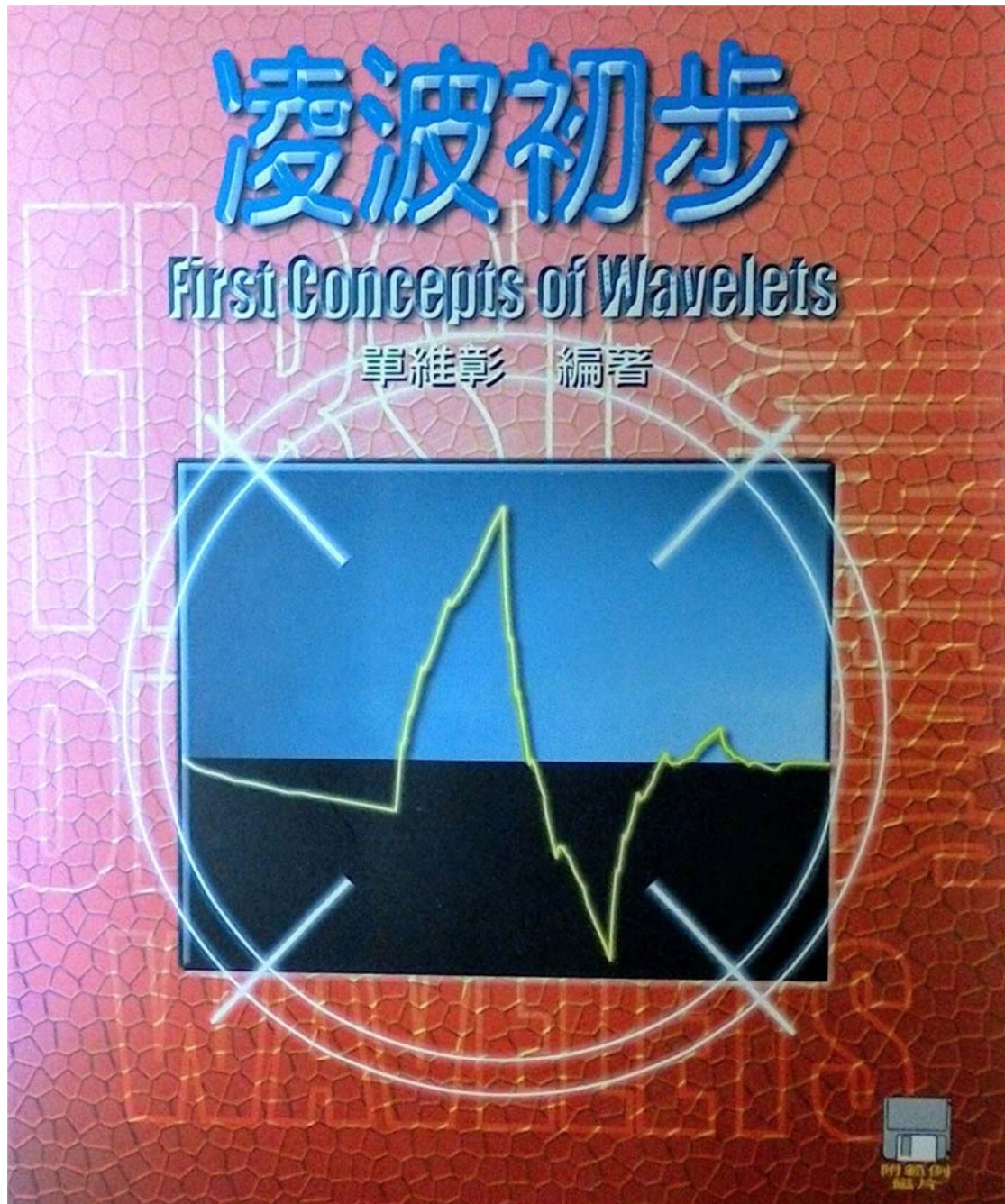


凌波初步

First Concepts of Wavelets

單維彰

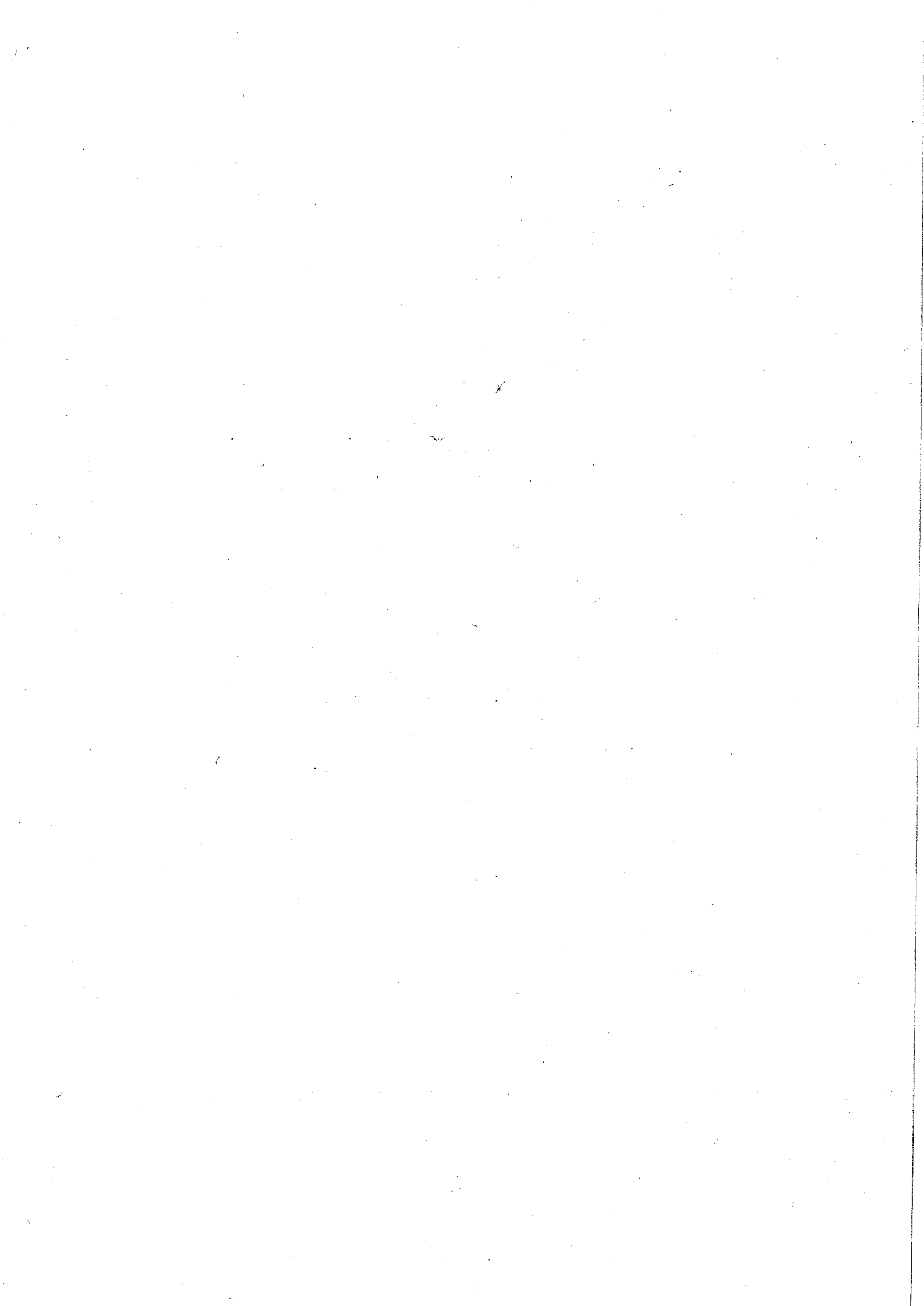


獻給

許進超教授

他在最關鍵的時刻

指導了我



前言

每一本書有它的使命。

這本書所企圖的使命有三：

- (1) 提倡以凌波當做 *wavelet* 的中文譯名。詳見第一章十三節。
- (2) 綜合整理入門的數學預備知識、凌波函數理論與應用方法之實作示範，並盡量降低數學的複雜度，希望能幫助非數學背景的學生或專業工作者，進入凌波函數論的創作與應用領域。
- (3) 將凌波函數的基本理論與應用範例，導引至大學和工專學生能力可及的範圍。使得這一門在將來可能會和傅立葉分析一樣被廣泛應用的數學科目，能夠及早成為大學部的選修課程。

基於上述之前兩項使命，這本書必須以中文寫作，而且假設講述的對象，僅僅具備普通的微積分和線性代數之數學基礎。這是在製作這本書的時候，所面臨的最大挑戰。在中文方面，因為長期以英文作為專業領域內之讀寫語言，使得我的中文語句非常不純正——尤其在談數學的時候。所以，在構句之時，但求詞意明確而不敢參言優美。在數學方面，我採取直覺、類比與描述動機的方式，帶領讀者以較快的步伐進入足夠抽象化的領域，而遺漏許多與主題不太相關的細節。我們只求讀者對於這些數學概念感到不陌生、會使用，而不能傳授一套完整的數學理論。如果您發現了中文不通順或者數學不正確之處，請直接給我建議。

在內容上，第一章利用很簡單的方盒函數和 Haar 函數，提示整個凌波理論的發展動機、理論要點和應用方式。第二章先複習線性代數與微積分中，跟本書內容比較相關的課題；然後分幾個單元介紹較深

的數學課題。第三章詳細而且完整地呈現正交凌波函數的基本理論。雖然我們沒有論及其他族群的凌波函數，但正交理論是各種凌波理論的共同基礎。第四、五章分別示範凌波函數在訊號處理與數值計算方面的應用。整本書的演算法都以 Matlab 當做實作平臺。在附錄中我們簡介 Matlab 的相關指令。

在結構上，第一章是低解析度的粗略概論，第三、四、五章則分三個主題，做高一層解析度的細述。第二章補足兩層解析度之間的差距。更高解析度的內容，就不在這本書裡了。這種「多層次結構」的安排，正反映了凌波函數的精神，是許進超教授提示給我的。

書中的某些文字或圖片，取自鄭儘女士或嚴健彰、徐敏堂、曾正男、陳聲偉、蘇芻穎五位先生的碩士論文。書中更有許多觀點，得自於和陳宜良教授、以及與前述六位研究生長期之研討結果。這些材料融於一爐很難一一指出，故在此一併聲明並致謝。

感謝嚴健彰、邱建明、吳舒婷、郭嘉慧，幫我校閱初稿、搜尋資料、製作美工圖片與函數圖形。在初版之後，還要感謝李宗元教授和鄭光哲先生的熱心指正。他們大量降低了這本書的失誤。但是所有漏網而出的錯誤，仍是作者我一個人的責任。

更要感謝我的家人在這段時間的諒解。這種話好像例行公式，若非親身經歷，確實難以理解個中的深義。

最後，要特別感謝中央大學數學系的陳弘毅教授。這整本書，除了封面以外，所有的排版與設計，完全是由陳教授的中文 $\text{T}_{\text{E}}\text{X}$ 數學文件排版系統製作出來的。因為這套中文 $\text{T}_{\text{E}}\text{X}$ 系統，使得中文的數學寫作成爲一件賞心悅目的樂事。

單維彰 中央大學數學系
一九九九年八月於臺灣中壢

<http://www.math.ncu.edu.tw/~shann>

章節目錄

前言	iii
章節目錄	v
圖例目錄	ix
重要程式目錄	xiii
重要數學式目錄	xv
第一章 一階凌波導覽	1
1.1 方盒函數	1
1.2 函數間的距離	3
1.3 階梯函數的逼近論	5
1.4 大 O 符號	10
1.5 函數的脹縮平移	12
1.6 階梯函數空間	14
1.7 抽象化的向量空間	17
1.8 多層解析空間	21
1.9 投影	22
1.10 正則基底	26
1.11 Haar 函數	29
1.12 多層解析空間的基底變換	33
1.13 凌波函數	37
1.14 Haar 凌波係數消散定理	39
1.15 數位訊號處理	44
1.16 DWT: 離散凌波轉換	47

1.17	DWT 之矩陣表達	55
1.18	DWT 演算法	61
第二章 數學理論基礎—複習與預習		71
2.1	基本符號與慣例	72
2.2	泰勒展開	74
2.3	向量空間	79
2.4	基底	81
2.5	向量的長度	85
2.6	向量的內積	91
2.7	再談基底	96
2.8	L^2 空間	102
2.9	勒貝格積分	112
2.10	疊積	119
2.11	傅立葉級數	122
2.12	傅立葉轉換	136
2.13	基本樣條函數	148
第三章 正交凌波理論概述		155
3.1	自格函數	156
3.2	自格係數	160
3.3	充分條件	164
3.4	凌波函數	167
3.5	凌波係數消散定理	179
3.6	再談自格係數	189
3.7	凌波轉換	193
3.8	傅立葉分析與頻譜	199

3.9	頻譜與尺度	203
3.10	相位平面	210
3.11	凌波轉換與泰勒展開	219
3.12	特徵函數	221
3.13	凌波函數與多項式	229
第四章	正交凌波函數的訊號處理範例	239
4.1	離散凌波轉換 (DWT)	240
4.2	離散凌波轉換 (DWT) 之矩陣表達	242
4.3	一維離散凌波轉換—演算法	246
4.4	一維離散凌波轉換—範例	260
4.5	與傅立葉轉換之比較	270
4.6	二維離散凌波轉換—演算法	278
4.7	二維離散凌波轉換—範例	287
4.8	CZO 型式的測試矩陣	299
4.9	一個數學之外的問題	308
第五章	正交凌波函數的數值運算範例	311
5.1	繪製 V_j 中的函數圖形	311
5.2	函數插值與投影	320
5.3	自格與凌波函數的數值	328
5.4	自格與凌波函數的圖形	337
5.5	數值積分	353
5.6	自格函數與多項式的積分	358
5.7	自格函數的連乘積分問題	369
5.8	自格函數的數值積分法	377
5.9	傅立葉技巧	385

