第一章 導論

一般來說，當我們在瀏覽網頁的時候，網頁上的圖像和文字都能恰如其份地傳達作者的想法。可是當我們有需要把網頁列印出來的時候，我們會發現在網頁中看起來賞心悅目的圖形，結果卻是差強人意的。這是因爲在電腦中每一張圖像都是由像素 (pixel) 所構成的。然而，在電腦中雖然是同一張圖像，隨著圖形呈現到我們的眼睛的媒介 (media) 或是裝置 (device) 的不同而有不同的視覺感受。造成這種視覺感受的差異的原因是裝置間的解析度不同。我們的構想即是將圖像經由增加解析度的處理，從而改善影像列印品質不佳的問題。增加解析度轉換成一種數學想法就是二維插值 (two-dimensional interpolation)。

1.1 起源

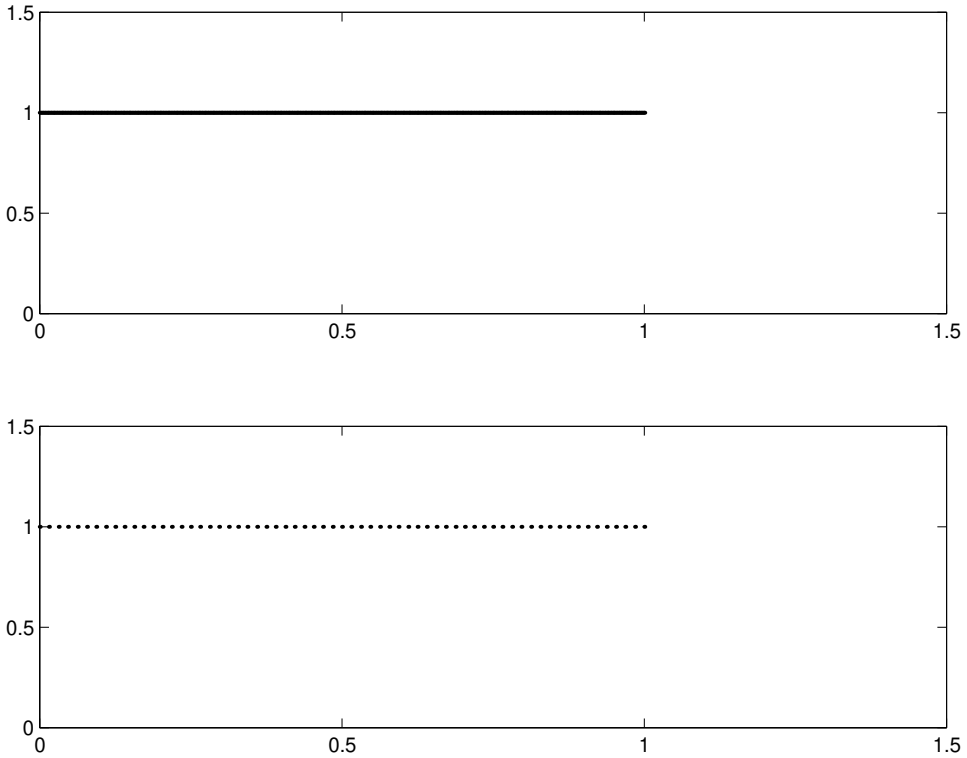
在應用上，一個函數是以其離散的函數點值，依照這個函數是單變數函數 $f(x)$ 或是雙變數函數 $f(x,y)$，而被紀錄成一個向量或是一個矩陣。舉一個簡單的例子：

$$f(x) = 1, \quad x \in [0,1]$$

在 Matlab 中我們輸入下列指令

```
x = linspace(0,1,513);
y = ones(size(x));
plot(x,y,'.');
```

第一列的指令是把 $f(x)$ 的定義域 $[0,1]$ 等分成 513 個點（包括 0,1 在內），也就是將 $[0,1]$ 均分爲 512 等份。因此，執行完第一列指令後， $\mathbf{x} = (0, \frac{1}{512}, \frac{2}{512}, \cdots, 1)$, 是一個 1×513 的列向量。第二列指令讓 $\mathbf{y}$ 是一個全是 1 的向量，其維度與 $\mathbf{x}$ 相同。最後我們讓 Matlab 替我們畫出 x-y 的圖形（圖一上），可以看到這張圖就如同我們平常在紙張或黑板討論時所畫的圖形。



圖一

有一個地方要特別說明的是，在第一列指令的第三個參數似乎是一個謎樣的數字 (magic number)。實際上，它代表的是取樣率 (sampling rate)。我們試著改變這個數字，而在其他的地方都保持不變。我們發現在取樣率越來越小的情形下，圖形開始有變化；當取樣率只有原來的 1/8 時，我們看到的並不是一條直線。正確地說，它是水平的 65 個點（圖一下）。總結來說，前面提到一個函數是以其離散的函數點值而被紀錄，就是先對函數的定義域做取樣，再計算其對應的函數值。而要將它的圖形繪製出來，就是在螢幕打點；點打得越密，看起來越像原來的函數。

在提到剛才的例子時，我們總是說『在電腦中描述一個函數』，而不稱爲『逼近一個函數』。我們之前的工作涉及了函數的離散化。一般而言，在眞實的世界裡，函數的定義域（不論是時間域或是空間域）總是連續的。依照 [1, 第 4.9 節]，所謂離散化，是在函數的定義域中取出有限多個節點。而這個由有限解析度的點來代表無窮解析度的區段的步驟也被稱爲逼近。因此，『在電腦中描述一個函數』的確是『逼近一個函數』，但是『逼近一個函數』不一定是『在電腦中描述一個函數』；更多數的情況是讓其他人認爲我們在使用泰勒多項式之類的數學工具來逼近一個函數。這是我們必須採取較拗口的說法的原因，因爲逼近的語意比我們所做的事要來得多。

1.2 多項式插值

我們將傳統的插值問題敘述如下 [2]：

> 給定 $n+1$ 個函數點對 $(x_i, f(x_i))$，找出一函數 $\Phi(x)$ 使得
> $\Phi(x_i) = f(x_i), \quad i = 0, \cdots, n$。

比較簡單的作法是考慮 $\Pi_n f(x)$ 是唯一的一條通過 $\{f(x_i)\}_{i=0}^n$ 的 n 階多項式，使得 $\Pi_n(x_i) = f(x_i)$。我們稱 $\Pi_n f(x)$ 爲 $f(x)$ 的內插多項式 (interpolation polynomial) [1, p.187]。可以想見，若多項式 $\Pi_n f(x)$ 是通過節點 $\{x_i\}_{i=0}^n$ 的 $f(x)$ 的內插多項式，那麼在其他 $x \neq x_i$ 的點上，$\Pi_n f(x)$ 不一定相等於 $f(x)$。這代表了 $f(x) \approx \Pi_n f(x)$，$\Pi_n f(x)$ 是 $f(x)$ 的一個多項式逼近。換句話說，插值就是一種逼近，只不過插值必須滿足插值條件。這就是我們在前一節吹毛求疵的原因；我們不能在整篇論文中都以逼近帶過，這樣勢必會帶來混淆。

值得注意的是，多項式插值的理論保證了對應函數 $f(x)$ 的插值多項式 $\Pi_n f(x)$ 不但存在而且是唯一的。由於多項式插值的理論發展得很成熟，我們在此不抄錄它的證明。有意思的是，這個定理的證明是以建構式的方法得到插值多項式的存在性。雖然插值多項式只有一條，卻可以有不只一種的表示式。常見的有牛頓形式的插值多項式 (Newton form of the interpolation polynomial) 以及拉格朗形式的插值多項式 (Lagrange form of the interpolation polynomial)。

給定 $(x_i, y_i), 0 \leq i \leq n$，拉格朗形式的插值多項式如下：

$$P(x) = \sum_{i=0}^{n} y_k l_i(x)$$

其中

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \quad , 0 \leq i \leq n$$

而牛頓形式的插值多項式爲

$$P(x) = \sum_{i=0}^{n} c_i \prod_{j=0}^{i-1} (x - x_j)$$

一般來說，這兩個形式各有擅場。以拉格朗形式而言，只要給定 $(x_i, y_i), 0 \leq i \leq n$，立刻就能得到拉格朗形式的插值多項式；而牛頓形式則必須算出每個係數 c_i。不過

牛頓形式的係數由 divided difference 可以很快地得到。整體來說，牛頓形式的計算效率會比拉格朗形式好一些。不過這只是大略的比較，在 [2, p.237] 有更深入的討論。

當資料點序列只有少數幾個點，多項式插值是很有用的。但是考慮實際的情形。給定資料點序列 $\{x_i\}_{i=0}^n$ 以及對應的函數點值 $\{y_i\}_{i=0}^n$，此時的 n 通常會大於 5。根據這些函數點值 $\{y_i\}_{i=0}^n$ 所計算的插值多項式是 n 階多項式。由於高階多項式有隨著資料點值的改變而劇烈振動的特性；此外，少部分函數點值的改變會造成整條插值多項式都受到影響。因此，實際上並不是很實用。針對這個問題，有一個改善的方法：

> 將原本整個區間分割成較小的區段，再分別計算這些子區間上
> 的插值多項式。

相對於原本廣域的多項式插值，這種改善的方法被稱爲分片多項式插值 (piecewise polynomial interpolation)。

分片多項式插值在 Matlab 已有對應的實作。我們可以在 Matlab 中使用函數 interp1

```
yi = interp1(x,y,xi,'method');
```

來作一維插值。 Matlab 總共提供四個方法作插值，分別以 'nearest', 'linear', 'cubic', 'spline' 這四個字串代表。當我們要指定使用哪一種方法，就將參數 'method' 的地方置換成相對應的字串。預設的方法是 'linear'，一般通稱這個方法爲線性插值 (linear interpolation)。除了 'spline' 之外，其餘三種方法 'nearest', 'linear', 'cubic' 都是分片多項式的實作。雖然 'spline' 也是一種分片多項式，不過它的概念與這裡的分片多項式不同，加上它本身的理論與發展很豐富，我們選擇把 'spline' 方法留到下一節討論。

先前提到分片插值多項式的想法是將廣域的區間分成多個較小的區段。給定資料點序列 $\{x_i\}_{i=0}^n$，其中 $x_0 \le x_1 \le \cdots \le x_n$，而我們要在 $[x_0, x_n]$ 之間做插值。以線性插值爲例，把資料點 $\{x_i\}_{i=0}^n$ 以兩點作爲一個區間，分爲 $[x_0, x_1]$, $[x_1, x_2]$, $\cdots$, $[x_{n-2}, x_{n-1}]$, $[x_{n-1}, x_n]$。每一個小區間都被視爲廣域的區間來計算插值多項式；以線性插值而言，就是在每個小區間計算個別的一次多項式。如果我們想知道的函數點值 y 其對應資料點 x 落在 $[x_0, x_1]$， y 的值可被下列一次多項式函數求出

$$y = \frac{y_0(x_1 - x) + y_1(x - x_0)}{x_1 - x_0}$$

以同樣的資料點序列，若採用 'cubic' 方法，就先把 $\{x_i\}_{i=0}^{n}$ 分成 $\{x_0, x_1, x_2, x_3\}$, $\{x_3, x_4, x_5, x_6\}$, $\cdots$, $\{x_{n-3}, x_{n-2}, x_{n-1}, x_n\}$。因此，我們要在 $[x_0, x_3]$, $[x_3, x_6]$, $\cdots$, $[x_{n-3}, x_n]$ 上就每個區間分別計算一個三次多項式。而最簡單的 'nearest' 方法則是以每一個資料點 x_i 爲基準，使用分片常數多項式計算函數點值

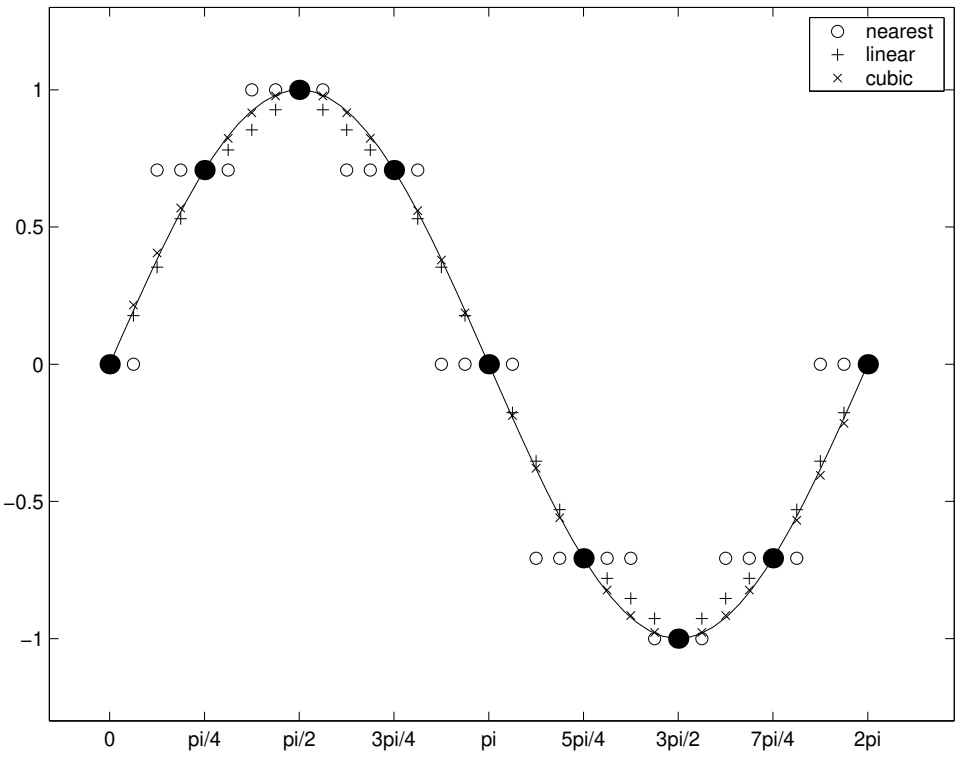
$$y = y_i \quad 若\ x \in [xi - \frac{x_i - x_{i-1}}{2}, x_i + \frac{x_{i+1} - x_i}{2}]$$

要注意的是，我們所使用的函數 interp1 要求資料點序列必須是單調的。這是因爲我們在廣域上計算插值多項式並不需要對資料點排序；而使用分片插值多項式，排序的動作是不可避免的。我們以正弦函數 $\sin(x), x \in [0, 2\pi]$ 作示範

```
x = linspace(0,2*pi,9);
y = sin(x);
xx = linspace(0,2*pi,60);
yy = sin(xx);
xi = linspace(0,2*pi,33); flops(0);
yn = interp1(x,y,xi,'nearest'); fps(1) = flops; flops(0);
yl = interp1(x,y,xi,'linear'); fps(2) = flops; flops(0);
yc = interp1(x,y,xi,'cubic'); fps(3) = flops; flops(0);
fps
fps =
    224     390     1061
```

我們把執行的結果繪製成圖形（圖二）。

從圖形上看來，'nearest' 的效果最差，'linear' 好一些，'cubic' 則是最好的。而從計算步驟來看，'nearest'所花費的步驟最少，'linear' 次之，'cubic' 則要比其他兩種方法花費更多的步驟。這樣的結果符合我們的期望：如果要得到更好的結果就得花費更多的計算步驟。

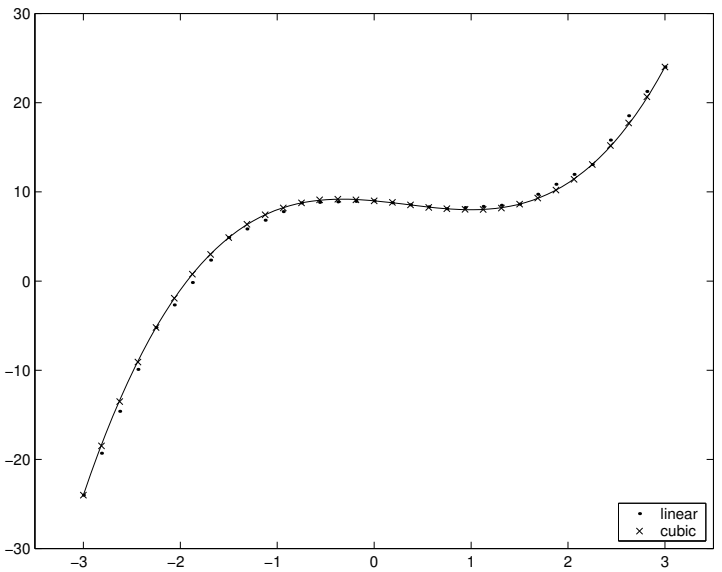


圖二

爲了闡釋效果與計算複雜度的關係，我們再看一個例子

$$g(x) = x^3 + x^2 - x + 3 \quad x \in [-3, 3]$$

我們重複前一個例子的步驟，因此這裡不另外列出程式碼。此外，我們省略 'nearest' 方法，把焦點集中在 'linear' 與 'cubic' 方法上（圖三）。



