有限位能井來進行模擬， $V_0 = 0.3 \text{ eV}$ 。在之後的程式碼裡，以事先選取適當的單位來減低運算的複雜度，可以自由更動模擬的間距以及總區間。至於質量項則考慮成 z 之函數，利用

$$\begin{aligned}
& \frac{1}{m_z(z)} \frac{\partial^2 \phi}{\partial z^2} \Big|_{z_i} \simeq \frac{\partial}{\partial z} \left[\frac{1}{m_z(z)} \frac{\partial \phi}{\partial z} \right] \\
&= \frac{1}{\Delta z} \left(\frac{1}{m_z(z)} \frac{\partial \phi}{\partial z} \Big|_{z_{i+1}} - \frac{1}{m_z(z)} \frac{\partial \phi}{\partial z} \Big|_{z_i} + \frac{1}{m_z(z)} \frac{\partial \phi}{\partial z} \Big|_{z_i} - \frac{1}{m_z(z)} \frac{\partial \phi}{\partial z} \Big|_{z_{i-1}} \right) \\
&= \frac{1}{2(\Delta z)^2} \left\{ \begin{aligned} & + \left(\frac{1}{m_z(z_{i+1})} + \frac{1}{m_z(z_i)} \right) \phi(z_{i+1}) \\ & - \left(\frac{1}{m_z(z_{i+1})} + \frac{2}{m_z(z_i)} + \frac{1}{m_z(z_{i-1})} \right) \phi(z_i) \\ & + \left(\frac{1}{m_z(z_i)} + \frac{1}{m_z(z_{i-1})} \right) \phi(z_{i-1}) \end{aligned} \right. \quad (A.6)
\end{aligned}$$

其中假設了 $1/m_z(z_i)$ 的微分乘上 $\partial\phi/z$ 遠小於 $\partial^2\phi/z^2$ ；四項中的頭兩項採用後向差分 $\partial\phi(z_i)/\partial z = (\phi(z_i) - \phi(z_{i-1}))/\Delta z$ ，後兩項採用前向差分 $\partial\phi(z_i)/\partial z = (\phi(z_{i+1}) - \phi(z_i))/\Delta z$ 。當然，取 m_z 爲常數以後仍然可以得到上一節 eq. (A.5) 的結果。程式碼中預留了將 m_z 擴充爲函數的空間，設爲 GaAs 的 effective mass $m_{\text{eff}} = 0.086 m_e$ 進行模擬。

A.1.3 Scilab code

運算部份

```

clear;
//[a(z)z'+v(z)]*phi(z)=En*phi(z)

// constant table
mass_e=0.067*9.10938188; //electron mass    10^-31 kg
hbar1=6.58211899; //10^-16 eV sec
hbar2=1.054571628; //10^-34 J sec

// simulation parameter

zsize=20; // total simulation range
zsize_order=-9; // zsize = 10^order m ; -9->nm
fdsize=1; // delta z
fdsize_order=-10; // astrom
total_steps=zsize*10^(zsize_order-fdsize_order); // simulation index
z=1:total_steps+2;//simu index

```

```
// potential wall config

barrier=0.3;    // eV
vwall_lngth=10; //nm
vwall=barrier*ones(1,total_steps);
vwall_step=vwall_lngth*10^(zsize_order-fdsize_order);
for i=total_steps/2+1-vwall_step/2:total_steps/2+1+vwall_step/2,
    vwall(i)=0;
end;

// mass function ->treat as constant

function [eff_mass]=mass(m0,z),
    eff_mass=m0;
endfunction

// 3 point FDM Matrix element assignment

function [coef_3fdm]=fdmnode3(z),
    coef_3fdm=-1/mass(mass_e,z+1)-1/mass(mass_e,z);
endfunction

function [coef_3fdm]=fdmnode2(z),
    coef_3fdm=1/mass(mass_e,z+1)+2/mass(mass_e,z)+1/mass(mass_e,z-1);
endfunction

function [coef_3fdm]=fdmnode1(z),
    coef_3fdm=-1/mass(mass_e,z)-1/mass(mass_e,z-1);
endfunction

// forming Matrix

H_fdw_pre=zeros(total_steps,total_steps);
```

```

for i=1:total_steps,
H_fdw_pre(i,i)=fdmnode1(z(i));
H_fdw_pre(i,i+1)=fdmnode2(z(i));
H_fdw_pre(i,i+2)=fdmnode3(z(i));
end;

H_fdw=H_fdw_pre(:,2:total_steps+1);

for i=1:total_steps,    //add potential term
    H_fdw(i,i)=H_fdw(i,i)+vwall(i)/cst_a;
end;

// compute eigenvalue&eigenvector
tic
[vecz engylvl]=spec(H_fdw);
toc

// correct the eigenvalue with cst_a

energy=zeros(1,total_steps);
for i=1:total_steps,
    energy(i)=engylvl(i,i)*cst_a;
end;

