

國立交通大學

電子物理系

碩士論文

單一原子石墨層電子結構的計算模擬及大型矩陣對
角化技術研究

Simulation of the electronic structure of graphene nanostructures and
studies of the techniques of large scale matrix diagonalization

研究生：陳勇達

指導教授：鄭舜仁 教授

中華民國九十八年七月

單一原子石墨層電子結構的計算模擬及大型矩陣對角化技術研究

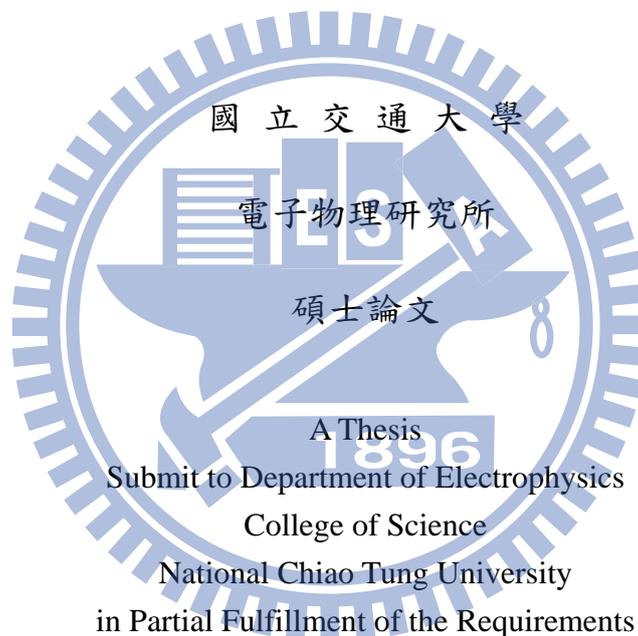
Simulation of the electronic structure of graphene nanostructures and
studies of the techniques of large scale matrix diagonalization

研究生：陳勇達

Student : Yung-Da Chen

指導教授：鄭舜仁 教授

Advisor : Shun-Jen Cheng



in Partial Fulfillment of the Requirements

for the Degree of

Master

in

Electrophysics

April 2009

HsinChu, Taiwan, Republic of China

中華民國九十八年七月

致謝

感謝鄭舜仁老師兩年來不厭其煩的細心教導，讓我在研究上能夠更上層樓，也讓我了解固態領域的淵博，並且從而學習到嚴謹的態度，了解每個式子更深層的意涵，也感謝老師在各個方面對我的教導，受益匪淺。感謝關肇正老師、邱博文老師、簡紋濱老師於口試時提出的寶貴意見。

在學兩年期間，非常感謝書楷學長在各個方面給我的幫忙與意見，也感謝彥廷學長、志彬學長帶領我進研究室，還有下來拿便當的上瑜學長；感謝同窗虔震、文廷、瑞甫你們不時的幫忙與熱烈討論，解開我許多疑惑；感謝學弟們克銘、禹淮、徐燁、浣鈞你們在實驗室的幫忙，讓我能好好完成論文。感謝我的父母支持我，讓我一路完成學業；感謝采玲，豐富了我的感情生活，一路在我艱苦的時候支持我。最後，感謝在我碩士生活中曾經幫過我的每一位，讓我有個美好的碩士生涯。

單一原子石墨層電子結構的計算模擬及大型矩陣對角化技術研究

學生：陳勇達

指導教授：鄭舜仁 博士

國立交通大學電子物理研究所碩士班

摘要

在這份論文中，我們將計算奈米尺度的單一原子石墨層(graphene)的電子結構並研究大規模矩陣(Large scale matrix)對角化的技術，首先我們介紹 Tight-Binding 的基本理論，建立 K 空間與實空間矩陣之後，我們建構出單一原子石墨層的奈米結構(graphene & graphene nanoribbon)的 Tight-Binding 矩陣，並只計算出單一原子石墨層與 nanoribbon 的能帶結構。接著再介紹大規模稀疏矩陣(Large scale sparse matrix)對角化技術中的 LANCZOS 以及使用最多時間研究的 ARPACK 演算法，應用在簡單的一維簡諧振子問題並利用有限差分法(finite difference)計算其能階，檢驗其準確度，會隨著矩陣增大而增加其準確度，使用一般配置 8G 記憶體電腦，目前 ARPACK 可對角化約十五萬乘十五萬的大矩陣，未來可以計算更大的矩陣。

Simulation of the electronic structure of graphene nanostructures and studies of the techniques of large scale matrix diagonalization

Student : Yung-Da Chen

advisor : Shun-Jen Cheng

Department of Electrophysics
National Chiao Tung University

ABSTRACT

In this thesis, we theoretically study the electronic structures of graphene nanostructures and the techniques of numerical diagonalization of large scale matrix, i.e., Lanczos and Arpack eigensolvers. In the first part of this thesis, the electronic structures of infinite two-dimensional(2D) graphene systems and one-dimensional(1D) graphene nano-ribbons are calculated by using one-orbital tight binding theory. The general Hamiltonian matrices for the 2D and 1D graphene systems are explicitly derived. The electronic structures of the graphene nano-systems are calculated by carrying out direct diagonalization. According to the study, we identify the localization of electron density at the zigzag edge in the zigzag ribbon. In the second part, we review the basic theory and algorithm of Lanczos and ARPACK eigensolvers for the diagonalization of large scale sparse matrix. The ARPACK package is applied to solve the problem of 1D simple harmonic oscillation. The maximum size of matrix diagonalized by the package is 8 million by 8 million on PC with 8G memory. Typically, the high accuracy 0.00002 % for the numerically calculated ground states can be achieved.

目 錄

致謝	I
中文摘要	II
英文摘要	III
目錄	IV
第一章、導論	1
1.1 graphene 簡介	1
1.2 研究動機	1
1.3 章節概要	2
第二章、Tight-binding Method	3
2.1 Tight-binding 理論	3
2.2 k-space Hamiltonian	5
2.3 Real space Hamiltonian	11
第三章、數值結果	13
3.1 graphene	13
3.2 graphene nanoribbon	17
第四章、數值方法	24
4.1 Lanczos Method	25
4.2 Arpack Method	30
第五章、結論與未來工作	44
參考文獻	46
附錄 A	47
附錄 B	50

第一章、導論

1.1 graphene 簡介

單層石墨(graphene)為一個單一原子厚度、外觀為蜂巢狀結構的石墨薄膜，這樣的材料從2004年英國的曼徹斯特(Manchester)大學的Novoselov K.S發表了可以利用力學的方法製作出單層石墨(graphene)[1,2]之後，開始引起大家廣泛的注意，其中，在單層石墨中的電子必須用滿足相對論的Dirac fermion去描述，在導電帶與價電帶交會的Dirac point上，能量與動量有線性的關係，且電子在單一石墨層上群速度移動接近光速的三百分之一，相當高的電子傳輸速度有可能成為新一代的電晶體材料。量子霍爾效應(quantum hall effect)[3,4]，也是單層石墨發現有趣的特性之一，包含了室溫下便能觀測到的整數量子霍爾效應，以及2007年有文章宣稱觀測到分數量子霍爾效應[5,6,7]，也是最近新穎材料的研究當中最令人訝異的，有如此特性的材料，開啟了各種研究的大門。

1.2 研究動機

主要的研究動機，是發展大規模矩陣的計算工具，以利後人能夠以現有電腦速度與規模，用於計算由原子數決定矩陣大小的電子結構，對於奈米材料的計算中，Tight-Binding方法去計算能帶理論時，在以往大約只能算到幾奈米大小，若現今引入此技術，大約能夠計算數十奈米，甚至到一百奈米的大小，大幅度增加計算尺度，所以此技術對研究是非常重要的。在

開始發展之後，便能使用此計算方法及此數值計算軟體去模擬新穎材料單層石墨的計算，以及單層石墨奈米緞帶(graphene nanoribbon)，了解此兩種材料的電子結構。

1.3 章節概要

第一章簡單介紹完單層石墨之後，第二章便會開始介紹本篇論文的主要理論的計算方法 Tight-Binding 法，針對於物理上的解釋，以及滿足 Bloch 理論的波函數，進而推導出 K 空間的 Hamiltonian，這部份是計算塊材主要的理論。但是針對實際上材料大小並非無窮大，所以在實際材料上必須考慮到材料邊界的效應，便產生在實空間中所需的理論，進而產生實空間的 Hamiltonian。也由於在實空間中，矩陣的大小會隨著材料增大而增加，因此在數值方法中，如何縮減矩陣元素，甚至是縮減矩陣大小，以增加數值運算的大小，提升目前電腦規格能夠計算的範圍，便是一個很重要的課題。所以第三章會介紹數值方法常用的 Lanczos[8,9]與 ARPACK[9,10]，其中 Lanczos 是將矩陣縮減成三對角線矩陣(tridiagonal matrix)，而 ARPACK 不僅僅縮減矩陣，還利用到稀疏矩陣的記錄方式，以及切割矩陣的大小，用以分段計算。而第四章開始使用 Tight-Binding 方法計算單層石墨(graphene)與單層石墨奈米緞帶(graphene nanoribbon) [11,12]的電子結構，一窺究竟。第五章是結論與其後續的研究工作內容。

第二章、Tight-Binding Method

這一章我們主要介紹的是我們基本的理論架構，如何將單電子在晶格中的漢米爾量(Hamiltonian)化簡，其滿足 Bloch 函數的波函數形式，使用 Tight-Binding 方法需要有哪些近似?皆會在這一章探討。

2.1 Tight-Binding Method

從量子力學中，可以了解在單電子單原子的電子能量與波函數的關係，為一特徵值方程，如(2.1.1)

$$H_0\varphi_\alpha(\vec{r}) = \left[-\frac{\hbar^2}{2m}\nabla^2 + V(\vec{r}) \right] \varphi_\alpha(\vec{r}) = \varepsilon_\alpha\varphi_\alpha(\vec{r}) \quad (2.1.1)$$

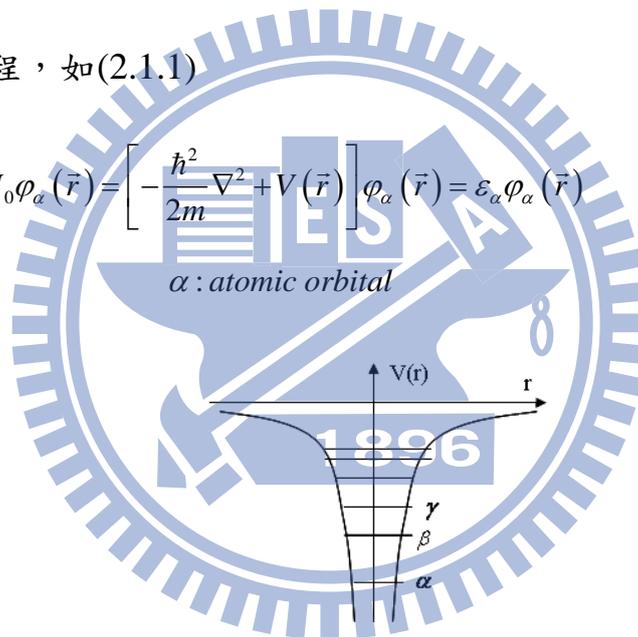


圖 2.1.1、單原子單電子位能

這是一個在量子力學中已知答案的問題，我們了解氫原子系統(單原子單電子)的能量與波函數，但在固態物理中，並沒有非常了解單電子在多原子中的行為，如果，我們能用單原子系統描述多原子系統，便可以簡化在固態物理中複雜的情形。

在固態的系統中，我們知道它是由原子以週期性來排列的。換句話說，

電子或其他帶電荷的粒子在固態的系統中所受到的位能也是成週期性的排列如公式(2.1.2)式

$$U(\vec{r}) = U(\vec{r} + \vec{R}) \quad (2.1.2)$$

$$\vec{R} = n_1 \hat{a}_1 + n_2 \hat{a}_2 + n_3 \hat{a}_3$$

$U(\vec{r})$ 是來自晶格的位能，而不是外加的位能。 \vec{R} 是一個滿足晶格週期的向量，其中 n_1, n_2, n_3 是整數，而 $\hat{a}_1, \hat{a}_2, \hat{a}_3$ 是晶格向量。根據 Bloch's theorem，當電子在週期性的位能中，它的波函數可以表示成 Bloch's function，如(2.1.3)式

$$\psi_{\vec{k}}(\vec{r}) = e^{i\vec{k} \cdot \vec{r}} u_{\vec{k}}(\vec{r}) \quad (2.1.3)$$

$$u_{\vec{k}}(\vec{r} + \vec{R}) = u_{\vec{k}}(\vec{r})$$

在方程式中 $e^{i\vec{k} \cdot \vec{r}}$ 所表示的是電子在晶體中具有平面波的特性，而 $u_{\vec{k}}(\vec{r})$ 是一個週期為 \vec{R} 的函數。所以，系統 Hamiltonian 可以分成單原子系統與微擾項，其中微擾來自於位能影響。如(2.1.4)式

$$\begin{aligned} H &= -\frac{\hbar^2}{2m} \nabla^2 + U(\vec{r}) \\ &= \left(-\frac{\hbar^2}{2m} \nabla^2 + V(\vec{r}) \right) + (U(\vec{r}) - V(\vec{r})) \\ &= H_0 + \Delta U(\vec{r}) \end{aligned} \quad (2.1.4)$$

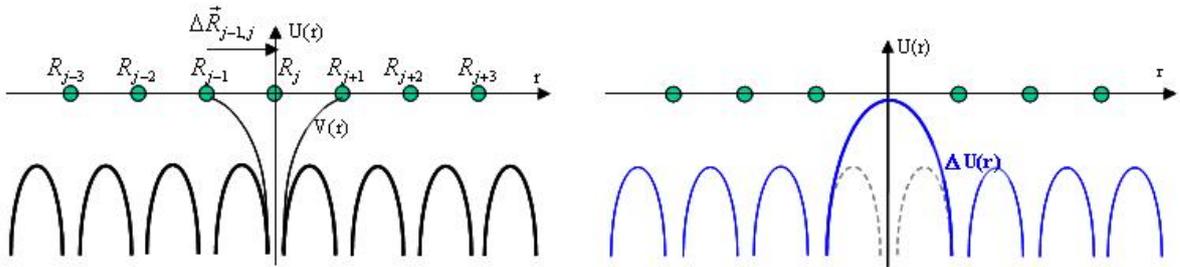


圖 2.1.2、晶格位能(不連續)， R_j 代表 unit cell 所在的位置

將(2.1.4)代入薛丁格方程，可得：

$$H\psi_{\vec{k}}(\vec{r}) = [H_0 + \Delta U(\vec{r})]\psi_{\vec{k}}(\vec{r}) = E\psi_{\vec{k}}(\vec{r}) \quad (2.1.5)$$

2.2 k-space Hamiltonian

在此這樣的位能下，滿足(2.1.3)的波函數如下：[13,14]

$$\psi_{\vec{k}}(\vec{r}) = \frac{1}{\sqrt{N}} \sum_{j=1}^N e^{i\vec{k}\cdot\vec{R}_j} \phi(\vec{r} - \vec{R}_j) \quad (2.2.1)$$

滿足晶格週期性的波函數，可看成 Wannier 函數 $\phi(\vec{r} - \vec{R}_j)$ 乘上相位 $e^{i\vec{k}\cdot\vec{R}_j}$ 的線性組合，其中 N 是總原子數， $\frac{1}{\sqrt{N}}$ 是歸一化常數，其意義為我們利用很多個相同的單位晶胞去表達整個空間，兩兩不同晶胞之間距離為 \vec{R}_j ，分布在第 j 個單位晶胞內 Wannier 函數 $\phi(\vec{r} - \vec{R}_j)$ (用來表達電子在單一晶胞中的行為)，再做線性疊加，即總波函數 $\psi_{\vec{k}}(\vec{r})$ 。

要表達電子在單位晶胞內只有單原子中的行為，Wannier 函數可以利用原子軌域的線性疊加 LCAO(Linear Combination of Atomic Orbital)，如(2.2.2)

$$\phi(\vec{r}) = \sum_{\alpha} C_{\alpha} \varphi_{\alpha}(\vec{r}) \quad (2.2.2)$$

代入(2.1.4)，重新整理可得：

$$\psi_{\vec{k}}(\vec{r}) = \sum_{\alpha} C_{\alpha} \Phi_{\alpha,\vec{k}}(\vec{r}) \quad (2.2.3)$$

$$\Phi_{\alpha,\vec{k}}(\vec{r}) \equiv \frac{1}{\sqrt{N}} \sum_{j=1}^N e^{i\vec{k}\cdot\vec{R}_j} \varphi_{\alpha}(\vec{r} - \vec{R}_j) \quad (2.2.4)$$