圖三

從圖中可以看到由 'cubic' 方法所產生的插值幾乎與原函數重合，而使用 'linear' 方法的插值則有些微的差距。這是因爲原函數 $g(x)$ 在廣域上是一個三次多項式，將原本的定義域分成較小的子區間後，$g(x)$ 在每個子區間上仍然是三次多項式。這與 'cubic' 方法不謀而合，只不過這時每個子區間上的三次多項式都是原本的 $g(x)$。

在 [1, 第 187 頁] 提供了多項式插值的誤差分析。如果 $f(x)$ 在區間 $[x_0, x_n]$ 內是 C^{n+1} 的函數，則逼近誤差是

$$|f(x) - \pi_n(x)| \leq Ch^{n+1} \max_{\xi \in [x_0, x_n]} \left| f^{n+1}(\xi) \right|$$

因此，我們可以知道插值的結果好壞與下列因素有關

1. 使用插值方法的函蓋 h
2. 使用插值方法所包含的多項式階數 n
3. 原函數 $f(x)$ 的可微程度

由於先前實驗所使用的函數不是多項式就是正弦函數。它們的可微程度分別是 C^{n+1} 及 C^{∞}，因此函數的可微程度這項因素並不明顯。不過，通常我們所處理的函數沒有這麼好的可微程度，這項因素就會對插值的結果產生影響。而由前面的實驗可知，我們所使用的插值方法包含越多的多項式，就會有更好的效果。另一方面，雖然計算越高階的多項式會增加計算複雜度，不過就分片多項式插值而言，所用的插值多項式通常只涉及到三階多項式。與其說是計算多項式本身的複雜度，更重要的原因是函蓋 (support) 的大小。給定資料點序列 $\{x_i\}_{i=0}^n$，使用 'linear' 方法的函蓋是 $[x_0, x_1]$, $[x_1, x_2]$, $\cdots$, $[x_{n-1}, x_n]$，每次在子區間上計算插值多項式都涉及一對資料點。而使用 'cubic' 方法函蓋是 $[x_0, x_3]$, $[x_3, x_6]$, $\cdots$, $[x_{n-3}, x_n]$，要得到每一段子區間的插值多項式就得要涉及四個資料點。如果要降低計算複雜度，勢必要選擇函蓋小 (short support) 的方法。

1.3 樣條函數

在 Matlab 中可以用來作一維插值的方法共有四種。在前一節我們先介紹分片多項式插值的理論以瞭解它們在 Matlab 中的實作情形。對於 'spline' 方法的介紹，我們打算如法泡製。

給定區間 $[a, b]$，$\{x_i\}_{i=0}^n$ 是在 [a,b] 區間上 $(n+1)$ 個不同的節點 (knot)，也就是說 $a = x_0 < x_1 < \cdots < x_n = b$。

若函數 $S_k(x)$, $x \in [a,b]$ 符合下列條件

1. $S_k|_{[x_j,x_{j+1}]} \in \mathbb{P}_k \quad j = 0,1,\cdots,n-1$
2. $S_k \in C^{k-1}[a,b]$

我們稱這樣的函數 $S_k(x)$ 爲 k 階的樣條函數。由它的定義可知，樣條函數其實也是分片多項式。前面所提到的 'linear' 方法就是一階樣條函數。另一方面，樣條函數的建造方式與之前的分片多項式不太一樣。分片多項式插值所用的方法包含的多項式越多，所涉及的資料點也越多。而不論是多少階的樣條函數，所涉及的計算都是資料點對 $\{x_i, x_{i+1}\}_{i=0}^{n-1}$。

Matlab 的 'spline' 採用了三階樣條函數 (cubic splines)，這也是一般最常被使用的選擇。給定點值 (x_i, y_i), $i = 0,\cdots,n$，我們來建造對應這些點值的三階樣條函數 $S(x)$。根據定義 (1) $S(x)$ 在每個子區間 $[t_0,t_1]$, $[t_1,t_2]$, $[t_{n-1},t_n]$上都是一條三階多項式 $S_i(x)$, $i = 0,\cdots,n-1$。 因此

$$S(x) = \begin{cases} S_0(x) & x \in [t_0,t_1] \\ S_1(x) & x \in [t_1,t_2] \\ \vdots & \vdots \\ S_{n-1}(x) & x \in [t_{n-1},t_n] \end{cases}$$

如果我們要建造出 $S(x)$ 也就是要求出每一條三階多項式 $S_i(x)$ 的係數。由於組合成 $S(x)$ 的三階多項式總共有 n 條，因此要解出 $4n$ 個係數。我們審視 $S(x)$ 是否擁有夠多的條件來求出這 $4n$ 個係數。由於 $S(x)$ 滿足插值條件，在區間 $[t_i,t_{i+1}]$ 上

$$S_i(t_i) = y_i,\ S_i(t_{i+1}) = y_{i+1} \quad i = 0,\cdots,n$$

這提供了 $2n$ 個條件。

另外，根據樣條函數的定義 (2)，$S(x) \in C^2[t_0,t_n]$。因此 $S(x)$, $S'(x)$, $S''(x)$ 都在 $[t_0,t_n]$ 上連續。以 t_i這一點來看，由 $S(x)$ 的插值條件，其實 $S_{i-1}(x)$ 和 $S_i(x)$ 都對 t_i 作插值。所以

$$S_{i-1}(t_i) = S_i(t_i) = y_i$$

因此我們無法從 $S(x)$ 的連續性得到更多的條件。不過由 $S'(x)$ 的連續性

$$S'_{i-1}(t_i) = S'_i(t_i) \quad i = 1,\cdots,n-1$$

以及 $S''(x)$ 的連續性

$$S''_{i-1}(t_i) = S''_i(t_i) \quad i = 1, \cdots, n-1$$

共可得到 $2(n-1)$ 個條件。

綜合以上的推論，我們總共有 $(4n-2)$ 個條件來決定 $4n$ 個係數。缺少的兩個條件給予我們兩個自由度 (freedom)，可以額外地指定兩個邊界條件。 Matlab 所選擇的方法是指定 $S'''(x)$ 在 t_1 及 t_{n-1} 這兩點上連續，亦即 $S'''_0(t_1) = S'''_1(t_1)$ 和 $S'''_{n-2}(t_{n-1}) = S'''_{n-1}(t_{n-1})$。

原本的

$$S(x)|_{[t_0,t_2]} = \begin{cases} S_0(x) & x \in [t_0, t_1] \\ S_1(x) & x \in [t_1, t_2] \end{cases}$$

$$S(x)|_{[t_{n-2},t_n]} = \begin{cases} S_{n-2}(x) & x \in [t_{n-2}, t_{n-1}] \\ S_{n-1}(x) & x \in [t_{n-1}, t_n] \end{cases}$$

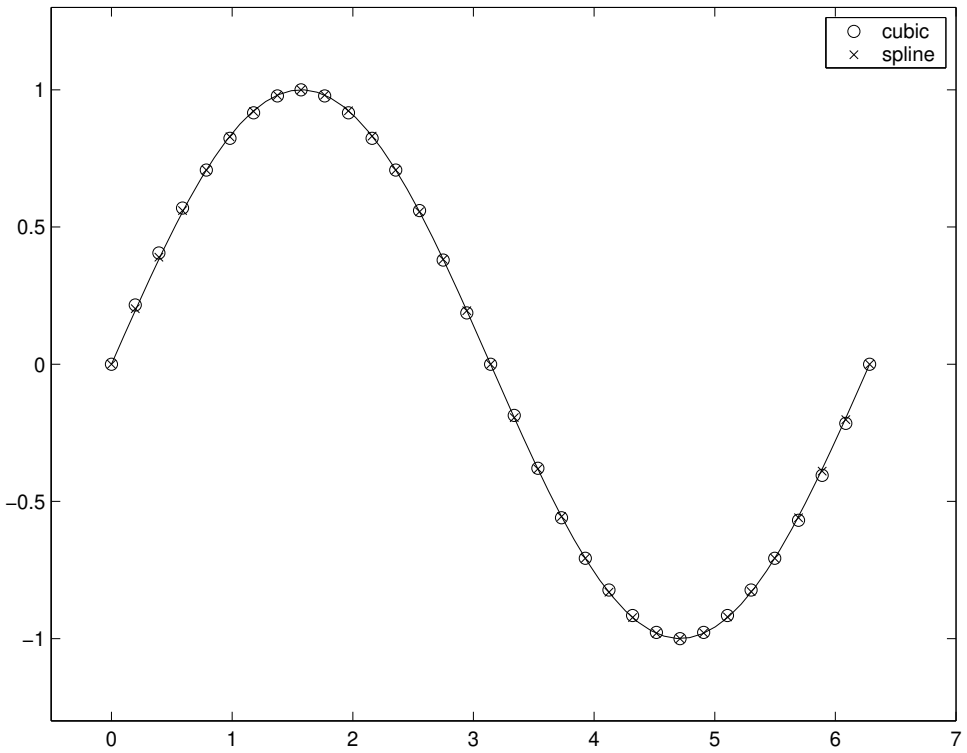
而這個條件會造成

$$T(x) = \begin{cases} S_0(x) & x \in [t_0, t_1] \\ S_1(x) & x \in [t_1, t_2] \end{cases}$$

$$W(x) = \begin{cases} S_{n-2}(x) & x \in [t_{n-2}, t_{n-1}] \\ S_{n-1}(x) & x \in [t_{n-1}, t_n] \end{cases}$$

其中 $T(x), W(x)$ 各爲一個三階多項式。這表示給定 $\{t_i\}_{i=0}^n$ 建造對應的三階樣條函數並不需要 t_1 和 t_{n-1}，亦即 t_1 和 t_{n-1} 不再是節點。所以依照這個條件而建造的三階樣條函數被稱爲 **not-a-knot spline**。

我們再看正弦函數 $\sin(x), x \in [0, 2\pi]$ 的例子。這一次只比較 'cubic' 和 'spline' 方法 (fig4.m)。就效果而言，'spline' 會好一些（圖四）。雖然它們都包含了三階多項式，不過 'spline' 的函蓋比 'cubic' 方法來得小，在函數變動較劇烈的地方 'spline' 方法會好一些。也就是說，'spline' 比 'cubic' 更具有局部性 (local)。 就計算複雜度而言，'cubic' 用了 1061 個 flops，而 'spline' 則只花了 737 個 flops。可見使用函蓋小的方法的確可以降低計算複雜度。



圖四

1.4 正交凌波函數理論

近幾年來由於凌波理論所取得的成功及迴響很大，關於凌波理論的專著已經爲數不少。這裡列了幾本值得推薦的書 [1, 4, 5, 6]。

我們從多層解析空間 (multiresolution analysis) 出發 [6, chap 6]。

若 $\{V_j\}, j \in \mathbb{Z}$ 滿足以下條件

1. $V_j \subset V_{j+1}$
2. $\cap V_j = \{0\}$以及$\overline{\cup V_j} = \mathcal{L}^2$
3. $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$
4. $f(x) \in V_0 \Leftrightarrow f(x-k) \in V_0$
5. $\{\phi(x-k)\}, k \in \mathbb{Z}$ 是 V_0 的一組正則基底

我們稱 $\{V_j\}$ 組成一個多層解析空間。根據多層解析空間的定義，我們可以得到

$$\{\phi_{jk}(x) = \sqrt{2^j}\phi(2^j x - k)\} \quad j,k \in \mathbb{Z}是\{V_j\}的一組正則基底$$

其中 $\phi_{jk}(x)$ 是由 $\phi(x)$ 的脹縮平移而產生的。我們記作

$$V_j = \text{span}\,\{\sqrt{2^j}\phi(2^j x - k) \mid k \in \mathbb{Z}\}$$

如果 $\phi(x) \in V_0$，由於 $\{V_j\}, j \in \mathbb{Z}$ 有層層相套的關係， $\phi(x)$ 也會落在 V_1。而 $\{\phi(2x-k)\}, k \in \mathbb{Z}$ 是 V_1 的基底，所以 $\phi(x)$ 可以表示成以下的等式

$$\phi(x) = \sqrt{2}\sum_k c_k \phi(2x-k) \qquad k \in \mathbb{Z}$$

我們稱這個等式爲自格等式 (scaling equation)。而分別滿足這個等式的 c_k 是一組自格係數 (scaling coefficients)， $\phi(x)$ 爲一個自格函數 (scaling function)。

有了 $\phi(x)$，可以根據它在 $\mathcal{L}^2(\mathbb{R})$ 中形成一組多層解析空間 V_j，但是光是只有 V_j 無法提供我們有效的逼近過程 [1, p.167]。由多層解析空間的定義，$V_{j-1} \subset V_j$。由於 $V_{j-1} \subset V_j$，定義 W_{j-1} 是 V_{j-1} 的垂直補空間 (orthogonal complement)。

$$\begin{aligned} V_j &= V_{j-1} \oplus W_{j-1} \\ &= V_{j-2} \oplus W_{j-2} \oplus W_{j-1} \\ &\vdots \\ &= V_0 \oplus W_0 \oplus \cdots \oplus W_{j-1} \end{aligned}$$

這就是一個有用的逼近過程。在 [1, 第 3.4 節] 證明了存在一個函數 $\psi(x)$，它的脹縮平移

$$\{\psi_{jk}(x) = \sqrt{2^j}\phi(2^j x - k) \mid j, k \in \mathbb{Z}\}$$

是 $W_j, j \in \mathbb{Z}$ 的一組正則基底。

由 $V_1 = V_0 \oplus W_0$，$\{\psi(x-k)\}$ 是 W_0 的一組正則基底，我們可以得到

$$\begin{aligned} \psi(x) &= \sum_k d_k \phi(2x-k) \\ &= \sum_{k=0}^{2p-1} (-1)^k c_{2p-1-k} \phi(2x-k) \end{aligned}$$

因爲 supp $\phi = [0, 2p-1]$, supp ψ 也是 $[0, 2p-1]$。一般而言，存在一個函數 $\psi(x)$ 符合

1. $\langle 1, \psi \rangle = \int \psi \, dx = 0$
2. $\|\psi\| = 1$
3. $\{\psi_{jk}(x)\}$ 形成 $\mathcal{L}^2(\mathbb{R})$ 的一組基底

我們稱 $\psi(x)$ 爲一個凌波函數 (wavelets)。當 $\{\psi(x-k)\}$ 形成一個正則集合， $\psi(x)$ 是一個正交凌波函數 (orthogonal wavelets)；特別地來說，這就是 Daubechies 的 p 階有限函蓋正交凌波函數。

我們現在討論如何在解析空間 V_j 繪製函數圖形。給定函數 $f(x) \in \mathcal{L}^2(\mathbb{R})$，定義正交投影算子 P_j 使得

$$\begin{aligned} f_j(x) &\equiv P_j f(x) \\ &= \sum_k \langle f, \phi_{jk} \rangle \phi_{jk}(x) \\ &= \sum_k \sqrt{(2^j)} \langle f, \phi_{jk} \rangle \phi(2^j x - k) \end{aligned}$$

在第 1.1 節時，我們繪製的是 $(x_n, f(x_n))$，而在解析空間 V_j 則是製作 $(x_n, f_j(x_n))$ 的圖形。要計算 $f_j(x_n)$ 牽涉了兩個問題：

1. 計算 $\phi(2^j x - k)$, $x \in \text{supp } f(x)$ 的值
2. 計算 $\langle f, \phi_{jk} \rangle$ 的值

第一個問題還好解決。我們可以針對常用的凌波函數事先計算它的點值，即使處理的函數不同也可繼續使用。但是計算 $\langle f, \phi_{jk} \rangle$ 的問題就令人感到棘手了。計算 $\langle f, \phi_{jk} \rangle$ 需要使用針對自格函數的數值積分技巧。此外，只要函數改變，我們必須針對不同的函數重新計算。不過，在 [1, 第 5.2 節] 推導了

若給定的函數$f(x) \in C^1$，則$\|f_j(x) - \tilde{f}_j(x)\| = O(2^{-j})$其中$\tilde{f}_j(x) = \sum_k f(\frac{k}{2^j})\phi(2^j x - k)$