```

解析解

若位能井寬度為 a ，則波函數於 $-a/2 < z < a/2$ 時

$$\text{when } n \text{ is } \begin{cases} \text{odd}, \\ \text{even}, \end{cases} \quad \phi(z) \propto \begin{cases} \cos kz \\ \sin kz \end{cases} \quad (\text{A.7})$$

其中

$$k = \sqrt{\frac{2mE_n}{\hbar^2}}, \quad E_n = \frac{\hbar^2 \pi^2}{2ma^2} n^2 \quad (\text{A.8})$$

```
// analytic solution
```

```
cst_k=2*mass_e/(hbar1*hbar2)*10^(-1) //1/(10^-10*astrom*eV)
z_barrier = -vwall_lngth/2*10:vwall_lngth/2*10; //astrom
phi_analytic=zeros(1,zsize);

for i=1:3; //first three level
k_binding(i)=sqrt(cst_k*(energy(i)));//
if modulo(i,2) then
phi_b(i,:)=cos(k_binding(i)*z_barrier);
else
phi_b(i,:)=sin(k_binding(i)*z_barrier);
end;
end;
```

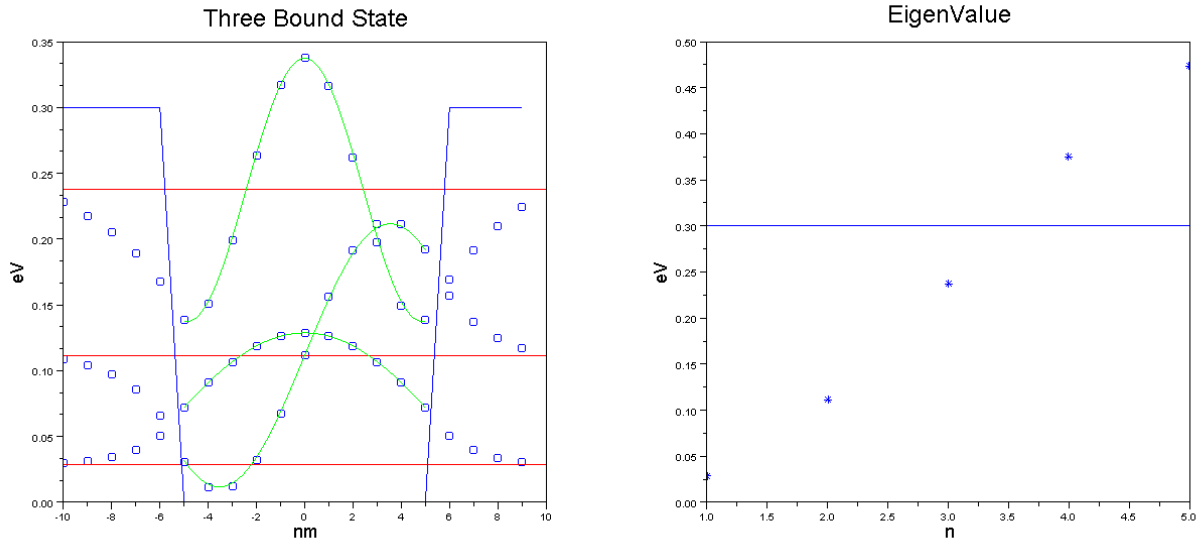
繪圖與模擬結果

將解析解與模擬結果相比對，在同一張圖中繪出三個 bound state 的波函數。

```
subplot(121)
for i=1:3
plot(z_scale,vecz(:,i)/max(vecz(:,i))*0.1+energy(i),'o');
plot((-zsize/2):(zsize/2),energy(i)*ones(1,zsize+1),'r')
end;

for i=1:3,
plot(z_barrier/10,phi_b(i,:)*0.1+energy(i),'g');
end;

plot(z_scale,vwall);
```