附錄 計算工具	393
A.1 Matlab 概要	394
A.2 Matlab 的製圖功能	403
A.3 Matlab 的顯像功能	408
後記	415
參考書目	417
英漢名詞對照	419
索引	421

圖例目錄

圖 1	方盒函數	2
圖 2	$f(x)$ 和 f_0, f_1, f_5, f_6 的逼近情形	9
圖 3	函數脹縮平移之圖例	13
圖 4	垂直投影的圖示	19
圖 5	Haar 函數	31
圖 6	兩層尺度間的分解與合成圖示	34
圖 7	一維 Haar 凌波分解的例子	65
圖 8	一維 Haar 凌波合成的例子	66
圖 9	二維 Haar 凌波分解的例子	67
圖 10	微分均值定理圖示	78
圖 11	$\ell^1, \ell^2, \ell^\infty$ 長度意義之下的單位圓	87
圖 12	$\langle \cdot, \cdot \rangle_A$ 內積意義之下的正交向量	95
圖 13	平面上的一對對偶基底	101
圖 14	勒貝格積分與黎曼積分圖示	115
圖 15	積分理論發展之示意圖	117
圖 16	發現 Gibbs 現象的不連續週期函數	130
圖 17	Gibbs 現象，取二十項或四十項部分和	131
圖 18	十六項傅立葉級數部分和	133
圖 19	$B_2(x)$ (三角帽函數)	150
圖 20	中間值定理不成立之範例	165
圖 21	高低頻係數測試函數	200
圖 22	豎笛與小喇叭的頻譜分析	202

圖 23	兩種尺度的 $\phi(2^j x - k)$	205
圖 24	理想中 V_{j+1} 、 V_j 、 W_j 之頻率反應範圍	209
圖 25	五線譜就是相位平面上的一個函數	211
圖 26	相位平面上之各種有效函蓋示意圖	216
圖 27	鋼琴的音階與頻率	218
圖 28	$p = 2, 4, 6, 8$ 的 $ m_0(\omega) $ 在 $[-\pi, \pi]$ 之內	234
圖 29	$p = 2, 4, 6, 8$ 的 $m_0(\omega)$ 在複數平面上的曲線	236
圖 30	$f(x) = x$ 的原始訊號與轉換後訊號	262 263
圖 31	$\cos x$ 的轉換後訊號	264
圖 32	一維測試函數	265
圖 33	一維測試訊號的 $\frac{1}{8}$ 解析度	266
圖 34	凌波轉換後的一維測試訊號	267
圖 35	凌波縮減除雜訊法之示意圖	269
圖 36	$H(x) - H(-x)$ 的 16 項與 50 項傅立葉級數	272
圖 37	數值黏滯性之圖例	273
圖 38	$H(x) - H(-x)$ 在 V_2 和 V_4 的正交投影	275
圖 39	一維測試訊號以 FFT 做 1:4 與 1:8 之壓縮與還原	276
圖 40	一維測試訊號以 DWT 做 1:4 與 1:8 之壓縮與還原	277
圖 41	Tiffany at $V_8(x, y)$	288
圖 42	$V_8(x) \times (V_7(y) \oplus W_7(y))$ 和 $(V_7(x) \oplus W_7(x)) \times V_8(y)$	289
圖 43	Tiffany at $V_7(x, y) \oplus W_7(x, y)$	290
圖 44	split 程式分解矩陣之圖示	292
圖 45	Tiffany 經過兩次和最多次的轉換結果	294
圖 46	Tiffany at $V_7(x, y)$ 、 $V_6(x, y)$ 、 $V_5(x, y)$ 、 $V_4(x, y)$	295
圖 47	Tiffany 經過 1:6 和 1:10 模擬壓縮後的合成影像	296
圖 48	Tiffany 經過 1:30 和 1:100 模擬壓縮後的合成影像	297

圖 49	CZO 型式的測試圖像	300
圖 50	CZO 測試矩陣經 $p = 2$ 和 $p = 4$ 凌波轉換	302
圖 51	CZO 測試矩陣經 $p = 6$ 和 $p = 8$ 凌波轉換	303
圖 52	分層轉換與全轉換的圖示	304
圖 53	CZO 測試矩陣經 $p = 2$ 和 $p = 6$ 凌波做全轉換	305
圖 54	週期化凌波轉換的邊框效應	306
圖 55	Tiffany 影像和 CZO 矩陣經過區間內之凌波轉換	307
圖 56	$\frac{4}{9}x(3-x)$ 在 $[0, 3]$ 以 $\frac{1}{64}$ 為間隔的折線圖	322
圖 57	$\frac{4}{9}x(3-x)$ 在 V_6 上做投影後的折線圖	323
圖 58	$\sum f(\frac{k}{64})\phi(64x-k)$ 以 $\frac{1}{64}$ 為間隔的折線圖	324
圖 59	$f(x) = x(1-x)$ 和 $\tilde{f}_4(x)$ 、 $\tilde{f}_5(x)$	326
圖 60	一個對稱正定 2×2 矩陣的特徵向量圖示	331
圖 61	Daubechies $\phi(x)$ 和 $\psi(x)$ $p = 2$	338
圖 62	Daubechies $\phi(x)$ 和 $\psi(x)$ $p = 3$	339
圖 63	Daubechies $\phi(x)$ 和 $\psi(x)$ $p = 4$	340
圖 64	$p = 3$ 的 $\phi(x)$ 在 $x = 1$ 附近的局部函數圖形	341
圖 65	$p = 1, 2, 3, 4$ 的 Daubechies $\phi(x)$ 與對應的 $\psi(x-p+1)$	342
圖 66	$p = 5, 6, 7, 8$ 的 Daubechies $\phi(x)$ 與對應的 $\psi(x-p+1)$	343
圖 67	$p = 9, 10, 11, 12$ 的 Daubechies $\phi(x)$ 與對應的 $\psi(x-p+1)$	344
圖 68	$p = 3$ 的 $\phi'(x)$	346
圖 69	$p = 6$ 的 $\phi'(x)$	346
圖 70	$p = 2$ 的 $\phi^{(1)}(x) \cdots \phi^{(6)}(x)$ 階梯函數逼近	349
圖 71	$p = 2$ 的 $\phi^{(1)}(x) \cdots \phi^{(6)}(x)$ 折線函數逼近	352

重要程式目錄

dwt()	離散凌波轉換的控制程式	247-250
dwt1d()	一維離散凌波轉換	251-258
dwt2d()	二維離散凌波轉換	283-286
Code-2	reflect() 鏡射成 $2^j \times 2^l$ 矩陣	250
Code-4	dwt1d 的分解執行過程	254
Code-5	pwd() 執行分解過程之週期化	254
Code-6	wf() 真正執行 DWT 的計算步驟	255
Code-8	dwt1d 的分解執行過程	258
Code-9	interlace() 將兩矩陣之列向量交叉重組	258
Code-10	pwc() 執行合成過程之週期化	259
Code-11	dwt2d 的分解執行過程	285
Code-12	dwt2d 的合成執行過程	286
Code-13	split() 將一個矩陣拆成三個子矩陣	291
Code-14	saturate() 飽和性切除	293
Code-15	用離散疊積畫 V_j 中的函數圖形	316
Code-17	用區間組合畫 V_j 中的函數圖形	318
Code-18	以特徵向量計算 $\phi(x)$ 的整數點值	333
Code-19	階梯函數迭代法畫 $\phi(x)$ 圖形	350
Code-20	折線函數迭代法畫 $\phi(x)$ 圖形	351
Code-21	row2mat() 向量轉成矩陣	401

重要數學式目錄

凌波係數消散定理	182
海森堡測不準定理	213
Haar 凌波係數消散定理	40
席瓦茲不等式	92
L^1 傅立葉轉換定理	139
L^2 傅立葉轉換定理 (Plancherel 定理、Parseval 等式)	142
勒貝格收斂定理	116
Villemoes 充分條件	226
Lawton 充分條件	166
單農取樣定理	152
Paley-Wiener 定理	147
微分均值定理	77
中間值定理	165
性質 2.1 席瓦茲不等式之積分形式	88
性質 2.2 垂直投影是最短距離	93
性質 2.3 樣條函數的基本性質	149
性質 2.4 基本樣條函數的遞迴性質	149
性質 3.1 基本樣條函數之自格等式 158, (證明)	223
性質 3.2 自格係數之奇數項和為 1	162
性質 3.3 凌波投影係數與 Hölder 指數的關係	181
性質 3.4 正則自格函數之充要條件	224
性質 3.5 正則自格函數之特徵函數性質	225

性質 3.6	自格函數可組成多項式	233
性質 5.1	$\sum \tau(k)\phi(x - k) = x^m$	360
性質 5.2	$M_2 = (M_1)^2$	360
性質 5.3	Strang 矩陣的長度 $\leq \sqrt{2}$	366
性質 5.4	積分形成新的自格函數	370
性質 5.5	Lawton 矩陣的長度 $\leq \sqrt{2}$	376
性質 5.6	凌波函數之導函數內積	387
性質 5.7	導函數之傅立葉轉換	388
性質 5.8	正交凌波函數的質量中心在中點	389

(1.8)	8	(2.17)	104
(1.16)	17	(2.18)	105
(1.20)	21	(2.21)	113
(1.27)	27	(2.22)	120
(1.34)	32	(2.23)	120
(1.37)	35	(2.24)	123
(1.39)	39	(2.28)	126
(1.41)	40	(2.30)	128
(1.44)	42	(2.32)	132
(1.49)	49	(2.33)	134
(1.50)	50	(2.34)	135
(1.53)	51	(2.36)	136
(1.58)	53	(2.43)	142
(1.60)	53	(2.44)	143
(1.63)	58	(2.47)	145
(2.11)	95	(2.49)	146

(2.51)	148	(3.64)	224
(2.53)	149	(3.67)	225
(3.4)	158	(3.70)	229
(3.6)	161	(3.71)	230
(3.10)	162	(3.76)	233
(3.11)	163	(3.77)	234
(3.17)	171	(3.78)	234
(3.19)	171	(4.3)	242
(3.20)	172 171	(4.5)	243
(3.24)	174	(4.10)	252
(3.25)	176	(5.6)	315
(3.29)	179	(5.10)	321
(3.35)	189	(5.11)	322
(3.36)	191	(5.14)	329
(3.43)	196	(5.15)	335
(3.47)	198	(5.17)	354
(3.48)	204	(5.23)	358
(3.60)	222	(5.26)	360
(3.61)	222	(5.30)	366
(3.62)	223	(5.35)	369

第一章

一階凌波導覽

在這開卷第一章裡面，我們探討所謂的方盒函數，並利用它來推導多層解析空間和 Haar 凌波函數，以及它們的性質和計算方法。在這一章裡面，我們從最單純的設定出發，一路上慢慢地引進抽象理論與專門術語，並且做小範圍的推廣。這一章所需的數學基礎，只有最基本的微積分和線性代數。

可以說，方盒函數帶領我們在整個凌波函數的園地中做最初步的流覽。希望讀者能從這裡獲得一個粗略的理論架構和發展動機。在第二章裡面，我們補充一些數學知識。然後在第三章，我們將方盒函數及 Haar 凌波函數的性質抽象化並做推廣。

1.1 方盒函數

所謂方盒函數就是

$$\phi(x) = \begin{cases} 1 & \text{if } x \in [0, 1) \\ 0 & \text{otherwise} \end{cases} \quad (1.1)$$

參照圖 1，由於 $\phi(x)$ 的圖形像個方盒子，所以它常被暱稱為方盒函數 (*box functions*)。 ϕ 是個希臘字母，希臘人讀 fee 的音，但是在英語國

idx:box

eq:box

2 凌波初步

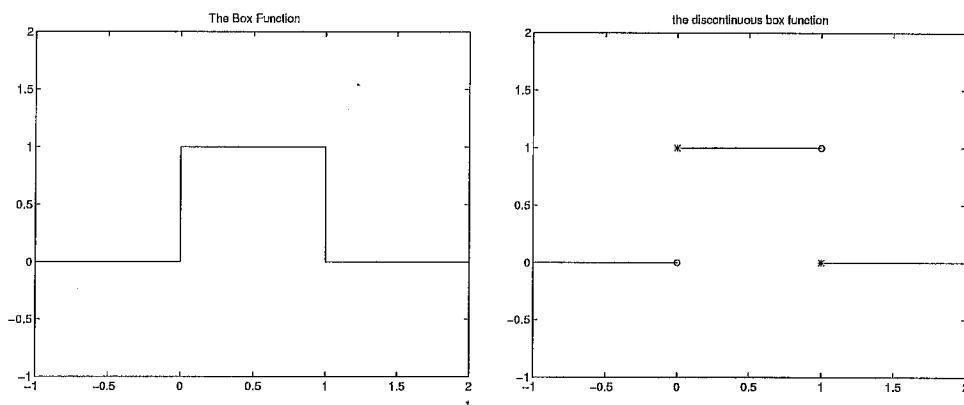


fig:box 圖 1 方盒函數，兩種圖示

家也有人讀 fai。就好像 π 也是個希臘字母，希臘人讀 pee，但是在臺灣大部分人讀 pai。 ϕ 的英文字母拼寫是 phi。)