也就是說，只要 j 足夠大，

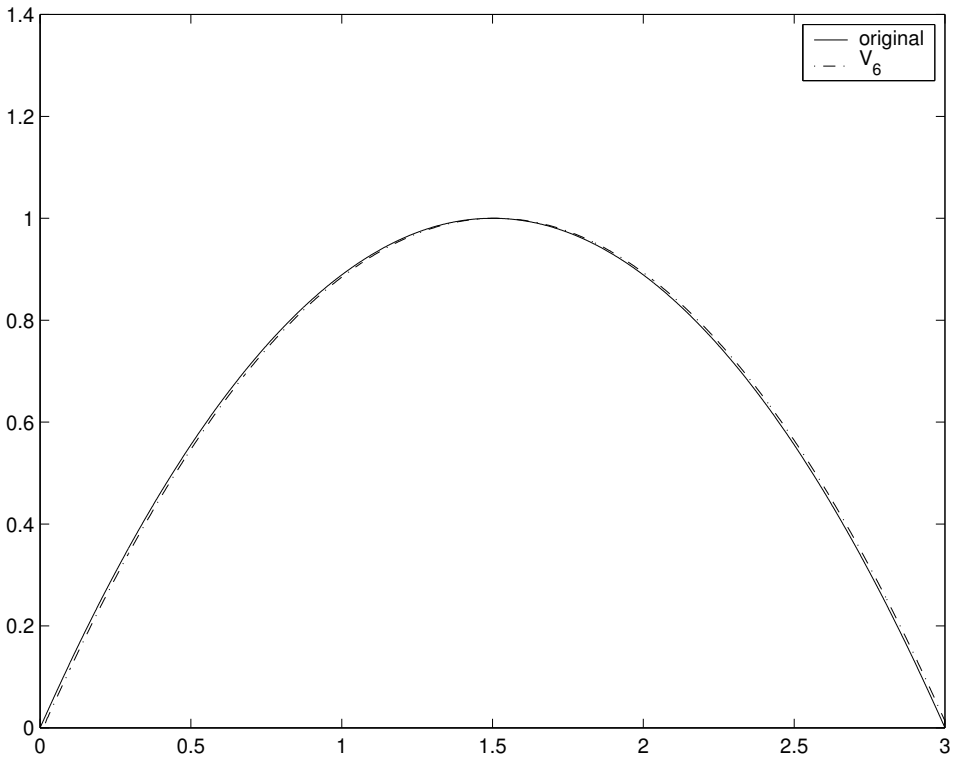
$$\sqrt{2^j} \langle f, \phi_{jk} \rangle \approx f(\frac{k}{2^j})$$

我們就可以藉由 $(x_n, \tilde{f}(x_n))$ 來繪製函數圖形。

我們看一個例子 $f(x) = \frac{4}{9}x(3-x)$, $x \in [0, 3]$（圖五）。實線代表的是採用第 1.1 節的方法。當取樣的點數夠多，我們可以相信那就是函數 $f(x)$ 的圖形。虛線代表的是 $f(x)$ 在解析空間 V_6 的樣子。這裡採用的是階數爲 2 的正交自格函數，$j = 6$ 的解析空間。此外，若 $\mathbf{f}$ 儲存的是 $f(x)$, $x \in [0, 3]$ 的函數點值，而 $\mathbf{y}$ 儲存的是 $f(x)$ 在解析空間的點值。在 Matlab 中下列指令可以得到相對誤差 (relative error)。

```
relative_error = norm(f-y)/norm(f)
relative_error =
            0.0104
```

這相當是 1% 的誤差。



圖五

雖然無論在視覺上或是進行誤差分析，我們都可以察覺得到在解析空間所看到的函數與原函數的確有差異。不過我們可以期望隨著解析度提高（也就是 j 越大），兩者的差異會越來越小。

有一點要注意的是，在解析空間 V_j 繪製圖形所使用的已經不是眞正的函數點值，而是使用了近似值。這是使用凌波函數所無法避免的。這是因爲自格函數 $\phi(x)$ 和凌波函數 $\psi(x)$ 都不是『插值函數』。在之後我們使用離散凌波轉換，這個問題會更令人感到困擾。不過我們最終的問題是增加解析度。即使插值能產生正確的函數點值，那也是針對多項式而言是如此。我們只要掌握一個原則

不論是插值或是凌波理論，就增加解析度而言都是逼近

綜合來看，光是繪製函數圖形，凌波理論好像就顯得差人一截。這是因爲我們已經知道函數 $f(x)$ 的形式。當已經知道函數是什麼，我們的確不必大費周章地運用凌波理論來繪製圖形；前面介紹的方法已經很實用了。不過，因爲我們將要處理的是影像，它是屬於二維的數位訊號。而凡是已經被離散化或是數位化的數位訊號而言，通常我們不可能知道決定它的函數。此外，我們將要在第二章討論的離散凌波轉

換是專門用來處理數位訊號的利器。因此，凌波理論的出現並不是取代之前的理論 (replacement)。毋寧說，針對問題的需要，凌波理論提供了另外的選擇 (alternative)。

第二章 區段上的離散凌波轉換

給定一個最高解析度的解析空間 V_J，我們可以在 $j \le J$ 解析空間利用 $V_j = V_{j-1} \oplus W_{j-1}$ 作基底之間的轉換。而針對投影係數和凌波係數也符合這個轉換的關係，這發展出離散凌波轉換。由於這裡建造的是週期化的離散凌波轉換，根據 [1, 第四章]，對二維的數位訊號（也就是圖片），經由離散凌波轉換的分解步驟在高頻部分會產生邊框效應。由於我們的應用主要在處理圖像，因此我們根據 [3] 建造區段上的離散凌波轉換。

2.1 離散凌波轉換

在介紹如何在解析空間 V_j 製作函數圖形的時候，我們已經看到函數點值 $f(x_n)$ 如何取代函數在解析空間 V_j 的投影係數 $\langle f, \phi_{jk} \rangle$。如果我們要製作在不同解析空間的函數圖形，還得再計算一次函數點值。但是由於 $V_j = V_{j-1} \oplus W_{j-1}$ 的關係，我們可以不必重新計算函數點值。而將 V_j 中的函數從 $\{\phi_{j,k}(x)\}$ 基底變換到 $V_{j-1} \oplus W_{j-1}$ 的基底 $\{\phi_{j-1,k}(x)\} \cup \{\psi_{j-1,k}(x)\}$ 的過程就是所謂的凌波轉換。我們稱從 $\{\phi_{j,k}(x)\}$ 轉換到 $\{\phi_{j-1,k}(x)\} \cup \{\psi_{j-1,k}(x)\}$ 的過程爲分解 (decomposition)，而從 $\{\phi_{j-1,k}(x)\} \cup \{\psi_{j-1,k}(x)\}$ 轉換到 $\{\phi_{j,k}(x)\}$ 的過程爲合成 (reconstruction)。

針對已被離散化的函數點值或是數位訊號

$$\boldsymbol{v} = (v_0, v_1, \cdots, v_{N-1}) \in \mathbb{R}^N$$

爲了方便，我們假設 $N = 2^j$。則凌波轉換就是在以下的等式之間作轉換。

$$\sum_{n=0}^{N-1} v_n \phi(2^j x - k) = \sum_{n=0}^{\frac{N}{2}-1} u_n \phi(2^{j-1} x - n) + \sum_{n=0}^{\frac{N}{2}-1} w_n \psi(2^{j-1} x - n)$$

而離散凌波轉換 (discrete wavelet transform) [1, 第 241 頁] 就是將係數 $\{v_n\}$ 轉換成 $\{u_n, w_n\}$ 和 $\{u_n, w_n\}$ 轉換成 $\{v_n\}$；前者是分解，後者稱爲合成。在 [1, 第 4.1 節] 有所有的推導的結果，我們不再抄錄。

我們講解離散凌波轉換是如何作用的。假設輸入一個向量 $\mathbf{v}$

$$\boldsymbol{v} = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{n-1} \end{pmatrix}$$

其中 $N = 2^j$。我們對它做離散凌波轉換，把 $\boldsymbol{v} \in V_j$ 分解成

$$\boldsymbol{u} = \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{\frac{N}{2}-1} \end{pmatrix} \in V_{j-1} \quad \text{和} \quad \boldsymbol{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{\frac{N}{2}-1} \end{pmatrix} \in W_{j-1}$$

而分解的步驟相當於兩個矩陣的乘法 $\boldsymbol{u} = \mathbf{H}\boldsymbol{v}$，$\boldsymbol{w} = \mathbf{G}\boldsymbol{v}$，其中 $\mathbf{H}$ 是低頻濾波器，而 $\mathbf{G}$ 是高頻濾波器。而我們把 $\mathbf{H}$ 和 $\mathbf{G}$ 堆在一起形成凌波轉換矩陣

$$\mathcal{W} = \begin{pmatrix} \mathbf{H} \\ \mathbf{G} \end{pmatrix}$$

其中 $\mathcal{W}$ 的維度是 $N \times N$。所以離散凌波轉換的分解過程相當是一個矩陣乘法

$$\mathcal{W}\boldsymbol{v} = \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{w} \end{pmatrix}$$

由於這裡的離散凌波轉換是依照 Daubechies 的正交凌波函數，因此離散凌波轉換的合成步驟就是 $\mathcal{W}^{-1} = 2\mathcal{W}^T$。

根據 [1, 第 61 頁]，在實作上我們並不眞的組成凌波轉換矩陣 $\mathcal{W}$，而是利用程式設計技巧把 $\mathbf{H}$ 和 $\mathbf{G}$ 的列向量交錯排列，再以每個列向量對 $\boldsymbol{v}$ 做向量乘法。那麼只需要一個暫存空間就可以把結果直接覆寫在 $\boldsymbol{v}$ 的儲存空間，然後把 $\boldsymbol{v}$ 的奇數列排在前半部，偶數列排在後半部，回到

$$\begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{w} \end{pmatrix}$$

的型態，就完成分解一次的離散凌波轉換。這個步驟可以重複地執行，直到滿意的解析度爲止。而在離散凌波轉換的合成過程，一樣先把 $\mathbf{H}$ 和 $\mathbf{G}$ 的列向量交錯排列，再將它們做轉置。要小心的是，這裡的 $\begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{w} \end{pmatrix}$ 也要交錯排列，形成

$$\begin{pmatrix} u_0 \\ w_0 \\ u_1 \\ w_1 \\ \vdots \\ u_{\frac{N}{2}} \\ w_{\frac{N}{2}} \end{pmatrix}$$

再以先前那個轉置矩陣的每一列對這個交錯排列的 $\begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{w} \end{pmatrix}$ 做向量乘法。

由 [1, 第 4.1 節] 的推導結果可知組成矩陣 $\mathbf{H}$ 和 $\mathbf{G}$ 的元素全都是自格係數 c_k。以 $p=2$，$j=3$ 爲例

$$\mathbf{H}=\frac{1}{2}\begin{pmatrix} c_0 & c_1 & c_2 & c_3 & & & & \\ & & c_0 & c_1 & c_2 & c_3 & & \\ & & & & c_0 & c_1 & c_2 & c_3 \\ & & & & & & \vdots & \\ c_2 & c_3 & & & & & c_0 & c_1 \end{pmatrix}$$

$$\mathbf{G}=\frac{1}{2}\begin{pmatrix} c_{2p-1} & -c_{2p-2} & c_{2p-3} & -c_{2p-4} & & & & \\ & & c_{2p-1} & -c_{2p-2} & c_{2p-3} & -c_{2p-4} & & \\ & & & & c_{2p-1} & -c_{2p-2} & c_{2p-3} & -c_{2p-4} \\ & & & & & & \vdots & \\ c_{2p-3} & -c_{2p-4} & & & & & c_{2p-1} & -c_{2p-2} \end{pmatrix}$$

矩陣 $\mathbf{H}$ 和 $\mathbf{G}$ 的最後一列會折回來，這就是所謂的週期化。只要階數大於 1 的時候，都免不了使用週期化的離散凌波轉換。下圖（圖六）是用一個二維影像 Tiffany 做一次週期化的離散凌波轉換。

圖六

我們可以看到在高頻的部分有兩條明顯的黑點，這是由週期化的離散凌波轉換所產生的邊框效應。以下的幾節，我們將建造區段上的離散凌波轉換來消除邊框效應。

2.2 區段上的自格函數

這一章的內容主要取材自鄭儘的論文 [3]。她的論文是以第一章所介紹的 Daubechies 的 2 階緊緻函蓋、正則的自格和凌波函數作爲根據，大部份按照 Andersson, Hall, Jawerth 及 Peters 的指引，建造靠近邊界的自格和凌波函數。我們在這一節先建造在區段上的自格函數，而在下一節建造在區段上的凌波函數。而本文推廣至 3 階的情形以作爲第三章的凌波演算法設計與評估。

先考慮特殊情形。$\phi(x)$ 和 $\psi(x)$ 是 Daubechies 的自格和凌波函數，階數 $p=3$，基準尺度 $j=0$ 時， $\operatorname{supp}\phi(x)=[0,5]$, $\operatorname{supp}\psi(x)=[0,5]$。在此我們令尺度 $j=3$ 爲最低基準。也就是當我們說 $\phi_{jk}(x)=\sqrt{2^j}\phi(2^jx-k)$，總是假設 $j\geq 3$ 的。如此，這樣的 ϕ_{jk} 才不會同時碰到 $[0,1]$ 的兩個端點。若直接使用 $\phi_{jk}|_{[0,1]}$ 作爲 $V_j|_{[0,1]}$ 的基底，則 $\dim V_j|_{[0,1]}=2^j+2(p-1)$。現在我們要定義新的 $V_j[0,1]$ 使其『內涵』不變，但 $\dim V_j[0,1]=2^j$，所謂的 $V_j[0,1]$ 的『內涵』不變，最主要的就是使 $V_j[0,1]$ 與 $V_j|_{[0,1]}$ 所含的多項式要一樣多。 首先，設定 $\delta_L=\delta_R=1$，當 $j\geq 3$

$$(\delta_L=)\ 1\leq k\leq 2^j-6\ (=2^j-2p+1-\delta_R)$$

ϕ_{jk} 的函蓋是整個落在$(0,1)$之內。我們保留這些函數爲基底。所以已經有了 2^j-6 個維度, 還差六個。我們將會在左右兩端各找三個。

投影函數

我們將靠邊界點(0或1)的 ϕ_{jk} 改成用多項式表達，再將它們正則化。

靠左端點 $(x=0)$，考慮多項式

$$\mathcal{X}_j^m(x)=\sqrt{2^j}(2^jx)^m,\quad 0\leq m\leq 2(p-1);$$

靠右端點 $(x=1)$，則考慮多項式

$$\mathcal{Y}_j^m(x)=\mathcal{X}_j^m(x-1)=\sqrt{2^j}(2^j(x-1))^m,\quad 0\leq m\leq 2(p-1).$$

對任一層 $j\geq 3$，要把 $\mathcal{X}_j^m(x)$ 投影到

$$\operatorname{span}\{\phi_{jk}(x)\mid \operatorname{supp}\phi_{jk}(x)\text{ 跨過左端點}0\ \}$$

$$=\operatorname{span}\{\phi_{jk}(x)\mid k=-4\ (=2-2p),-3,-2,-1,0\ (=\delta_L-1)\}$$

把 $\mathcal{Y}_j^m(x)$ 投影到

$$\begin{aligned}&\text{span }\{\phi_{jk}(x) \mid \text{supp }\phi_{jk}(x) \text{ 跨過右端點1}\}\\ =&\text{span }\{\phi_{jk}(x) \mid k = 2^j - 5\ (= 2^j - 2p + 2 - \delta_R), 2^j - 4, 2^j - 3, 2^j - 2, 2^j - 1\}\end{aligned}$$

$\mathcal{X}_j^m$ 的投影係數

$$\begin{aligned}\langle \mathcal{X}_j^m, \phi_{jk}\rangle &= 2^j \int (2^j x)^m \phi(2^j x - k)\, dx\\ &= \int (x + k)^m \phi(x)\, dx\\ &= \sum_{i=0}^{m} \binom{m}{i} k^i M_{m-i}\end{aligned}$$

根據 [1,(5.23)]，定義 $\phi(x)$ 的 m 次動量

$$M_m = \int x^m \phi(x)\, dx$$

其中，在此份論文中若無特別註明,我們以 $\int$ 來代表整個實數軸的積分，i.e. $\int = \int_{-\infty}^{\infty}$ 因此 $\mathcal{X}_j^m$ 的投影係數與 j 無關；將投影係數定爲 x_k^m，$m = 0, 1, 2$， $k = -4, -3, -2, -1, 0$。$\mathcal{Y}_j^m$ 的投影係數

$$\begin{aligned}\langle \mathcal{Y}_j^m, \phi_{jk}\rangle &= 2^j \int (2^j(x-1))^m \phi(2^j x - k)\, dx\\ &= \int (x + k - 2^j)^m \phi(x)\, dx.\end{aligned}$$