式(2.2.4)稱為 *Tight-Binding Bloch function*，為展開波函數的基底。

將(2.2.3)、(2.2.4)代入(2.1.4)，並乘上 $\Phi_{\alpha',\vec{k}}^*(\vec{r})$ 取積分，可得：

$$\sum_{\alpha} C_{\alpha} \frac{1}{N} \sum_j \sum_{j'} e^{i\vec{k}\cdot(\vec{R}_j - \vec{R}_{j'})} \int \varphi_{\alpha'}^*(\vec{r} - \vec{R}_{j'}) H \varphi_{\alpha}(\vec{r} - \vec{R}_j) d\vec{r} = \sum_{\alpha} C_{\alpha} E \int \Phi_{\alpha',\vec{k}}^*(\vec{r}) \Phi_{\alpha,\vec{k}}(\vec{r}) d\vec{r} \quad (2.2.5)$$

處理右手邊的積分時，假設不同原子之間波函數不重疊，並利用不同軌域之間互相正交(2.2.6)。

$$\int \varphi_{j'\alpha'}^*(\vec{r}) \varphi_{j\alpha}(\vec{r}) d\vec{r} = \delta_{jj'} \delta_{\alpha'\alpha} \quad (2.2.6)$$

處理左手邊積分時，考慮任意一顆原子週遭環境所產生的影響皆相同，則可以消去一個總和(summation)，便可以得到下式(2.2.7)：

$$\sum_{\alpha} C_{\alpha} \sum_{j'} e^{i\vec{k}\cdot(\vec{R}_j - \vec{R}_{j'})} \int \varphi_{\alpha'}^*(\vec{r} - \vec{R}_{j'}) H \varphi_{\alpha}(\vec{r} - \vec{R}_j) d\vec{r} = \sum_{\alpha} C_{\alpha} E \delta_{\alpha\alpha'} \quad (2.2.7)$$

經過一些簡單的計算與代換，可得

$$\sum_{\alpha} C_{\alpha} H_{\alpha\alpha'}(\vec{k}) = E C_{\alpha'} \quad (2.2.8)$$

$$H_{\alpha\alpha'}(\vec{k}) \equiv \sum_{\vec{d}_i} e^{i\vec{k}\cdot\vec{d}_i} \int \varphi_{\alpha'}^*(\vec{r}') H \varphi_{\alpha}(\vec{r}' - \vec{d}_i) d\vec{r}' \quad (2.2.9)$$

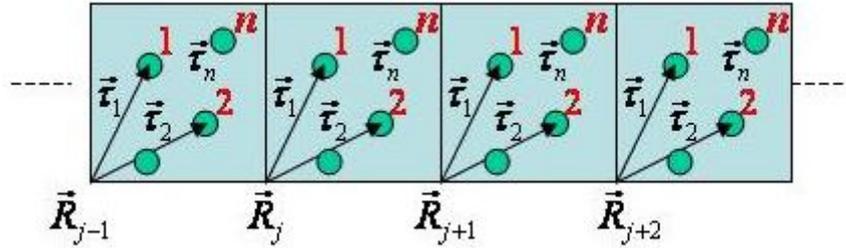
其中， \vec{d}_i 為其他原子與參考原子的距離。當我們將一個一個原子慢慢組合成晶體，並考慮原子間的交互作用時，可以將參考原子週遭的原子依距離分成第一近鄰(Nearest Neighbor)、第二近鄰(2NN)、第三近鄰(3NN)...等，考慮愈多，精確度愈高，計算便會更複雜。

將(2.2.8)用矩陣表示，如(2.2.11)

$$\begin{bmatrix} \langle s|H|s \rangle & \langle s|H|p_x \rangle & \cdots & \langle s|H|\alpha \rangle \\ \langle p_x|H|s \rangle & \langle p_x|H|p_x \rangle & \cdots & \langle p_x|H|\alpha \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \alpha|H|s \rangle & \langle \alpha|H|p_x \rangle & \cdots & \langle \alpha|H|\alpha \rangle \end{bmatrix} \begin{bmatrix} C_s \\ C_{p_x} \\ \vdots \\ C_{\alpha} \end{bmatrix} = E \begin{bmatrix} C_s \\ C_{p_x} \\ \vdots \\ C_{\alpha} \end{bmatrix} \quad (2.2.11)$$

這是單位晶胞中具有一個原子的矩陣。

讓我們討論更一般的狀況，如果一個單位晶胞中有 n 個原子，則該單位晶胞的電子波函數應與每個原子有關，其 Wannier 函數應修正為[13,14]：



$$\phi(\vec{r}) = \sum_{n\alpha} C_{n\alpha} e^{i\vec{k}\cdot\vec{r}_n} \varphi_{n\alpha}(\vec{r} - \vec{r}_n) \quad (2.2.12)$$

圖 2.2.1、單位晶胞有 n 個原子示意圖

(2.2.12) 中 $\varphi_{n\alpha}$ 為第 n 顆原子的 α 軌域， \vec{r}_n 為相對於晶格向量 \vec{R}_j 的位置向量，並且乘上相對於 \vec{R}_j 的相位。再將 Wannier 函數代入滿足 Bloch 函數的波函數，可得：

$$\psi_{\vec{k}}(\vec{r}) = \sum_{n\alpha} C_{n\alpha} \Phi_{n\alpha,\vec{k}}(\vec{r}) \quad (2.2.13)$$

$$\Phi_{n\alpha,\vec{k}}(\vec{r}) \equiv \frac{1}{\sqrt{N}} \sum_{j=1}^N e^{i\vec{k}\cdot\vec{R}_{jn}} \varphi_{n\alpha}(\vec{r} - \vec{R}_{jn}) \quad \vec{R}_{jn} = \vec{R}_j + \vec{r}_n \quad (2.2.14)$$

將(2.2.13)、(2.2.14)代入 (2.1.4)，並乘上 $\Phi_{n'\alpha',\vec{k}}^*(\vec{r})$ 取積分，可得：

$$\int \Phi_{n'\alpha',\vec{k}}^*(\vec{r}) H \sum_{n\alpha} C_{n\alpha} \Phi_{n\alpha,\vec{k}}(\vec{r}) d\vec{r} = \int \Phi_{n'\alpha',\vec{k}}^*(\vec{r}) E \sum_{n\alpha} C_{n\alpha} \Phi_{n\alpha,\vec{k}}(\vec{r}) d\vec{r} \quad (2.2.14)$$

等式右邊，假設兩兩不同原子之間的波函數重疊的範圍很小，趨近於零，

且不同軌域之間互相正交(2.2.15)：

$$\int \varphi_{j'\alpha'}^*(\vec{r}) \varphi_{j\alpha}(\vec{r}) d\vec{r} = \delta_{jj'} \delta_{\alpha'\alpha} \quad (2.2.15)$$

所以可得

$$\sum_{n\alpha} C_{n\alpha} \frac{1}{N} \sum_j \sum_{j'} e^{i\vec{k}\cdot(\vec{R}_j - \vec{r}_n - \vec{R}_{j'} + \vec{r}_{n'})} \int \varphi_{n'\alpha'}^*(\vec{r} - \vec{R}_{j'} - \vec{r}_{n'}) H \varphi_{n\alpha}(\vec{r} - \vec{R}_j - \vec{r}_n) d\vec{r} = \sum_{n\alpha} C_{n\alpha} E \delta_{n\alpha} \delta_{\alpha\alpha'} \quad (2.2.16)$$

利用 $\vec{r}' \equiv \vec{r} - \vec{R}_{j'} - \vec{r}_{n'}$ 與 $\vec{d}_i \equiv \Delta\vec{R}_{jn;j'n'} = (\vec{R}_j - \vec{r}_n) - (\vec{R}_{j'} - \vec{r}_{n'})$ ，此時所有相對位置關係已轉成原子之間的近鄰關係，則可得下式

$$\sum_{\alpha} C_{\alpha} H_{n\alpha;n'\alpha'} = E C_{n'\alpha'} \quad (2.2.17)$$

$$H_{n\alpha;n'\alpha'}(\vec{k}) \equiv \sum_{\vec{d}_i} e^{i\vec{k}\cdot\vec{d}_i} \int \varphi_{n'\alpha'}^*(\vec{r}') H \varphi_{n\alpha}(\vec{r}' - \vec{d}_i) d\vec{r}' \quad (2.2.18)$$

(2.2.16)經過整理，可得到 Hamiltonian 的通式：

$$H_{n\alpha;n'\alpha'}(\vec{k}) \equiv \left(\varepsilon_{n\alpha} + \int \varphi_{n'\alpha'}^*(\vec{r}') \Delta U \varphi_{n\alpha}(\vec{r}') d\vec{r}' \right) \delta_{n\alpha} \delta_{\alpha\alpha'} + \sum_{\vec{d}_i \neq 0} e^{i\vec{k}\cdot\vec{d}_i} \int \varphi_{n'\alpha'}^*(\vec{r}') \Delta U \varphi_{n\alpha}(\vec{r}' - \vec{d}_i) d\vec{r}' \quad (2.2.19)$$

若表達成矩陣形式，如下：

$$\begin{bmatrix} \langle 1,s|H(k)|1,s\rangle & \langle 1,s|H(k)|1,p_x\rangle & \cdots & \langle 1,s|H(k)|2,s\rangle & \langle 1,s|H(k)|2,p_x\rangle & \cdots \\ \langle 1,p_x|H(k)|1,s\rangle & \langle 1,p_x|H(k)|1,p_x\rangle & \cdots & \langle 1,p_x|H(k)|2,s\rangle & \langle 1,p_x|H(k)|2,p_x\rangle & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ \langle 2,s|H(k)|1,s\rangle & \langle 2,s|H(k)|1,p_x\rangle & \cdots & \langle 2,s|H(k)|2,s\rangle & \langle 2,s|H(k)|2,p_x\rangle & \cdots \\ \langle 2,p_x|H(k)|1,s\rangle & \langle 2,p_x|H(k)|1,p_x\rangle & \cdots & \langle 2,p_x|H(k)|2,s\rangle & \langle 2,p_x|H(k)|2,p_x\rangle & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} C_s(k) \\ C_{p_x}(k) \\ \vdots \\ C_s(k) \\ C_{p_x}(k) \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} = E \begin{bmatrix} C_s(k) \\ C_{p_x}(k) \\ \vdots \\ C_s(k) \\ C_{p_x}(k) \\ \vdots \\ \vdots \\ \vdots \end{bmatrix} \quad (2.2.20)$$

此矩陣大小為 $N\alpha \times N\alpha$ ，由原子個數與軌域個數決定。若依照不同原子將此

矩陣元素分類，則此複雜的矩陣可看成下式：

$$\begin{array}{c} \text{atom} \\ 1 \quad 2 \quad \cdots \end{array} \begin{bmatrix} H_{11} & H_{12} & \cdots \\ H_{21} & H_{22} & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \end{bmatrix} = E \begin{bmatrix} C_1 \\ C_2 \\ \vdots \end{bmatrix} \quad (2.2.21)$$

其中 H_{11} 與 H_{22} 代表 on-site 能量，非對角線 H_{12} 與 H_{21} 為不同原子之間的

hopping 項，這矩陣表示了每個原子之間的關係與其交互作用，但在複雜的 Hamiltonian 中並不了解如何將式子填入矩陣中。若今有兩種一維的原子結構如圖(2.2.2)

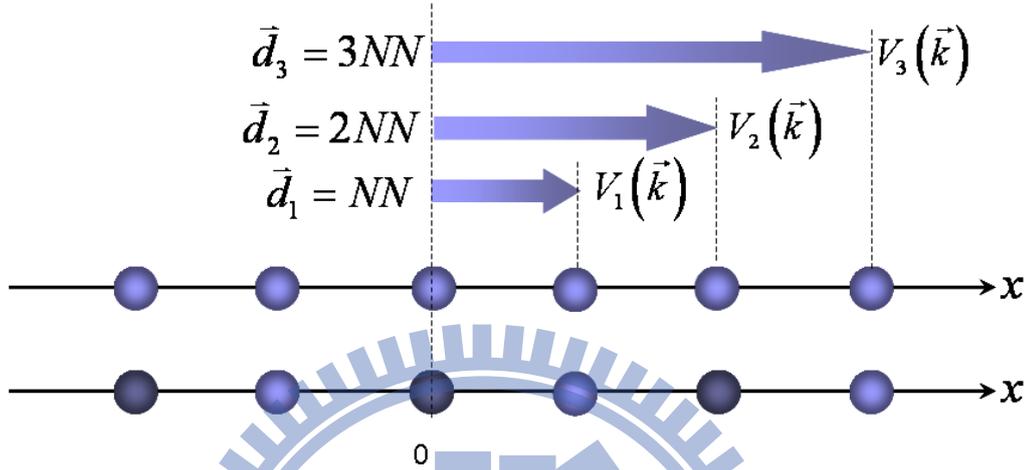


圖 2.2.2、上序列為一維相同原子的排列，下序列為一維相異原子的排列，NN(Nearest Neighbor)為近鄰之意， $V_m(\vec{k})$ 為第 m 近鄰的交互作用項。

若考慮相同原子(單位晶胞中有一顆原子)，將 Hamiltonian 依照原子間距區分為單原子能量、第一近鄰(Nearest Neighbor) \vec{d}_1 、第二近鄰(2NN) \vec{d}_2 、第三近鄰(3NN) \vec{d}_3 等等，所以(2.2.19)可以寫成(2.2.22)式

$$H(\vec{k}) = E_\alpha \cdot \delta_{\alpha\alpha'} + V_1(\vec{k}) + V_2(\vec{k}) + \dots \quad (2.2.22)$$

$V_m(\vec{k})$ 如(2.2.23)列舉兩項如下

$$V_1(\vec{k}) \equiv t_{\alpha,\alpha'}(\vec{d}_1) \sum_{\vec{d}_1} e^{i\vec{k} \cdot \vec{d}_1}, \quad t_{\alpha,\alpha'}(\vec{d}_1) \equiv \int \varphi_\alpha^*(\vec{r}) \Delta U \varphi_{\alpha'}(\vec{r} - \vec{d}_1) \cdot d^3 r$$

$$V_2(\vec{k}) \equiv t_{\alpha,\alpha'}(\vec{d}_2) \sum_{\vec{d}_2} e^{i\vec{k} \cdot \vec{d}_2}, \quad t_{\alpha,\alpha'}(\vec{d}_2) \equiv \int \varphi_\alpha^*(\vec{r}) \Delta U \varphi_{\alpha'}(\vec{r} - \vec{d}_2) \cdot d^3 r \quad (2.2.23)$$

其中 $t_{\alpha,\alpha'}(\vec{d}_m)$ 為實驗或使用第一原理所決定的參數

若單位晶胞中有兩顆原子，2x2 的 Hamiltonian 則如下圖(2.2.3)所示

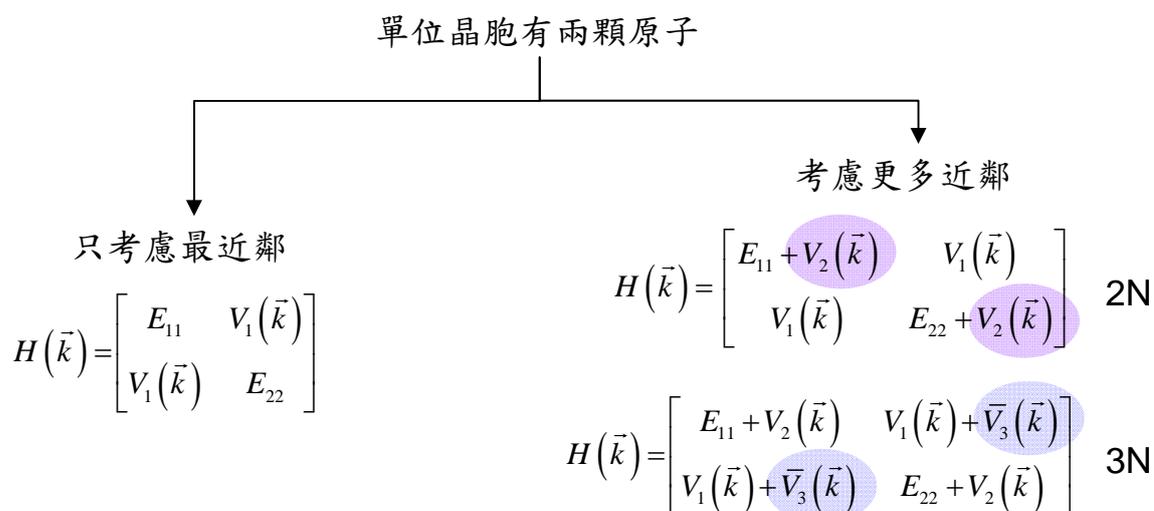


圖 2.2.3、增加近鄰數對矩陣的影響

屆此已可以將任意結構的半導體利用週期性條件將 Hamiltonian 矩陣化，只要知道原子間交互作用的參數、晶體結構，便能決定每一個矩陣元素的數值，在進行對角化，便可以得到相對應的特徵值與特徵向量，稍後在第三章，我們將會針對單位晶胞有兩顆原子的單層石墨(graphene)當為示範。