在圖 1 中我們呈現兩種方盒函數的圖形。左邊的是方塊波圖示。 $B_1(x)$ 是不連續函數，方塊波中的垂直線不應該在那裡，它只是為了視覺上的好看。右邊的是提醒函數定義 (1.1) 的圖示。它標明了 $\phi(0) = 1$ 而 $\phi(1) = 0$ 。

讓我們介紹一個名詞：函蓋。就函數 $y = f(x)$ 的圖形看來，可以想像那條曲線是比重均勻的鐵鏈，或是有彈性的薄膜。在沒有外力的情況下，它應該靜止在 x -軸上。它之所以離開了 x -軸 (有上下起伏)，乃是因為有能量施予其上。所以，使得函數 $f(x) \neq 0$ 的區間，表示有力量支撐著它，那段區間就稱為 $f(x)$ 的支架 (*support*) 換另外一個角度來看，使得 $f(x) \neq 0$ 的區間，就好像是被函數圖形覆蓋而有影子的部分。因此那段區間又稱為 $f(x)$ 的函蓋 (*support*)。我們選擇函蓋這個名詞，看起來似乎比支架文雅一點。但是支架所賦予的聯想，是比較有意義的。符號上記做 $\text{supp } f(x)$ 。

idx:supp

爲了數學上的其他理由，我們定義函蓋爲最小的包含所有 $f(x) \neq 0$

之 x 的閉區間之聯集。所以，雖然當 $x \in [0, 1)$ 時 $\phi(x) \neq 0$ ，我們仍說 $\phi(x)$ 的函蓋是 $[0, 1]$ 。當兩個函數的函蓋之交集只有一個點的時候，我們說它們兩個函蓋幾乎沒有交集。

令

$$V = \{ \text{閉區間 } [0, 1] \text{ 內所有連續實值函數} \} \quad (1.2) \quad \text{eq:Vdef}$$

(函數值在 \mathbb{R} 內，我們暫不考慮複數值的函數)。任取一個 $f(x) \in V$ ，我們要提這個問題：哪一個數 α ，使得 $f(x)$ 和 $\alpha\phi(x)$ 的距離最近？要回答這個問題，我們必須先問，什麼是兩個函數之間的距離？

習題 1.1 若 $g(x) = \max\{1 - x^2, 0\}$ ，請問 $\text{supp } g(x)$ ？

1.2 函數間的距離

回顧微積分課程中可能會提到的柯西積分定理：閉區間內的連續函數都是可積的。所以，若 $f(x)$ 與 $g(x)$ 是 $[a, b]$ 內的連續函數，則它們是可積函數。因為連續函數的平方還是連續函數，所以 $f^2(x)$ 和 $g^2(x)$ 也是可積的。

若 $u = (u_1, u_2, \dots, u_n)$ 和 $v = (v_1, v_2, \dots, v_n)$ 是兩個向量，我們知道， u 和 v 的距離是

$$\left(\sum_{i=1}^n |u_i - v_i|^2 \right)^{\frac{1}{2}}$$

比照向量的距離定義，以此為動機，我們定義 $f(x)$ 和 $g(x)$ 的距離是 idx:dist

$$\left(\int_a^b |f(x) - g(x)|^2 dx \right)^{\frac{1}{2}} \quad (1.3) \quad \text{eq:fgdist}$$

可見距離一定是個非負的實數，而當 $f(x) = g(x)$ 的時候距離是 0。反過來，如果距離是 0，表示

$$\int_a^b |f(x) - g(x)|^2 dx = 0$$

4 凌波初步

但 $|f(x) - g(x)|^2$ 是個非負的連續函數，它底下所圍的面積是 0，所以必須 $|f(x) - g(x)|^2 = 0$ ，也就是 $f(x) = g(x)$ 。

如果 $f(x)$ 是 $[a, b]$ 內的連續函數， α 代表 $[a, b]$ 內的常數函數。根據 (1.3) 的距離意義，我們就是要找一個 α 使得

$$\int_a^b (f(x) - \alpha)^2 dx \quad \alpha \in \mathbb{R}$$

發生極小值。上述對 x 之積分結果為 α 的函數，所以我們對 α 微分，求其臨界點：

$$\frac{d}{d\alpha} \int_a^b (f(x) - \alpha)^2 dx = -2 \int_a^b f(x) - \alpha dx = 0$$

故得

$$\int_a^b f(x) dx = \int_a^b \alpha dx = \alpha(b - a)$$

因此看到，和 $f(x)$ 距離最近的 α 就是

$$\alpha = \frac{1}{b - a} \int_a^b f(x) dx \quad (1.4) \quad \text{eq: fmean}$$

也就是 $f(x)$ 在 $[a, b]$ 區間內的積分平均值。

回到前面的問題：若 $f(x)$ 是 $[0, 1]$ 內的連續函數，什麼時候 $f(x)$ 和 $\alpha\phi(x)$ 的距離最近？嚴格來說 $\alpha\phi(x)$ 並不是 $[0, 1]$ 區間內的連續函數，它在 1 那一點不連續。但是根據 (1.3) 的距離定義，這並沒有影響。因為 $\alpha\phi(x)$ 還是在 $[0, 1]$ 內可積，而且那一個點的不連續對積分值根本沒有影響。所以，我們可以援用前面的結論：與 $f(x)$ 最近的 $\alpha\phi(x)$ 就是 $\alpha = \int_0^1 f dx$ 。

習題 1.2 令 α 如 (1.4) 所定，亦即 $\alpha\phi(x)$ 距離 $f(x)$ 最近，請寫出它們之間的距離與 $\int f^2 dx$ 、 $\int f dx$ 的關係。

1.3 階梯函數的逼近論

在 $[0, 1]$ 區間內，我們定義 0 次階梯函數為以下類型的函數：

$$\begin{cases} v_{00} & \text{if } x \in [0, 1) \\ 0 & \text{if } x = 1 \end{cases} \quad (1.5) \quad \text{eq:stepzero}$$

定義 1 次階梯函數為以下類型的函數：

$$\begin{cases} v_{10} & \text{if } x \in [0, \frac{1}{2}) \\ v_{11} & \text{if } x \in [\frac{1}{2}, 1) \\ 0 & \text{if } x = 1 \end{cases} \quad (1.6) \quad \text{eq:stepone}$$

依此類推。其中 v_{00} 、 v_{10} 、 v_{11} 是常數，而 00 、 10 、 11 是足標，就像我們在寫矩陣元素的時候所用的雙足標一樣。

可見，所謂 j 次階梯函數，就是將 $[0, 1]$ 區間等分切成 2^j 段：

$$[\frac{k}{2^j}, \frac{k+1}{2^j}], \quad \text{for } k = 0, 1, \dots, 2^j - 1$$

在每一段裡面幾乎就是一個常數函數 (除了右端點之外)。也就是

$$\begin{cases} v_{j0} & \text{if } x \in [0, \frac{1}{2^j}) \\ v_{j1} & \text{if } x \in [\frac{1}{2^j}, \frac{2}{2^j}) \\ \vdots & \\ v_{j, 2^j-1} & \text{if } x \in [1 - \frac{1}{2^j}, 1) \\ 0 & \text{if } x = 1 \end{cases}$$

像這樣的函數，我們又稱它為分片常數函數，因為它是由一片一片的常數所組成的函數。讀者可以想像它的函數圖形，好像一階一階的臺階 (可以參考後面的圖 2 或)，它因此而得名 (step functions)。

在前面我們學習了函數的距離，因此知道，在所有 0 次階梯函數中，最靠近 $f(x)$ 的就是當 $v_{00} = \int_0^1 f dx$ 的時候。由 (1.4)，我們可以推

6 凌波初步

廣這個觀念到任意 j 次階梯函數。在所有 j 次階梯函數中，令 $f_j(x)$ 是最靠近 $f(x)$ 的，它的 2^j 片常數分別是 $f(x)$ 在每一小段 $[\frac{k}{2^j}, \frac{k+1}{2^j}]$ 裡面的積分平均值：

$$v_{jk} = 2^j \int_{\frac{k}{2^j}}^{\frac{k+1}{2^j}} f(x) dx$$

階梯函數雖然不連續，而且結構簡單（一片片的常數），但是它們卻可以逼近 (*approximate*) $[0, 1]$ 內的連續函數。乍聽之下似乎匪夷所思。但是讓我們先觀察一個例子。在圖 2 中，我們看到一個函數 idx: approx

$$f(x) = -\frac{16}{9}x^2 + \frac{8}{3}x$$

和最靠近它的 $f_0(x)$ 、 $f_1(x)$ 、 $f_5(x)$ 、 $f_6(x)$ 。從視覺的直覺上，我們應該可以看得出來，當 j 越來越大的時候， $f_j(x)$ 就圖形就長得越來越像 $f(x)$ 。這差不多就是逼近的意思了。

要說得明確，就必須用數學觀念。我們檢查 $f(x)$ 和在所有 j 次階梯函數中最靠近它的 $f_j(x)$ ，到底有多靠近？這也就是問， $f(x)$ 和 $f_j(x)$ 的距離是多少？如果我們知道 $f(x)$ 的表達式，比如說 $f(x) = x(1-x)$ ，則根據 (1.3) 的定義，我們可以明確計算 $f(x)$ 和 $f_j(x)$ 的距離。但是當我們要對一般的 $f(x)$ 做理論性的探討時，必須做一些數學上的假設，並且使用數學分析的工具。

因為我們將要常常提到距離，所以值得為它定義一個符號，以簡化我們的行文。我們用 idx: norm

$$\|f(x) - g(x)\|$$

表示， $f(x)$ 和 $g(x)$ 的距離。而其定義如同 (1.3)。

如果 $f(x)$ 是一次可微，而且其一次導函數連續，簡單地記做

$$f(x) \in C^1([0, 1])$$

則它可以做一階的泰勒展開 (參考第二章二節)。根據連續函數的中間值定理，在每一小段區間 $[\frac{k}{2^j}, \frac{k+1}{2^j}]$ 內，都至少有一個點 x_{jk}^* 使得 $f(x_{jk}^*)$ 等於 $f(x)$ 在此區間內的積分平均值；也就是說 $f(x_{jk}^*) = v_{jk}$ 。那麼以 x_{jk}^* 為參考點做一階泰勒展開，就得到

$$f(x) = f(x_{jk}^*) + f'(\xi)(x - x_{jk}^*) = v_{jk} + f'(\xi)(x - x_{jk}^*),$$

$$\text{for } x \in [\frac{k}{2^j}, \frac{k+1}{2^j}]$$

其中 ξ 是某個落在開區間 $(\frac{k}{2^j}, \frac{k+1}{2^j})$ 內的點，我們不確定它是誰。因為我們假設 $f'(x)$ 也是 $[0, 1]$ 內的連續函數，而連續函數在閉區間內必有一最大值。所以可以說，存在一個 M ，使得 $|f'(x)| \leq M, \forall x \in [0, 1]$ 。同時，注意

$$|x - x_{jk}^*| \leq |\frac{k+1}{2^j} - \frac{k}{2^j}| = \frac{1}{2^j}$$

又因為積分可以分段來做，所以

$$\int_0^1 (f(x) - f_j(x))^2 dx = \sum_{k=0}^{2^j-1} \int_{\frac{k}{2^j}}^{\frac{k+1}{2^j}} (f(x) - v_{jk})^2 dx$$

有了這些預備知識，我們可以推導

$$\begin{aligned} \|f(x) - f_j(x)\|^2 &= \sum_{k=0}^{2^j-1} \int_{\frac{k}{2^j}}^{\frac{k+1}{2^j}} f'(\xi)^2 (x - x_{jk}^*)^2 dx \\ &\leq M^2 \left(\frac{1}{2^j}\right)^2 \sum_{k=0}^{2^j-1} \int_{\frac{k}{2^j}}^{\frac{k+1}{2^j}} 1 dx = M^2 \left(\frac{1}{2^j}\right)^2 \sum_{k=0}^{2^j-1} \frac{1}{2^j} \\ &= M^2 \left(\frac{1}{2^j}\right)^2 \end{aligned}$$

8 凌波初步

根據 (1.3)，開根號之後就是距離。所以

$$\|f(x) - f_j(x)\| \leq \frac{M}{2^j} \quad (1.7) \quad \text{eq:apx}$$

因此我們知道 $\|f(x) - f_j(x)\|$ 的上界。這就是數學分析所能明確描述的理論。如果 $f(x) \notin C^1$ ，並不是 $f(x)$ 和 $f_j(x)$ 就一定不互相靠近，而是沒有數學理論可以保證 $f_j(x)$ 往 $f(x)$ 靠近的速度。例如圖 2 中的 $f(x)$ ，它的 $M = \frac{8}{3}$ ，所以 $\|f(x) - f_6(x)\| \leq \frac{1}{24}$ 。

其實上述 $\|f(x) - f_j(x)\|$ 之估計，可以做得更精確。 $|f'(x)|$ 的上界可以在每個小區段中討論。參照圖 2 的 $f_1(x)$ 或 $f_5(x)$ 的圖形，可以看得出來，當區段靠近 $x = 0$ 的時候， $|f'(x)|$ 的值比較大，因此在這些區段內，常數函數和 $f(x)$ 就離得比較遠。當區段靠近 $x = \frac{3}{4}$ 的時候， $|f'(x)|$ 的值比較小，因此在這些區段內，常數函數和 $f(x)$ 就靠得比較近。目前我們不需要做更精確的估計。