但是 k 的值是 $2^j - 2p + l, l = 1, 2, \ldots, 2p - 1$，使得 y_k^m 與 j 無關。令 $y_k^m = \langle \mathcal{Y}_k^m, \phi_{jk}\rangle$, $m = 0, 1, 2$, $k = 2^j - 5, 2^j - 4, \ldots, 2^j - 1$。 x_k^m 和 y_k^m 的數值可以由 M_m 精確的算出來 (xphi.m)。

令投影函數爲

$$\begin{aligned}X_j^m &= \sum_{k=-4}^{0} x_k^m \phi_{jk} \mid_{[0,1]}\\ Y_j^m &= \sum_{k=2^j-5}^{2^j-1} y_k^m \phi_{jk} \mid_{[0,1]}\end{aligned}$$

可驗證 $\{X_j^m\}_{m=0}^2$ 和 $\{Y_j^m\}_{m=0}^2$ 這兩個集合的函數是線性無關的。而且由於正交的關係，此二集合和 $\{\phi_{jk}\}_{k=1}^{2^j-4}$ 相互正交。而且後者已經是一個正則集合，只須再把前二個集合中的線性無關函數作正則化處理，就有一整個正則集合，個數爲 2^j。定義 $V_j[0,1]$ 就是這個正則集合所展開的線性空間。

正則化

我們在此使用一個標準的正則化步驟。亦即，算出 $\{X_j^m\}_{m=0}^2$ 的質量矩陣 $\mathbf{X} = (X_{mn})$，其中 $X_{mn} = \langle X_j^m, X_j^n \rangle$。 注意: $\mathbf{X}$ 其實與 j 無關。計算這些數時，我們須要這些半積分:

$$\begin{aligned}
N_1 &= \int_\infty^0 \phi(x+4)\phi(x+4)\,dx & N_6 &= \int_\infty^0 \phi(x+4)\phi(x+2)\,dx \\
N_2 &= \int_\infty^0 \phi(x+4)\phi(x+3)\,dx & N_7 &= \int_\infty^0 \phi(x+3)\phi(x+1)\,dx \\
N_3 &= \int_\infty^0 \phi(x+4)\phi(x+2)\,dx & N_8 &= \int_\infty^0 \phi(x+2)\phi(x+2)\,dx \\
N_4 &= \int_\infty^0 \phi(x+4)\phi(x+1)\,dx & N_9 &= \int_\infty^0 \phi(x+2)\phi(x+1)\,dx \\
N_5 &= \int_\infty^0 \phi(x+3)\phi(x+3)\,dx & N_{10} &= \int_\infty^0 \phi(x+1)\phi(x+1)\,dx
\end{aligned}$$

推導 $\mathbf{X}_{mn}$如下:

$$X = \begin{pmatrix} \langle X_j^0, X_j^0 \rangle & \langle X_j^0, X_j^1 \rangle & \langle X_j^0, X_j^2 \rangle \\ \langle X_j^1, X_j^0 \rangle & \langle X_j^1, X_j^1 \rangle & \langle X_j^1, X_j^2 \rangle \\ \langle X_j^2, X_j^0 \rangle & \langle X_j^2, X_j^1 \rangle & \langle X_j^2, X_j^2 \rangle \end{pmatrix}$$

其中

$$\begin{aligned}
\langle X_j^m, X_j^n \rangle = {} & x_{-4}^m x_{-4}^n N_1 + \left(x_{-4}^m x_{-3}^n + x_{-3}^m x_{-4}^n\right) N_2 + \left(x_{-4}^m x_{-2}^n + x_{-2}^m x_{-4}^n\right) N_3 \\
& + \left(x_{-4}^m x_{-1}^n + x_{-1}^m x_{-4}^n\right) N_4 + x_{-3}^m x_{-3}^n N_5 + \left(x_{-3}^m x_{-2}^n + x_{-2}^m x_{-3}^n\right) N_6 \\
& + \left(x_{-3}^m x_{-1}^n + x_{-1}^m x_{-3}^n\right) N_7 + x_{-2}^m x_{-2}^n N_8 + \left(x_{-2}^m x_{-1}^n + x_{-1}^m x_{-2}^n\right) N_9 \\
& + x_{-1}^m x_{-1}^n N_{10} + x_0^m x_0^n
\end{aligned}$$

推導 Y_{mn} 需要有以下的半積分的值：

$$\begin{aligned}
T_1 &= \int_{-\infty}^0 \phi(x+4)\phi(x+4)\,dx = 1 - N_1 & T_6 &= \int_{-\infty}^0 \phi(x+4)\phi(x+2)\,dx = -N_3 \\
T_2 &= \int_{-\infty}^0 \phi(x+4)\phi(x+3)\,dx = -N_2 & T_7 &= \int_{-\infty}^0 \phi(x+3)\phi(x+1)\,dx = -N_7 \\
T_3 &= \int_{-\infty}^0 \phi(x+4)\phi(x+2)\,dx = -N_3 & T_8 &= \int_{-\infty}^0 \phi(x+2)\phi(x+2)\,dx = 1 - N_8 \\
T_4 &= \int_{-\infty}^0 \phi(x+4)\phi(x+1)\,dx = -N_4 & T_9 &= \int_{-\infty}^0 \phi(x+2)\phi(x+1)\,dx = -N_9 \\
T_5 &= \int_{-\infty}^0 \phi(x+3)\phi(x+3)\,dx = 1 - N_5 & T_{10} &= \int_{-\infty}^0 \phi(x+1)\phi(x+1)\,dx = 1 - N_{10}
\end{aligned}$$

推導 $\mathbf{Y}_{mn}$ 如下:

$$Y = \begin{pmatrix} \langle Y_j^0,\, Y_j^0\rangle & \langle Y_j^0,\, Y_j^1\rangle & \langle Y_j^0,\, Y_j^2\rangle \\ \langle Y_j^1,\, Y_j^0\rangle & \langle Y_j^1,\, Y_j^1\rangle & \langle Y_j^1,\, Y_j^2\rangle \\ \langle Y_j^2,\, Y_j^0\rangle & \langle Y_j^2,\, Y_j^1\rangle & \langle Y_j^2,\, Y_j^2\rangle \end{pmatrix}$$

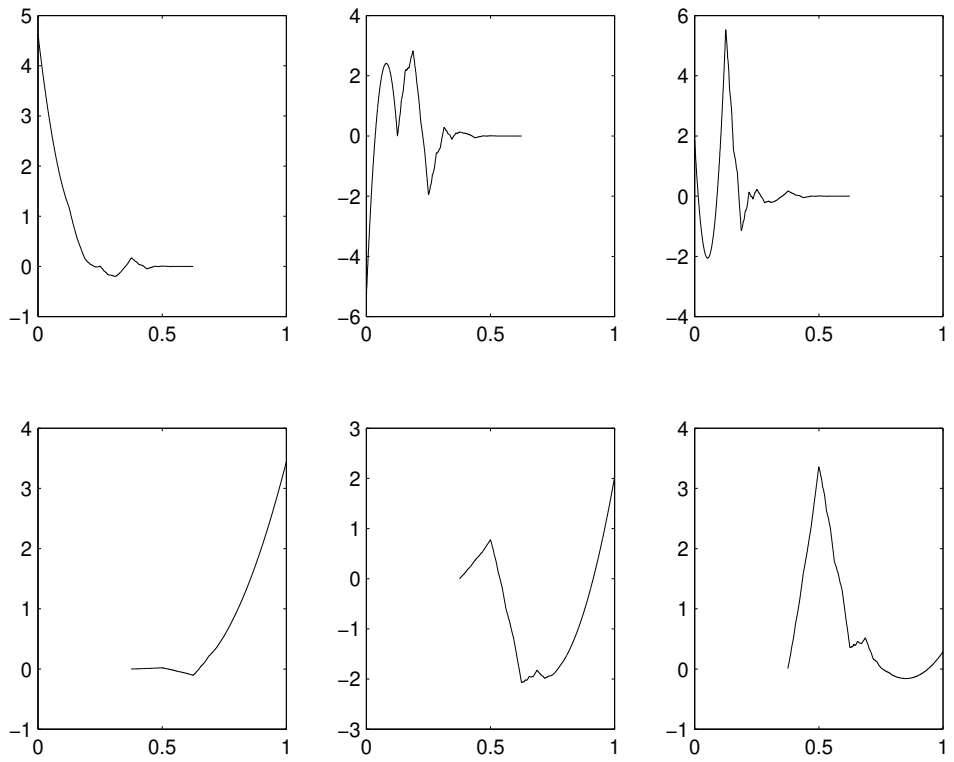
其中

$$\begin{aligned} \langle Y_j^m, Y_j^n\rangle = {} & y_{r_4}^m y_{r_4}^n T_1 + \left(y_{r_4}^m y_{r_3}^n + y_{r_3}^m y_{r_4}^n\right) T_2 + \left(y_{r_4}^m y_{r_2}^n + y_{r_2}^m y_{r_4}^n\right) T_3 \\ & + \left(y_{r_4}^m y_{r_1}^n + y_{r_1}^m y_{r_4}^n\right) T_4 + y_{r_3}^m y_{r_3}^n T_5 + \left(y_{r_3}^m y_{r_2}^n + y_{r_2}^m y_{r_3}^n\right) T_6 \\ & + \left(y_{r_3}^m y_{r_1}^n + y_{r_1}^m y_{r_3}^n\right) T_7 + y_{r_2}^m y_{r_2}^n T_8 + \left(y_{r_2}^m y_{r_1}^n + y_{r_1}^m y_{r_2}^n\right) T_9 \\ & + y_{r_1}^m y_{r_1}^n T_{10} + y_{r_5}^m y_{r_5}^n \end{aligned}$$

而 $r_1 = 2^j - 1$, $r_2 = 2^j - 2$, $r_3 = 2^j - 3$, $r_4 = 2^j - 4$, $r_5 = 2^j - 5$(mass.m)。明顯地，$\mathbf{X}$ 和 $\mathbf{Y}$ 都是對稱正定矩陣。所以存在 $\mathbf{S} = \mathbf{X}^{-1/2}$ 和 $\mathbf{T} = \mathbf{Y}^{-1/2}$。定義:

$$\begin{pmatrix} \varphi_{j,-2}(x) \\ \varphi_{j,-1}(x) \\ \varphi_{j,0}(x) \end{pmatrix} = \mathbf{S} \begin{pmatrix} X_j^0(x) \\ X_j^1(x) \\ X_j^2(x) \end{pmatrix} \qquad \begin{pmatrix} \varphi_{j,r_1}(x) \\ \varphi_{j,r_2}(x) \\ \varphi_{j,r_3}(x) \end{pmatrix} = \mathbf{T} \begin{pmatrix} Y_j^0(x) \\ Y_j^1(x) \\ Y_j^2(x) \end{pmatrix}$$

則 $\{\varphi_{j,-2}, \varphi_{j,-1}, \varphi_{j,0}, \phi_{j,1}, \phi_{j,2}, \ldots, \phi_{j,r_0}, \varphi_{j,r_1}, \varphi_{j,r_2}, \varphi_{j,r_3}\}$ 形成 $V_j[0,1]$ 的一組正則基底 (or_phi.m)。我們畫出在 $V_3[0,1]$ 我們所建造的在區段上的自格函數（圖七）。圖七上是靠區段左邊的自格函數，圖七下則是右邊的部分。



圖七

在此 $r_0 = 2^j - 6$。爲符號上的方便， r_0 永遠代表第 j 層屬於內部基底的最後一個足標， $r_n = r_0 + n$。而 j 可以由前後文中闡明。 注意: 靠邊界的自格函數在靠邊界點的 $\frac{1}{2^j}$ 長度實際上是多項式。

低頻濾波器

我們所製造的 $V_j[0,1]$ 並不等於 $V_j|_{[0,1]}$，但 $\cup V_j[0,1]$ 對 $L^2([0,1])$ 是稠密的(dense)，而且 $V_j[0,1]$ 仍保有原始凌波函數中層層包含之關係:

$$V_j[0,1] \subset V_{j+1}[0,1] \subset V_{j+2}[0,1] \subset \dots$$

因此，$V_j[0,1]$ 的基底可由 $V_{j+1}[0,1]$ 的基底重建回來。其關係如下:

考慮 $j = 3$ 之情況:

$$\Phi_3 = \begin{pmatrix} \varphi_{3,-2} \\ \varphi_{3,-1} \\ \varphi_{3,0} \\ \vdots \\ \varphi_{3,3} \\ \varphi_{3,4} \\ \varphi_{3,5} \end{pmatrix} \qquad \Phi_4 = \begin{pmatrix} \varphi_{4,-2} \\ \varphi_{4,-1} \\ \varphi_{4,0} \\ \vdots \\ \varphi_{4,11} \\ \varphi_{4,12} \\ \varphi_{4,13} \end{pmatrix}$$

令 $\Phi_3 = \mathbf{H}\Phi_4$, 則 $\mathbf{H}$ 是8×16 的矩陣。矩陣 $\mathbf{H}$ 就是對應於第四層 $(j = 4)$ 的低頻濾波(lowpass filter)。因爲 $\{\varphi_{4,-2}, \varphi_{4,-1}, \varphi_{4,0}, \phi_{4,1}, \dots, \phi_{4,10}, \varphi_{4,11}, \varphi_{4,12}, \varphi_{4,13}\}$ 是 $V_4[0,1]$ 的一組正則基底。因此 $\mathbf{H}$ 的一般項 $\mathbf{H}_{mn} = \langle \varphi_{3,m}, \varphi_{4,n} \rangle \quad m = -2, -1, 0, \dots, 5 \quad n = -2, -1, 0, \dots, 13$。若將 $\mathbf{H}$ 按 $\mathbf{H}_{mn} = \langle \varphi_{3,m}, \varphi_{4,n} \rangle$ 中的 $\varphi_{3,m}, \varphi_{4,n}$ 分爲左、中、右，而劃分爲九個區域，則

$$\mathbf{H} = \begin{pmatrix} \mathcal{H}_{\mathcal{LL}} & \mathcal{H}_{\mathcal{LI}} & \mathcal{H}_{\mathcal{LR}} \\ \mathcal{H}_{\mathcal{IL}} & \mathcal{H}_{\mathcal{II}} & \mathcal{H}_{\mathcal{IR}} \\ \mathcal{H}_{\mathcal{RL}} & \mathcal{H}_{\mathcal{RI}} & \mathcal{H}_{\mathcal{RR}} \end{pmatrix}$$

我們以這九個子矩陣來分析矩陣 H 的結構:

$\mathcal{H}_{\mathcal{LL}}$

$\mathcal{H}_{\mathcal{LL}} = \langle \varphi_{3,m}, \varphi_{4,n} \rangle \quad m = -2,-1,0 \quad n = -2,-1,0$ 是 3×3 的矩陣,

$$H_{LL} = \begin{pmatrix} \langle \varphi_{3,-2},\, \varphi_{4,-2} \rangle & \langle \varphi_{3,-2},\, \varphi_{4,-1} \rangle & \langle \varphi_{3,-2},\, \varphi_{4,0} \rangle \\ \langle \varphi_{3,-1},\, \varphi_{4,-2} \rangle & \langle \varphi_{3,-1},\, \varphi_{4,-1} \rangle & \langle \varphi_{3,-1},\, \varphi_{4,0} \rangle \\ \langle \varphi_{3,0},\, \varphi_{4,-2} \rangle & \langle \varphi_{3,0},\, \varphi_{4,-1} \rangle & \langle \varphi_{3,0},\, \varphi_{4,0} \rangle \end{pmatrix}$$

$$= S \begin{pmatrix} \langle X_3^0,\, X_4^0 \rangle & \langle X_3^0,\, X_4^1 \rangle & \langle X_3^0,\, X_4^2 \rangle \\ \langle X_3^1,\, X_4^0 \rangle & \langle X_3^1,\, X_4^1 \rangle & \langle X_3^1,\, X_4^2 \rangle \\ \langle X_3^2,\, X_4^0 \rangle & \langle X_3^2,\, X_4^1 \rangle & \langle X_3^2,\, X_4^2 \rangle \end{pmatrix} S^t$$