2.3 Real space Hamiltonian

如果只計算一部份材料，在任意方向上都不具有週期性位能時，波函數便與 K 無關，Bloch 理論便不適用，所以此時薛丁格方程為：

$$H\psi(\vec{r}) = E\psi(\vec{r}) \quad (2.3.1)$$

然而總波函數可以由每一個單電子單原子波函數組成，每一個單電子單原子可以由原子軌域的線性疊加 LCAO(Linear Combination of Atomic Orbital) 所組成，因此其展開形式便假設為：

$$\psi(\vec{r}) = \sum_{j=1}^N \sum_{\alpha} C_{j\alpha} \phi_{j\alpha}(\vec{r}) \quad (2.3.2)$$

將(2.3.2)代入(2.3.1)並乘上 $\phi_{j'\alpha'}^*(\vec{r})$ 取積分，可得

$$\sum_{j=1}^N \sum_{\alpha} C_{j\alpha} \int \phi_{j'\alpha'}^*(\vec{r}) H \phi_{j\alpha}(\vec{r}) d\vec{r} = \sum_{j=1}^N \sum_{\alpha} C_{j\alpha} \int \phi_{j'\alpha'}^*(\vec{r}) E \phi_{j\alpha}(\vec{r}) d\vec{r} \quad (2.3.3)$$

假設兩兩不同原子之間的波函數重疊的範圍很小，趨近於零，且不同軌域之間互相正交：

$$\int \phi_{j'\alpha'}^*(\vec{r}) \phi_{j\alpha}(\vec{r}) d\vec{r} = \delta_{jj'} \delta_{\alpha'\alpha} \quad (2.3.4)$$

(2.3.3)考慮(2.3.4)後，再將 H 拆開，可得：

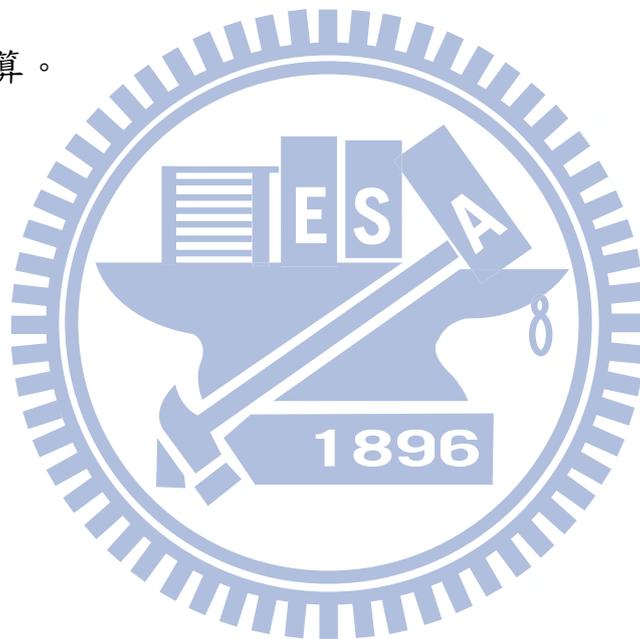
$$\begin{aligned} \sum_{j=1}^N \sum_{\alpha} C_{j\alpha} \left[\varepsilon_{\alpha} \delta_{jj'} \delta_{\alpha'\alpha} + \int \phi_{j'\alpha'}^*(\vec{r}) \Delta U \phi_{j\alpha}(\vec{r}) d\vec{r} \right] &= E C_{j'\alpha'} \\ \Rightarrow \sum_{j=1}^N \sum_{\alpha} C_{j\alpha} H_{j\alpha; j'\alpha'} &= E C_{j'\alpha'} \end{aligned} \quad (2.3.5)$$

$$\Rightarrow H_{j\alpha; j'\alpha'} = \varepsilon_{\alpha} \delta_{jj'} \delta_{\alpha'\alpha} + \int \phi_{j'\alpha'}^*(\vec{r}) \Delta U \phi_{j\alpha}(\vec{r}) d\vec{r} \quad (2.3.6)$$

將(2.3.6)用矩陣表達

$$\begin{bmatrix}
 \langle 1, s | H | 1, s \rangle & \langle 1, s | H | 1, p_x \rangle & \cdots & \langle 1, s | H | 2, s \rangle & \langle 1, s | H | 2, p_x \rangle & \cdots & \cdots \\
 \langle 1, p_x | H | 1, s \rangle & \langle 1, p_x | H | 1, p_x \rangle & \cdots & \langle 1, p_x | H | 2, s \rangle & \langle 1, p_x | H | 2, p_x \rangle & \cdots & \cdots \\
 \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
 \hline
 \langle 2, s | H | 1, s \rangle & \langle 2, s | H | 1, p_x \rangle & \cdots & \langle 2, s | H | 2, s \rangle & \langle 2, s | H | 2, p_x \rangle & \cdots & \cdots \\
 \langle 2, p_x | H | 1, s \rangle & \langle 2, p_x | H | 1, p_x \rangle & \cdots & \langle 2, p_x | H | 2, s \rangle & \langle 2, p_x | H | 2, p_x \rangle & \cdots & \cdots \\
 \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
 \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots
 \end{bmatrix}
 \begin{bmatrix}
 C_s \\
 C_{p_x} \\
 \vdots \\
 C_s \\
 C_{p_x} \\
 \vdots \\
 \vdots
 \end{bmatrix}
 = E
 \begin{bmatrix}
 C_s \\
 C_{p_x} \\
 \vdots \\
 C_s \\
 C_{p_x} \\
 \vdots \\
 \vdots
 \end{bmatrix}
 \quad (2.3.8)$$

至此已經介紹了在 Tight-Binding 法中，如何將複雜的 Hamiltonian 轉為單原子 Hamiltonian 與晶格微擾項，並將其展開為矩陣進行運算，接下來利用 Graphene 來計算。



第三章、單層石墨、單層石墨奈米緞帶

本章節將會展示對基本的單層石墨以及單層石墨奈米緞帶 (graphene nanoribbon) 的計算結果。接下來就介紹一下單層石墨的基本特性。

3.1 Graphene

單一石墨層是由碳以六角形晶格(Hexagonal lattice)所組成[11]，如圖

3.1.1 所示，其晶格向量為：

$$\vec{a}_1 = \frac{a}{2}(3, \sqrt{3}), \vec{a}_2 = \frac{a}{2}(3, -\sqrt{3}) \quad (3.1.1)$$

a 為兩碳原子間距 1.42 \AA ，三個同心圓由內到外為第一近鄰、第二近鄰、第三近鄰，所以單一石墨層的單位晶胞中有兩顆原子。且經由計算，倒晶格空間(Reciprocal Space)的倒晶格向量(Reciprocal lattice vector)為

$$\vec{b}_1 = \frac{2\pi}{3a}(1, \sqrt{3}), \vec{b}_2 = \frac{2\pi}{3a}(1, -\sqrt{3}) \quad (3.1.2)$$

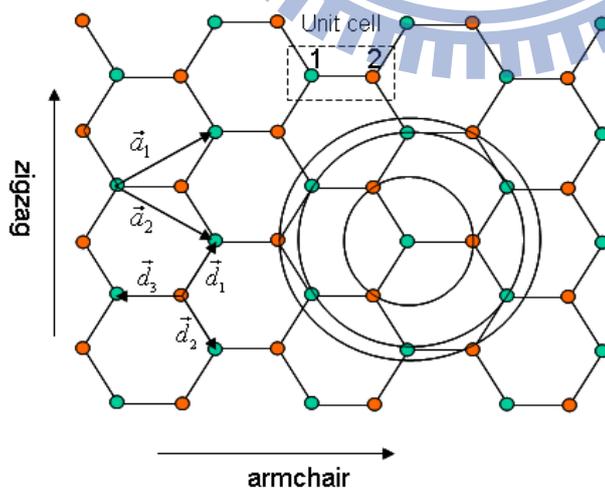


圖 3.1.1、graphene 結構

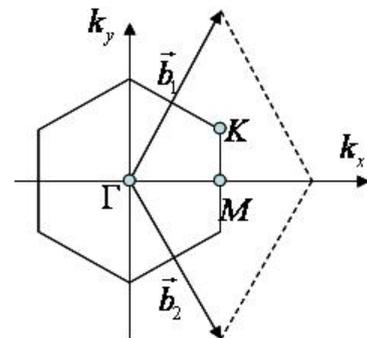


圖 3.1.2、graphene k 空間結構

在 k 空間中，有重要物理意義的點在 K 點上，也就是布里淵區的六角形頂

點上，這被稱為 Dirac Point，其 k 空間座標為：

$$K = \left(\frac{2\pi}{3a}, \frac{2\pi}{3\sqrt{3}a} \right) \quad (3.1.3)$$

第一近鄰的三個向量分別是(依據不同材料有不同向量)

$$\vec{d}_1 = \frac{a}{2}(1, \sqrt{3}), \quad \vec{d}_2 = \frac{a}{2}(1, -\sqrt{3}), \quad \vec{d}_3 = a(-1, 0) \quad (3.1.4)$$

所以便能開始使用第二章 Tight-Binding 法所推導出的 Hamiltonian 公式

(2.2.20)如下

$$H_{n\alpha; n'\alpha'}(\vec{k}) \equiv \left(\varepsilon_{n\alpha} + \int \varphi_{n'\alpha'}^*(\vec{r}') \Delta U \varphi_{n\alpha}(\vec{r}') d\vec{r}' \right) \delta_{m'} \delta_{\alpha\alpha'} + \sum_{\vec{d}_i \neq 0} e^{i\vec{k} \cdot \vec{d}_i} \int \varphi_{n'\alpha'}^*(\vec{r}') \Delta U \varphi_{n\alpha}(\vec{r}' - \vec{d}_i) d\vec{r}'$$

相對原子關係 \vec{d}_i 為(3.1.4)式的三個向量。

其中所使用到的條件有

1. 每個單位晶胞有兩顆原子
2. 單軌域近似(One orbital approximation (Pz))
3. 只考慮第一近鄰(Nearest Neighbor)

產生矩陣大小為 2x2 的矩陣如下

$$\begin{bmatrix} \langle 1, p_z | H | 1, p_z \rangle & \langle 1, p_z | H | 2, p_z \rangle \\ \langle 2, p_z | H | 1, p_z \rangle & \langle 2, p_z | H | 2, p_z \rangle \end{bmatrix} \begin{bmatrix} C_{1p_z} \\ C_{2p_z} \end{bmatrix} = E \begin{bmatrix} C_{1p_z} \\ C_{2p_z} \end{bmatrix} \quad (3.1.5)$$

由於 graphene 本身是 P_z 主導，可以只取 P_z 軌域，矩陣元素的參數由[17]可得

$$\langle 1, p_z | H | 1, p_z \rangle = \langle 2, p_z | H | 2, p_z \rangle = \varepsilon_\alpha + \int \varphi_{n\alpha}^*(\vec{r}') \Delta U \varphi_{n\alpha}(\vec{r}') d\vec{r}' \equiv E_p = 0 \quad (3.1.6)$$

並因有效鍵結為 P_z 所產生的 π 鍵，因此針對非對角線上的元素，定義

$$\int \varphi_{2p_z}^*(\vec{r}') \Delta U \varphi_{1p_z}(\vec{r}' - \vec{d}_i) d\vec{r}' \equiv t = -3.033eV \quad (3.1.7)$$

並考慮以二號原子為中心，與周圍有三顆一號原子有鍵結，所以

$$\langle 1, p_z | H | 2, p_z \rangle = t \left[e^{i\vec{k}\cdot\vec{d}_1} + e^{i\vec{k}\cdot\vec{d}_2} + e^{i\vec{k}\cdot\vec{d}_3} \right] = t \left[e^{-ik_x a} + 2e^{ik_x \frac{a}{2}} \cos \left(k_y \frac{\sqrt{3}}{2} a \right) \right] \quad (3.1.8)$$

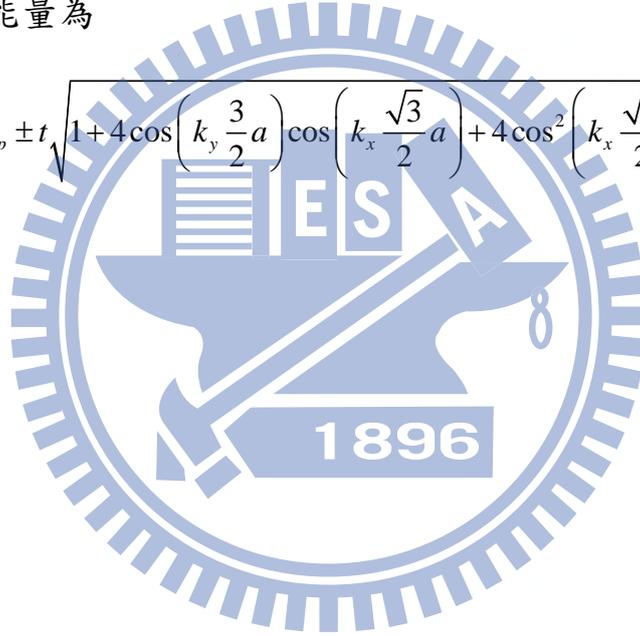
Hamiltonian 便為

$$H = \begin{bmatrix} E_p & t \left[e^{-ik_x a} + 2e^{ik_x \frac{a}{2}} \cos \left(k_y \frac{\sqrt{3}}{2} a \right) \right] \\ t \left[e^{ik_x a} + 2e^{-ik_x \frac{a}{2}} \cos \left(k_y \frac{\sqrt{3}}{2} a \right) \right] & E_p \end{bmatrix} \quad (3.1.9)$$

此結果可從[16]中驗證，在此使用到的 $t=3.033\text{eV}$ 、 $E_p=0$ 參數參考[17]。

所計算出來的能量為

$$E_{\pm} = E_p \pm t \sqrt{1 + 4 \cos \left(k_y \frac{3}{2} a \right) \cos \left(k_x \frac{\sqrt{3}}{2} a \right) + 4 \cos^2 \left(k_x \frac{\sqrt{3}}{2} a \right)} \quad (3.1.10)$$