現在，我們可以闡述階梯函數的逼近理論。

如果 $f(x) \in C^1([0, 1])$ 而 $f_j(x)$ 是最靠近 $f(x)$ 的 j 次階梯函數，

$$\lim_{j \rightarrow \infty} \|f(x) - f_j(x)\| = 0 \quad (1.8) \quad \text{eq:xpm}$$

這就是我們曾說階梯函數可以逼近連續函數的明確意義了。其實我們還可以說得更明確，使用所謂的大 O 符號，我們可以這樣表達：

$$f_j(x) = f(x) + O\left(\frac{1}{2^j}\right) \quad (1.9) \quad \text{eq:xpo}$$

在圖 2 中，我們呈現一個階梯函數逼近連續函數的範例。圖中的 $f_5(x)$ 和 $f_6(x)$ 因為每片常數很短，所以就不再細述它們左右端點的值。如果想知道製作這些圖片的程序，請參考附錄 A.2。

習題 1.3 令 $f(x) = x^2$ ， $x \in [0, 1]$ 。請寫出 $f_2(x)$ 的四片常數。根據 (1.7)， $\|f(x) - f_2(x)\|$ 的上界是多少？而真實的 $\|f(x) - f_2(x)\|$ 是多少？

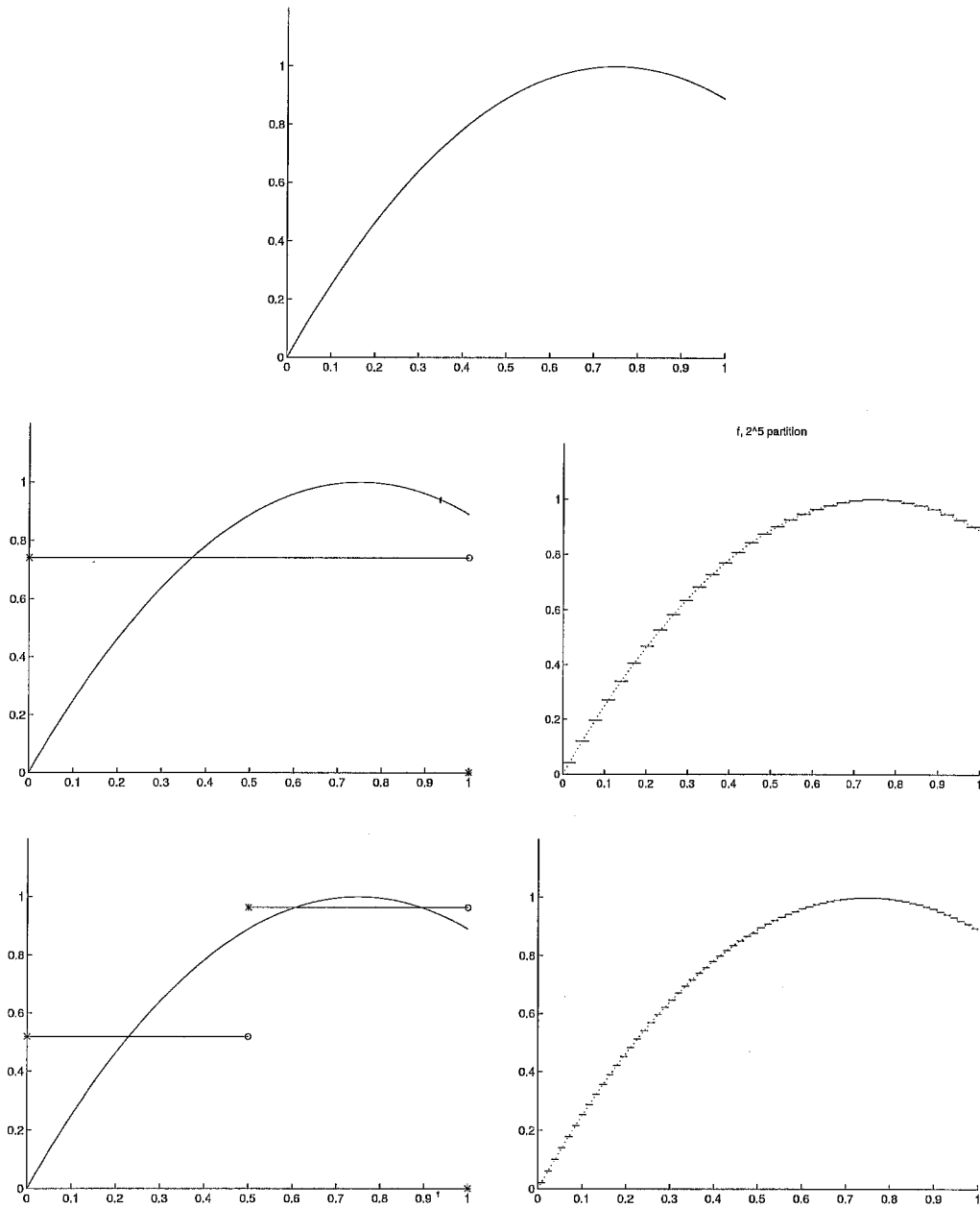


fig:steps 圖 2 $f(x)$ 和 f_0 、 f_1 、 f_5 、 f_6 的逼近情形

1.4 大 O 符號

在 (1.8) 裡面，我們只看到 $\|f(x) - f_j(x)\|$ 收斂到 0，在那個式子裡面，並沒有透露它收斂到 0 的速度是多快 (或是多慢)。讓我們看一些更簡單的情形。當 $x \rightarrow 0$ 的時候， x 和 x^2 都趨近於 0。但是

$$\lim_{x \rightarrow 0} \frac{x^2}{x} = 0, \quad \lim_{x \rightarrow 0} \frac{x}{x^2} = \infty$$

在這裡，我們要提出收斂速度的看法。固然 x 和 x^2 都趨近於 0，但是速度不同。直覺上，我們會同意 $x^2 \rightarrow 0$ 的速度遠快於 $x \rightarrow 0$ 的速度。所以，在 $\frac{x^2}{x}$ 中，分子趨近於 0 的速度遠快於分母趨近於 0 的速度。因此 $\frac{x^2}{x} \rightarrow 0$ 。在 $\frac{x}{x^2}$ 中，分母趨近於 0 的速度遠快於分子趨近於 0 的速度，因此它發散。

應用數學常常是計量的數學，所以，除了關心收斂不收斂的問題之外，收斂的速度，更是我們所要仔細研究的。比如說，當我們要用某個計算方法，寫個電腦程式算出一個數列 $\{x_n\}$ ，使得 $\lim x_n$ 逼近某數 x 。在數學上，當然需要證明 $\lim x_n = x$ 。基於這種數學理論上的保證，我們才能相信，當 n 夠大之後 (也就是說，電腦迭代了夠多次之後)， x_n 的值就很靠近 x 的值。這時候，如果 $x_n \rightarrow x$ 的速度有如 $\frac{1}{\sqrt{n}} \rightarrow 0$ 的速度，電腦要花 100,000,000 個迭代步驟才能得到 0.1% 誤差的估計值，但是如果 $x_n \rightarrow x$ 的速度有如 $\frac{1}{n^2} \rightarrow 0$ 的速度，電腦可能只要花 100 個迭代步驟就能得到 0.1% 誤差的估計值了。其間的差異非常大。所以當我們探討應用數學的收斂或逼近理論時，特別關心速度的估計。

大 O 符號的目的就是不但要表達收斂，還要描述其速度。若兩個數列 x_n 與 y_n 符合以下關係

$$0 < \lim_{n \rightarrow \infty} \frac{x_n}{y_n} < \infty$$

則記做

$$x_n = O(y_n) \quad (1.10) \quad \text{eq:big0}$$

意思是說 x_n 和 y_n 以大約同樣的速度趨近於它們的極限。所以，回顧 (1.9)，它的意思就是

$$\|f_j(x) - f(x)\| = O(2^{-j})$$

或者說 $f_j(x) \rightarrow f(x)$ 的速度和 $2^{-j} \rightarrow 0$ 的速度一樣快。

其實這個速度並不非常令人滿意。因為 $2^{-j} \rightarrow 0$ 的速度並不很快。例如當 $j = 20$ 的時候 $2^{-20} \approx 10^{-6}$ 。我們希望能夠更快地收斂 (人總是不滿足的)。只要能夠提高一個冪數，對收斂的速度就大有幫助。比如說，如果能夠有 2^{-2j} 的收斂速度，那麼當 $j = 20$ 的時候，誤差大約就是 $2^{-40} \approx 10^{-12}$ 。

我們看到，雖然階梯函數可以逼近 C^1 函數，但是速度差強人意。有沒有更高效率的逼近方法呢？當然是有的，我們將在第三章回答這個問題。

方盒函數和 j 次階梯函數之間，有什麼關係？我們已經發現 0 次階梯函數就是 $c\phi(x)$ ， c 是某個實數。但是一般的 j 次階梯函數是否可以用方盒函數 $\phi(x)$ 來表達？當我們瞭解函數的脹縮平移之後，這個問題就會很容易回答了。

習題 1.4 如果 $x_n = O\left(\frac{1}{n \log n}\right)$ ， $y_n = O\left(\frac{1}{n^2}\right)$ ，則

$$\lim_{n \rightarrow \infty} \frac{x_n}{y_n} = ?$$

習題 1.5 考慮積分 $\int_1^\infty \frac{1}{x^r} dx$ 只有當 $r > 1$ 時可積。則以下哪些函數在 $[1, \infty)$ 之中可積？這個習題的用意在於，學習判別哪些函數是 $O\left(\frac{1}{x^r}\right)$ 而

12 凌波初步

其中 $r > 1$ 。

$$\frac{\sqrt{x}}{x^2 + 1}, \quad \frac{x}{\sqrt{x^4 + 1}}, \quad \sqrt{\frac{x}{x^2 + 1}}, \quad \frac{\sin x}{x^2 + 1}$$

1.5 函數的脹縮平移

在這一節裡面，我們要複習幾種基本的函數變形。假設 $f(x)$ 是一個有限函蓋函數，其函蓋是 $[a, b]$ 。首先考慮 $f(x - k)$ ，因為

$$\text{supp } f(x - k) = [a + k, b + k] \quad (1.11) \quad \text{eq:shift}$$

可見 $f(x - k)$ 保持 $f(x)$ 的圖形不變，只是整個圖形移動了 k 個單位。如果 $k > 0$ ，則表示向右移，如果 $k < 0$ ，則表示向左移。所以 $f(x - k)$ 形式的變形叫做函數的平移。

idx:shift

注意不要和 $f(x) - k$ 弄糊塗了。 $f(x) - k$ 是將 $f(x)$ 的圖形下降 (若 $k > 0$) 或上升 (若 $k < 0$)，我們稱之為函數的升降。但是在凌波函數的課題中，我們不需要升降。

接著，考慮 $f(2x)$ 。只有 $x \in [\frac{a}{2}, \frac{b}{2}]$ 的時候 $f(2x)$ 才有非零值。也就是說

$$\text{supp } f(2x) = \left[\frac{a}{2}, \frac{b}{2}\right]$$

可見 $f(2x)$ 的寬度是 $f(x)$ 的一半，就好像整個圖形被壓縮變瘦了一樣。反之， $\text{supp } f(\frac{x}{2}) = [2a, 2b]$ ，所以 $f(\frac{x}{2})$ 的寬度是 $f(x)$ 的兩倍，就好像整個圖形被膨脹變胖了一樣。依此類推， $f(2^j x)$ 就是 $f(x)$ 被壓縮 ($j > 0$) 或膨脹 ($j < 0$) 2^j 倍。因此我們稱 $f(2^j x)$ 形式的變形叫做函數的脹縮。

idx:dilate

注意不要和 $2f(x)$ 弄糊塗了。 $2f(x)$ 是將 $f(x)$ 的圖形拉高，而 $2f(x)$ 並沒有將圖形變胖或變瘦。函數的脹縮並沒有改變原來的高度。函數