$\mathcal{H}_{\mathcal{LI}}$

$H_{LI} = \langle \varphi_{3,m}, \phi_{4,n} \rangle \quad m = -2,-1,0 \quad n = 1,2,\ldots,10$ 是 3×10的矩陣，最前面5行是:

$$\begin{pmatrix} \langle \varphi_{3,-2},\, \phi_{4,1} \rangle & \langle \varphi_{3,-2},\, \phi_{4,2} \rangle & \langle \varphi_{3,-2},\, \phi_{4,3} \rangle & \langle \varphi_{3,-2},\, \phi_{4,4} \rangle & \langle \varphi_{3,-2},\, \phi_{4,5} \rangle \\ \langle \varphi_{3,-1},\, \phi_{4,1} \rangle & \langle \varphi_{3,-1},\, \phi_{4,2} \rangle & \langle \varphi_{3,-1},\, \phi_{4,3} \rangle & \langle \varphi_{3,-1},\, \phi_{4,4} \rangle & \langle \varphi_{3,-1},\, \phi_{4,5} \rangle \\ \langle \varphi_{3,0},\, \phi_{4,1} \rangle & \langle \varphi_{3,0},\, \phi_{4,2} \rangle & \langle \varphi_{3,0},\, \phi_{4,3} \rangle & \langle \varphi_{3,0},\, \phi_{4,4} \rangle & \langle \varphi_{3,0},\, \phi_{4,5} \rangle \end{pmatrix}$$

$$= \frac{1}{\sqrt{2}} \cdot S \begin{pmatrix} c_1 + c_3 + c_5 & c_2 + c_4 & c_3 + c_5 & c_4 & c_5 \\ x_0^1 c_1 + x_{-1}^1 c_3 + x_{-2}^1 c_5 & x_0^1 c_2 + x_{-1}^1 c_4 & x_0^1 c_3 + x_{-1}^1 c_5 & x_0^1 c_4 & x_0^1 c_5 \\ x_0^2 c_1 + x_{-1}^2 c_3 + x_{-2}^2 c_5 & x_0^2 c_2 + x_{-1}^2 c_4 & x_0^2 c_3 + x_{-1}^2 c_5 & x_0^2 c_4 & x_0^2 c_5 \end{pmatrix}$$

其餘的 3×5 部分因爲正交的關係是零矩陣。

$\mathcal{H}_{\mathcal{II}}$

$H_{II} = \langle \phi_{3,m}, \phi_{4,n} \rangle$， $m = 1,2$，$n = 1,2,\ldots,10$ 是 2×10 的矩陣。

$$H_{II} = \begin{pmatrix} \langle \phi_{3,1},\, \phi_{4,1} \rangle & \langle \phi_{3,1},\, \phi_{4,2} \rangle & \langle \phi_{3,1},\, \phi_{4,3} \rangle & \ldots & \langle \phi_{3,1},\, \phi_{4,10} \rangle \\ \langle \phi_{3,2},\, \phi_{4,1} \rangle & \langle \phi_{3,2},\, \phi_{4,2} \rangle & \langle \phi_{3,2},\, \phi_{4,3} \rangle & \ldots & \langle \phi_{3,2},\, \phi_{4,10} \rangle \end{pmatrix}$$

$$= \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 0 & c_0 & c_1 & c_2 & c_3 & c_4 & c_5 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_0 & c_1 & c_2 & c_3 & c_4 & c_5 & 0 \end{pmatrix}$$

$\mathcal{H}_{\mathcal{RI}}$

$H_{RI} = \langle \varphi_{3,m}, \phi_{4,n} \rangle$， $m = 3,4,5$，$n = 1,2,\ldots,10$ 是 3×10 的矩陣，最後面 5 行是:

$$H_{RI} = \begin{pmatrix} \langle \varphi_{3,3},\, \phi_{4,6} \rangle & \langle \varphi_{3,3},\, \phi_{4,7} \rangle & \langle \varphi_{3,3},\, \phi_{4,8} \rangle & \langle \varphi_{3,3},\, \phi_{4,9} \rangle & \langle \varphi_{3,3},\, \phi_{4,10} \rangle \\ \langle \varphi_{3,4},\, \phi_{4,6} \rangle & \langle \varphi_{3,4},\, \phi_{4,7} \rangle & \langle \varphi_{3,4},\, \phi_{4,8} \rangle & \langle \varphi_{3,4},\, \phi_{4,9} \rangle & \langle \varphi_{3,4},\, \phi_{4,10} \rangle \\ \langle \varphi_{3,5},\, \phi_{4,6} \rangle & \langle \varphi_{3,5},\, \phi_{4,7} \rangle & \langle \varphi_{3,5},\, \phi_{4,8} \rangle & \langle \varphi_{3,5},\, \phi_{4,9} \rangle & \langle \varphi_{3,5},\, \phi_{4,10} \rangle \end{pmatrix}$$

$$= \frac{1}{\sqrt{2}} \cdot T \begin{pmatrix} c_0 & c_1 & c_2 + c_0 & c_3 + c_1 & c_4 + c_2 + c_0 \\ y_3^1 c_0 & y_3^1 c_1 & y_3^1 c_2 + y_4^1 c_0 & y_3^1 c_3 + y_4^1 c_1 & y_3^1 c_4 + y_4^1 c_2 + y_5^1 c_0 \\ y_3^2 c_0 & y_3^2 c_1 & y_3^2 c_2 + y_4^2 c_0 & y_3^2 c_3 + y_4^2 c_1 & y_3^2 c_4 + y_4^2 c_2 + y_5^2 c_0 \end{pmatrix}$$

前面的3×5部分因爲正交的關係是零矩陣。

$\mathcal{H}_{\mathcal{RR}}$

$H_{RR}=\langle\varphi_{3,m},\varphi_{4,n}\rangle$， $m=3,4,5$，$n=11,12,13$ 是 3×3的矩陣，

$$H_{RR}=\begin{pmatrix}\langle\varphi_{3,3},\varphi_{4,11}\rangle & \langle\varphi_{3,3},\varphi_{4,12}\rangle & \langle\varphi_{3,3},\varphi_{4,13}\rangle\\ \langle\varphi_{3,4},\varphi_{4,11}\rangle & \langle\varphi_{3,4},\varphi_{4,12}\rangle & \langle\varphi_{3,4},\varphi_{4,13}\rangle\\ \langle\varphi_{3,5},\varphi_{4,11}\rangle & \langle\varphi_{3,5},\varphi_{4,12}\rangle & \langle\varphi_{3,5},\varphi_{4,13}\rangle\end{pmatrix}$$

$$=T\begin{pmatrix}\langle Y_3^0,Y_4^0\rangle & \langle Y_3^0,Y_4^1\rangle & \langle Y_3^0,Y_4^2\rangle\\ \langle Y_3^1,Y_4^0\rangle & \langle Y_3^1,Y_4^1\rangle & \langle Y_3^1,Y_4^2\rangle\\ \langle Y_3^2,Y_4^0\rangle & \langle Y_3^2,Y_4^1\rangle & \langle Y_3^2,Y_4^2\rangle\end{pmatrix}T^t$$

其餘的 $\mathcal{H}_{\mathcal{LR}}$, $\mathcal{H}_{\mathcal{IL}}$, $\mathcal{H}_{\mathcal{IR}}$, $\mathcal{H}_{\mathcal{RL}}$ 均因爲正交的關係都是零矩陣(lpfilter.m)。雖然矩陣 $\mathbf{H}$ 的大小隨 j 而變 ($\mathbf{H}$ 是 $2^j\times2^{j+1}$ 的矩陣)，但是其非零數值不隨 j 而變。

2.3 區段上的凌波函數

首先，我們定義 $W_j[0,1]$ 是 $V_{j+1}[0,1]$ 中與 $V_j[0,1]$ 垂直的空間。因爲 $\dim V_j[0,1]=2^j$ 而且 $V_{j+1}[0,1]=V_j[0,1]\oplus W_j[0,1]$，所以 $\dim W_j[0,1]=2^j$。亦即，$W_j[0,1]$ 應當有 2^j 個基底。

第一步，我們先確定有那些原始的凌波函數 $\psi_{j,k}(x)$，仍然在我們定義的 $V_{j+1}[0,1]$ 裡。在原始的自格函數 $\phi(x)$ 和凌波函數 $\psi(x)$ 有下列關係:

$$\psi(x)=\sum_{k=0}^{2p-1}(-1)^{k+1}c_{2p-1-k}\phi(2x-k)$$

因爲 $\operatorname{supp}\phi(x)=[0,2p-1]$，$\operatorname{supp}\psi(x)$ 也是 $[0,2p-1]$，可以得到 $\operatorname{supp}\psi_{j,k}=[\frac{k}{2^j},\frac{2p-1+k}{2^j}]$。而我們所要找尋的足標 k 就是使得 $\operatorname{supp}\psi_{j,k}\subseteq(0,1)$ 的那些足標

$$\operatorname{supp}\psi_{j,k}=[k/2^j,(2p-1+k)/2^j]\subseteq(0,1)$$

$$\Rightarrow\quad k=1,\ldots,2^j-2p$$

$$\Rightarrow\quad 1\le k\le 2^j-6$$

$$\Rightarrow\quad \{\psi_{j,k}\}_{k=1}^{2^j-6}\in W_j[0,1]$$

現在我們確定有 2^j-6 個正則基底仍然在 $W_j[0,1]$ 裏面。因此同樣需要在左右兩邊各找三個基底。到目前爲止，我們已有了 $V_j[0,1]$ 以及 $V_{j+1}[0,1]$ 的正則基底，也知道 $W_j[0,1]$ 的 2^j-6 個基底。因爲 $V_{j+1}[0,1]=V_j[0,1]\oplus W_j[0,1]$，所以 $V_j[0,1]$ 的基底聯集

$W_j[0,1]$ 的基底會構成 $V_{j+1}[0,1]$ 的一組基底。因此我們檢查 $V_{j+1}[0,1]$ 的基底中有哪些不能由 $V_j[0,1]$ 及 $\{\psi_{j,k}\}_{k=1}^{2^j-4}$ 線性組合所展開，再把它們投影至 $W_j[0,1]$ 成爲 $W_j[0,1]$ 的基底。

我們推測 $\varphi_{j+1,-2}(x)$, $\varphi_{j+1,-1}(x)$,$\varphi_{j+1,0}(x)$, $\varphi_{j+1,r_1}(x)$, $\varphi_{j+1,r_2}(x)$,$\varphi_{j+1,r_3}(x)$ 這六個原本不在 $V_{j+1}|_{[0,1]}$ 的函數並不能由 $V_j[0,1]$ 及 $\{\psi_{j,k}\}_{k=1}^{2^j-6}$ 的線性組合所展開。亦即:

$\varphi_{j+1,-2}(x) \notin \text{span } S$

$\varphi_{j+1,-1}(x) \notin \text{span } \{S \cup \varphi_{j+1,-2}(x)\}$

$\varphi_{j+1,0}(x) \notin \text{span } \{S \cup \varphi_{j+1,-2}(x) \cup \varphi_{j+1,-1}(x)\}$

$\varphi_{j+1,r_1}(x) \notin \text{span } \{S \cup \varphi_{j+1,-2}(x) \cup \varphi_{j+1,-1}(x) \cup \varphi_{j+1,0}(x)\}$

$\varphi_{j+1,r_2}(x) \notin \text{span } \{S \cup \varphi_{j+1,-2}(x) \cup \varphi_{j+1,-1}(x) \cup \varphi_{j+1,0}(x) \cup \varphi_{j+1,r_1}(x)\}$

$\varphi_{j+1,r_3}(x) \notin \text{span } \{S \cup \varphi_{j+1,-2}(x) \cup \varphi_{j+1,-1}(x) \cup \varphi_{j+1,0}(x) \cup \varphi_{j+1,r_1}(x) \cup \varphi_{j+1,r_2}(x)\}$

其中集合 S 爲 $\{V_j[0,1] \cup \{\psi_{j,k}(x)\}_{k=1}^{2^j-6}\}$

我們將只驗證 $\varphi_{j+1,-2}(x) \notin \text{span } \{V_j[0,1] \cup \{\psi_{j,k}(x)\}_{k=1}^{2^j-6}\}$，其餘的驗證均相似。驗證:

假設 $\varphi_{j+1,-2}(x) \in \text{span } \{V_j[0,1] \cup \{\psi_{j,k}(x)\}_{k=1}^{2^j-6}\}$，亦即

$$\exists a_{-2}, a_{-1}, a_0, a_1, a_2, b_1, b_2 \in \mathbb{R}$$

不全爲零，使得

$$\varphi_{j+1,-2} = a_{-2}\varphi_{j,-2} + a_{-1}\varphi_{j,-1} + a_0\varphi_{j,0} + a_1\phi_{j,1} + b_1\psi_{j,1} + a_2\phi_{j,2} + b_2\psi_{j,2}$$

因爲supp $\varphi_{j+1,-1} = [0, \frac{5}{2^{j+1}}]$。但只有兩個函數 $\phi_{j,2}, \psi_{j,2}$ 的函蓋包含 $[\frac{6}{2^j}, \frac{7}{2^j}]$，所以

$$a_2\phi_{j,2} + b_2\psi_{j,2}\,|_{[\frac{6}{2^j}, \frac{7}{2^j}]} = 0$$
$$\Rightarrow \quad a_2 = b_2 = 0$$

如果 $a_2 \neq 0$， $\phi_{j,2} = \frac{b_2}{a_2}\psi_{j,2}\,|_{[6/2^j, 7/2^j]}$。但是由於 ϕ 正交於 ψ，這是不可能的。同樣地，考慮 $[\frac{5}{2^j}, \frac{6}{2^j}]$ 得到 $a_1 = b_1 = 0$ 以及 $[\frac{4}{2^j}, \frac{5}{2^j}]$ 得到 $a_{-2} = a_{-1} = a_0 = 0$。

■

我們將這六個函數投影至 $W_j[0,1]$。定義投影函數:

$$\Psi^0_{jL} = \varphi_{j+1,-2} - P_j|_{V_j[0,1]}\varphi_{j+1,-2}$$

$$\Psi^1_{jL} = \varphi_{j+1,-1} - P_j|_{V_j[0,1]}\varphi_{j+1,-1}$$

$$\Psi^2_{jL} = \varphi_{j+1,0} - P_j|_{V_j[0,1]}\varphi_{j+1,0}$$

$$\Psi^0_{jR} = \varphi_{j+1,r_1} - P_j|_{V_j[0,1]}\varphi_{j+1,r_1}$$

$$\Psi^1_{jR} = \varphi_{j+1,r_2} - P_j|_{V_j[0,1]}\varphi_{j+1,r_2}$$

$$\Psi^2_{jR} = \varphi_{j+1,r_3} - P_j|_{V_j[0,1]}\varphi_{j+1,r_3}$$

則 Ψ^*_{jL}, Ψ^*_{jR} 彼此線性獨立 (psi_proj.m) 也與 $\{\psi_{j,k}\}_{k=1}^{2^j-6}$ 線性獨立，相互正交。由於 $\{\psi_{j,k}\}_{k=1}^{2^j-6}$ 已是正則集合，所以我們只需正則化 Ψ^*_{jL}, Ψ^*_{jR}，那麼就可得到一組爲數 2^j 的 $W_j[0,1]$ 的正則基底。

分別計算 Ψ^*_{jL} 的質量矩陣 $\hat{\mathbf{X}} = (\hat{\mathbf{X}}_{mn})$， Ψ^*_{jR} 的質量矩陣$\hat{\mathbf{Y}} = (\hat{\mathbf{Y}}_{mn})$

$$\hat{\mathbf{X}}_{mn} = \langle \Psi^m_{jL}, \Psi^n_{jL} \rangle$$

$$\hat{\mathbf{Y}}_{mn} = \langle \Psi^m_{jR}, \Psi^n_{jR} \rangle$$

我們可利用第二章第一節的 $\mathbf{H}$，推導得

$$\hat{\mathbf{X}} = I - H^t_{LL} H_{LL}$$

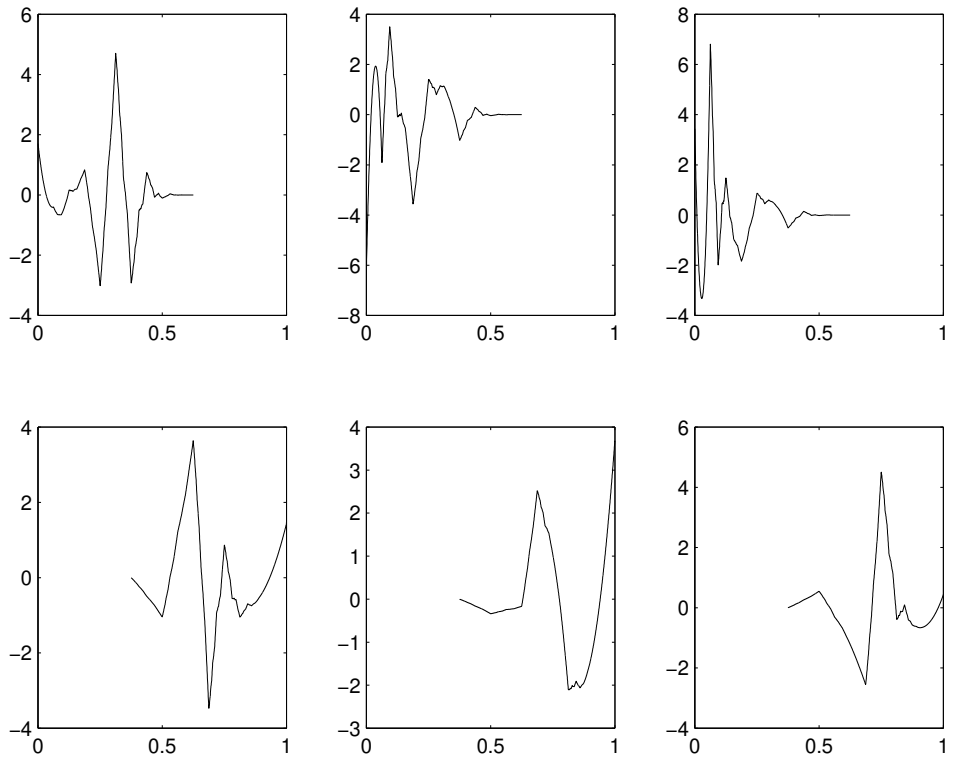
$$\hat{\mathbf{Y}} = I - H^t_{RR} H_{RR}$$