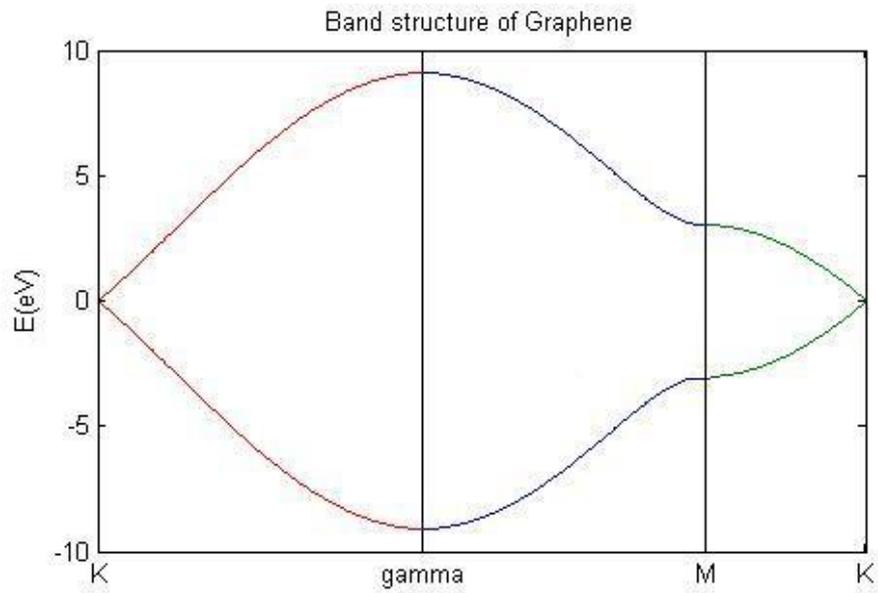


圖 3.1.3、graphene 能帶結構

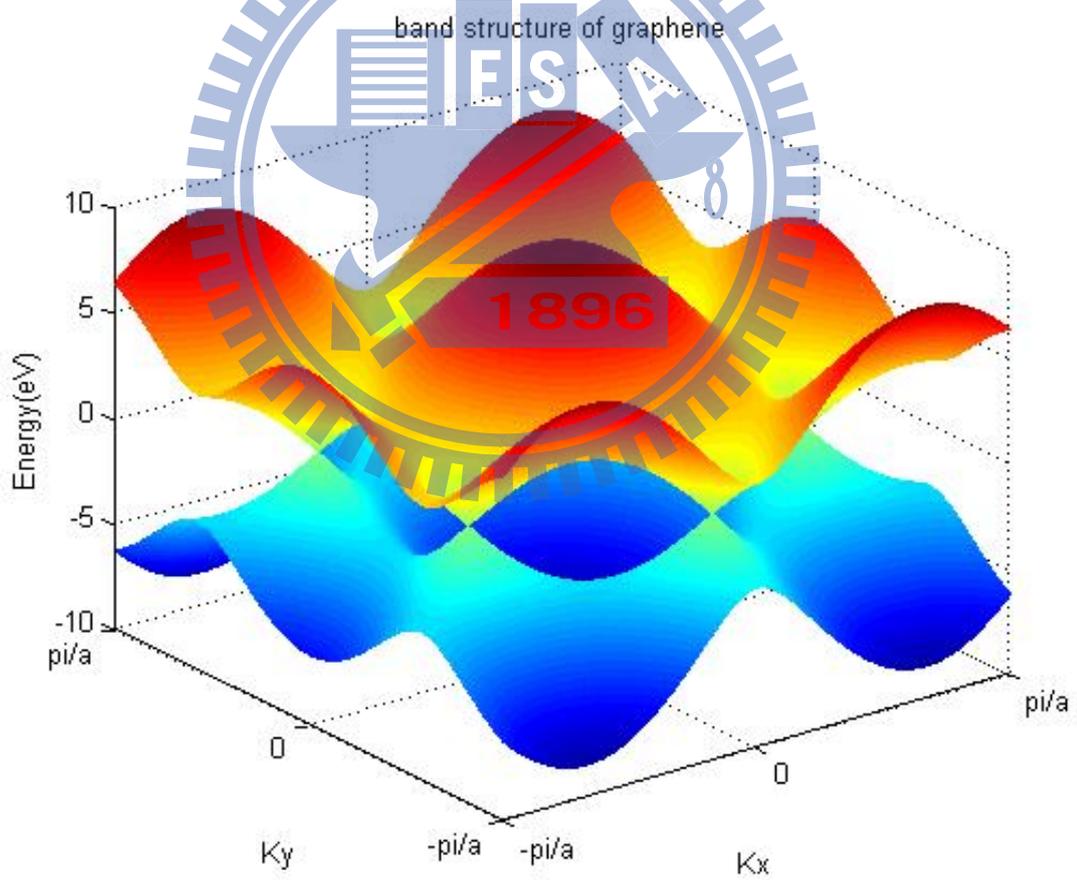


圖 3.1.4、graphene 能帶結構

3.2 graphene nanoribbon

計算完 graphene 之後，若是帶狀結構的 graphene nanoribbon，也就是一個方向無窮延伸，另一個方向有限，所以在 x 方向上具有週期性，y 方向上不具有週期性，所以 Tight-Binding 需要修正 k 只存在 kx 方向。即

$$\vec{k} = (k_x, 0) \quad (3.2.1)$$

因此若考慮傳輸方向為 zigzag，如下圖之結構

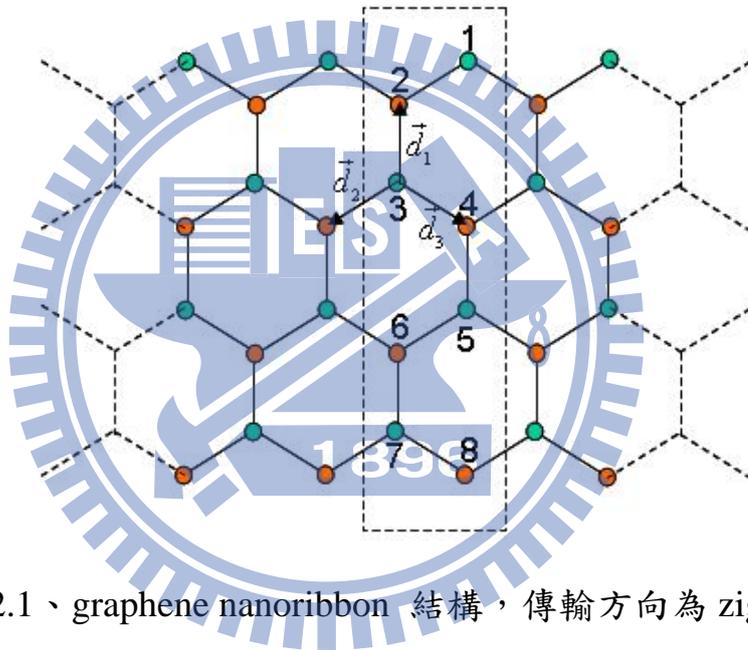


圖 3.2.1、graphene nanoribbon 結構，傳輸方向為 zigzag

在單位晶胞中原子數有八個原子，個別定為一號到八號，所以為 8*8 大小的矩陣，並且每個原子只考慮 Pz 軌域，而每個原子最近鄰的向量為

$$\vec{d}_1 = a(0,1), \quad \vec{d}_2 = \frac{a}{2}(-\sqrt{3}, -1), \quad \vec{d}_3 = \frac{a}{2}(\sqrt{3}, -1) \quad (3.2.2)$$

考慮各個原子之間的鍵結關係之後，並依據(2.2.20)，矩陣元素如下

$$\langle i, p_z | H | i, p_z \rangle = \varepsilon_\alpha + \int \varphi_{n\alpha}^*(\vec{r}') \Delta U \varphi_{n\alpha}(\vec{r}') d\vec{r}' \equiv E_p = 0 \quad i=1,2,\dots,8 \quad (3.2.3)$$

$$\langle i, p_z | H | i+1, p_z \rangle = \langle i+1, p_z | H | i, p_z \rangle^\dagger = \sum_{j' \neq 0} e^{i\vec{k} \cdot \Delta \vec{R}_{jn'; j'n'}} \overbrace{\int \varphi_{n'\alpha'}^*(\vec{r}') \Delta U \varphi_{n\alpha}(\vec{r}' - \Delta \vec{R}_{jn'; j'n'}) d\vec{r}'}^t = \begin{cases} te^{i\vec{k} \cdot \vec{d}_3} + te^{i\vec{k} \cdot \vec{d}_2} & i = 1, 3, 5, 7 \\ te^{-i\vec{k} \cdot \vec{d}_1} & i = 2, 4, 6 \end{cases} \quad (3.2.4)$$

$$\begin{bmatrix} E_p & te^{i\vec{k} \cdot \vec{d}_3} + te^{i\vec{k} \cdot \vec{d}_2} & 0 & 0 & 0 & 0 & 0 & 0 \\ te^{-i\vec{k} \cdot \vec{d}_3} + te^{-i\vec{k} \cdot \vec{d}_2} & E_p & te^{-i\vec{k} \cdot \vec{d}_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & te^{i\vec{k} \cdot \vec{d}_1} & E_p & te^{i\vec{k} \cdot \vec{d}_3} + te^{i\vec{k} \cdot \vec{d}_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & te^{-i\vec{k} \cdot \vec{d}_3} + te^{-i\vec{k} \cdot \vec{d}_2} & E_p & te^{-i\vec{k} \cdot \vec{d}_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & te^{i\vec{k} \cdot \vec{d}_1} & E_p & te^{i\vec{k} \cdot \vec{d}_3} + te^{i\vec{k} \cdot \vec{d}_2} & 0 & 0 \\ 0 & 0 & 0 & 0 & te^{-i\vec{k} \cdot \vec{d}_3} + te^{-i\vec{k} \cdot \vec{d}_2} & E_p & te^{-i\vec{k} \cdot \vec{d}_1} & 0 \\ 0 & 0 & 0 & 0 & 0 & te^{i\vec{k} \cdot \vec{d}_1} & E_p & te^{i\vec{k} \cdot \vec{d}_3} + te^{i\vec{k} \cdot \vec{d}_2} \\ 0 & 0 & 0 & 0 & 0 & 0 & te^{-i\vec{k} \cdot \vec{d}_3} + te^{-i\vec{k} \cdot \vec{d}_2} & E_p \end{bmatrix} \quad (3.2.5)$$

因為 ribbon 本身只有 x 方向上具有週期性，依據(3.2.1)

$$\begin{bmatrix} E_p & 2t \cos(k_x a / 2) & 0 & 0 & 0 & 0 & 0 & 0 \\ 2t \cos(k_x a / 2) & E_p & t & 0 & 0 & 0 & 0 & 0 \\ 0 & t & E_p & 2t \cos(k_x a / 2) & 0 & 0 & 0 & 0 \\ 0 & 0 & 2t \cos(k_x a / 2) & E_p & t & 0 & 0 & 0 \\ 0 & 0 & 0 & t & E_p & 2t \cos(k_x a / 2) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2t \cos(k_x a / 2) & E_p & t & 0 \\ 0 & 0 & 0 & 0 & 0 & t & E_p & 2t \cos(k_x a / 2) \\ 0 & 0 & 0 & 0 & 0 & 0 & 2t \cos(k_x a / 2) & E_p \end{bmatrix} \quad (3.2.6)$$

其中參數部分也是使用[17]，經由計算，可以得到下圖結果，

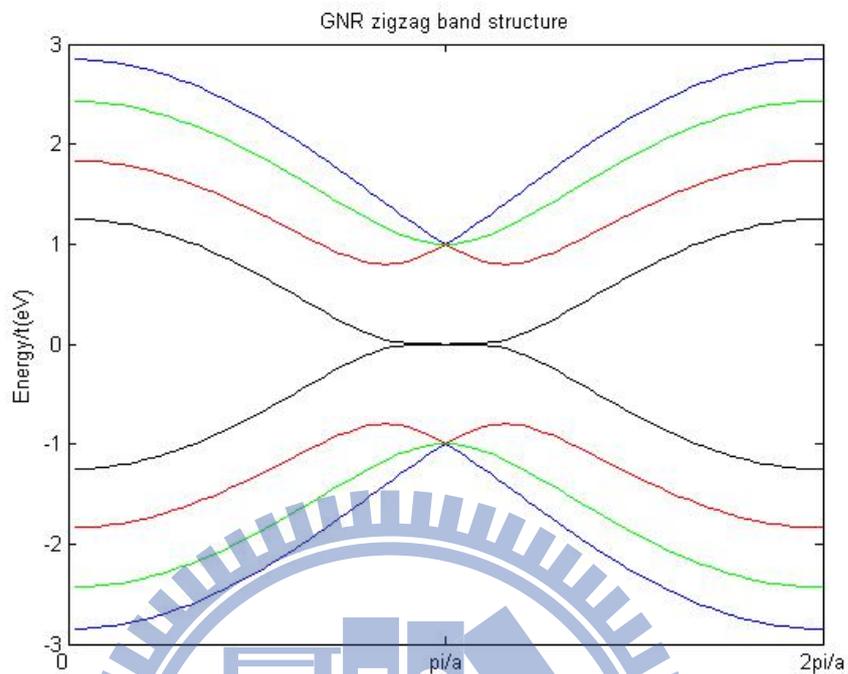


圖 3.2.2、graphene nanoribbon zigzag 方向的能帶結構，為靠近零的八個能量。

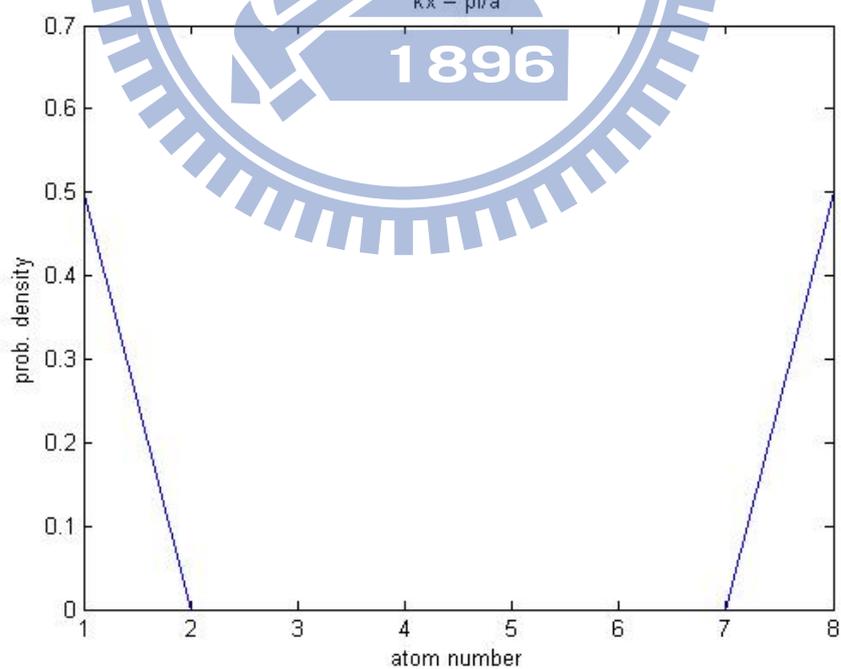


圖 3.2.3、graphene nanoribbon zigzag 方向的能帶結構，各個原子在 $Kx=\pi/a$ 的導電帶機率密度。

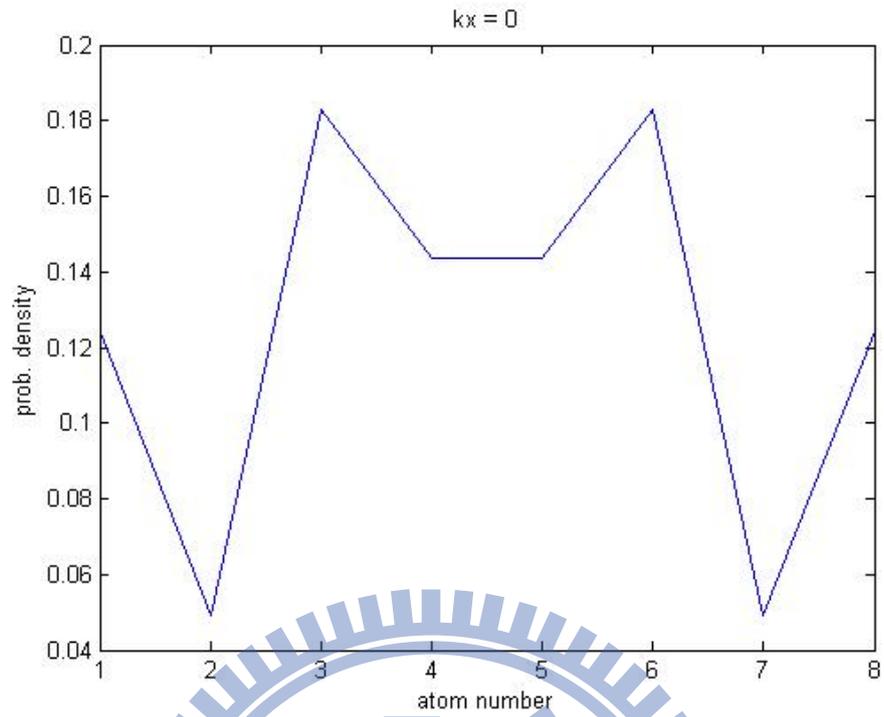


圖 3.2.4、graphene nanoribbon zigzag 方向的能帶結構，各個原子在 $K_x=0$ 的

導電帶機率密度。



若考慮傳輸方向為 armchair，如下圖之結構

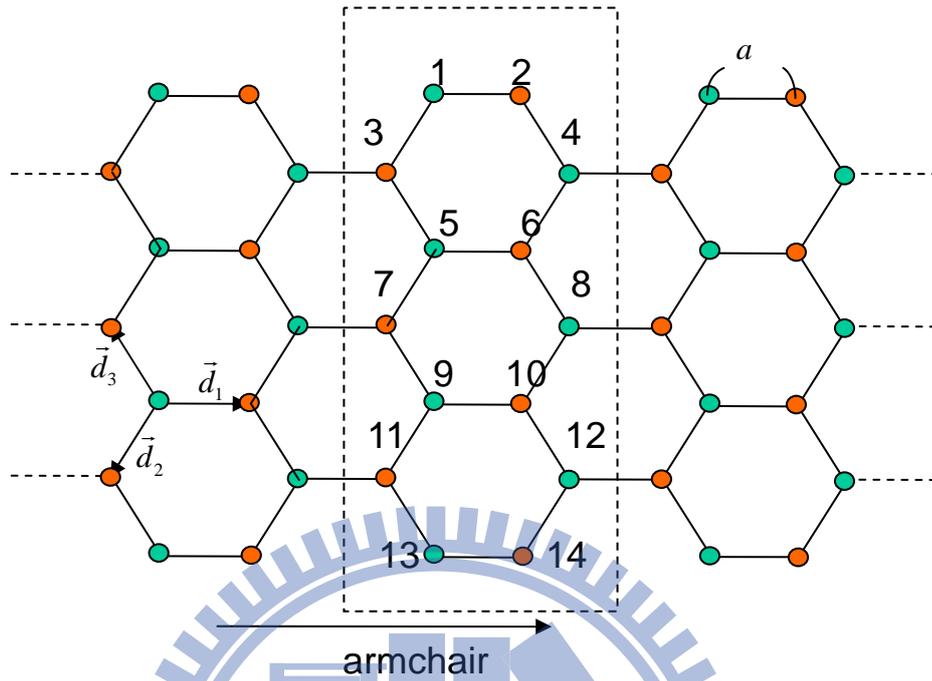


圖 3.2.5、graphene nanoribbon 結構，傳輸方向為 armchair

在單位晶胞中原子數有十四個原子，個別定為一號到十四號，一樣只考慮 Pz 軌域，鄰近向量考慮如下

$$\vec{d}_1 = a(1,0), \quad \vec{d}_2 = \frac{a}{2}(-1, -\sqrt{3}), \quad \vec{d}_3 = \frac{a}{2}(-1, \sqrt{3}) \quad (3.2.7)$$