Figure A.1: 以 $\Delta z = 1$ 進行模擬

```

xtitle('Three Bound State','nm','eV');

subplot(122)
plot(0.3*ones(1,5));
plot(energy(1:5),'*');
xtitle('EigenValue','n','eV');

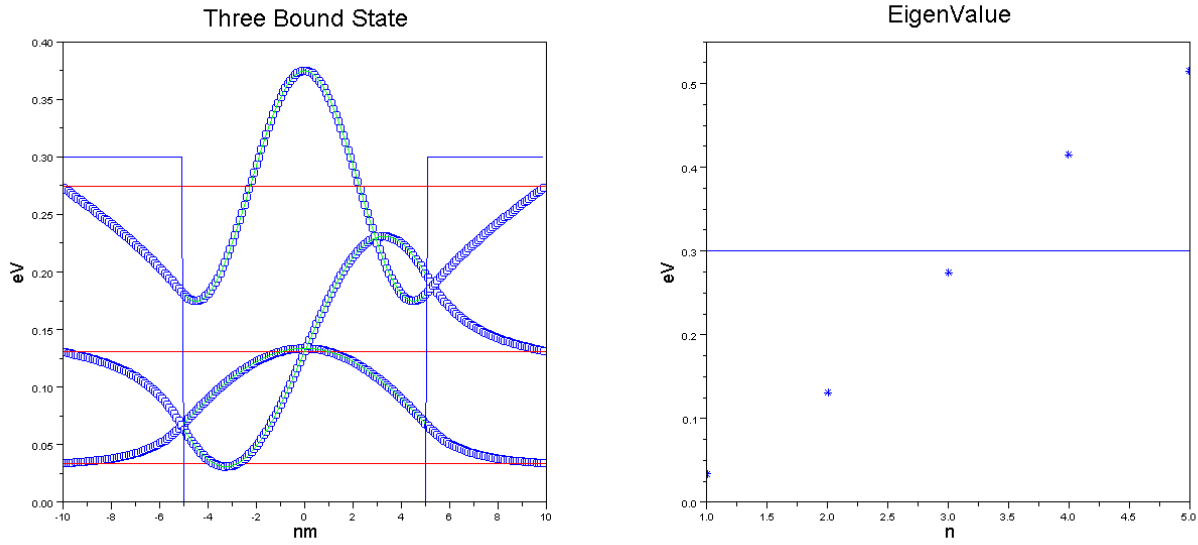
```

A.1.4 思考問題

1. 如果要用來模擬週期性結構，矩陣要如何修改？
2. 使用更高階的 FWM 進行計算：

$$\frac{\partial^2 \phi_i}{\partial z^2} = \frac{-\phi_{i-2} + 16\phi_{i-1} - 30\phi_i + 16\phi_{i+1} - \phi_{i+2}}{12(\Delta z)^2} \quad (\text{A.9})$$

3. 模擬不同的 potential well，比方說 Harmonic, pöschl-Teller 或是 Morse potential，請參考 Supplementary Notes on General Physics, Jyhpyng Wang 內 Bound States in Quantum Models 之章節。

Figure A.2: 以 $\Delta z = 1 \text{ nm}$ 進行模擬

4. 針對 tridiagonal Matrix 的性質自行撰寫一個計算特徵值與特徵向量的子程式。

A.2 四階 Runge-Kutta 法的證明

A.2.1 推導過程

參考《數值方法與程式》林聰悟、林佳慧一書中第十二章有關常微分方程數值解法分析，利用泰勒展開式即可導出 Runge-Kutta 法之係數。假設足碼代表偏微分， $f_t = \partial f / \partial t$ ， $f_{yy} = \partial^2 f / \partial y^2$ ，全微分（對 t ）以 f' 表示。已知 $dy/dt = f(t, y)$ ，

$$y_{i+1} = y_i + hf_i + h^2 \frac{f'_i}{2!} + h^3 \frac{f''_i}{3!} + h^4 \frac{f'''_i}{4!} + O(h^5) \quad (\text{A.10})$$

其中

$$\begin{aligned}
f' &= f_t + f f_y \\
f'' &= f_{tt} + 2f f_{ty} + f^2 f_{yy} + f_t f_y + f f_y^2 \\
f''' &= (f_{ttt} + 3f f_{tty} + 3f^2 f_{tyy} + f^3 f_{yyy}) + f_y (f_{tt} + 2f f_{ty} + f f_{yy}) \\
&\quad + 3(f_t f_{ty} + f f_y f_{ty} + f f_t f_{yy} + f^2 f_y f_{yy}) + (f_t f_y^2 + f f_y^3)
\end{aligned} \tag{A.11}$$