其中，I 是 3×3 的單位矩陣 (psi_mass.m)。

$\hat{\mathbf{X}}$ 和 $\hat{\mathbf{Y}}$ 都是對稱正定矩陣，且與 j 無關。定義:

$$\begin{pmatrix} \psi_{j,-2}(x) \\ \psi_{j,-1}(x) \\ \psi_{j,0}(x) \end{pmatrix} = \hat{\mathbf{X}}^{-1/2} \begin{pmatrix} \Psi^0_{jL}(x) \\ \Psi^1_{jL}(x) \\ \Psi^2_{jL}(x) \end{pmatrix} \qquad \begin{pmatrix} \psi_{j,r_1}(x) \\ \psi_{j,r_2}(x) \\ \psi_{j,r_3}(x) \end{pmatrix} = \hat{\mathbf{Y}}^{-1/2} \begin{pmatrix} \Psi^0_{jR}(x) \\ \Psi^1_{jR}(x) \\ \Psi^2_{jR}(x) \end{pmatrix}$$

則 $\{\psi_{j,-2}, \psi_{j,-1}, \psi_{j,0}, \psi_{j,1}, \ldots, \psi_{j,r_0}, \psi_{j,r_1}, \psi_{j,r_2}\}$ 形成 $W_j[0,1]$ 的一組正則基底 (or_psi.m)。我們畫出在 $W_3[0,1]$ 我們所建造的在區段上的凌波函數（圖八）。圖八上是靠區段左邊的凌波函數，圖八下則是右邊的部分。



圖八

由於 $W_j[0,1]$ 是 $V_{j+1}[0,1]$ 的一個子空間，因此 $W_j[0,1]$ 的基底可由 $V_{j+1}[0,1]$ 的基底重建回來。考慮 $j=3$ 之情況。

$$\Psi_3 = \begin{pmatrix} \psi_{3,-2} \\ \psi_{3,-1} \\ \psi_{3,0} \\ \vdots \\ \psi_{3,3} \\ \psi_{3,4} \\ \psi_{3,5} \end{pmatrix} \qquad \Phi_4 = \begin{pmatrix} \varphi_{4,-2} \\ \varphi_{4,-1} \\ \varphi_{4,0} \\ \vdots \\ \varphi_{4,11} \\ \varphi_{4,12} \\ \varphi_{4,13} \end{pmatrix}$$

令 $\Psi_3 = \mathbf{G}\Phi_4$，則 $\mathbf{G}$ 是 8×16 的矩陣。 $\mathbf{G}_{mn} = \langle \psi_{3,m}, \varphi_{4,n} \rangle$, $m = -2, -1, \ldots, 5$, $n = -2, -1, \ldots, 13$。

$$\mathbf{G} = \begin{pmatrix} \mathbf{G}_{LL} & \mathbf{G}_{LI} & \mathbf{G}_{LR} \\ \mathbf{G}_{IL} & \mathbf{G}_{II} & \mathbf{G}_{IR} \\ \mathbf{G}_{RL} & \mathbf{G}_{RI} & \mathbf{G}_{RR} \end{pmatrix}$$

$$= \begin{pmatrix} (\mathrm{I} - \mathbf{H}_{LL}^t \mathbf{H}_{LL})^{\frac{1}{2}} & -(\mathrm{I} - \mathbf{H}_{LL}^t \mathbf{H}_{LL})^{\frac{-1}{2}} (\mathbf{H}_{LL}^t \mathbf{H}_{LI}) & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_{II} & \mathbf{0} \\ \mathbf{0} & -(\mathrm{I} - \mathbf{H}_{RR}^t \mathbf{H}_{RR})(\mathbf{H}_{RR}^t \mathbf{H}_{RI}) & (\mathrm{I} - \mathbf{H}_{RR}^t \mathbf{H}_{RR})^{\frac{1}{2}} \end{pmatrix}$$

其中

$$\mathbf{G}_{II} = \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 0 & c_5 & -c_4 & c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_5 & -c_4 & c_3 & -c_2 & c_1 & -c_0 & 0 \end{pmatrix}$$

I 是 3×3 的單位矩陣 (lpfilter.m)。

附註 *1:* 在一般的情況下，

$$\dim V_j[0,1] = 2^j - \delta_L - \delta_R + 2, \quad \dim W_j[0,1] = 2^j$$

選擇越大的 δ_L, δ_R 會使得靠邊界的自格函數 φ 函蓋越大。所以我們選擇 $\delta_L = \delta_R = 1$ 除了可以使 $V_j[0,1]$ 與 $W_j[0,1]$ 有相同的維度外，還使得靠邊界的自格函數和內部的自格函數有相同長度的函蓋。

附註 *2:* 另外，H 是低頻濾波器 (low pass filter)，G 是高頻濾波器 (high pass filter)。在一般的情況下，H 和 G 是 $2^j \times 2^{j+1}$ 的矩陣， H_{LL}、G_{LL}、H_{RR} 和 G_{RR} 是 $p \times p$ 的矩陣， H_{LI}、G_{LI}、H_{RI} 和 G_{RI} 有值的部分是 $p \times (2p-1)$ 的矩陣，它們的值都與 j 無關。 H_{II} 和 G_{II} 是 $(2^j - 2p) \times (2^{j+1} - 2p)$ 的矩陣，雖然矩陣的大小由 j 決定，但是它們的每一列都有相同的形式。總而言之低頻濾波器 $\mathbf{H}$ 和高頻濾波器 $\mathbf{G}$ 滿足

$$(\frac{1}{\sqrt{2}}H)^t(\frac{1}{\sqrt{2}}H) + (\frac{1}{\sqrt{2}}G)^t(\frac{1}{\sqrt{2}}G) = I$$

2.4 區段上的離散凌波轉換及前濾波器

在 [1, 第 4.2 節] 中已經詳細地探討了週期化的離散凌波轉換，在此我們只討論以區段上凌波函數作爲離散凌波轉換的工具的情形，其中有幾個細節敘述於此:

1. 原本週期化的離散凌波轉換只需要一組低頻濾波器及一組高頻濾波器，而區段上的離散凌波轉換多了一組低頻濾波器及一組高頻濾波器，因此區段上的離散凌波轉換有較多的係數必須紀錄。
2. 前述的分解和合成步驟，利用到的矩陣 $\mathcal{L}$，並非紀錄一整個矩陣，因爲相對應於輸入的向量 $\boldsymbol{v}$ 長度是 N，則 $\mathcal{L}$ 是 $N \times N$ 的矩陣，$\mathcal{L}$ 是稀疏而有規則的，所以我們只紀錄 $\mathcal{L}$ 裡非零的部分，把相對應的 $\boldsymbol{v}$ 或 $(\boldsymbol{u}, \boldsymbol{w})^t$ 拿出來作乘法，以節省記憶空間。
3. 如前所述，作分解與合成時，都須乘上 $\frac{1}{\sqrt{2}}$，而又可寫成

$$(\frac{1}{\sqrt{2}}L)^t(\frac{1}{\sqrt{2}}L) = I$$
$$(\frac{1}{2}L)^tL = L^t(\frac{1}{2}L) = I$$

所以，我們選擇了在分解時除 2，以保持低頻數值的相對大小和原資料一致，把分解那組係數除 2，合成時不再除 $\frac{1}{\sqrt{2}}$，節省計算量。

接著考慮二維的情況，我們使用最常見的直積模式 (Cartesian product)，將雙變數的自格函數視爲

$$\varphi(x,y) = \varphi(x)\varphi(y)$$

則雙變數的凌波函數爲

$$\psi(x,y) = \psi(x)\psi(y)$$

也就是說，對於二維的資料把它視爲兩個獨立的一維資料的直積。假設輸入一個 $N \times N$ 的矩陣，作離散凌波轉換時，就把它視爲一組 $N \times 1$ 的行向量和 $1 \times N$ 的列向量，套用一維的模式，分別對每個行向量和列向量作離散凌波轉換，即可完成二維的離散凌波轉換。

在離散凌波轉換的處理過程中，如同 [7] 所指出， 區段化的離散凌波轉換比週期化的離散凌波轉換留下更大的量在高頻的部分。在此我們解釋 $p = 3$ 時的原因。就理論而言， $p = 3$ 時，階數 0 次, 1 次和 2 次的多項式是在 V_j 裡， 也就是一維時

$$1 = \sum_{k=0}^{2^j-1} \langle 1, \phi_{jk} \rangle \phi_{jk}$$
$$x = \sum_{k=0}^{2^j-1} \langle x, \phi_{jk} \rangle \phi_{jk}$$
$$x^2 = \sum_{k=0}^{2^j-1} \langle x^2, \phi_{jk} \rangle \phi_{jk}$$

其投影係數分別是

$$\langle 1, \phi_{jk} \rangle = 2^{\frac{-j}{2}}, \quad \langle x, \phi_{jk} \rangle = 2^{\frac{-3j}{2}}(k + M_1), \quad \langle x^2, \phi_{jk} \rangle = 2^{\frac{-5j}{2}}(k^2 + 2kM_1 + M_2)$$

所以對應在 V_j 裡的多項式 $1, x, x^2$ 分別是

$$x_0 = (\cdots, 1, 1, 1, \cdots)^t, \quad x_1 = (\cdots, 1, 2, 3, \cdots)^t, \quad x_2 = (\cdots, 1, 4, 9, \cdots)^t$$

而在區段上的凌波函數, 雖然仍然保持

$$1 = \sum_{k=-2}^{2^j-3} \langle 1, \varphi_{jk} \rangle \varphi_{jk}$$
$$x = \sum_{k=-2}^{2^j-3} \langle x, \varphi_{jk} \rangle \varphi_{jk}$$
$$x^2 = \sum_{k=-2}^{2^j-3} \langle x^2, \varphi_{jk} \rangle \varphi_{jk}$$

但其投影係數

$$\begin{pmatrix} \int \varphi_{j,-2} \\ \int \varphi_{j,-1} \\ \int \varphi_{j,0} \end{pmatrix} = 2^{\frac{-j}{2}} \cdot S \cdot \tilde{\mathbf{X}} \cdot \begin{pmatrix} K_4^0 \\ K_3^0 \\ K_2^0 \\ K_1^0 \\ K_0^0 \end{pmatrix}, \quad \text{其中 } K_n^m = \int_0^\infty x^m \phi(x+n)\, dx$$

不再是如此的整齊，使其對應在 $V_j[0,1]$ 裡的多項式 $1, x, x^2$ 不再是類似函數點值的 x_0, x_1 和 x_2，所以當輸入 x_0, x_1 和 x_2，區段上的高頻濾波器不再完美的消滅這三個量，而在高頻部分留下不可忽視的量。因此我們導入前濾波器的觀念，將多項式先轉換成在區段上凌波函數應有的形式，才作區段凌波轉換，以降低留在 $W_j[0,1]$ 的量。而前濾波器的製作如下:

定義

$$F_L: \quad \text{span}\left\{ \begin{pmatrix} \int \phi_{j,-2} \\ \int \phi_{j,-1} \\ \int \phi_{j,0} \end{pmatrix}, \begin{pmatrix} \int x\phi_{j,-2} \\ \int x\phi_{j,-1} \\ \int x\phi_{j,0} \end{pmatrix}, \begin{pmatrix} \int x^2\phi_{j,-2} \\ \int x^2\phi_{j,-1} \\ \int x^2\phi_{j,0} \end{pmatrix} \right\}$$

$$\mapsto \text{span}\left\{ \begin{pmatrix} \int \varphi_{j,-2} \\ \int \varphi_{j,-1} \\ \int \varphi_{j,0} \end{pmatrix}, \begin{pmatrix} \int x\varphi_{j,-2} \\ \int x\varphi_{j,-1} \\ \int x\varphi_{j,0} \end{pmatrix}, \begin{pmatrix} \int x^2\varphi_{j,-2} \\ \int x^2\varphi_{j,-1} \\ \int x^2\varphi_{j,0} \end{pmatrix} \right\}$$

$$F_R: \quad \text{span}\left\{ \begin{pmatrix} \int \phi_{j,2^j-5} \\ \int \phi_{j,2^j-4} \\ \int \phi_{j,2^j-3} \end{pmatrix}, \begin{pmatrix} \int x\phi_{j,2^j-5} \\ \int x\phi_{j,2^j-4} \\ \int x\phi_{j,2^j-3} \end{pmatrix}, \begin{pmatrix} \int x^2\phi_{j,2^j-5} \\ \int x^2\phi_{j,2^j-4} \\ \int x^2\phi_{j,2^j-3} \end{pmatrix} \right\}$$

$$\mapsto \text{span}\left\{ \begin{pmatrix} \int \varphi_{j,2^j-5} \\ \int \varphi_{j,2^j-4} \\ \int \varphi_{j,2^j-3} \end{pmatrix}, \begin{pmatrix} \int x\varphi_{j,2^j-5} \\ \int x\varphi_{j,2^j-4} \\ \int x\varphi_{j,2^j-3} \end{pmatrix}, \begin{pmatrix} \int x^2\varphi_{j,2^j-5} \\ \int x^2\varphi_{j,2^j-4} \\ \int x^2\varphi_{j,2^j-3} \end{pmatrix} \right\}$$

也就是說

$$F_L : 2^{\frac{-j}{2}} \begin{pmatrix} 1 & 2^{-j}(M_1-2) & 2^{-2j}(M_2-4M_1+4) \\ 1 & 2^{-j}(M_1-1) & 2^{-2j}(M_2-2M_1+1) \\ 1 & 2^{-j}M_1 & 2^{-2j}M_2 \end{pmatrix}$$

$$\mapsto \begin{pmatrix} \int \varphi_{j,-2} & \int x\varphi_{j,-2} & \int x^2\varphi_{j,-2} \\ \int \varphi_{j,-1} & \int x\varphi_{j,-1} & \int x^2\varphi_{j,-1} \\ \int \varphi_{j,0} & \int x\varphi_{j,0} & \int x^2\varphi_{j,0} \end{pmatrix}$$

$$F_R : 2^{\frac{-j}{2}} \begin{pmatrix} 1 & 2^{-j}(M_1+2^j-5) & 2^{-2j}(M_2-10M_1+25+2^{j+1}(M_1-5)+2^{2j}) \\ 1 & 2^{-j}(M_1+2^j-4) & 2^{-2j}(M_2-8M_1+16+2^{j+1}(M_1-4)+2^{2j}) \\ 1 & 2^{-j}(M_1+2^j-3) & 2^{-2j}(M_2-6M_1+9+2^{j+1}(M_1-3)+2^{2j}) \end{pmatrix}$$

$$\mapsto \begin{pmatrix} \int \varphi_{r_1} & \int x\varphi_{r_1} & \int x^2\varphi_{r_1} \\ \int \varphi_{r_2} & \int x\varphi_{r_2} & \int x^2\varphi_{r_2} \\ \int \varphi_{r_3} & \int x\varphi_{r_3} & \int x^2\varphi_{r_3} \end{pmatrix}$$