所以為一 14*14 大小的矩陣，考慮各個原子之間的鍵結關係之後，並依據 (2.2.19)與(2.2.20)，矩陣元素如下

$$\langle i, p_z | H | i, p_z \rangle = \varepsilon_\alpha + \int \varphi_{n\alpha}^*(\vec{r}') \Delta U \varphi_{n\alpha}(\vec{r}') d\vec{r}' \equiv E_p = 0 \quad i=1,2,\dots,14 \quad (3.2.3)$$

$$\begin{bmatrix}
 E_p & te^{i\vec{k}\cdot\vec{d}_1} & te^{i\vec{k}\cdot\vec{d}_2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 te^{-i\vec{k}\cdot\vec{d}_1} & E_p & 0 & te^{-i\vec{k}\cdot\vec{d}_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 te^{-i\vec{k}\cdot\vec{d}_2} & 0 & E_p & te^{-i\vec{k}\cdot\vec{d}_1} & te^{-i\vec{k}\cdot\vec{d}_3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & te^{i\vec{k}\cdot\vec{d}_3} & te^{i\vec{k}\cdot\vec{d}_1} & E_p & 0 & te^{i\vec{k}\cdot\vec{d}_2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & te^{i\vec{k}\cdot\vec{d}_3} & 0 & E_p & te^{i\vec{k}\cdot\vec{d}_1} & te^{i\vec{k}\cdot\vec{d}_2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & te^{-i\vec{k}\cdot\vec{d}_2} & te^{-i\vec{k}\cdot\vec{d}_1} & E_p & 0 & te^{-i\vec{k}\cdot\vec{d}_3} & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & te^{-i\vec{k}\cdot\vec{d}_2} & 0 & E_p & te^{-i\vec{k}\cdot\vec{d}_1} & te^{-i\vec{k}\cdot\vec{d}_3} & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & te^{i\vec{k}\cdot\vec{d}_3} & te^{i\vec{k}\cdot\vec{d}_1} & E_p & 0 & te^{i\vec{k}\cdot\vec{d}_2} & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & te^{i\vec{k}\cdot\vec{d}_3} & 0 & E_p & te^{i\vec{k}\cdot\vec{d}_1} & te^{i\vec{k}\cdot\vec{d}_2} & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & te^{-i\vec{k}\cdot\vec{d}_2} & te^{-i\vec{k}\cdot\vec{d}_1} & E_p & 0 & te^{-i\vec{k}\cdot\vec{d}_3} & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & te^{-i\vec{k}\cdot\vec{d}_2} & E_p & te^{-i\vec{k}\cdot\vec{d}_1} & te^{-i\vec{k}\cdot\vec{d}_3} & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & E_p & te^{i\vec{k}\cdot\vec{d}_1} & 0 & te^{i\vec{k}\cdot\vec{d}_2} \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & te^{i\vec{k}\cdot\vec{d}_3} & 0 & E_p & te^{i\vec{k}\cdot\vec{d}_1} \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & te^{-i\vec{k}\cdot\vec{d}_2} & te^{-i\vec{k}\cdot\vec{d}_1} & E_p
 \end{bmatrix} \quad (3.2.8)$$

將矩陣進行對角化之後，可得到能帶圖。

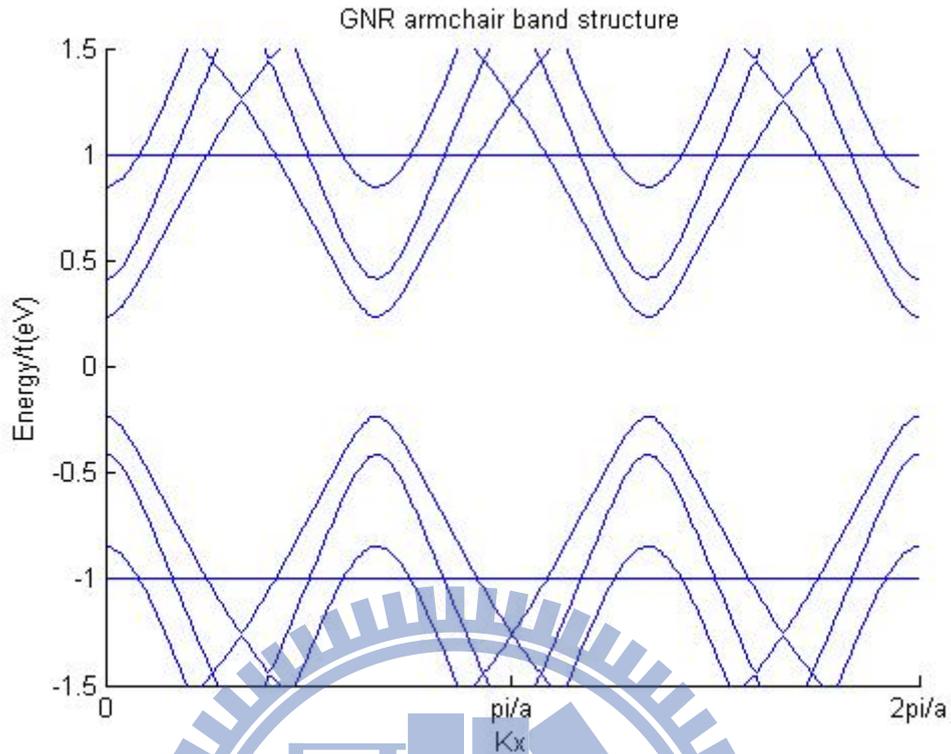


圖 3.2.6、graphene nanoribbon armchair 方向的能帶結構

此時，矩陣的大小會因計算材料的原子數多寡與每個原子考慮幾個 orbital 而決定，舉例來說，如果要計算的材料有 N 個原子，每個原子考慮 α 個軌域，則矩陣大小為 $N\alpha \times N\alpha$ ，其中的非零矩陣元就要看考慮到第幾近鄰而決定，然而，目前電腦記憶體大小仍舊無法計算過大材料，因此，一個好的計算方法便很重要，所以接下來要介紹一個處理大型矩陣對角化的演算法。

第四章、數值方法

現今電腦科技蓬勃發展，記憶體容量急速上升，用於一般電腦綽綽有餘，但適用於科學計算方面仍然稍嫌貧瘠，如何以有限的電腦資源解決無窮的基底需求的問題，是數值計算中的一大課題。在這章節，我們將討論如何化簡一個大型矩陣的方法。

大型矩陣的解決方法

一般而言，處理物理系統時常將算子(operator)使用基底轉為矩陣，利用矩陣對角化求得能量 E ，進而進行物理討論，但如果今天數值離散化的方法如果不夠精確，往往勢必將增加基底，以增加其精確度。目前(2009)的電腦記憶體雖然已經便宜又好用，對於物理系統而言，卻還是只能處理有限大小的問題，因此，一個好的處理大型矩陣的方法便很重要。以下，將介紹如何解決這樣的問題。

相似變換(similar transformation)

若矩陣 A 能夠找到一個么正(unitary)矩陣 U ，使得 $A = UBU^{-1}$ ，則可稱 A 相似於 B (A is similar to B ， $A \sim B$)，滿足相似變換的矩陣 A 與 B 有下列性質。

1. A 與 B 具有相同特徵值
2. $\text{tr}(A) = \text{tr}(B)$
3. $\det(A) = \det(B)$
4. $\text{rank}(A) = \text{rank}(B)$

因此，如果 A 的特徵值很不好解，若能使用相似變換將 A 轉換成 B，而 B 是比較好求特徵值的，則為相同解 A 的特徵值。

其中解決大型矩陣知名的方法有二：

1. 若 A 是對稱(赫密特)矩陣

Lanczos 法：
$$AU = UT$$

T 為 tridiagonal 矩陣(對角線三條矩陣)

2. 若 A 是非對稱(非赫密特)矩陣

Arnoldi 法：
$$AU = UH + fe^T$$

H 為上海森堡矩陣(upper Heisenberg matrix)，f 為殘餘向量

若 A 為對稱，則 Arnoldi 法會變成 Lanczos 法。

透過這兩個方法，可以將 A 矩陣轉換為 H 與 T，並且使用 QR 分解法與高斯消去法各別去求 H 與 T 特徵值相較於 A 簡單的多。但為何能這樣子做呢？以下介紹這兩個方法。

4.1 Lanczos Method

首先，考慮標準特徵值問題

$$Ax = \mu x \tag{4.1.1}$$

A 為一個 N x N 的矩陣， μ 為特徵值，X 為特徵向量，如果我們從任意向

$$a_1 = \mathbf{u}_1^+ \mathbf{H} \mathbf{u}_1 \quad (4.1.6)$$

a_1 也就是 $\mathbf{H} \mathbf{u}_1$ 與 \mathbf{u}_1 的內積。由於所得新向量尚未歸一，所以歸一化常數為

$$b_1^2 = \mathbf{w}_1^+ \mathbf{w}_1 \quad (4.1.7)$$

所以第一個正交向量為

$$\mathbf{u}_2 = \frac{\mathbf{w}_1}{b_1} \quad (4.1.8)$$

現在我們開始建構 u_3 ，讓我們定義一個暫時向量 w_2

$$\mathbf{w}_2 = \mathbf{H} \mathbf{u}_2 - a_2 \mathbf{u}_2 - b_2 \mathbf{u}_1 = (\mathbf{H} - a_2) \mathbf{u}_2 - b_2 \mathbf{u}_1 \quad (4.1.9)$$

其中

$$a_2 = \mathbf{u}_2^+ \mathbf{H} \mathbf{u}_2 \quad (4.1.10)$$

$b_2 b_1$ 由前面計算可知為正交歸一係數，所以

$$b_2^2 = \mathbf{w}_2^+ \mathbf{w}_2 \quad (4.1.11)$$

所以第二個正交新向量為

$$\mathbf{u}_3 = \frac{\mathbf{w}_2}{b_2} \quad (4.1.12)$$

相同方法可求得

$$\mathbf{w}_3 = \mathbf{H} \mathbf{u}_3 - a_3 \mathbf{u}_3 - b_3 \mathbf{u}_2 - c_3 \mathbf{u}_1 \quad (4.1.13)$$

注意，我們定義 $c = \mathbf{u}_1^+ \mathbf{H} \mathbf{u}_3 \equiv 0$ ，因此，此迭代程序便導致為一個三項關係式。

在迭代 N 次後，正交的新向量 $\mathbf{H} \mathbf{u}_p$ 只與兩個向量有關係，即 \mathbf{u}_p 與 \mathbf{u}_{p-1} ，得

$$\mathbf{w}_{p+1} = \mathbf{H} \mathbf{u}_p - a_p \mathbf{u}_p - b_p \mathbf{u}_{p-1} \quad (4.1.14)$$

且

$$\mathbf{b}_{p+1}^2 = \mathbf{w}_{p+1}^+ \mathbf{w}_{p+1}, \quad a_{p+1} = \mathbf{u}_{p+1}^+ \mathbf{H} \mathbf{u}_{p+1} \quad (4.1.15)$$

我們使用此方法決定其他三條對角線(tridiagonal)矩陣 T 的其他元素。

所以

$$\begin{aligned} \mathbf{A} \cdot [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p] &= [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p] \cdot \mathbf{T} \\ \Rightarrow \mathbf{A} \mathbf{U} &= \mathbf{U} \mathbf{T} \\ \Rightarrow \mathbf{U}^+ \mathbf{A} \mathbf{U} &= \mathbf{T} \end{aligned} \quad (4.1.16)$$

此為一個相似變換(similar transformation)，其中一種特性為 A 與 T 有相同特徵值，便可以利用 Lanczos 方法去找出 u_i 使 A 能夠轉成 T，讓對角化簡單化，這即為 Lanczos 方法的精神。

接下來，我們現在就可以將此演算法公式化：

1. 選擇開始向量 u_1 ，選擇是任意的，唯一條件是必須歸一化。
2. 計算 a_1 ： $a_1 = \mathbf{u}_1^+ \mathbf{A} \mathbf{u}_1$
3. 計算 b_1 ： $b_1^2 = [(\mathbf{A} - a_1 \mathbf{I}) \mathbf{u}_1]^+ [(\mathbf{A} - a_1 \mathbf{I}) \mathbf{u}_1]$
4. 導出向量 u_2 ： $\mathbf{u}_2 = \frac{(\mathbf{A} - a_1 \mathbf{I}) \mathbf{u}_1}{b_1}$
5. 計算 a_2 ： $a_2 = \mathbf{u}_2^+ \mathbf{A} \mathbf{u}_2$
6. 計算 b_2 ： $b_2^2 = [(\mathbf{A} - a_2 \mathbf{I}) \mathbf{u}_2 - b_1 \mathbf{u}_1]^+ [(\mathbf{A} - a_2 \mathbf{I}) \mathbf{u}_2 - b_1 \mathbf{u}_1]$
7. 導出向量 u_3 ： $\mathbf{u}_3 = \frac{(\mathbf{A} - a_2 \mathbf{I}) \mathbf{u}_2 - b_2 \mathbf{u}_1}{b_2}$

重新運算第 5~7 步驟，當 i 從 1 到 n。

因此，基本的 Lanczos 的演算法中，做為大型矩陣計算方面，需要再做許多修改，接下來介紹的 ARPACK，克服了某些 Lanczos 的問題。

4.2 ARPACK 簡介

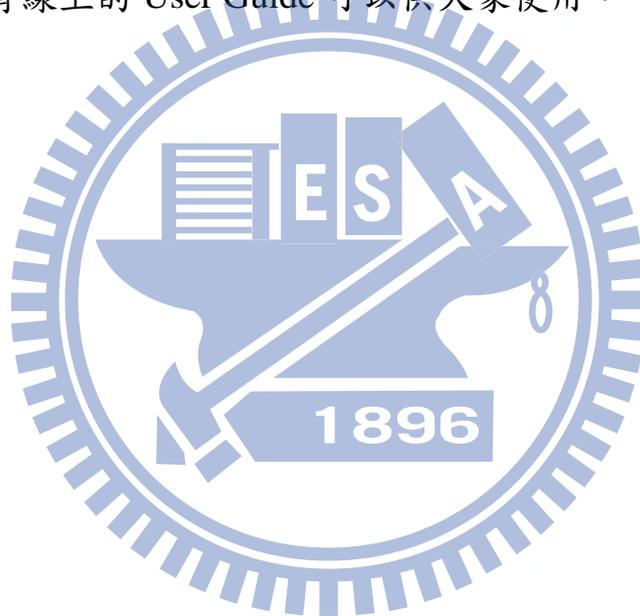
ARPACK 是一個用來求解大規模特徵值問題的軟體。ARPACK 是由 Fortran77 所寫的副程式的集合，主要可以解決來自各應用領域的大規模對稱、非對稱與廣義特徵值問題。可以求特定數量的特徵值，其中包含最大或最小實部特徵值、最大或最小虛部特徵值、以及最大或最小振幅的特徵值。ARPACK 軟體是基於隱式重開始 Arnoldi/Lanczos 方法 (Implicitly Restarted Arnoldi/Lanczos Method)。Arnoldi/Lanczos 是用來求解大規模 $n \times n$ 矩陣的幾個特徵值的重要方法，特別適合用於大規模稀疏矩陣的特徵值問題。隱式重開始 Arnoldi 方法採用了隱式位移 QR 法與 k 階 Arnoldi 分解結合的策略，再配合隱式重開始的方法，節省標準 Arnoldi 和 Lanczos 方法中存在的數值困難和儲存困難 (需大量儲存基底)，紀錄矩陣的方式是採用稀疏矩陣，也就是只紀錄矩陣不為零的位置與大小，進而達到處理大規模特徵值問題的需求。所以 ARPACK 本身是適合使用在矩陣元素中，為零元素很多的矩陣，例如解決固態物理中所使用到的 Tight-Binding Method、解決微分方程的有限差分法 (Finite Difference Method)，都是矩陣中具有相當多為零的元素，使用 ARPACK 才會有較佳的結果。