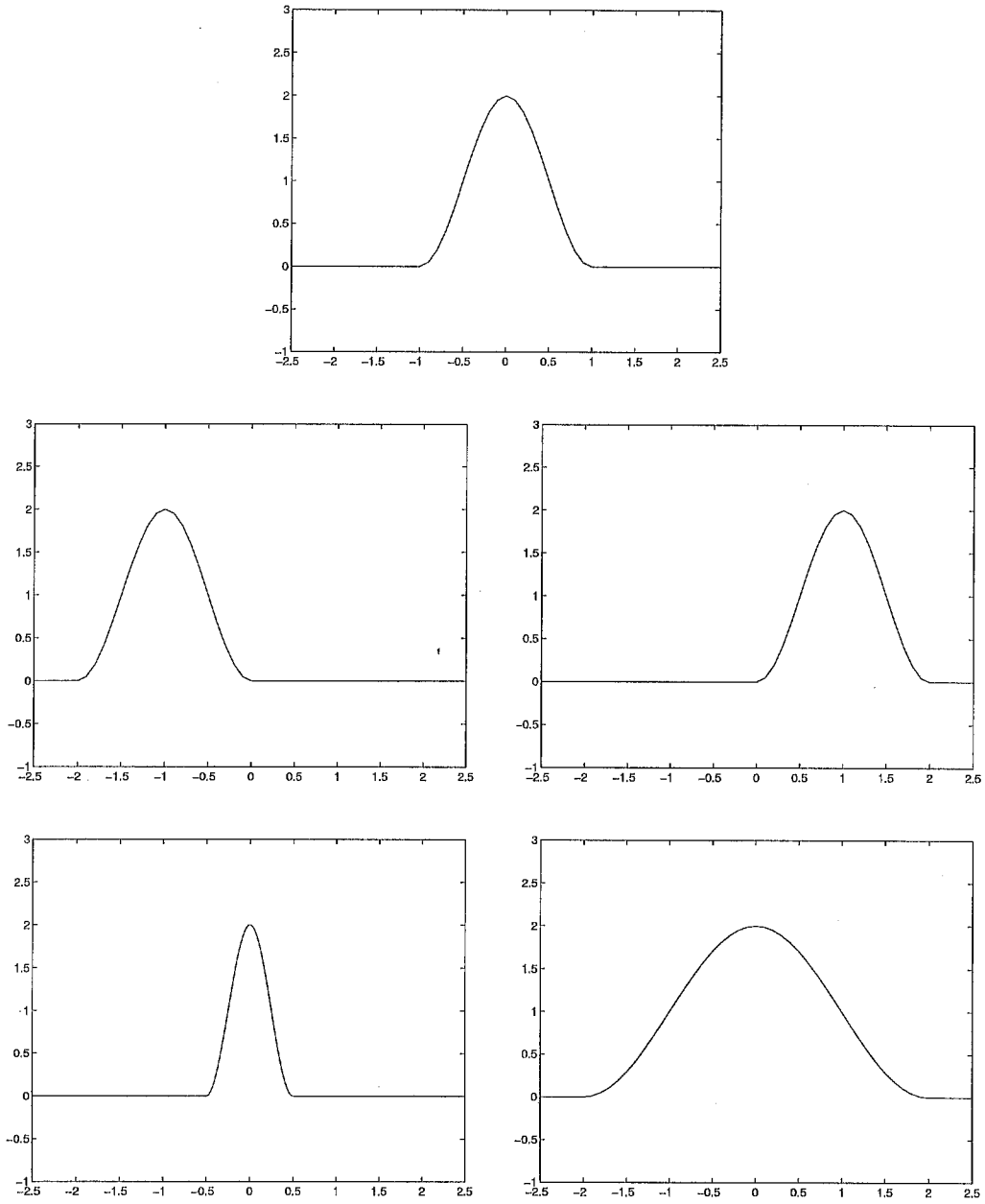


fig:dilshift 圖3 函數脹縮平移之圖例

的脹縮平移是凌波函數論中的基本操作。在圖 3 中我們示範幾個例子。圖中的上方是 $f(x)$ ，它是一個鐘形函數 (*bell function*)：

$$f(x) = \begin{cases} 1 + \cos \pi x & \text{if } |x| \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (1.12)$$

這是個一次可微的有限函蓋函數，其函蓋是 $[-1, 1]$ 。圖 3 的第二列，左邊是 $f(x+1)$ ，右邊是 $f(x-1)$ 。第三列圖的左邊是 $f(2x)$ ，右邊是 $f(\frac{x}{2})$ 。

脹縮平移混作一起的時候，要注意平移的單位長改變了。例如

$$\text{supp } f(2x - k) = \left[\frac{a+k}{2}, \frac{b+k}{2} \right] \quad (1.13)$$

所以，如果 $k > 0$ ， $f(2x - k)$ 就是 $f(2x)$ 向右平移 $\frac{1}{2}k$ 個單位，如果 $k < 0$ 就是向左平移 $\frac{1}{2}k$ 個單位。

習題 1.6 證明 (1.11)。

習題 1.7 若 $j, k > 0$ ，請問 $f(2^j x - k)$ 是將 $f(x)$ 縮窄 2^j 倍之後再向右平移多少單位？

習題 1.8 令 $\phi(x)$ 為方盒函數， $j, k \in \mathbb{Z}$ 為整數而 $j \geq 0$ ，請討論以下的積分值。(亦即，討論當 j, k 是什麼的時候，積分值是什麼。)

$$\begin{aligned} & \int_{-\infty}^{\infty} \phi(x - k) dx, & \int_{-\infty}^{\infty} \phi(x)\phi(x - k) dx \\ & \int_{-\infty}^{\infty} \phi(2^j x - k) dx, & \int_{-\infty}^{\infty} \phi(2^j x)\phi(2^j x - k) dx \end{aligned}$$

1.6 階梯函數空間

從 0 次階梯函數的定義 (1.5) 和方盒函數的定義 (1.1)，明顯可見 0 次階梯函數就是 $v_{00}\phi(x)$ 這種形式的函數。因為

$$\phi(2x) = \begin{cases} 1 & \text{if } x \in [0, \frac{1}{2}) \\ 0 & \text{otherwise} \end{cases}, \quad \phi(2x - 1) = \begin{cases} 1 & \text{if } x \in [\frac{1}{2}, 1) \\ 0 & \text{otherwise} \end{cases}$$

回顧 1 次階梯函數的定義 (1.6)，可見 1 次階梯函數就是

$$v_{10}\phi(2x) + v_{11}\phi(2x - 1)$$

這種形式的函數。依此類推， j 次階梯函數就是

$$\sum_{k=1}^{2^j-1} v_{jk}\phi(2^j x - k), \quad v_{jk} \in \mathbb{R}$$

這種形式的函數。

數學的發展，總是把具備某種關係的物件放在一起討論，稱之為集合。比如說平面上所有向量的集合，所有階數 ≤ 3 的多項式集合， $[0, 1]$ 中所有連續函數的集合，所有整數的集合 (\mathbb{Z})，等等。有了集合以後，我們可以討論它們的交集、聯集、差集之類的抽象關係。許多有用的性質就是這樣討論出來的。

現在，我們要將所有 j 次階梯函數收集到一個集合裡面。令

$$V_j = \{[0, 1] \text{ 內的 } j \text{ 次階梯函數}\}$$

用數學符號表達，就是

$$V_j = \left\{ v(x) \mid v(x) = \sum_{k=1}^{2^j-1} v_{jk}\phi(2^j x - k), \quad v_{jk} \in \mathbb{R} \right\} \quad (1.14)$$

eq:Vja

利用這些符號，階梯函數的逼近論就可以寫成一個比較抽象的形式：

任給一個 $[0, 1]$ 內的連續函數 $f(x)$ ，我們可以找到一個

$$f_j(x) \in V_j$$

使得

$$\|f(x) - f_j(x)\| \leq \|f(x) - v(x)\|, \quad \forall v(x) \in V_j \quad (1.15) \quad \text{eq:bestapp}$$

其中 (1.15) 的意思就是說 $f_j(x)$ 是 V_j 中最靠近 $f(x)$ 的函數。

數學結構中，集合算是最基礎的一個。但是在單純的集合之間，並沒有很豐富的關係可以探討。我們通常知道的集合之間的關係，只有屬於 (\in)、包含於 (\subset) 和等於 (其實等於和包含於都是從屬於推導出來的)；通常知道的集合之間的運算，只有交集 (\cap)、聯集 (\cup) 和差集 (其實差集是由交集和聯集推導出來的)。當我們將一些物件定義成集合之後，最好能抽出它們共同的性質，然後在集合的關係與運算之外，再定義它們之間的特殊關係和運算，這樣我們的數學結構才會更豐富，也才能使用比較多的工具、做更多的理論推導。

觀察 V_j 這個集合 (1.14)。因為 $v_{jk} \in \mathbb{R}$ ，所以它有無窮多個元素。所以它看似很複雜。但是，讓我們回憶向量空間 (或線性空間) 的定義 (或是翻閱第二章中的複習)。我們發現 V_j 就是一個抽象的向量空間。其中的向量就是

$$v \in V_j \iff v = \sum_{k=0}^{2^j-1} v_{jk} \phi(2^j x - k)$$

我們很容易可以檢查 $\phi(2^j x - k)$ 是一組線性無關的函數 (或向量)，因此 $\{\phi(2^j x - k)\}$ 就是 V_j 的一組基底；而 V_j 是 2^j 維度的向量空間。

數學為什麼要抽象化？抽象化有什麼好處？且看下回分解。

習題 1.9 證明

$$V_j \subset V_{j+1}$$

習題 1.10 令 $f(x)$ 和 $f_j(x)$ 如 (1.15) 所述。根據 $V_j \subset V_{j+1}$ 證明

$$\|f(x) - f_{j+1}(x)\| \leq \|f(x) - f_j(x)\|$$

1.7 抽象化的向量空間

抽象使事物變得具體。

所謂抽象化，就是將事物中最純粹的性質萃取出來。經過萃取之後，許多看起來井水不犯河水的不相干事物，就會變成相同。因此，這花花世界中的千奇百態，經過抽象化之後，就僅只在皮相上有所不同，而在本質上無有差異。所以無論它在皮相上多麼複雜，只要掌握了抽象的意義，我們可以將看似複雜的事物類比到看似簡單或熟悉的事物上，藉著我們在熟悉事物上的具體經驗，來操作或分析看似複雜的事物。

我們從小學習數數兒、學習算術，就是一個抽象化的過程。一旦我們掌握了一顆牛奶糖加兩顆牛奶糖，就是三顆牛奶糖；一片巧克力加兩片巧克力，就是三片巧克力。我們就會默默地發現並且相信， $1 + 2 = 3$ 的事實與物件無關。因此在日後，會不加猶豫地相信，看了一小時的卡通，再看兩小時的卡通，一共看了三小時的卡通。

抽象的向量空間，其實就是由坐標平面的向量空間 (\mathbb{R}^2) 萃取出來的。如果讀者需要複習，請參閱第二章三四五六節。所以一旦 V_j 被貼上向量空間的標籤，我們就可以利用平面向量的性質來類比 V_j 的性質。平面向量有基底向量 $\vec{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ 和 $\vec{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ ； V_j 有基底函數 $\phi(2^j x)$ 、 $\phi(2^j x - 1)$ 、 \dots 、 $\phi(2^j x - 2^j + 1)$ 。平面向量 $\vec{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ 的長度是

$$\|\vec{v}\| = \sqrt{v_1^2 + v_2^2}$$

(我們用 $\|\cdot\|$ 代表長度)； V_j 中的函數 $v(x)$ 也有長度

$$\|v(x)\| = \left(\int_0^1 |v(x)|^2 dx \right)^{\frac{1}{2}} \quad (1.16) \quad \text{eq:vnorm}$$

因此 $u(x)$ 和 $v(x)$ 之間的距離就是 $\|u(x) - v(x)\|$ 。

在平面向量中還有一個好的性質：內積。若 $\vec{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ 、 $\vec{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ 則將內積寫成

$$\vec{u} \cdot \vec{v} = \langle \vec{u}, \vec{v} \rangle = u_1 v_1 + u_2 v_2$$

而 $\langle \vec{v}, \vec{v} \rangle = \|\vec{v}\|^2$ 。參照 (1.16) 我們也可以類比地定義 V_j 中的內積 (*inner product*)：

idx:ip

$$\langle u(x), v(x) \rangle = \int_0^1 u(x) v(x) dx \quad (1.17)$$

eq:vip

在平面上，因為三角形的兩邊和不小於第三邊，所以有平面向量的三角不等式 $\|\vec{u} + \vec{v}\| \leq \|\vec{u}\| + \|\vec{v}\|$ 。類比於 V_j 中長度定義 (1.16)，也具備三角不等式：

$$\|u(x) + v(x)\| \leq \|u(x)\| + \|v(x)\|$$

根據內積定義，我們知道 $\langle \vec{u}, \vec{v} \rangle = \|\vec{u}\| \|\vec{v}\| \cos \theta$ ，因此 $|\langle \vec{u}, \vec{v} \rangle| \leq \|\vec{u}\| \|\vec{v}\|$ 。這個不等式有許多不同的名字，我們稱它為席瓦茲不等式 (*Schwarz inequality*)。類比於 V_j 中內積的定義 (1.17)，也具備席瓦茲不等式：

idx:Schwarz

$$|\langle u(x), v(x) \rangle| \leq \|u(x)\| \|v(x)\|$$

回顧 V 集合的定義 (1.2)。現在，我們要將 $f(x) \in V$ 類比成平面上的一個點 P 。而類比 V_j 是平面上的一條通過原點的直線 Γ 。(通過原點的直線是平面的向量子空間)。有些讀者可能比較喜歡將 $f(x)$ 類比為 \mathbb{R}^3 空間中的一個點，而 V_j 是 x - y 平面。這樣也可以。只是在後面的討論中請自行對應。

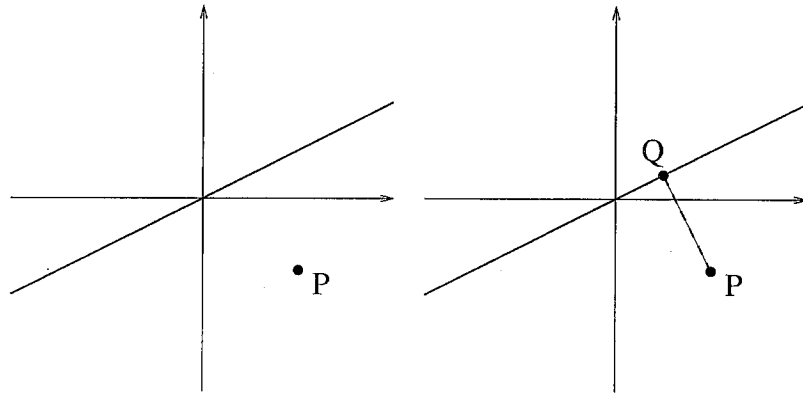


fig:perp 圖 4 垂直投影的圖示

通常 $f(x)$ 不在 V_j 裡面，就好像 P 不在 Γ 上。而 $f_j(x)$ 是 V_j 中最靠近 $f(x)$ 的函數，類比於在 Γ 上找到一點 Q ，而 Q 是 Γ 上最靠近 P 的點。參照圖 4。從平面幾何中，我們知道 \overrightarrow{PQ} 垂直於 Γ ，記做

$$\overrightarrow{PQ} \perp \Gamma$$

而互相垂直的向量的內積是 0，所以

$$\langle \overrightarrow{PQ}, \vec{v} \rangle = 0, \quad \forall \vec{v} \in \Gamma \tag{1.18} \quad \text{eq:pqproj}$$