經過整理

$$F_L : I \mapsto S \cdot \tilde{\mathbf{X}} \cdot \begin{pmatrix} K_4^0 & 2^{-j}K_4^1 & 2^{-2j}K_4^2 \\ K_3^0 & 2^{-j}K_3^1 & 2^{-2j}K_3^2 \\ K_2^0 & 2^{-j}K_2^1 & 2^{-2j}K_2^2 \\ K_1^0 & 2^{-j}K_1^1 & 2^{-2j}K_1^2 \\ K_0^0 & 2^{-j}K_0^1 & 2^{-2j}K_0^2 \end{pmatrix} \cdot$$

$$\begin{pmatrix} 1 & -M_1 & -M_2 \\ 0 & 2^j & 2^j \\ 0 & 0 & 2^{2j} \end{pmatrix} \cdot \begin{pmatrix} -\frac{1}{2}M_1 & M_1 & 1-\frac{1}{2}M_1 \\ -M_1 & 2M_1-1 & 1-M_1 \\ \frac{1}{2} & -1 & \frac{1}{2} \end{pmatrix}$$

$$F_R : I \mapsto T \cdot \tilde{\mathbf{Y}} \cdot \mathrm{PreFR} \cdot \begin{pmatrix} 1 & -(M_1+2^j-3 & 2^{2j}-(M_2-6M_1+9) \\ 0 & 2^j & -2^{2j+1} \\ 0 & 0 & 2^{2j} \end{pmatrix} \cdot$$

$$\begin{pmatrix} 1 & 0 & 2^{2j} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 0 & 1 \\ \frac{7}{2}-M_1 & 2M_1-8 & \frac{9}{2}-M_1 \\ \frac{1}{2} & -1 & \frac{1}{2} \end{pmatrix}$$

其中

$$K_n^m = \int_0^\infty x^m \phi(x+n)\,dx,\ 0 \le m \le 2\ (= p-1),\ 1 \le n \le 4\ (= 2p-2)$$

$$M_0 = 1,\ M_1 = \int x\phi(x)\,dx,\ M_2 = \int x^2\phi(x)\,dx$$

$$\tilde{\mathbf{X}} = \begin{pmatrix} x_{-4}^0 & x_{-3}^0 & x_{-2}^0 & x_{-1}^0 & x_0^0 \\ x_{-4}^1 & x_{-3}^1 & x_{-2}^1 & x_{-1}^1 & x_0^1 \\ x_{-4}^2 & x_{-3}^2 & x_{-2}^2 & x_{-1}^2 & x_0^2 \end{pmatrix}$$

$$\tilde{\mathbf{Y}} = \begin{pmatrix} y_{2^j-5}^0 & y_{2^j-4}^0 & y_{2^j-3}^0 & y_{2^j-2}^0 & y_{2^j-1}^0 \\ y_{2^j-5}^1 & y_{2^j-4}^1 & y_{2^j-3}^1 & y_{2^j-2}^1 & y_{2^j-1}^1 \\ y_{2^j-5}^2 & y_{2^j-4}^2 & y_{2^j-3}^2 & y_{2^j-2}^2 & y_{2^j-1}^2 \end{pmatrix}$$

我們以行向量的形式說明 PreFR。

PreFR 的第一行是

$$(1 \quad 1-K_4^0 \quad 1-K_3^0 \quad 1-K_2^0 \quad 1-K_1^0)^t$$

第二行是

$$\begin{pmatrix} 2^{-j}(M_1-5+2^j) \\ 2^{-j}(M_1-4-K_4^1+2^j(1-K_4^0)) \\ 2^{-j}(M_1-3-K_3^1+2^j(1-K_3^0)) \\ 2^{-j}(M_1-2-K_2^1+2^j(1-K_2^0)) \\ 2^{-j}(M_1-1-K_1^1+2^j(1-K_1^0)) \end{pmatrix}$$

而最後一行是

$$\begin{pmatrix} 2^{-2j}(M_2-10M_1+25+2^{j+1}(M_1-5)+2^{2j}) \\ 2^{-2j}[M_2-8M_1+16-K_4^2+2^{j+1}(M_1-4-K_4^1)+2^{2j}(1-K_4^0)] \\ 2^{-2j}[M_2-6M_1+9-K_3^2+2^{j+1}(M_1-3-K_3^1)+2^{2j}(1-K_3^0)] \\ 2^{-2j}[M_2-4M_1+4-K_2^2+2^{j+1}(M_1-2-K_2^1)+2^{2j}(1-K_2^0)] \\ 2^{-2j}[M_2-2M_1+1-K_1^2+2^{j+1}(M_1-1-K_1^1)+2^{2j}(1-K_1^0)] \end{pmatrix}$$

則這樣造出來的 F_L，F_R 都是與 j 無關的。因此，我們有了 $3\times 3(=p\times p)$ 的前濾波器 F_L, F_R。

我們從 Daubechies 的緊緻函蓋而且正則的自格函數出發，利用多項式投影的觀念而造出同樣是緊緻函蓋並且正則的在區段上的自格和凌波函數，也建造了相對應的低頻濾波器、高頻濾波器以及前濾波器 (pequalsthree.nb)。此外，我們依照鄭儘的論文重新計算階數爲 2 的濾波器及前濾波器 (pequalstwo.nb)。這些數值是由 Mathematica 計算出來的。雖然在每一個計算步驟都計算至 50 位準確位置，實際上並沒有達到這麼高的精確度。不過我們的目的只是將這些數值匯入 Matlab 以供離散凌波轉換使用，相對於 Matlab 的精確度 (2^{-52}) 而言，這些數值是非常精確的。

對於一維的資料，假設輸入

$$\boldsymbol{v} = (v_1, \cdots, v_n)^t$$

先把它經過前濾波器的處理，轉換成

$$\tilde{v} = (\tilde{v}_1, \tilde{v}_2, \tilde{v}_3, v_4, \cdots, v_{n-3}, \tilde{v}_{n-2}, \tilde{v}_{n-1}, \tilde{v}_n)^t$$

其中

$$\begin{pmatrix}\tilde{v}_1\\ \tilde{v}_2\\ \tilde{v}_3\end{pmatrix} = F_L \begin{pmatrix} v_1\\ v_2\\ v_3\end{pmatrix} \quad \begin{pmatrix}\tilde{v}_{n-2}\\ \tilde{v}_{n-1}\\ \tilde{v}_n\end{pmatrix} = F_R \begin{pmatrix} v_{n-2}\\ v_{n-1}\\ v_n\end{pmatrix}$$

我們把 $\tilde{v}$ 送去作分解，重建，再作反轉換，亦即把 $\tilde{v}$ 的頭尾分別乘上 F_L^{-1}, F_R^{-1}。則這樣經過前濾波器的處理，就可以使留在 $W_j[0,1]$ 的量減少。

接下來我們討論二維的情形。因爲我們採用直積模式，假設輸入一個 $n \times n$ 的矩陣 A， $A = (a_{ij})$，經過前濾波器的處理，把 A 先轉換成 $\tilde{A}$。

而 $\tilde{A}$ 的型式爲

$$\tilde{A} = (\tilde{a}_{ij})$$

則

$$\begin{pmatrix}\tilde{a}_{11} & \tilde{a}_{12} & \tilde{a}_{13}\\ \tilde{a}_{21} & \tilde{a}_{22} & \tilde{a}_{23}\\ \tilde{a}_{31} & \tilde{a}_{32} & \tilde{a}_{33}\end{pmatrix} = F_L \cdot \begin{pmatrix} a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{13}\\ a_{31} & a_{32} & a_{33}\end{pmatrix}^t \cdot F_L^t$$

$$\begin{pmatrix}\tilde{a}_{14} & \tilde{a}_{15} & \tilde{a}_{16} & \cdots & \tilde{a}_{1,n-3}\\ \tilde{a}_{24} & \tilde{a}_{25} & \tilde{a}_{26} & \cdots & \tilde{a}_{2,n-3}\\ \tilde{a}_{34} & \tilde{a}_{35} & \tilde{a}_{36} & \cdots & \tilde{a}_{3,n-3}\end{pmatrix} = F_L \begin{pmatrix} a_{14} & a_{15} & a_{16} & \cdots & a_{1,n-3}\\ a_{24} & a_{25} & a_{26} & \cdots & a_{2,n-3}\\ a_{34} & a_{35} & a_{36} & \cdots & a_{3,n-3}\end{pmatrix}$$

$$\begin{pmatrix}\tilde{a}_{1,n-2} & \tilde{a}_{1,n-1} & \tilde{a}_{1,n}\\ \tilde{a}_{2,n-2} & \tilde{a}_{2,n-1} & \tilde{a}_{2,n}\\ \tilde{a}_{3,n-2} & \tilde{a}_{3,n-2} & \tilde{a}_{3,n}\end{pmatrix} = F_R \cdot \begin{pmatrix} a_{1,n-2} & a_{1,n-1} & a_{1,n}\\ a_{2,n-2} & a_{2,n-1} & a_{2,n}\\ a_{3,n-2} & a_{3,n-2} & a_{3,n}\end{pmatrix}^t \cdot F_L^t$$

$$\begin{pmatrix}\tilde{a}_{n-2,1} & \tilde{a}_{n-2,2} & \tilde{a}_{n-2,3}\\ \tilde{a}_{n-1,1} & \tilde{a}_{n-1,2} & \tilde{a}_{n-1,3}\\ \tilde{a}_{n,1} & \tilde{a}_{n,2} & \tilde{a}_{n,3}\end{pmatrix} = F_L \cdot \begin{pmatrix} a_{n-2,1} & a_{n-2,2} & a_{n-2,3}\\ a_{n-1,1} & a_{n-1,2} & a_{n-1,3}\\ a_{n,1} & a_{n,2} & a_{n,3}\end{pmatrix}^t \cdot F_R^t$$

$$\begin{pmatrix}\tilde{a}_{n-2,4} & \tilde{a}_{n-2,5} & \cdots & \tilde{a}_{n-2,n-3}\\ \tilde{a}_{n-1,4} & \tilde{a}_{n-1,5} & \cdots & \tilde{a}_{n-1,n-3}\\ \tilde{a}_{n,4} & \tilde{a}_{n,5} & \cdots & \tilde{a}_{n,n-3}\end{pmatrix} = F_R \begin{pmatrix} a_{n-2,4} & a_{n-2,5} & \cdots & a_{n-2,n-3}\\ a_{n-1,4} & a_{n-1,5} & \cdots & a_{n-1,n-3}\\ a_{n,4} & a_{n,5} & \cdots & a_{n,n-3}\end{pmatrix}$$

$$\begin{pmatrix}\tilde{a}_{41} & \tilde{a}_{42} & \tilde{a}_{43}\\ \tilde{a}_{51} & \tilde{a}_{52} & \tilde{a}_{53}\\ \vdots & \vdots & \vdots\\ \tilde{a}_{n-3,1} & \tilde{a}_{n-3,2} & \tilde{a}_{n-3,3}\end{pmatrix} = F_L \cdot \begin{pmatrix} a_{41} & a_{42} & a_{43}\\ a_{51} & a_{52} & a_{53}\\ \vdots & \vdots & \vdots\\ a_{n-3,1} & a_{n-3,2} & a_{n-3,3}\end{pmatrix}$$

$$\begin{pmatrix} \tilde{a}_{4,n-2} & \tilde{a}_{4,n-1} & \tilde{a}_{4,n} \\ \tilde{a}_{5,n-2} & \tilde{a}_{5,n-1} & \tilde{a}_{5,n} \\ \vdots & \vdots & \vdots \\ \tilde{a}_{n-3,n-2} & \tilde{a}_{n-3,n-1} & \tilde{a}_{n-3,n} \end{pmatrix} = F_R \cdot \begin{pmatrix} a_{4,n-2} & a_{4,n-1} & a_{4,n} \\ a_{5,n-2} & a_{5,n-1} & a_{5,n} \\ \vdots & \vdots & \vdots \\ a_{n-3,n-2} & a_{n-3,n-1} & a_{n-3,n} \end{pmatrix}$$

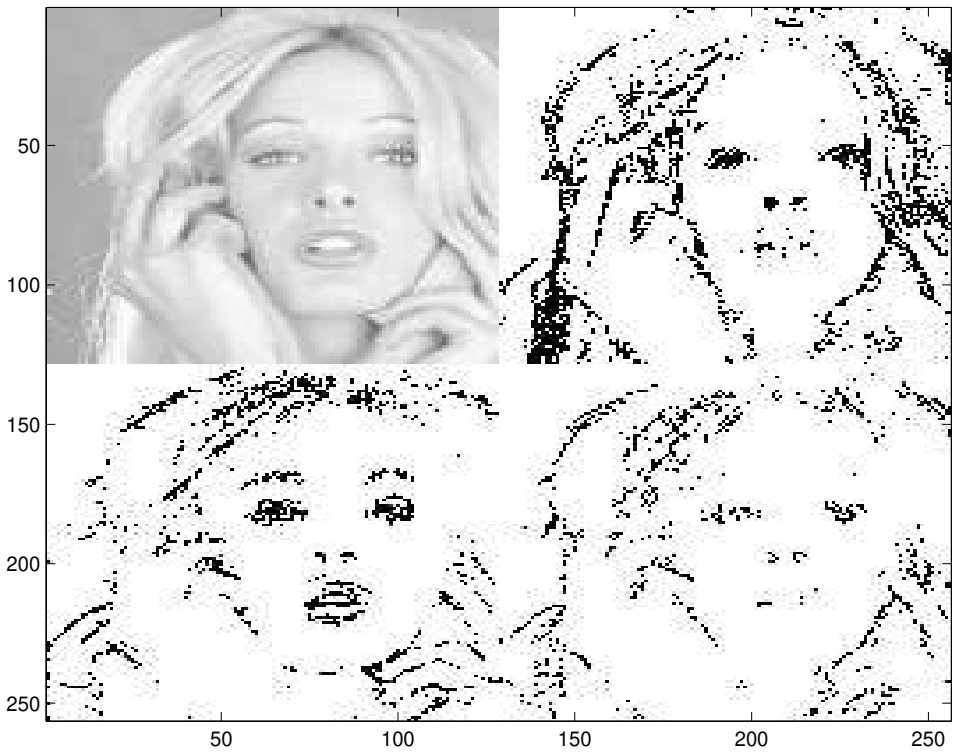
$$\begin{pmatrix} \tilde{a}_{n-2,n-2} & \tilde{a}_{n-2,n-1} & \tilde{a}_{n-2,n} \\ \tilde{a}_{n-1,n-2} & \tilde{a}_{n-1,n-1} & \tilde{a}_{n-1,n} \\ \tilde{a}_{n,n-2} & \tilde{a}_{n,n-1} & \tilde{a}_{n,n} \end{pmatrix} = F_R \cdot \begin{pmatrix} a_{n-2,n-2} & a_{n-2,n-1} & a_{n-2,n} \\ a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} \\ a_{n,n-2} & a_{n,n-1} & a_{n,n} \end{pmatrix}^t \cdot F_R^t$$

在此我們看到四個角落的 3×3 矩陣都需要乘上兩次的前濾波器，這是因爲這四個 3×3 的矩陣同時是行或列的最前或最後的部分。這樣先經過前濾波器的處理，再作轉換就可以使留在高頻部分的值降低。

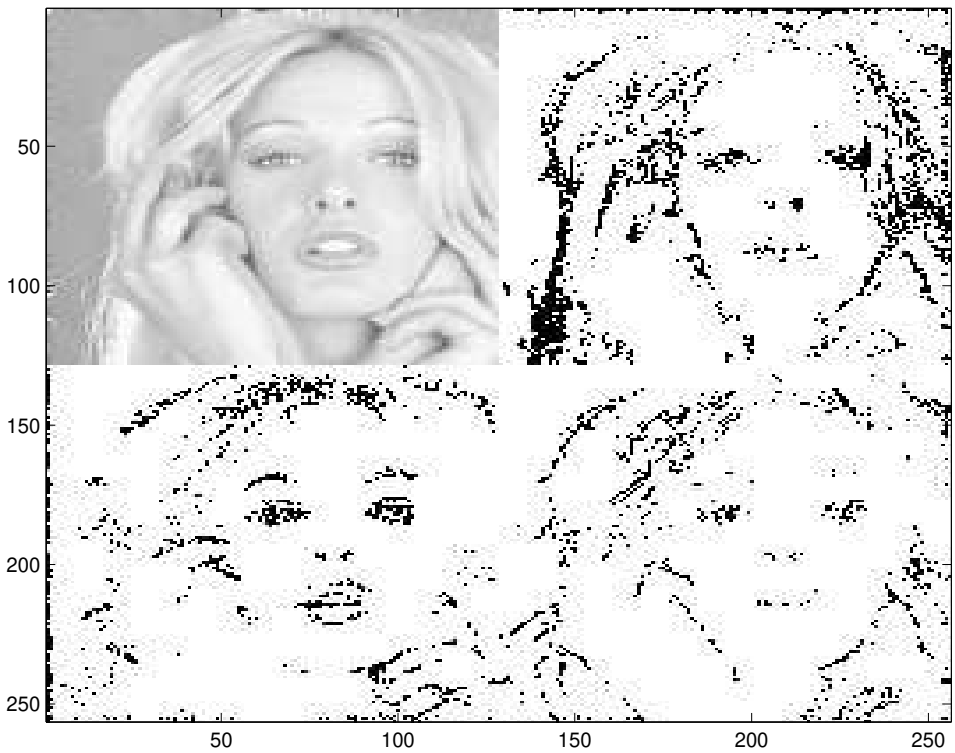
我們以一幅 256×256 的灰階影像 Tiffany 爲例（圖九）來展示區段上的離散凌波轉換的情形。