主要功能包括：

1. 求解單雙精度實對稱矩陣特徵值問題
2. 求解雙精度複數矩陣特徵值問題(Hermitian & non-Hermitian)
3. 廣義特徵值問題
4. 奇異值分解(SVD)問題

最新軟體版本可上 <http://www.caam.rice.edu/software/ARPACK/> 自行下載使

用，網站上也有線上的 User Guide 可以供大家使用。



Implicit Restart Arnoldi Method (IRAM)

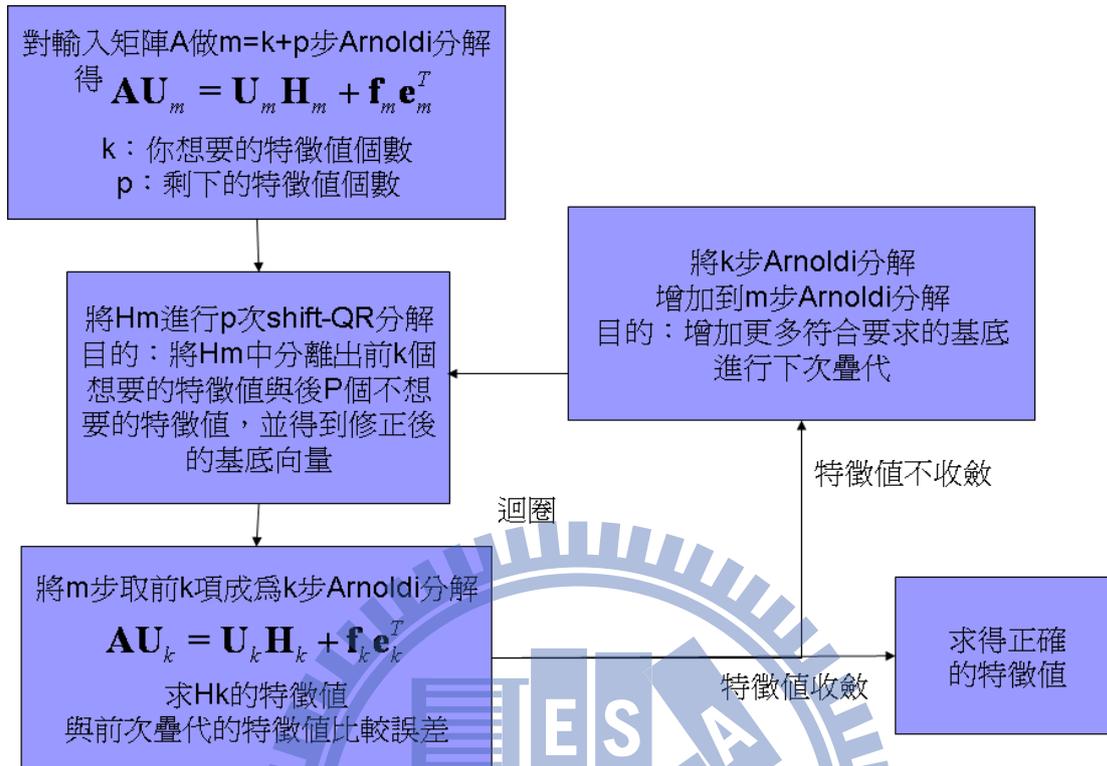


圖 4.2.1、IRAM 流程圖

Arnoldi Factorization

在介紹 IRAM 之前，首先要瞭解 k 階 Arnoldi Factorization。先考慮一個一般特徵值問題，定義：如果待求矩陣 $A \in C^{n \times n}$ ，且滿足

$$AU_k = U_k H_k + f_k e_k^T \quad (4.2.1)$$

其中 $U_k \in C^{n \times k}$ 為正交歸一的行向量，稱為 Arnoldi 向量，且 $U_k^H f_k = 0$ ， $H_k \in C^{k \times k}$ 是上海森堡矩陣(upper Heisenberg matrix)，為下對角線以上皆有值的矩陣， $e_k^T = (\dots 0 1 0 \dots)$ 在第 k 個位置才有 1 其它為零，此關係為 A 的 k 階 Arnoldi Factorization，為 Krylov 子空間的一種應用。若 A 是 Hermitian，則 H_k 是實數對稱三對角線矩陣(real symmetric tridiagonal matrix)，此關係稱

為 A 的 k 階 Lanczos 分解，所對應的行向量 U_k 稱為 Lanczos 向量。理論上來說，解決上海森堡矩陣會比直接解原始矩陣來的有效率。

使用 Arnoldi 分解時，需要儲存大量的 Krylov 子空間的基底，且理論上需要與 A 行向量個數一樣多的基底才會收斂，但電腦模擬計算是使用有限記憶體，如果計算矩陣很大，往往在收斂以前就已達記憶體上限，所以必須利用重開始 Arnoldi 分解，減少記憶基底的數目，經由每次疊代改變起始向量，改變相對產生的 Krylov 子空間，直到 Krylov 子空間將所有有興趣的特徵值與特徵空間包起來，進而達到收斂的結果。至於如何改變起始向量呢？

Filter polynomials

假設在 $n \times n$ 矩陣 A 中有特徵值 μ_i 與特徵向量 x_i ，而有興趣的是前 k 個特徵值與特徵向量，並且基本假設為使用特徵向量展開起始向量 v_1 如下列形式：

$$v_1 = \sum_{i=1}^k C_i x_i + \sum_{i=k+1}^m C_i x_i \quad (4.2.2)$$

若 f 是任意的多項式，乘上起始向量，則可有：

$$f(A)v_1 = \sum_{i=1}^k C_i f(\mu_i) x_i + \sum_{i=k+1}^m C_i f(\mu_i) x_i \quad (4.2.3)$$

選擇 f 使 $f(\mu_i)$ ($i=k+1, \dots, n$) 都比 $f(\mu_i)$ ($i=1, \dots, k$) 還要小，則乘上 v_1 後會使想要的特徵向量的係數變大，不想要的特徵向量係數變小，則稱 f 為 Filter polynomials，如果在 A 的特徵值 $\mu_1 \dots \mu_m$ 中， $\mu_{k+1} \dots \mu_m$ 是 p 個不感興趣的，若 f

與 A 有關，則 f 為 A 的特徵多項式

$$f(\mathbf{A}) = (\mathbf{A} - \mu_{k+1}\mathbf{I}) \cdots (\mathbf{A} - \mu_m\mathbf{I}) \quad (4.2.4)$$

利用此 Filter 多項式，碰到後 p 個不感興趣的向量時，此多項式會變成零，即可將起始向量中，不感興趣的特徵向量成分刪掉，使起始向量展延的 Krylov 子空間為我們想要的特徵空間，但由於此多項式為一矩陣向量乘積，直接處理將此 Filter 多項式乘上向量，稱為顯式重開始(Explicit Restarting)。若矩陣很大時，顯式重開始勢必會花費許多的時間，因此，隱式重開始便因應而生。

Implicit Restart Arnoldi Method

針對顯式重開始的不方便，因此發展了隱式重開始，先將 Arnoldi 分解從 k 階變為 $m=k+p$ 階(前 k 個有興趣，後 p 個沒興趣)，再進行隱式位移 QR 法，達到分類的效果，使矩陣能夠只解出想要的特徵值。將(3.2.1)右邊乘上 Q 後可以修改如下：

$$\mathbf{A}\mathbf{U}_m^+ = \mathbf{U}_m^+ \mathbf{H}_m^+ + \mathbf{f}_m \mathbf{e}_m^T \mathbf{Q} \quad (4.2.5)$$

其中 $\mathbf{U}_m^+ = \mathbf{U}_m \mathbf{Q}$ ， $\mathbf{H}_m^+ = \mathbf{Q}^H \mathbf{H}_m \mathbf{Q}$ ， $\mathbf{Q} = \mathbf{Q}_1, \mathbf{Q}_2, \dots, \mathbf{Q}_p$ ， \mathbf{Q}_j 為矩陣 $\mathbf{H}_m - \mu_j \mathbf{I}$ 的 QR 分解，此時利用 p 次隱式位移 QR 分解法取所求得 \mathbf{U}_m^+ 的第一行向量如下

$$\begin{aligned} \mathbf{U}_m^{+(1)} &= \mathbf{U}_m \mathbf{Q}_1 \rightarrow \mathbf{u}_1^{(1)} = (\mathbf{A} - \mu_1 \mathbf{I}) \mathbf{v}_1 \\ \mathbf{U}_m^{+(2)} &= \mathbf{U}_m^{+(1)} \mathbf{Q}_2 \rightarrow \mathbf{u}_1^{(2)} = (\mathbf{A} - \mu_2 \mathbf{I})(\mathbf{A} - \mu_1 \mathbf{I}) \mathbf{v}_1 \\ \mathbf{U}_m^{+(3)} &= \mathbf{U}_m^{+(2)} \mathbf{Q}_3 \rightarrow \mathbf{u}_1^{(3)} = (\mathbf{A} - \mu_3 \mathbf{I})(\mathbf{A} - \mu_2 \mathbf{I})(\mathbf{A} - \mu_1 \mathbf{I}) \mathbf{v}_1 \\ &\vdots \\ \mathbf{U}_m^{+(p)} &= \mathbf{U}_m^{+(p-1)} \mathbf{Q}_p \rightarrow \mathbf{u}_1^{(p)} = (\mathbf{A} - \mu_p \mathbf{I}) \cdots (\mathbf{A} - \mu_1 \mathbf{I}) \mathbf{v}_1 \end{aligned} \quad (4.2.6)$$

由此可以看出，藉由隱式位移 QR 分解，可以得到與顯式重開始相同效果的基底，詳細推導內容請參見附錄。由於 $\mathbf{e}_m^T \mathbf{Q}$ 的前 $k-1$ 項為零，所以此 m 階的前 k 項可以等價改寫成 k 階 Arnoldi 分解，或等同只取前 k 項，本質上是一樣的：

$$\mathbf{A}\mathbf{U}_k^+ = \mathbf{U}_k^+ \mathbf{H}_k^+ + \mathbf{f}_k^+ \mathbf{e}_k^{+T} \quad (4.2.7)$$

其中，殘餘向量 $\mathbf{f}_k^+ = \mathbf{U}_k^+ \mathbf{e}_{k+1} \hat{\beta}_k + \mathbf{f}_m \sigma$ ，此時的 k 階 Arnoldi 分解，可以包含此 m 階 Arnoldi 分解中，有興趣的特徵值所展延的特徵空間，並且起始向量也將不想要的特徵值過濾掉，進而再增加 p 階使 Arnoldi process 產生下一個新的 m 階 Arnoldi 分解，再進行隱式位移 QR 法，直到收斂為止。利用所得到的該 Krylov 子空間的基底向量與顯式重開始 \mathbf{v}_1 所產生的空間相同的特性，便可以大大減少產生 Krylov 子空間所需要的計算量。此為 ARPACK 重要的精神。

根據上述方法，可編成下列演算法：

Input : $(\mathbf{A}, \mathbf{U}, \mathbf{H}, \mathbf{f})$ with $\mathbf{A}\mathbf{U}_m = \mathbf{U}_m\mathbf{H}_m + \mathbf{f}_m\mathbf{e}_m^T$,

an m-step Arnoldi Factorization ;

For $l=1,2,3,\dots$ until convergence

1. *Computer* $\sigma(\mathbf{H}_m)$ and select set of p shifts $\mu_1, \mu_2, \dots, \mu_p$
based upon $\sigma(\mathbf{H}_m)$ or perhaps other information ;

2. *For* $j=1,2,\dots,p$

$$\text{Factor } [\mathbf{Q}, \mathbf{R}] = qr(\mathbf{H}_m - \mu_j \mathbf{I});$$

$$\mathbf{H}_m \leftarrow \mathbf{Q}^H \mathbf{H}_m \mathbf{Q}; \quad \mathbf{U}_m \leftarrow \mathbf{U}_m \mathbf{Q}; \quad \mathbf{q}_m \leftarrow \mathbf{q}^H \mathbf{Q};$$

End _ For

3. $\mathbf{f}_k \leftarrow \mathbf{v}_{k+1} \hat{\beta}_k + \mathbf{f}_m \sigma_k$;

4. $\mathbf{U}_k \leftarrow \mathbf{U}_m(1:n, 1:k)$; $\mathbf{H}_k \leftarrow \mathbf{H}_m(1:k, 1:k)$;

5. *Beginning with the k-step Arnoldi factorization*

$$\mathbf{A}\mathbf{U}_k = \mathbf{U}_k\mathbf{H}_k + \mathbf{f}_k\mathbf{e}_k^T$$

*apply p additional step of the Arnoldi process
to obtain a new m-step Arnoldi factorization*

$$\mathbf{A}\mathbf{U}_k = \mathbf{U}_k\mathbf{H}_k + \mathbf{f}_k\mathbf{e}_k^T$$

End _ For

圖 4.1.2、演算法

與 Lanczos 法的差別：

1. 起始向量經由疊代修正到最適合起始向量
2. 重開始以增加其基底正交性
3. 修改起始向量，求出想要的特徵值，避免假的特徵值出現

特徵空間、Krylov 子空間與稀疏矩陣紀錄方法

眾所皆知的，一個向量可以展延(span)成一維空間，二個獨立向量可以展延(span)成二維空間，以此類推，當有 N 個獨立向量時，便可以展延成無限維空間，在物理上稱為 Hilbert 空間(Hilbert space)。

特徵空間是用來描述一個由 A 的特徵值相對應特徵向量所形成的空間，稱為特徵空間。若要用另一個方法去解決此特徵問題，則必須 Krylov 子空間維度能等於 A 的特徵空間，才可在 Krylov 子空間中找到 A 的所有特徵值。

在先前介紹的 Arnoldi 法與 Lanczos 法所使用的空間稱為 Krylov 子空間，其定義為

$$K_v = \{v_1, Av_1, A^2v_1, \dots\}$$

其中 v_1 為自行決定的起始向量，也就是當起始向量決定時，Krylov 子空間便決定了。使用 Krylov 子空間最大的特性是，可以只紀錄矩陣中不為零的部份，因為要產生此空間只需要矩陣乘上向量而產生的基底向量，而基底向量中的元素皆是由 A 不為零的元素所產生，所以，只要使用 Krylov 子空間，便可以使用稀疏矩陣(sparse matrix)的記錄方式，進而節省記憶體儲存空間，便能在有限的電腦資源做到更大的運算。

4.2.2 Test Result

我們現在就拿簡單例子二維諧振子來測試使用結果，這是電子處在拋物線位能下的薛丁格方程式

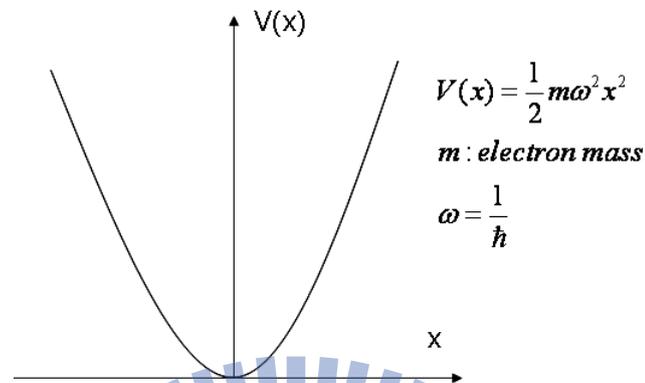


圖 3.2.1、位能與參數

$$\left[-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + \frac{1}{2} m \omega^2 x^2 \right] \psi(x) = E \psi(x) \quad (4.2.1)$$