我們將使用 $y_{i+1} = y_i + h(\gamma_0 f_i + \gamma_1 f_j^a + \gamma_2 f_j^b + \gamma_3 f_j^c)$ 來計算下個 step 的函數值，其中

$$\begin{aligned}
f_i &= f(t_i, y_i) \\
f_i^a &= f(t_i + \alpha_1 h, y_i + \beta_1 f_i h) \\
f_i^b &= f(t_i + \alpha_2 h, y_i + \beta_2 f_i h + \beta_3 f_i^a h) \\
f_i^c &= f(t_i + \alpha_3 h, y_i + \beta_4 f_i h + \beta_5 f_i^a h + \beta_6 f_i^b h)
\end{aligned} \tag{A.12}$$

我們同樣可以將 f_i^a, f_i^b, f_i^c 展開成泰勒級數，取四階近次似後再代回 Eq. (A.10)，透過比較係數找出適合的未定係數 $\alpha_{i,j}$ 與 γ_j 之值。

$$\begin{aligned}
f_i^a &= f^a(0) + h f^{a'}(0) + h^2 \frac{f^{a''}(0)}{2!} + h^3 \frac{f^{a'''}(0)}{3!} + o(h^4) \\
f^a(0) &= f \\
f^{a'}(0) &= \alpha_1 f_t + f f_y \\
f^{a''}(0) &= \alpha_1^2 f_{tt} + 2\alpha_1 \beta_1 f f_{ty} + \beta_1^2 f^2 f_{yy} \\
f^{a'''}(0) &= \alpha_1^3 f_{ttt} + 3\alpha_1^2 \beta_1 f f_{tty} + 3\alpha_1 \beta_1^2 f^2 f_{tyy} + \beta_1^3 f^3 f_{yyy}
\end{aligned} \tag{A.13}$$

f_i^b 與 f_j^c 類推。比較包含未定係數的計算式與其泰勒展開式同階次的係數，可以得到 11 條等式。由於總共只有 9 個變數，四階精度的 Runge-Kutta 法有多種係數組合可選擇。

Figure A.3: 常用係數組合

	α_1	α_2	α_3	β_1	β_2	β_3	β_4	β_5	β_6	γ_0	γ_1	γ_2	γ_3
Runge	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	1	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$
Kutta	$\frac{1}{3}$	$\frac{2}{3}$	1	$\frac{1}{3}$	$-\frac{1}{3}$	1	1	-1	1	$\frac{1}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{1}{8}$

$$\begin{aligned}
\alpha_1 &= \beta_1 \\
\alpha_2 &= \beta_2 + \beta_3 \\
\alpha_3 &= \beta_4 + \beta_5 + \beta_6 \\
f : 1 &= \gamma_0 \gamma_1 + \gamma_2 + \gamma_3 \\
f_t + f f_y : \frac{1}{2} &= \gamma_1 \alpha_1 + \gamma_2 \alpha_2 + \gamma_3 \alpha_3 \\
f_t t + 2f f_{ty} + f^2 f_{yy} : \frac{1}{3} &= \gamma_1 \alpha_1^2 + \gamma_2 \alpha_2^2 + \gamma_3 \alpha_3^2 \\
f_{ttt} + 3f f_{f t t y} + 3f^2 f_{t y y} + f^3 f_{y y y} : \frac{1}{4} &= \gamma_2 \alpha_1 \beta_3 + \gamma_3 (\alpha_1 \beta_5 + \alpha_{25}) \\
f_y (f_{tt} + 2f f_{ty} + f^2) f_{yy} : \frac{1}{12} &= \gamma_2 \alpha_1^2 \beta_3 + \gamma_3 (\alpha_1^2 \beta_5 + \alpha_{25}^2) \\
f_{ty} (f_t + f f_y) + f_{yy} (f f_t + f^2 f_y) : \frac{1}{8} &= \gamma_2 \alpha_1 \alpha_1 \beta_3 + \gamma_3 \alpha_3 (\alpha_1 \beta_5 + \alpha_{25}) \\
f_t f_y^2 + f f_y^3 : \frac{1}{24} &= \gamma_3 \alpha_1 \beta_3 \beta_6
\end{aligned}$$

(A.14)

A.2.2 思考問題

1. 試著自己推導出完整的計算過程
2. 試著推導五階 Runge-Kutta 法的係數