其中 $\vec{v} \in \Gamma$ 的意義就是所有 Γ 上的向量。類比於 $f(x)$ 與 $f_j(x)$ 的關係，就是

$$f(x) - f_j(x) \perp V_j$$

或者用內積的形式來寫，就是

$$\langle f(x) - f_j(x), v(x) \rangle = 0, \quad \forall v(x) \in V_j \tag{1.19} \quad \text{eq:vproj}$$

就像 Q 是 Γ 中唯一最靠近 P 的點， $f_j(x)$ 是 V_j 中唯一最靠近 $f(x)$ 的函數。

或許讀者沒有見過前述事實在抽象形式下的證明。所以我們示範證明 $f_j(x)$ 是 V_j 中唯一最靠近 $f(x)$ 的函數。如果 $\tilde{f}_j(x) \in V_j$ 也是最靠近 $f(x)$ 的函數，那麼 $\tilde{f}_j(x)$ 也會符合 (1.19)。因此

$$\langle f(x) - \tilde{f}_j(x), v(x) \rangle = 0, \quad \forall v(x) \in V_j$$

所以

$$\begin{aligned} 0 &= \langle f(x) - f_j(x), v(x) \rangle - \langle f(x) - \tilde{f}_j(x), v(x) \rangle \\ &= \langle \tilde{f}_j(x) - f_j(x), v(x) \rangle, \quad \forall v(x) \in V_j \end{aligned}$$

因為 $\tilde{f}_j(x) - f_j(x) \in V_j$ ，所以可以代入 $v(x) = \tilde{f}_j(x) - f_j(x)$ ，得到

$$\langle \tilde{f}_j(x) - f_j(x), \tilde{f}_j(x) - f_j(x) \rangle = \|\tilde{f}_j(x) - f_j(x)\|^2 = 0$$

因此 $\|\tilde{f}_j(x) - f_j(x)\| = 0$ 故 $\tilde{f}_j(x) = f_j(x)$ 。

敏銳的讀者或許要問，如何確定 (1.16) 中定義的長度就真的會是長度呢？這牽涉到什麼是長度的抽象意義，第二章再談。或許又問，連續函數 $f(x)$ 真的可以類比成平面上的一個點嗎？在實用上差不多可以，嚴格來說是不行的。我只是暫時打個馬虎。也是留到第二章再談。嚴肅的讀者或許認為我應該要證明上面所有的敘述。但是幾乎所有高等微積分或工程數學的書裡都有詳細的推導證明，而且上述性質的證明，並不會增加讀者對本書主題的瞭解。所以我們就省略那些證明的篇幅吧。

習題 1.11 請討論什麼情況下 $v(x) \in V_j$ 會是個連續函數。

習題 1.12 一般來說，如果 B 是平面上的一個區域，而非一個向量子空間（亦即，通過原點的直線），而 P 是 B 外的一點。則 B 中到 P 的最近點未必是唯一的。請舉出一個例子。

1.8 多層解析空間

很明顯地， V_0 不足以構成整個 V 空間。而 V_1 的維度比 V_0 多了一倍，因為一片常數函數 ($v_{00}\phi(x)$) 可以被兩片常數函數組成 (只要令 $v_{10} = v_{11} = v_{00}$ 就可以了)。所以明顯可見 $V_0 \subset V_1$ 。同理可推

$$V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_j \subset \cdots$$

因此 V_j 是一層套著一層的多層空間 (就像俄羅斯娃娃)。

其實，從基底的觀點，我們也可以證明 $V_j \subset V_{j+1}$ 。因為

$$\phi(x) = \phi(2x) + \phi(2x - 1) \quad (1.20) \quad \text{eq:scone}$$

我們可以推廣

$$\phi(2^j x - k) = \phi(2^{j+1} x - 2k) + \phi(2^{j+1} x - 2k - 1) \in V_{j+1}$$

而 $\phi(2^j x - k)$ 是 V_j 的基底，既然這些基底都在 V_{j+1} 裡面了，那麼當然整個 V_j 都在 V_{j+1} 裡面。等式 (1.20) 看似簡單得無聊，其實以後會很重要，請對它留下印象。

回顧圖 2 的圖例，我們觀察當 j 越來越大的時候， V_j 的維度越來越高，而 V_j 中的函數就越來越能夠逼近 V 中的函數。換句話說，當 j 越來越大的時候， V_j 空間就越來越靠近 V 空間。數學的術語是說，

$$\bigcup_{j=0}^{\infty} V_j \text{ 稠密於 } V$$

嚴格地說，任給一個 $f(x) \in V$ ，不管要求多小的靠近程度 $\epsilon > 0$ ，必定存在一個 j ，使得 V_j 中有一個函數 $f_j(x)$ 符合要求；也就是

$$\|f(x) - f_j(x)\| < \epsilon$$

22 凌波初步

如果我們設想 $f(x)$ 是自然界的一個波或是訊號，它在每個 $x \in [0, 1]$ 上都對應一個值。反觀 V_j 中的函數，它也可以是一個波（方塊波）或是訊號，但是此時在 $[0, 1]$ 內它只有 2^j 個值，亦即在每段區間 $[\frac{k}{2^j}, \frac{k+1}{2^j}]$ 上對應一個值。在這個意義之下，我們可以說

V 是個無窮解析度的空間， $f(x)$ 具備無窮的解析度，而 V_j 是個有限解析度的空間， $f_j(x)$ 是 $f(x)$ 在有限解析度情況下的最佳寫照。

我們可以設想 V_j 中的基底函數 $\phi(2^j x - k)$ 就是探測的儀器。每當 j 升高一個數字，每個基底函數的寬度縮窄一倍，就好像探測儀器的精確度提高了一倍、同時探測的範圍縮小了一倍；而基底函數的個數增加了一倍，就好像探測儀器的分佈密度增加了一倍。在這個意義下， V_0 是最低解析度的空間， V_1 比它高一倍的解析度， V_2 又比 V_1 提高一倍的解析度；依此類推這些有限解析度的空間越來越精密，所以這些空間一層套著一層 ($V_0 \subset V_1 \subset V_2 \subset \dots$)，解析度越來越高，空間的維度越來越大，距離自然界的無窮解析度越來越靠近。

idx:resol

我們稱空間族 $\{V_j\}$ 為多層解析空間 (*multiresolution space*)。它們將在未來扮演重要的角色。

idx:mra

習題 1.13 證明 (1.20)。

習題 1.14 這一節裡面沒有什麼數學，但是提示的解析度想法是很重要的觀點，請多想一想，多看一眼圖 2。

1.9 投影

因為 (1.18) 我們說 Q 是 P 在 Γ 上的垂直投影，新的名詞是正交投影。(互相垂直的向量又稱為互相正交。) 類比於 (1.19) 我們也可以說

idx:orthproj

$f_j(x)$ 是 $f(x)$ 在 V_j 的正交投影。我們定義正交投影算子 (*orthogonal projection operator*) 為 \mathcal{P}_j :

$$\mathcal{P}_j f(x) \in V_j \text{ 使得 } \langle f(x) - \mathcal{P}_j f(x), v(x) \rangle = 0, \quad \forall v(x) \in V_j$$

所以，我們現在有個新的看法：

在 V_j 中最靠近 $f(x)$ 的函數就是它在 V_j 內的正交投影 $f_j(x) = \mathcal{P}_j f(x)$

我們知道投影算子 \mathcal{P}_j 的值域是 V_j ，但是並不明白它的定義域是什麼？這個問題必須留在第二章解說。

根據 (1.19)，我們可以將 $\mathcal{P}_j f(x)$ 的正交投影性質改寫成

$$\langle \mathcal{P}_j f(x), v(x) \rangle = \langle f(x), v(x) \rangle, \quad \forall v(x) \in V_j \quad (1.21) \quad \text{eq:Pj}$$

我們要闡述這條式子的兩個意義。

第一，我們想像 $f(x)$ 有無窮的解析度。想像一個在 V_j 空間中的生物，他的視覺與知覺只能局限在 V_j 這個有限解析度的空間之內。(我認為人類自己就是在某個有限解析度之內的生物，讀者同意嗎？) 那麼，這個生物所看到的 $f(x)$ 是什麼樣子？什麼是 $f(x)$ 在 V_j 這個解析度之下的表相呢？假設生物的知覺能夠自我調適到最好的接收狀況，那麼它所看到的應該是 $f(x)$ 在 V_j 裡的最佳逼近。也就是說，他應該看到的就是 $\mathcal{P}_j f(x)$ 。為什麼可以假設最好的狀況就是看到 $\mathcal{P}_j f(x)$ 呢？因為

最好的狀況就是，在這個有限解析度空間內的生物，根本無法察覺他所看到的 $\mathcal{P}_j f(x)$ 與真正的 $f(x)$ 有任何差別。

再次利用平面向量的類比，我們知道 $\langle f(x), v(x) \rangle$ 就是 $f(x)$ 在 $v(x)$ 方向上的投影。換個位置來看待，我們可以說是站在 $v(x)$ 的角度來探

測 $f(x)$ 的長度。現在，我們

設想 $\langle f(x), v(x) \rangle$ 就是拿 $v(x)$ 去探測 $f(x)$ 。

那麼 (1.21) 的意義就是說，用盡了 V_j 中所有的探測角度 $v(x)$ ，都無法分辨 $\mathcal{P}_j f(x)$ 與 $f(x)$ 的不同。因此，對於生存而局限在 V_j 解析度空間內的生物 (或電腦) 而言， $\mathcal{P}_j f(x)$ 與 $f(x)$ 可以說是一樣的。這就好像某人被某個謊言矇騙一輩子。他用盡一生的所有可能方式，亦不能發現那是個謊言。那麼對當事人而言，即使是謊言又與真言何異？(我的意思並不是鼓勵大家去試試看；有太多的史實與經驗告訴我們，謊言總是會被拆穿的。)

等式 (1.21) 的第二個意義就是，它提供了一般性的求 $\mathcal{P}_j f(x)$ 的算法。我們從第三節以來，就知道 $\mathcal{P}_j f(x)$ 怎麼算：求積分平均值。但是這個方法只對方盒函數或階梯函數有效。而 (1.21) 所提供的方法是一般性的。讓我們先介紹一般性的方法，然後再拿階梯函數空間來驗證。

令 \mathcal{V} 是一個向量空間， $\langle \cdot, \cdot \rangle$ 代表一個 \mathcal{V} 中的內積。若 S 是 \mathcal{V} 的一個向量子空間， S 的一組基底向量是 u_1, u_2, \dots, u_n 。若 \mathcal{P} 是從 \mathcal{V} 到 S 的正交投影。則任給一 $v \in \mathcal{V}$ ， $\mathcal{P}v$ 滿足

$$\langle \mathcal{P}v, x \rangle = \langle v, x \rangle, \quad \forall x \in S \quad (1.22) \quad \text{eq: vpa}$$

因為 $\mathcal{P}v \in S$ ，所以

$$\mathcal{P}v = \sum_{i=1}^n \alpha_i u_i \quad (1.23) \quad \text{eq: vpb}$$

現在的問題只是如何決定投影係數 α_k 。

讀者應該注意到了，在此我們用 v 表示一個抽象的向量。不同於

第七節中以 \vec{v} 表示一個平面向量。而且，我們用 $\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ 表示平面向量的 x - y 坐標值，用 $\begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ 表示抽象向量的陣列。參閱第二章四節。

將 (1.23) 代入 (1.22)，然後一一將探測向量 x 代入基底向量 u_j ，則 (1.22) 產生 n 個未知數 α_i 的 n 條線性聯立方程組：

$$\left\langle \sum_{i=1}^n \alpha_i u_i, u_j \right\rangle = \sum_{i=1}^n \alpha_i \langle u_i, u_j \rangle = \langle v, u_j \rangle, \text{ for } j = 1, 2, \dots, n$$

寫成矩陣-向量形式，就是

$$\mathbf{A} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} \langle v, u_1 \rangle \\ \langle v, u_2 \rangle \\ \vdots \\ \langle v, u_n \rangle \end{pmatrix} \quad (1.24)$$

eq: vpc

其中 $\mathbf{A} = (a_{ij})$ 是一個 $n \times n$ 的方陣，其元素

$$a_{ij} = \langle u_i, u_j \rangle$$

(由於 v 和 u_i 是已知，所以矩陣和等號右端是常數向量)

只要求解線性系統 (1.24)，就得知投影係數 α_i ，也就得知 $\mathcal{P}v$ 。因為 u_i 是一組基底，也就是說它們線性獨立，或稱線性無關，故矩陣 \mathbf{A} 必定是對稱正定的矩陣，因此必為可逆。所以 (1.24) 必定有唯一解。也就是說，必定有唯一的正交投影 $\mathcal{P}v \in \mathcal{S}$ 。這就是一個一般性的方法。 idx: lininde