該例子的能量大家熟知為

$$\frac{E}{\hbar \omega} = n + \frac{1}{2} \quad (4.2.2)$$

我們是使用有限差分法(finite difference method)[15]將微分方程轉成矩陣，再利用 ARPACK 去執行對角化所產生的結果，其正解應為 0.5、1.5、2.5... 等，以下是測試的結果

	$E/(hbar*w)$	Arpack	exact solution	誤差
Eigenvalue 0:	0.500001	0.5	0.5	0.00002%
Eigenvalue 1:	1.5	1.5	1.5	0.0%
Eigenvalue 2:	2.5	2.5	2.5	0.0%
Eigenvalue 3:	3.5	3.5	3.5	0.0%
Eigenvalue 4:	4.500001	4.5	4.5	0.00002%
Eigenvalue 5:	5.500001	5.5	5.5	0.00002%
Eigenvalue 6:	6.500001	6.5	6.5	0.00002%
Eigenvalue 7:	7.500001	7.5	7.5	0.00002%
Eigenvalue 8:	8.500001	8.5	8.5	0.00002%
Eigenvalue 9:	9.500001	9.5	9.5	0.00002%
Eigenvalue 10:	10.5	10.5	10.5	0.0%
Eigenvalue 11:	11.5	11.5	11.5	0.0%
Eigenvalue 12:	12.5	12.5	12.5	0.0%
Eigenvalue 13:	13.5	13.5	13.5	0.0%
Eigenvalue 14:	14.5	14.5	14.5	0.0%

圖 4.2.1、矩陣 10 萬*10 萬的特徵值

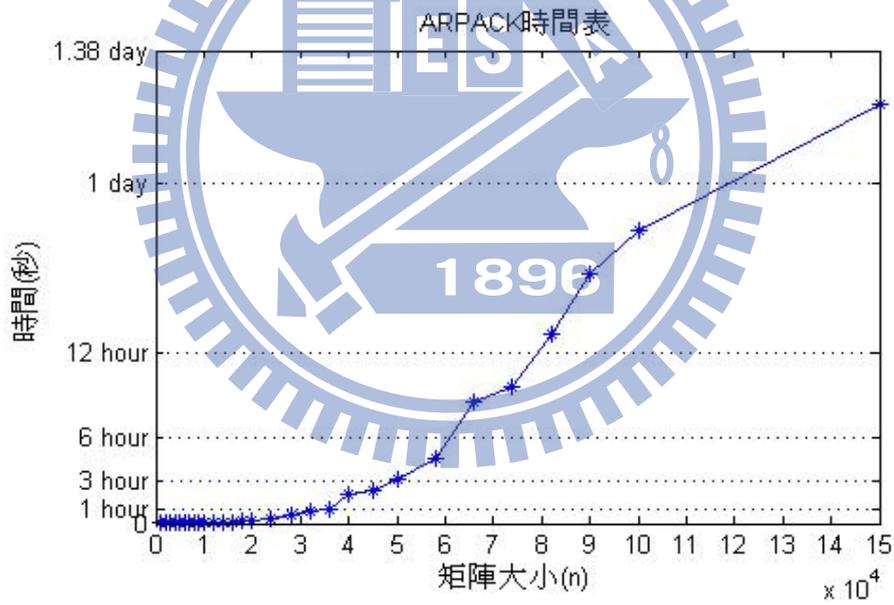


圖 4.2.2、ARPACK 的矩陣大小與時間分析圖

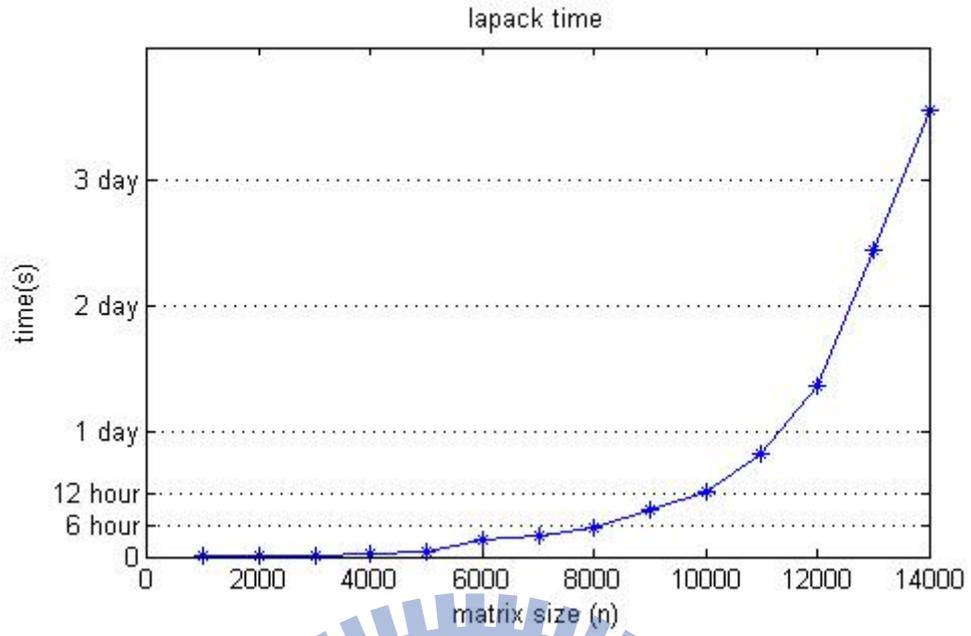


圖 4.2.3、LAPACK 的矩陣大小與記憶體分析圖

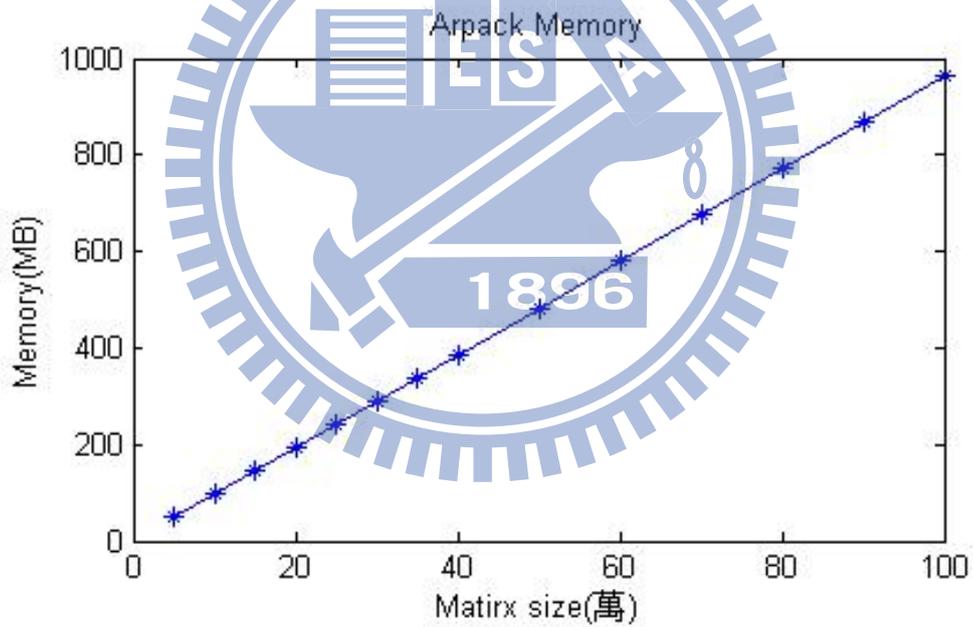


圖 4.2.4、ARPACK 的矩陣大小與記憶體分析圖

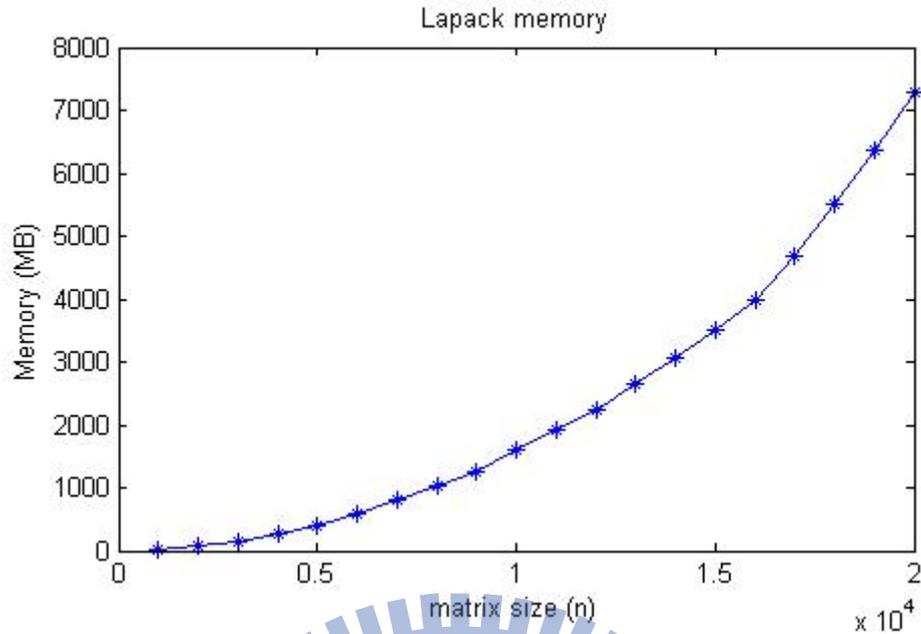


圖 4.2.5、LAPACK 的矩陣大小與記憶體分析圖

可以看出以 ARPACK 與 LAPACK 相比之下，記憶體大小大幅度的降低，計算大小大幅度的增加。

ARPACK 的所有參數皆列在附錄 A 中，其中最主要需要更改的參數如下所示：

參數	功能描述
n	矩陣大小
nev	求解特徵值的個數
ncv	Arnoldi 分解長度，即基底數量；預設值=4*nev
which	指定求解特徵值的特性(參考附錄 A)
iparam(3)	允許疊代的最大次數；預設值=3*n

其中，主要控制運算時間長短的兩個參數為 ncv 與 iparam(3)，增加基底

數 ncv 能降低運算時間，其主要是將待解矩陣分割成小矩陣的大小， ncv 越大，所產生小矩陣的大小越大，疊代次數會較少，但是每次疊代後解小矩陣的時間越多；反之， ncv 越小，所產生的小矩陣的大小越小，解小矩陣的時間越少，但疊代次數會較多。舉例來說：若待解矩陣為 1000×1000 的大小， ncv 假設是 200，則每次的小矩陣為 200×200 ，可能的疊代次數假設是 5 次，即解 5 次 200×200 大小的矩陣；若 ncv 假設是 10，則每次的小矩陣為 10×10 ，可能的疊代次數假設是 100 次，即解 100 次 10×10 的矩陣。

疊代次數($iparam(3)$)只需要修改到滿足程式未判定疊代次數不夠即可。

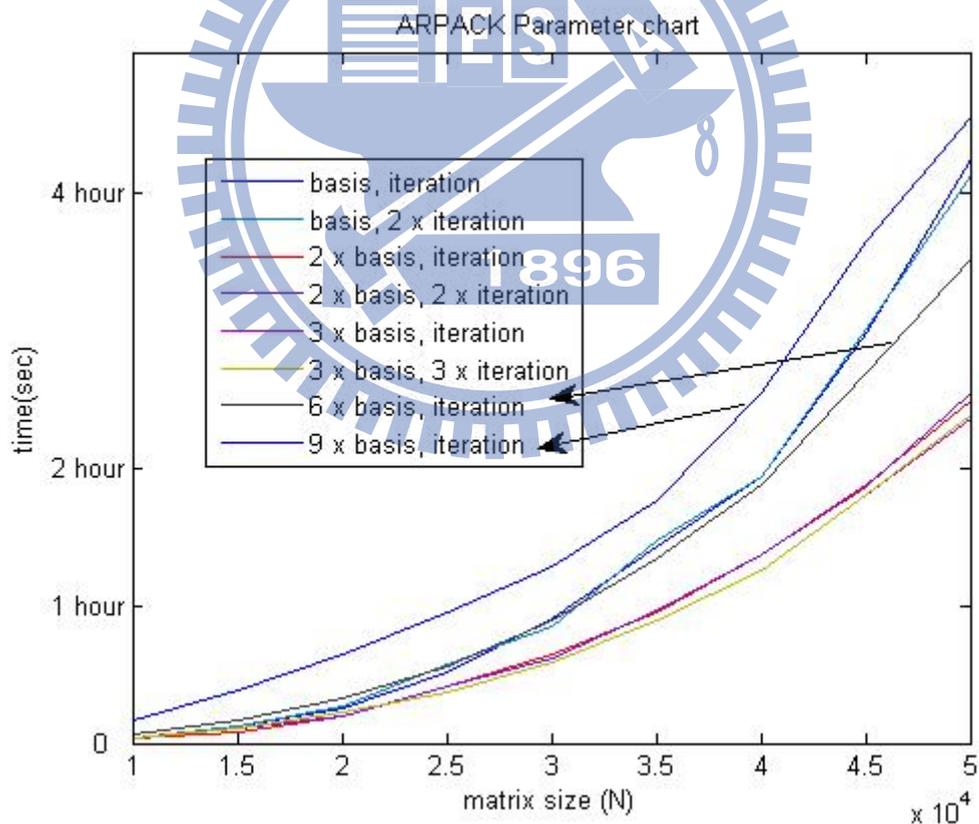
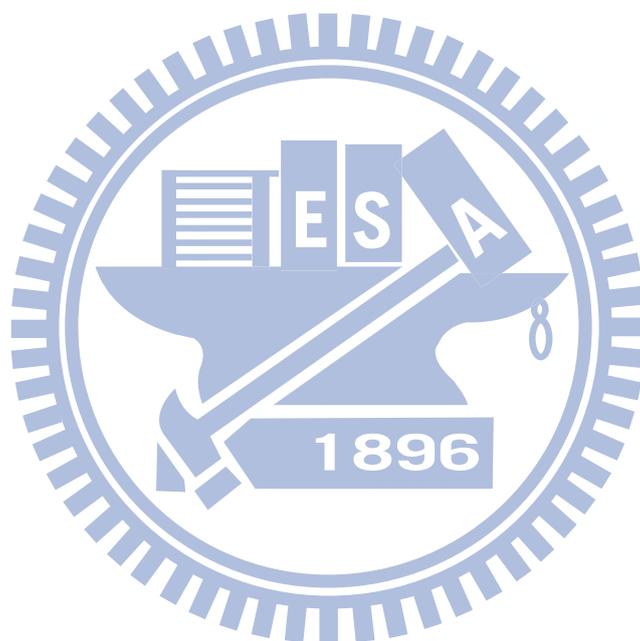


圖 4.2.6、ARPACK 參數修改對時間的影響，前項參數為 $ncv(basis)=4 \cdot ncv$ ，
後項參數為 $iparam(3) = 3 \cdot N$

在經過一連串的測試之後，針對求一維簡諧振子最低的 15 個能量，測試出基底數量取 $12 * nev$ ($nev=15$) 是可以節省最多時間的，在此提供給大家參考。但是一般來說，不同的矩陣對於參數的優化並沒有一個固定的公式，端看不同矩陣的稀疏程度、特徵值的結構與個數的而決定，因此，針對任何一個新的問題，便必須經由反覆的測試才能得到最佳化的參數。



第五章、總結與未來工作

總結：

由 ARPACK 的計算結果，可以了解到我們現在已經能計算超過十萬大小的矩陣，遠遠超過 LAPACK 再目前能計算的一萬，並且節省大量時間，然而其精確度也是非常高的，可以看到矩陣大小十萬的精確度已達 0.0002%，在計算更大的矩陣時，應會更準確。在投資非常多時間之後，我們已有一個可信賴的大型對角化軟體，在未來，便能計算更多的材料，這是一個非常有用的軟體。