圖九

我們假設它是落在 V_8 的投影係數。經過一次的離散凌波轉換，把 V_J 分解成 V_{J-1} 和 W_{J-1}，再把 W_{J-1} 中絕對值大於 3 的數值，以黑色點表現出來，形成類似描邊的效果。圖十是 $p = 2$ 所做的區段上的離散凌波轉換，而圖十一是 $p = 3$ 的情形。



圖十



圖十一

我們可以看到不論是 p＝2 是 p＝3 都可以消除邊框效應。不過在 p＝3 的時候在原

本發生邊框效應的另外兩邊有長條的黑點。這是由於 Daubechies 的正交凌波函數是不對稱的，因此會產生偏移的情形。我們會在第三章再詳細討論這對我們會有怎樣的影響。

我們在下列這個網頁

http://www.math.ncu.edu.tw/ zhenggz/download.html

提供本章所有的資料檔以及程式。

第三章 增加解析度的凌波演算法

一般而言，凌波演算法都依賴在兩個基石上。一個就是在第一章就介紹過的多層解析空間觀念，這是凌波理論的理論架構。而另外一個則是幾乎與所有的應用都有直接的關係的凌波係數消散定理。相對於傳統的插值方法，我們在這一章提出根據多層解析空間的概念所發展的凌波演算法。我們分別計算了階數爲 2 及階數爲 3 的凌波演算法，以比較效果與計算複雜度之間的關係。此外，在第一章中所介紹的三階樣條函數包含了三階多項式，而它的函蓋只有一個步距 (step size)。根據我們在第一章的實驗，這是一個相當好的方法。而階數爲 3 的正交凌波函數包含了二階多項式，對應的離散凌波轉換的函蓋則有三個步距。由於前述凌波理論的兩個概念，雖然包含的多項式的階數較少，但是凌波演算法仍然有相當好的效果。

3.1 增加解析度的凌波演算法

根據凌波轉換

$$V_j = V_{j-1} \oplus W_{j-1}$$

我們知道 V_{j-1} 是較低解析度的解析空間， V_j 是較高解析度的解析空間，而 W_{j-1} 則是 V_{j-1} 和 V_j 之間的差異。當 j 夠大時，我們可以相信 j 也有夠高的解析度，因此 V_{j-1} 和 V_j 的差異應該不大。凌波係數消散定理 [1, 第 182 頁] 證實了上述的猜想，並且提供定量的關係。

凌波係數消散定理

令 $\psi(x)$ 爲一凌波函數，其正交於階數 $\le n$ 之多項式 $\int x^m \psi(x)\,dx = 0, \quad \text{for } 0 \le m \le n$。令 $\psi_{jk} = \sqrt{2^j}\psi(2^j x - k)$。若 $f(x) \in C^{m,\sigma}(\text{supp}\ \psi_{j,k})$ 則 $|\langle f(x), \psi_{jk}(x)\rangle| = O(2^{-(n+\sigma+\frac{1}{2})j})$, 其中 $n = \min(m, n)$

因爲它們最後的消散速率是

$$|\langle f(x), \psi_{jk}(x)\rangle| = O(2^{-n+\sigma+\frac{1}{2})j})$$

而這個 n 可以表示成

$$n = \min(f(x)\text{是}m\text{次可微}, \psi(x)\text{正交於}n\text{次多項式})$$

因此，我們要注意定理的假設中，『凌波函數 $\psi(x)$ 正交於 n 階多項式』以及『函數 $f(x)$ 是 n 次可微』這兩個條件是相輔相成的。

以 Daubechies 的 p 階正交凌波函數而言，由於它是有限含蓋 $[0, 2p-1]$，決定一組自格係數 $\{c_k\}$ 一共需要 $2p$ 個條件。由自格函數 $\{\phi(x-k)\}$ 是一個正則集合，可以得到

$$\sum_k c_k c_{k+2m} = 2\delta_{0m} \quad \forall m \in \mathbb{Z}$$

這給了我們 p 個條件。此外，將自格等式左右兩邊都積分，得到

$$\begin{aligned}\int \phi(x)\,dx &= \sum_{k=0}^{N} c_k \int \phi(2x-k)\,dx \\ &= \frac{1}{2}\sum_{k=0}^{N} c_k \int \phi(x)\,dx\end{aligned}$$

由於 $\int \phi\,dx = 1$，因此

$$\sum_k c_k = 2$$

這提供了 1 個條件。因此我們還剩下 $p-1$ 個自由度。原本的凌波函數 $\psi(x)$ 已經正交於零階多項式

$$\langle 1, \psi \rangle = \int \psi\,dx = 0$$

根據凌波係數消散定理，Daubechies 的 p 階正交凌波函數用這些 $p-1$ 個自由度製造

$$\int x^m \psi(x)\,dx = 0 \quad \text{for } 0 \le m \le p-1$$

也就是

Daubechies 的 p 階正交凌波函數正交於 $p-1$ 階的多項式

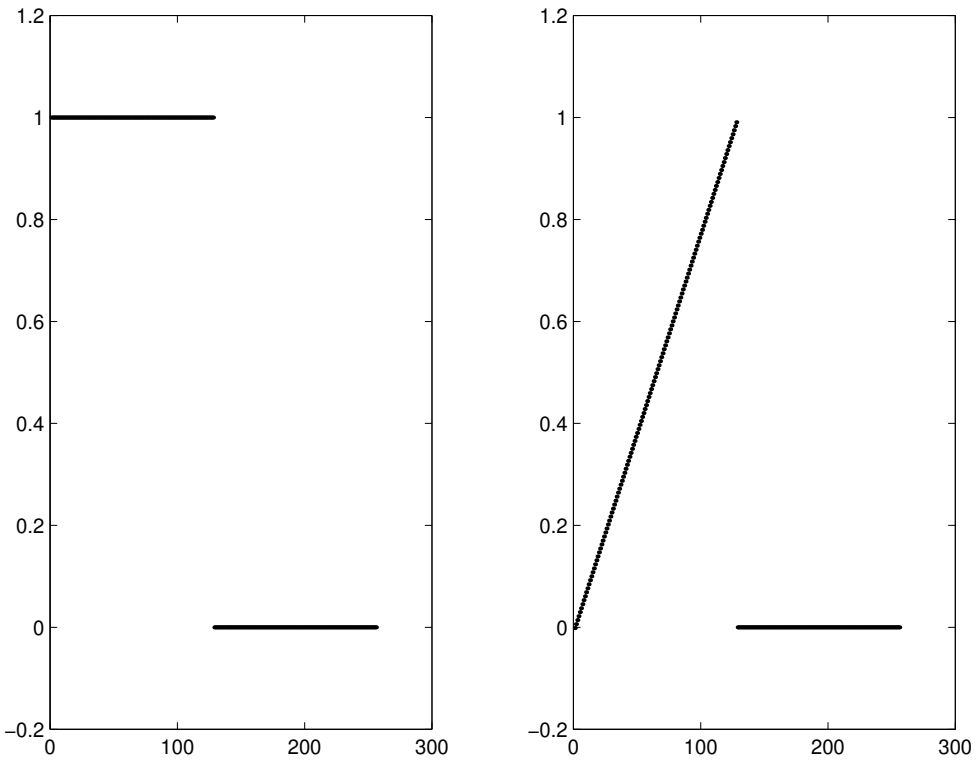
以 $p=2$ 的 Daubechies 正交凌波函數爲例。由於階數爲 2 的凌波函數 $\psi(x)$ 正交於常數函數以及一階多項式，因此高頻濾波係數也會正交於常數訊號以及線性訊號。而所謂係數的正交性也是經由離散凌波轉換的分解過程後，落在 W_j 中的分量全是

零。舉例來說，輸入一個 1×256 的序列，其元素全是 1，使用前述的階數爲 2 的凌波轉換做一次分解過程（圖十二左）。

```
y = jdwt(2,ones(1,256),1);
```

再輸入一個 1×256 的線性序列做一次做一次分解過程（圖十二右）。

```
yy = jdwt(2,ones(1,256),1);
```



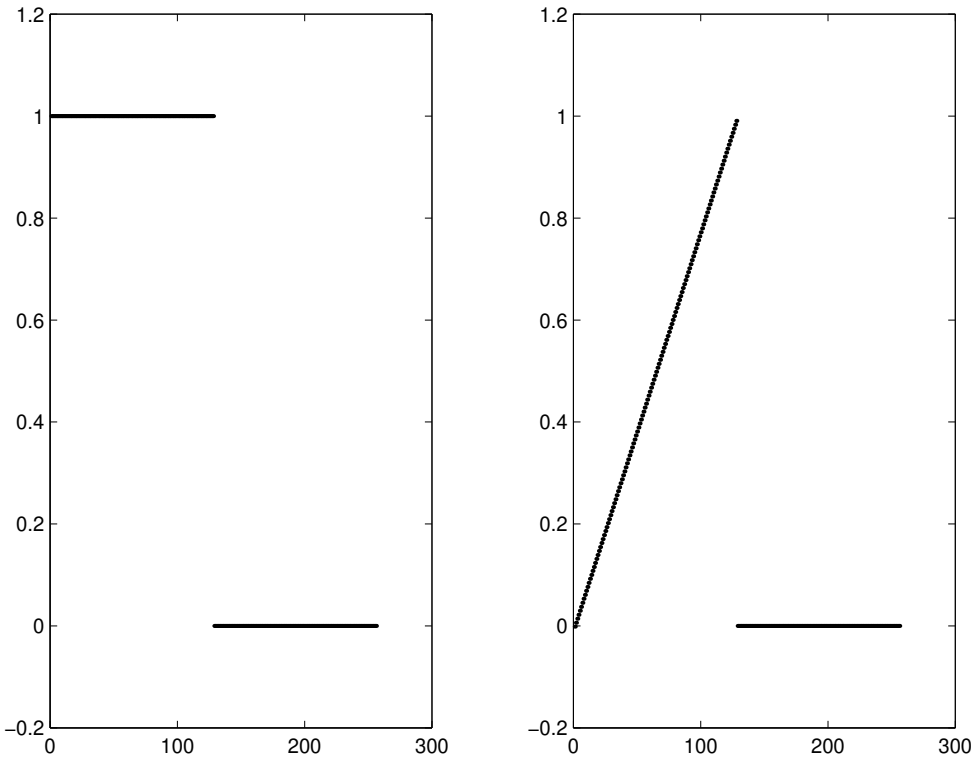
圖十二

圖中前一半（前 128 個點）是在 V_7 中的係數，後一半（後 128 個點）是在 W_7 中的係數。由於我們所使用的是區段上的離散凌波轉換，在 W_7 中的係數全是零。順便一提，如果我們使用週期化的離散凌波轉換，在 W_7 中的係數會有一小部分出現較大的值。由於階數爲 2 的離散凌波轉換最多可以保持一階多項式。因此對二階多項式做同樣的步驟，在高頻會出現非零的值。不過隨著解析度的提高，也就是 j 越大，這些值會越小。前述的作法是用在資料壓縮，也就是減少資訊量。我們假設這個想法也可用在增加解析度的問題。

以一個一維數位訊號 $\boldsymbol{v}$ 爲例。若 $\boldsymbol{v} \in V_J$，我們設定原本不存在的高頻資訊

$$\mathbf{w} = (0, 0, \cdots, 0) \in W_J$$

其中 $\boldsymbol{w}$ 是維度爲 2^J 的向量。合併 $\boldsymbol{v}$ 和 $\boldsymbol{w}$，再使用離散凌波轉換做一次合成步驟。我們分別對常數函數，一階多項式及二階多項式做測試。此外，我們使用 $\mathrm{p}=2$ 的離散凌波轉換（圖十三上）及 $\mathrm{p}=3$ 的離散凌波轉換做比較（圖十三下）。



圖十三

從圖形看來，函數的樣子被保持住了。我們看數值測試的結果。假設由原函數取樣所得到的向量是 $\mathbf{f}$，而由我們的方法得到的向量是 $\boldsymbol{u}$，則

```
norm(f−u)
```

得到

	常數函數	一階多項式	二階多項式
$\mathrm{p}=2$	$3.582082587e-015$	$2.801798351e-002$	$3.237127066e-002$
$\mathrm{p}=3$	$8.399627802e-015$	$8.069804530e-003$	$9.334349993e-003$

雖然階數爲 3 的離散凌波轉換比階數爲 2 的離散凌波轉換多包含二階多項式。但是根據實驗結果，階數爲 3 的時候並沒有比使用 $\mathrm{p}=2$ 的離散凌波轉換好多少。我們曾在第二章最後提到使用 Daubechies 的正交凌波函數會有偏移的現象。因此，階數爲 3 的離散凌波轉換在包含多項式的優勢被這個因素所消耗了。不過這個實驗告訴我們，這個想法基本上是可行的。

3.2 數值實驗

我們在前一節提出了以『多層解析空間』爲架構，應用『凌波係數消散定理』以及『Daubechies 的 p 階正交凌波函數正交於 (p-1) 階多項式』的凌波演算法。

以前面使用過的 Tiffany 圖像爲例。我們特別以 Tiffany 的眼睛部分做增加解析度的處理。

我們分別以 'spline' 方法，階數爲 2 的離散凌波轉換以及階數爲 3 的離散凌波轉換做測試。

由於 'spline' 方法包含到三階多項式，而階數爲 2 及階數爲 3 的離散凌波轉換分別只有包含到一階和二階多項式。。我們看計算複雜度

	spline	p = 2 wavelets	p = 3 wavelets
flops	515354	278569	434654

可以看到不論是階數爲 2 或是階數爲 3 的凌波演算法，所使用的 flops 都比三階樣條函數來的少。

有一點要說明的是，這裡只測試一張圖像的計算複雜度。這是因爲本文一開始提到，由於解析度不同的緣故，從螢幕看一張圖像與觀看列印出來的圖像是不一樣的。爲了避免這項因素的影響，我們在下列網頁

http://www.math.ncu.edu.tw/ zhenggz/test.html

提供較多的測試與比較

3.3 結論

從多項式插值的誤差分析可以知道

1. 使用插值方法的函蓋
2. 使用插值方法所包含的多項式階數
3. 原函數 f(x) 的平滑程度

這個結果對我們所提出的凌波演算法仍然適用。本文以插值的觀點比較了分片多項式插值，三階樣條函數以及階數爲 2 和階數爲 3 的離散凌波轉換。我們也藉著實驗闡述了函蓋與所包含的多項式階數的關係。一般而言，函蓋越大計算複雜度就越高。而方法所包含的多項式階數越高，所得的效果越好。此外，原本誤差分析的結果還包含原

函數的平滑程度，本文並沒有針對這點做處理。若不論我們所處理的圖像的特性，而純就插值的方法而言，處理插值問題的核心課題即是如何在效果與計算複雜度之間取得平衡。以數學的觀點而言

> 考慮插值方法必須在方法本身所包含的多項式與它的函蓋大小之間作取捨 *(trade-off)*。

我們在前面曾提到過，就圖像而言我們不可能知道它的函數。我們所知道的就是它的點值。我們這裡所發展的凌波演算法只是雛形，它揭示了凌波理論應用在增加解析度的問題的確是可行的。

參考書目

[1] 單維彰，凌波初步，全華科技圖書公司，1999。

[2] D. Kincaid and W. Cheney, *Numerical Analysis*, Second ed., Brooks/Cole, 1996.

[3] 鄭儘，區段上的凌波函數，中央大學數學研究所碩士論文，1995。

[4] Ingrid Daubechies, *Ten lectures on Wavelets*, SIAM, 1992.

[5] Stéphane Mallat, *a Wavelet Tour of Signal Processing*, Second ed., Academic Press, 1999.

[6] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1995.