習題 1.15 證明 (1.24) 中的 \mathbf{A} 是對稱矩陣。

習題 1.16 令 \mathbf{A} 是 (1.24) 中的矩陣。若

$$x = \sum_{k=1}^n x_k u_k \in \mathcal{S}$$

證明

$$\langle \mathbf{x}, \mathbf{x} \rangle = (x_1, x_2, \dots, x_n) \mathbf{A} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

從這裡，再引用內積的定義（參見第二章），就可以證明 \mathbf{A} 是正定矩陣。

1.10 正則基底

什麼時候線性系統 (1.24) 是最容易求解的呢？最大膽的要求，就是 \mathbf{A} 是單位矩陣 \mathbf{I} 。果真如此，則根本不須解線性系統，直接得到答案

$$\alpha_i = \langle \mathbf{v}, \mathbf{u}_i \rangle$$

也就是說

$$\mathcal{P}\mathbf{v} = \sum_{i=1}^n \langle \mathbf{v}, \mathbf{u}_i \rangle \mathbf{u}_i$$

什麼時候 $\mathbf{A} = \mathbf{I}$ ？那就是當 \mathbf{u}_i 是 S 的一組基底，而且它們滿足

$$\langle \mathbf{u}_i, \mathbf{u}_j \rangle = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases} \quad (1.25) \quad \text{eq:ON}$$

類比於平面向量，內積為 0 的兩個向量稱為彼此垂直，或是用個新名詞叫做正交 (*orthogonal*)。所以彼此正交的基底稱為正交基底。比如說平面上的自然基底 $\vec{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ 和 $\vec{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ 為一組正交基底。而 $\vec{u}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ 和 $\vec{u}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ 也是一組正交基底。

當一組正交基底的每個基底向量本身的長度是 1 的時候，也就是當 $\langle \mathbf{u}_i, \mathbf{u}_i \rangle = \|\mathbf{u}_i\|^2 = 1$ 的時候，我們稱這組基底為正則基底 (*ortho-*

normal basis)。比如前面說的 \vec{e}_1 和 \vec{e}_2 是一組正則基底， $\begin{bmatrix} \cos \theta \\ \sin \theta \end{bmatrix}$ 和 $\begin{bmatrix} -\sin \theta \\ \cos \theta \end{bmatrix}$ 也是一組正則基底。但 \vec{u}_1 和 \vec{u}_2 就不是一組正則基底。 idx:ON

因此 (1.25) 的意思就是說，

如果 S 有一組正則基底 u_1, u_2, \dots, u_n ，則

$$Pv = \sum_{i=1}^n \langle v, u_i \rangle u_i \quad (1.26) \quad \text{eq:Porth}$$

注意，(1.26) 只有對正則基底才成立，不要隨便套用在其他的基底上。

現在我們回頭來考慮階梯函數空間 V_j 。它的一組基底函數是

$$\{\phi(2^j x - k) \mid 0 \leq k \leq 2^j - 1\}$$

當 $m \neq n$ ，因為 $\phi(2^j x - m)$ 和 $\phi(2^j x - n)$ 的函蓋幾乎沒有交集（交集最多只是一個點），所以當然 $\langle \phi(2^j x - m), \phi(2^j x - n) \rangle = 0$ ，亦即 $\{\phi(2^j x - k)\}$ 是一組正交基底。但是

$$\int_{\frac{k}{2^j}}^{\frac{k+1}{2^j}} \phi^2(2^j x - k) dx = \frac{1}{2^j} \int_0^1 \phi^2(x) dx$$

所以 $\|\phi(2^j x - k)\| = \frac{1}{\sqrt{2^j}}$ 。因此 $\{\phi(2^j x - k)\}$ 並非正則基底。

但是，從正交基底轉變到正則基底很簡單，只要把每個基底向量的長度除掉即可。令

$$\phi_{jk}(x) = \sqrt{2^j} \phi(2^j x - k) \quad (1.27) \quad \text{eq:phijk}$$

則 $\{\phi_{jk}(x)\}$ 就是 V_j 的一組正則基底：

$$V_j = \text{span} \{\phi_{jk}(x) \mid 0 \leq k \leq 2^j - 1\}$$

所以，根據 (1.21) 和 (1.26)，我們知道

$$\mathcal{P}_j f(x) = \sum_{k=0}^{2^j-1} \langle f(x), \phi_{jk}(x) \rangle \phi_{jk}(x)$$

而

$$\langle f(x), \phi_{jk}(x) \rangle \phi_{jk}(x) = \left(\int_{\frac{k}{2^j}}^{\frac{k+1}{2^j}} \sqrt{2^j} f(x) dx \right) \sqrt{2^j} = 2^j \int_{\frac{k}{2^j}}^{\frac{k+1}{2^j}} f(x) dx$$

就是 $f(x)$ 在 $[\frac{k}{2^j}, \frac{k+1}{2^j}]$ 內的積分平均值。並無意外，這就是我們在第三節獲致的結論。

我們已經知道，當 j 越來越大的時候， V_j 的解析度越來越高；所以， $\{\phi_{jk}(x)\}$ 這組正則基底也就越來越有效地逼近 $f(x) \in V$ 。問題是，當 $j \rightarrow \infty$ 的時候， $\phi_{jk}(x)$ 變得又細又高，它的函蓋寬度 $\rightarrow 0$ 。可以說，當逼近的程度越來越完美的時候，用以逼近的工具卻變得越來越難以捉摸。所以實際上我們幾乎不能使用 $j \rightarrow \infty$ 的這組正則基底。

$\{\phi_{jk}(x)\}$ 的另外一個應用上的問題是，即使已經計算出來 $\mathcal{P}_j f(x) \in V_j$ ，要計算低一層解析度的 $\mathcal{P}_{j-1} f(x)$ 的時候，還是要重算一遍。這是很不經濟的，因為我們已經知道 $V_{j-1} \subset V_j$ ，所以 $\mathcal{P}_{j-1} f(x) \in V_j$ ，我們應該要能夠從 $\mathcal{P}_j f(x)$ 中得到 $\mathcal{P}_{j-1} f(x)$ 的資料才對。

這兩個問題引導了下一節的發展。

習題 1.17 如果 u_1, u_2, \dots, u_n 是一組正交基底，證明

$$\mathcal{P}v = \sum_{i=1}^n \frac{1}{\|u_i\|^2} \langle v, u_i \rangle u_i$$

習題 1.18 查字典，orthogonal 和 orthonormal 的發音和重音節。

1.18

1.11 Haar 函數

我們知道，若 $\mathcal{P}_0 f(x) = v_{00} \phi(x)$ ， $\mathcal{P}_1 f(x) = v_{10} \phi(2x) + v_{11} \phi(2x - 1)$ ，則

$$v_{00} = \int_0^1 f(x) dx, \quad v_{10} = 2 \int_0^{\frac{1}{2}} f(x) dx, \quad v_{11} = 2 \int_{\frac{1}{2}}^1 f(x) dx$$

所以

$$v_{10} + v_{11} = 2 \left(\int_0^{\frac{1}{2}} + \int_{\frac{1}{2}}^1 f(x) dx \right) = 2 \int_0^1 f(x) dx = 2v_{00}$$

或者說

$$v_{00} = \frac{v_{10} + v_{11}}{2}$$

也就是說 v_{00} 是 v_{10} 和 v_{11} 的平均值。

依理類推，在每一個小區段 $[\frac{k}{2^j}, \frac{k+1}{2^j})$ 之中， $\mathcal{P}_j f(x)$ 是一片常數： v_{jk} 。而 $\mathcal{P}_{j+1} f(x)$ 是兩片常數，左半是

$$v_{j+1,2k} = 2^{j+1} \int_{\frac{2k}{2^{j+1}}}^{\frac{2k+1}{2^{j+1}}} f(x) dx$$

右半是

$$v_{j+1,2k+1} = 2^{j+1} \int_{\frac{2k+1}{2^{j+1}}}^{\frac{2k+2}{2^{j+1}}} f(x) dx$$

這兩個數的平均值是

$$\frac{v_{j+1,2k} + v_{j+1,2k+1}}{2} = 2^j \int_{\frac{k}{2^j}}^{\frac{k+1}{2^j}} f(x) dx = v_{jk} \quad (1.28) \quad \text{eq: vjk}$$

也就是說

低解析度投影係數是高解析度投影係數的平均值。

那麼，高解析度投影係數和低解析度投影係數之間的差異，其和為零。因為，如果

$$v_{j+1,2k} = v_{jk} + \alpha, \quad v_{j+1,2k+1} = v_{jk} + \beta$$

則

$$\frac{v_{j+1,2k} + v_{j+1,2k+1}}{2} = v_{jk} + \frac{\alpha + \beta}{2} = v_{jk}$$

所以 $\alpha + \beta = 0$ ，或是 $\alpha = -\beta$ 。因此

高解析度投影係數和低解析度投影係數之間的差必為零和。

爲了行文與口語化的方便，我們引進一個新名詞：尺度 (*scale*)。以前說的高解析度，就是小尺度，因為每個基底函數的函蓋較窄，探測範圍較小但是精確程度較高；以前說的低解析度，就是大尺度，因為每個基底函數的函蓋較寬，探測範圍較大但是精確程度較低。用這個新名詞，我們把前面的結論在綜合一遍：

idx:scale

大尺度係數是小尺度係數的平均，小尺度係數與大尺度係數之間的差異為零和。

所以，延續前面的討論，令 $w_{jk} = \alpha = -\beta$ ，則

$$w_{jk} = v_{j+1,2k} - v_{jk} = v_{jk} - v_{j+1,2k+1}$$

或者

$$w_{jk} = \frac{v_{j+1,2k} - v_{j+1,2k+1}}{2} \quad (1.29) \quad \text{eq:wjk}$$

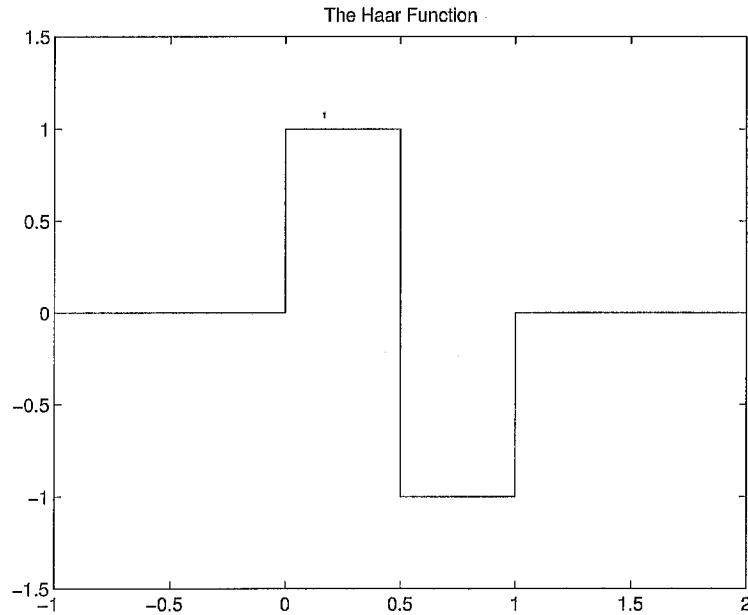


fig:haar 圖 5 Haar 函數

則在區間 $[\frac{k}{2^j}, \frac{k+1}{2^j})$ 之中，

$$\mathcal{P}_{j+1}f(x) = \mathcal{P}_j f(x) + \sqrt{2^j} w_{jk} \times \begin{cases} 1 & \text{for } x \in [\frac{2k}{2^{j+1}}, \frac{2k+1}{2^{j+1}}) \\ -1 & \text{for } x \in [\frac{2k+1}{2^{j+1}}, \frac{2k+2}{2^{j+1}}) \end{cases} \quad (1.30) \quad \text{eq:haarjk}$$

現在，我們定義一個新的函數

$$\psi(x) = \begin{cases} 1 & \text{for } x \in [0, \frac{1}{2}) \\ -1 & \text{for } x \in [\frac{1}{2}, 1) \end{cases} \quad (1.31) \quad \text{eq:Haar}$$

考慮 $\psi(x)$ 的脹縮平移

$$\psi(2^j x - k) = \begin{cases} 1 & \text{for } x \in [\frac{2k}{2^{j+1}}, \frac{2k+1}{2^{j+1}}) \\ -1 & \text{for } x \in [\frac{2k+1}{2^{j+1}}, \frac{2k+2}{2^{j+1}}) \end{cases}$$

並且仿照 (1.27) 令

$$\psi_{jk}(x) = \sqrt{2^j} \psi(2^j x - k) \quad (1.32) \quad \text{eq:psijk}$$

則 (1.30) 就可以用函數的形式寫出來：

$$\mathcal{P}_{j+1}f(x) = \mathcal{P}_j f(x) + w_{jk} \psi_{jk}(x), \text{ for } x \in \left[\frac{k}{2^j}, \frac{k+1}{2^j} \right)$$

換一套符號，我們可以寫

$$\begin{aligned} \mathcal{P}_{j+1}f(x) \Big|_{\left[\frac{2k}{2^{j+1}}, \frac{2k+2}{2^{j+1}} \right)} &= v_{j+1,2k} \phi_{j+1,2k}(x) + v_{j+1,2k+1} \phi_{j+1,2k+1}(x) \\ &= v_{jk} \phi_{jk}(x) + w_{jk} \psi_{jk}(x) \end{aligned} \quad (1.33) \quad \text{eq:Philo}$$