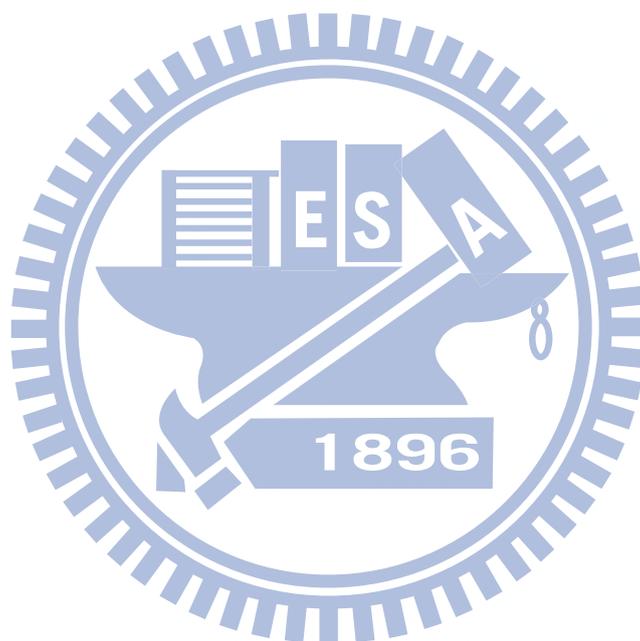
在單層石墨方面，我們也已經能計算出一些基本的能帶結構與其物理的結果，雖然目前物理方面的討論仍不足，但是在未來，相信是可以給更多的物理分析結果，以了解單層石墨更多的物理特性。

未來工作：

在單層石墨方面，目前大小還沒有算的很大，所以接下來的工作是計算非常龐大的單層石墨，以及計算 zigzag 與 armchair 方向的奈米帶狀單層石墨不同邊界的結果，或是算很大的島狀結構的單層石墨，都是屬於我們未來的課題，而且目前都是只有計算單電子模型，故未來還可以引入多電子模型，並加入電子間交互作用項，甚至是加入鎳基板[18]等等。至於 ARPACK 方面，還可以擴充不同的計算結構，目前是能計算實數矩陣最大最小值，

之後，可以慢慢增加計算複數矩陣，或是加入能平行處理矩陣的

PARPACK，都是未來的計算課題。



參考文獻

- [1]Novoselov, K.S. *et al.* "Electric Field Effect in Atomically Thin Carbon Films", *Science*, Vol **306** (5696), p. 666-669, 2004
- [2] Novoselov, K.S. *et al.* "Two-dimensional atomic crystals", PNAS, Vol **102** (30), p. 10451-10453, January 26, 2005
- [3] Mark Wilson, *Phys. Today* **59** January 21 (2006)
- [4] A.K. Geim & K.S. Novoselov, *Nature Materials* **6**, 183-191 (2007)
- [5] J. R. Williams, L. C. DiCarlo, and C. M. Marcus, *Science* **317**, 638 (2007)
- [6] D. A. Abanin and L. S. Levitov, *Science* **317**, 641 (2007) ◦
- [7]Barbaros Özyilmaz *et al.* , *Phys. Rev. Lett.* **99**, 166804 (2007)
- [8]Marek Korkusinski,"Application of the effective band-orbital model to semiconductor nanostructure"
- [9]Gene Howard Golub, Charles F. Van Loan,"Matrix computations"(1996)
- [10]R. B. Lehoucq, D. C. Sorensen, C. Yang," ARPACK Users' Guide:Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods"(1997)
- [11] A. H. Castro Neto, F. Guinea, N. M. R. Peres, K. S. Novoselov and A. K. Geim, *Rev. Mod. Phys.* **81**, 109 (2009)
- [12] K. Nakada, M. Fujita, G. Dresselhaus, M. S. Dresselhaus, *Phys. Rev. B* **54**, 17954 - 17961 (1996)
- [13] Peter Y. Yu,Manuel Cardona,"Fundamentals of Semiconductors:Physics and Materials Properties"(springer-Verlag 1996)
- [14] D. J. Chadi and M. L. Cohen Calculations of the Valence Bands *phys. stat. sol. (b)* **68**, 405 (1975)
- [15] L. V. Fausett, "Applied Numerical Analysis Using Matlab", Prentice-Hall(1999)
- [16] R. Saito, M. Fujita, G. Dresselhaus, M.S. Dresselhaus , *Phys. Rev. B* **46**, 1804 - 1811 (1992)
- [17] R. Saito, G. Dresselhaus, M.S. Dresselhaus ,"Physical Properties of Carbon Nanotubes",(Imperial College Press, Convent Garden, London 1998)
- [18] A. Varykhalov, J. Sa´nchez-Barriga, A. M. Shikin, C. Biswas, E. Vescovo, A. Rybkin, D. Marchenko, and O. Rader, *Phys. Rev. Lett* **101** 157601(2008)

附錄 A：隱式位移 QR 法

$$\mathbf{A}\mathbf{U}_m = \mathbf{U}_m\mathbf{H}_m + \mathbf{f}_m\mathbf{e}_m^T$$

$$\rightarrow (\mathbf{A} - \mu_1\mathbf{I})\mathbf{U}_m = \mathbf{U}_m(\mathbf{H}_m - \mu_1\mathbf{I}) + \mathbf{f}_m\mathbf{e}_m^T$$

$$\because \mathbf{H}_m - \mu_1\mathbf{I} = \mathbf{Q}_1\mathbf{R}_1 \quad (\text{對}\mathbf{H}_m\text{進行位移QR分解})$$

$$\rightarrow (\mathbf{A} - \mu_1\mathbf{I})\mathbf{U}_m = \mathbf{U}_m\mathbf{Q}_1\mathbf{R}_1 + \mathbf{f}_m\mathbf{e}_m^T$$

$$\rightarrow (\mathbf{A} - \mu_1\mathbf{I})\mathbf{U}_m\mathbf{Q}_1 = \mathbf{U}_m\mathbf{Q}_1\mathbf{R}_1\mathbf{Q}_1 + \mathbf{f}_m\mathbf{e}_m^T\mathbf{Q}_1$$

$$\rightarrow \mathbf{A}\mathbf{U}_m^{(1)} = \mathbf{U}_m^{(1)}\mathbf{H}_m^{(1)} + \mathbf{f}_m\mathbf{b}_{m+1}^{(1)H} \quad \text{回到Arnoldi分解的形式}$$

$$\text{其中}\mathbf{H}_m^{(1)} \equiv \mathbf{R}_1\mathbf{Q}_1 + \mu_1\mathbf{I}, \quad \mathbf{U}_m^{(1)} \equiv \mathbf{U}_m\mathbf{Q}_1, \quad \mathbf{b}_{m+1}^{(1)H} \equiv \mathbf{e}_m^T\mathbf{Q}_1 = (0 \quad \dots \quad 0 \quad *)$$

$\mathbf{H}_m^{(1)}$: one step of single shifted QR algorithm

$$\text{此時}\mathbf{H}_m^{(1)} = \begin{bmatrix} \mathbf{H}_{m-1}^{(1)} & * \\ 0 & \mu_1 \end{bmatrix} \quad * \text{表示不重要的數字}$$

proof easily:

$$\because \mathbf{H}_m^{(1)} - \mu_1\mathbf{I} = \mathbf{R}_1\mathbf{Q}_1$$

μ_1 為 $\mathbf{H}_m^{(1)}$ 的特徵值，故左邊為 singular ($\det(\mathbf{H}_m^{(1)}) = 0$)

$$\text{而}\mathbf{Q}_1\text{必為non-singular，則}\mathbf{R}_1\text{必為singular } \mathbf{R}_1 = \begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\therefore \mathbf{H}_m^{(1)} - \mu_1\mathbf{I} = \mathbf{R}_1\mathbf{Q}_1 = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\rightarrow \mathbf{H}_m^{(1)} = \mathbf{R}_1\mathbf{Q}_1 + \mu_1\mathbf{I} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & 0 & \mu_1 \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{m-1}^{(1)} & * \\ 0 & \mu_1 \end{bmatrix}$$

$$\begin{aligned}
(\mathbf{A} - \mu_1 \mathbf{I}) \mathbf{U}_m &= \mathbf{U}_m (\mathbf{H}_m - \mu_1 \mathbf{I}) + \mathbf{f}_m \mathbf{e}_m^T \\
\rightarrow (\mathbf{A} - \mu_1 \mathbf{I}) \mathbf{U}_m \mathbf{e}_1 &= \mathbf{U}_m (\mathbf{H}_m - \mu_1 \mathbf{I}) \mathbf{e}_1 + \cancel{\mathbf{f}_m \mathbf{e}_m^T \mathbf{e}_1} \\
\rightarrow (\mathbf{A} - \mu_1 \mathbf{I}) \mathbf{u}_1 &= \mathbf{U}_m \mathbf{Q}_1 \mathbf{R}_1 \mathbf{e}_1 = r_{11}^{(1)} \mathbf{U}_m^{(1)} \mathbf{e}_1 = r_{11}^{(1)} \mathbf{u}_1^{(1)}
\end{aligned}$$

其中 $r_{11}^{(1)}$ 為矩陣 $\mathbf{R}_1(1,1)$ 的值

$$\rightarrow \mathbf{u}_1^{(1)} // (\mathbf{A} - \mu_1 \mathbf{I}) \mathbf{u}_1$$

同理進行第二次位移QR分解

$$\mathbf{A} \mathbf{U}_m^{(1)} = \mathbf{U}_m^{(1)} \mathbf{H}_m^{(1)} + \mathbf{f}_m \mathbf{b}_{m+1}^{(1)H}$$

$$\rightarrow (\mathbf{A} - \mu_2 \mathbf{I}) \mathbf{U}_m^{(1)} = \mathbf{U}_m^{(1)} (\mathbf{H}_m^{(1)} - \mu_2 \mathbf{I}) + \mathbf{f}_m \mathbf{b}_{m+1}^{(1)H}$$

$$\therefore \mathbf{H}_m^{(1)} - \mu_2 \mathbf{I} = \mathbf{Q}_2 \mathbf{R}_2 \quad (\text{對 } \mathbf{H}_m^{(1)} \text{ 進行位移QR分解})$$

$$\rightarrow (\mathbf{A} - \mu_2 \mathbf{I}) \mathbf{U}_m^{(1)} = \mathbf{U}_m^{(1)} \mathbf{Q}_2 \mathbf{R}_2 + \mathbf{f}_m \mathbf{b}_{m+1}^{(1)H}$$

$$\rightarrow (\mathbf{A} - \mu_2 \mathbf{I}) \mathbf{U}_m^{(1)} \mathbf{Q}_2 = \mathbf{U}_m^{(1)} \mathbf{Q}_2 \mathbf{R}_2 \mathbf{Q}_2 + \mathbf{f}_m \mathbf{b}_{m+1}^{(1)H} \mathbf{Q}_2$$

$$\rightarrow \mathbf{A} \mathbf{U}_m^{(2)} = \mathbf{U}_m^{(2)} \mathbf{H}_m^{(2)} + \mathbf{f}_m \mathbf{b}_{m+1}^{(2)H}$$

$$\text{其中 } \mathbf{H}_m^{(2)} \equiv \mathbf{R}_2 \mathbf{Q}_2 + \mu_2 \mathbf{I}, \quad \mathbf{U}_m^{(2)} \equiv \mathbf{U}_m^{(1)} \mathbf{Q}_2, \quad \mathbf{b}_{m+1}^{(2)H} \equiv \mathbf{b}_{m+1}^{(1)H} \mathbf{Q}_2 = (\dots \quad 0 \quad * \quad *)$$

$$\text{此時 } \mathbf{H}_m^{(2)} = \left[\begin{array}{c|cc} \mathbf{H}_{m-2}^{(2)} & * & * \\ \hline 0 & \mu_1 & * \\ 0 & & \mu_2 \end{array} \right]$$

$$(\mathbf{A} - \mu_2 \mathbf{I}) \mathbf{U}_m^{(1)} = \mathbf{U}_m^{(1)} (\mathbf{H}_m^{(1)} - \mu_2 \mathbf{I}) + \mathbf{f}_m \mathbf{b}_{m+1}^{(1)H}$$

$$\rightarrow (\mathbf{A} - \mu_2 \mathbf{I}) \mathbf{U}_m^{(1)} \mathbf{e}_1 = \mathbf{U}_m^{(1)} (\mathbf{H}_m^{(1)} - \mu_2 \mathbf{I}) \mathbf{e}_1 + \cancel{\mathbf{f}_m \mathbf{b}_{m+1}^{(1)H} \mathbf{e}_1}$$

$$\rightarrow (\mathbf{A} - \mu_2 \mathbf{I}) \mathbf{u}_1^{(1)} = \mathbf{U}_m \mathbf{Q}_2 \mathbf{R}_2 \mathbf{e}_1 = r_{11}^{(2)} \mathbf{U}_m^{(2)} \mathbf{e}_1 = r_{11}^{(2)} \mathbf{u}_1^{(2)}$$

其中 $r_{11}^{(2)}$ 為矩陣 $\mathbf{R}_2(1,1)$ 的值

$$\rightarrow \mathbf{u}_1^{(2)} // (\mathbf{A} - \mu_2 \mathbf{I}) \mathbf{u}_1^{(1)} // (\mathbf{A} - \mu_2 \mathbf{I})(\mathbf{A} - \mu_1 \mathbf{I}) \mathbf{u}_1$$

進行第p次位移QR分解

$$\mathbf{A}\mathbf{U}_m^{(p)} = \mathbf{U}_m^{(p)}\mathbf{H}_m^{(p)} + \mathbf{f}_m\mathbf{b}_{m+1}^{(p)H}$$

$$\text{其中 } \mathbf{H}_m^{(p)} \equiv \mathbf{R}_p\mathbf{Q}_p + \mu_p\mathbf{I}, \quad \mathbf{U}_m^{(p)} \equiv \mathbf{U}_m^{(p-1)}\mathbf{Q}_p, \quad \mathbf{b}_{m+1}^{(p)H} \equiv \mathbf{b}_{m+1}^{(p-1)H}\mathbf{Q}_p = \begin{pmatrix} \overbrace{0 \cdots 0}^{k-1} & \overbrace{* \cdots *}^{p+1} \end{pmatrix}$$

$$\text{此時 } \mathbf{H}_m^{(p)} = \left[\begin{array}{c|ccc} \mathbf{H}_k^{(p)} & & & * \\ \hline & \mu_p & * & * \\ 0 & 0 & \ddots & * \\ & 0 & 0 & \mu_1 \end{array} \right]$$

$$\mathbf{u}_1^{(p)} // (\mathbf{A} - \mu_p\mathbf{I}) \cdots (\mathbf{A} - \mu_2\mathbf{I})(\mathbf{A} - \mu_1\mathbf{I})\mathbf{u}_1$$

$\mathbf{H}_k^{(p)}$ 中包含所有想要的特徵值的資訊

取 $\mathbf{A}\mathbf{U}_m^{(p)} = \mathbf{U}_m^{(p)}\mathbf{H}_m^{(p)} + \mathbf{f}_m\mathbf{b}_{m+1}^{(p)H}$ 的前K項

將此k階增加分解長度為m階，再重新進行QR分解



附錄 B：ARPACK 程式參數

n：矩陣大小。

nev：求解特徵值的個數。

ncv：Arnoldi 分解的分解長度，即基底數量。此部份影響特徵值的精確度，
分解長度高，則收斂越快，也越精確。

which：指定求解特徵值的特性：

Which 的設定	特徵值特性
'LA'	最大特徵值
'SA'	最小特徵值
'LM'	最大模(modulus)特徵值
'SM'	最小模特徵值
'BE'	兩端的特徵值

Info：初始向量產生方式。

Info=0：隨機產生初始向量；

Info=1：透過 resid 提供初始向量。

Sigma：將所有特徵值平移的位移量。

若設定了以上變數，則需相對修改 maxn、maxnev、maxncv 滿足下式：

$$\begin{aligned} \max n &\geq n & \max ncv &\geq ncv \\ \max nev &\geq nev & ncv &\geq nev+1 \end{aligned}$$

(在 C code 中並無 max 這些參數)

根據需要，下列變數可以依照其功能修改：

lwork1：陣列 work1 的大小，設定為工作的陣列大小，至少應被設置 為 $ncv*(ncv+8)$ 。

tol：收斂標準，此值用來控制收斂精確度。tol 越小，達到停止標準所需要的計算量越大

ido：逆通訊標誌，進入**aupd 之前要設為 0

bmat：表示解決的是標準特徵值問題(bmat='I')或是廣益特徵值問題 (bmat='G')

iparam(1)：是否在隱式 QR 法中使用位移。若不清楚可設定為(iparam(1)=1)

iparam(3)：允許的 IRAM 疊代的最大次數

iparam(7)：ARPACK 的運算模式