函數 $\psi(x)$ 並不算是新發明，因為在 1910 年就被 Haar 探討過。所以我們稱它為 *Haar* 函數。參見圖 5。注意 Haar 函數是不連續函數，圖 5 中的垂直線不應該在那裡，它只是為了視覺上的好看。（ ψ 是一個希臘字母，讀 psai 或 psee，用英文字母拼寫成 psi）。

利用方盒函數 $\phi(x)$ ，Haar 函數可以寫成

$$\psi(x) = \phi(2x) - \phi(2x - 1) \quad (1.34) \quad \text{eq:Haarpsi}$$

因此 $\psi(x) \in V_1$ 。但是它和 $\phi(x)$ 不同的是 $\int_0^1 \psi dx = 0$ 。很容易可以證明 $\{\psi_{jk}(x) \mid 0 \leq k \leq 2^j - 1\}$ 是個正則集合。令

$$W_j = \text{span} \{ \psi(2^j x - k) \mid 0 \leq k \leq 2^j - 1 \}$$

則 $\{\psi_{jk}(x)\}$ 是 W_j 的一組正則基底。

習題 1.19 證明 $\int_0^1 \psi(x) dx = 0$ 、 $\int_{-\infty}^{\infty} \psi(2^j x - k) dx = 0$ 。

習題 1.20 證明 $\{\psi_{jk}(x) \mid 0 \leq k \leq 2^j - 1\}$ 是一個正則集合。也就是說，當 $m \neq n$ ， $\langle \psi_{jm}, \psi_{jn} \rangle = 0$ ，而 $\|\psi_{jk}\| = 1$ 。

1.12 多層解析空間的基底變換

因為 $\psi(x) \in V_1$ ，可以推廣為 $\psi_{jk}(x) \in V_{j+1}$ ，而 W_j 是由 ψ_{jk} 所展成的空間，所以 W_j 中的函數都會在 V_{j+1} 。就是 $W_j \subset V_{j+1}$ 。另一方面，由多層解析空間的特性，我們也知道 $V_j \subset V_{j+1}$ 。更美好的是，從 (1.33) 我們得知 V_{j+1} 中的函數都可以分解成一個 V_j 中的函數加一個 W_j 中的函數。

讓我們先看簡單的例子。考慮 V_0 、 W_0 和 V_1 空間。已知 $\dim V_0 = 1$ ， $\dim W_0 = 1$ 和 $\dim V_1 = 2$ 。將 V_1 類比於坐標平面，而 V_0 類比於一條通過原點的直線，就假設它是 x -軸吧。 W_0 是另一個向量子空間，所以它是平面上另一條通過原點的直線。但是觀察

$$\langle \phi(x), \psi(x) \rangle = 0$$

所以 $\phi \perp \psi$ ，可見 $V_0 \perp W_0$ 。所以如果 V_0 是 x -軸，那麼 W_0 就應該是 y -軸。而將 x -軸和 y -軸上的向量相加，正好就是整個坐標平面：

$$\mathbb{R}^2 \ni \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ y \end{bmatrix} \in \mathbb{R}_x + \mathbb{R}_y$$

其中 \mathbb{R}_x 表示 x -軸向量空間， \mathbb{R}_y 表示 y -軸向量空間。用線性代數符號，寫成

$$\mathbb{R}^2 = \mathbb{R}_x \oplus \mathbb{R}_y$$

而 \oplus 中的 $+$ 表示每一個 \mathbb{R}^2 中的向量都可以寫成一個 \mathbb{R}_x 中的向量加上一個 \mathbb{R}_y 中的向量，也就是說 \mathbb{R}^2 可以拆成兩個向量子空間 \mathbb{R}_x 和 \mathbb{R}_y 的和。而 \oplus 的圈圈表示 \mathbb{R}_x 和 \mathbb{R}_y 是互相垂直的兩個向量空間。我們稱 \mathbb{R}^2 是 \mathbb{R}_x 和 \mathbb{R}_y 的直和。

任意取一個 $v(x) \in V_1$ ，若

$$v(x) = v_{10}\phi(2x) + v_{11}\phi(2x-1), \quad v_{10}, v_{11} \in \mathbb{R}$$

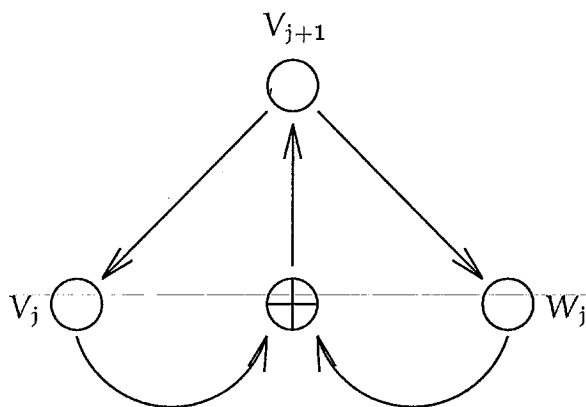


fig:Vone 圖 6 兩層尺度間的分解與合成圖示

我們取

$$v_{00} = \frac{v_{10} + v_{11}}{2}, \quad w_{00} = \frac{v_{10} - v_{11}}{2} \quad (1.35) \quad \text{eq:decompA}$$

則

$$\begin{aligned} & v_{00}\phi(x) + w_{00}\psi(x) \quad (\in V_0 + W_0) \\ &= \frac{v_{10} + v_{11}}{2} (\phi(2x) + \phi(2x-1)) + \frac{v_{10} - v_{11}}{2} (\phi(2x) - \phi(2x-1)) \\ &= v_{10}\phi(2x) + v_{11}\phi(2x-1) \\ &= v(x) \quad (\in V_1) \end{aligned}$$

根據 (1.28) 和 (1.29)，我們可以推廣上述的分解： V_{j+1} 可以分解成 V_j 和 W_j 兩部份，而 V_j 和 W_j 是互相垂直的兩個子空間。(兩個向量空間 $S \perp T$ 的意思是，任取 $u \in S, v \in T$ ，則 $u \perp v$ 。) 用線性代數的符號，寫成 idx:perp

$$V_{j+1} = V_j \oplus W_j \quad (1.36) \quad \text{eq:Vjone}$$

讀者應該檢查 $\dim V_j + \dim W_j = 2^j + 2^j = 2^{j+1} = \dim V_{j+1}$ 。

注意 (1.35)，它很容易可以被推廣：若 $v(x) \in V_{j+1}$ ，而

$$v(x) = \sum_{k=0}^{2^{j+1}-1} v_{j+1,k} \phi(2^{j+1}x - k)$$

則 $v(x)$ 可以被分解 (*decompose or analyze*) 成爲

idx:decomp

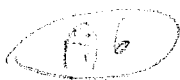
$$v(x) = u(x) + w(x), \quad u(x) \in V_j, w(x) \in W_j$$

$$u(x) = \sum_{k=0}^{2^j-1} u_{jk} \phi(2^j x - k), \quad \text{而 } u_{jk} = \frac{v_{j+1,2k} + v_{j+1,2k+1}}{2}$$

其中

$$w(x) = \sum_{k=0}^{2^j-1} w_{jk} \psi(2^j x - k), \quad \text{而 } w_{jk} = \frac{v_{j+1,2k} - v_{j+1,2k+1}}{2}$$

(1.37) eq:decomp



反過來，如果知道

$$u(x) = \sum_{k=0}^{2^j-1} u_{jk} \phi(2^j x - k) \in V_j$$

$$w(x) = \sum_{k=0}^{2^j-1} w_{jk} \psi(2^j x - k) \in W_j$$

則可以合成 (*synthesis*)，或說重建 (*reconstruct*) 一個 $v(x) \in V_{j+1}$ ：

idx:recons

$$v(x) = \sum_{k=0}^{2^{j+1}-1} v_{j+1,k} \phi(2^{j+1}x - k) \tag{1.38}$$

$$\text{其中 } v_{j+1,k} = \begin{cases} u_{jn} + w_{jn} & \text{if } k = 2n \\ u_{jn} - w_{jn} & \text{if } k = 2n + 1 \end{cases}$$

eq:reconst

←請參照圖 6 的圖示。→

由 (1.37) 和 (1.38)，當我們知道了 $\mathcal{P}_{j+1}f(x)$ ，不需要重新計算低解析度的 $\mathcal{P}_j f(x)$ ，從 $\mathcal{P}_{j+1}f(x)$ 就可以得知 $\mathcal{P}_j f(x)$ 。一般來說，我們不可能從低解析度去產生高解析度的訊號。但是如果在分解的過程中保留住大尺度與小尺度之間的差異（也就是 w_{jk} ），就可以從低解析度還原到高解析度的訊號。

這分解與合成的過程，其實就是線性代數中的基底變換而已。因為 (1.36)，所以 V_j 中的 (正交、正則) 基底聯合 W_j 中的 (正交、正則) 基底，就是 V_{j+1} 中的 (正交、正則) 基底：

idx:ch-basis

$$\begin{aligned} V_{j+1} &= \text{span} \{ \phi(2^{j+1}x' - k) \mid 0 \leq k \leq 2^{j+1} - 1 \} \\ &= \text{span} \left(\{ \phi(2^j x - k) \mid 0 \leq k \leq 2^j - 1 \} \right. \\ &\quad \left. \cup \{ \psi(2^j x - k) \mid 0 \leq k \leq 2^j - 1 \} \right) \end{aligned}$$

在這個意義之下，將 $v(x) \in V_{j+1}$ 從 $\{\phi_{j+1,k}(x)\}$ 的基底變換到 $\{\phi_{jk}(x)\} \cup \{\psi_{jk}(x)\}$ 的基底，就是分解；反過來，從 $\{\phi_{jk}(x)\} \cup \{\psi_{jk}(x)\}$ 的基底變換到 $\{\phi_{j+1,k}(x)\}$ 的基底，就叫做合成。而不論分解還是合成都是應用時的術語，在數學的眼光中，它們是同一個向量用不同的基底來展開。

分解了 $V_{j+1} = V_j \oplus W_j$ 之後，可以再分解 $V_j = V_{j-1} \oplus W_{j-1}$ ，依此類推。

$$\begin{aligned} V_{j+1} &= W_j \oplus (W_{j-1} \oplus V_{j-1}) \\ &\quad \dots \\ &= W_j \oplus W_{j-1} \oplus \dots \oplus W_0 \oplus V_0 \end{aligned}$$

所以，變換基底的過程也可以往大尺度 (或低解析度) 持續進行。分解到最後，相當於

$$\phi_{00}, \psi_{10}, \psi_{11}, \psi_{20}, \psi_{21}, \psi_{22}, \psi_{23}, \dots, \psi_{j0}, \psi_{j1}, \dots, \psi_{j,2^j-1}$$

是 V_{j+1} 的一組正則基底。而對所有的 $v(x) \in V_\ell$,

$$\begin{aligned} v(x) &= \sum_{k=0}^{2^\ell-1} v_k \phi_{\ell k}(x) \\ &= u_0 \phi(x) + \sum_{j=0}^{\ell-1} \sum_{k=0}^{2^j-1} w_{jk} \psi_{jk}(x) \end{aligned}$$

習題 1.21 證明 $V_j \perp W_j$ 。

習題 1.22 證明 (1.37) 和 (1.38)。

習題 1.23 證明 (1.37) 中的 w_{jk} 就是 $\sqrt{2^j} \langle f(x), \psi_{jk}(x) \rangle$ 。

1.13 凌波函數

究竟什麼是凌波函數？當我們有一組多層解析空間

$$V_0 \subset V_1 \subset V_2 \subset \cdots \subset V$$

而當 $j \rightarrow \infty$ 的時候 V_j 會稠密於 V 內。如果每一尺度的 V_{j+1} 可以分解成 $V_j \oplus W_j$ ，又如果存在一個函數 $\psi(x)$ 在某個尺度的平穩可以成爲 W_j 的正則基底。則 $\psi(x)$ 必將符合 (以後才證明) *see 35, 40*

$$\int_{-\infty}^{\infty} \psi(x) dx = 0$$

所以它的圖形一定上下起伏，如波狀；又因爲

$$\|\psi(x)\|^2 = \int_{-\infty}^{\infty} \psi(x)^2 dx = 1$$

所以 $|\psi(x)|$ 底下所圍的面積必然不會很大*。因此，它的圖形看起來就像一個小小的波浪（參照圖 5 Haar 函數）。這個 $\psi(x)$ ，法國人命名為 *ondelette*，英文就直譯成 *wavelet*，中文的直譯就是小波。

idx:small-wav

但是，作者本人覺得直譯小波不夠雅緻。近代中國的科學名詞多半譯自西方。有些經過細想的譯名，既合中國語言習慣，又能表達數學本體的含義。例如 *differentiation* 翻作微分，*integration* 翻作積分，而 *calculus* 卻合作微積分；又如 *conjugate* 翻作共軛，*fractal* 翻作碎形。我提議將 *wavelet* 翻作凌波。根據台灣三民書局在 1989 重新編輯的辭典，凌有八個意義，其中：(4) 雜亂，交錯；例如凌亂 (6) 迫近，接近；例如凌晨 這兩個意義很接近 *wavelets* 的圖形及應用。而且凌波也是個古典的形容詞。這本辭典列了兩個意義：(1) 起伏的波濤，(2) 形容輕盈的步履**。

idx:wavelet

作者在序言中明白表示（我知道很多人從來不讀序言），寫這本書的三個目的之一就是提倡這個譯名。

現在我們知道，Haar 函數就是一個凌波函數。到目前為止，我們也只知道這麼一個凌波函數。我們將會看到，Haar 函數是許多凌波函數族類中最基礎的一個，我們稱之為一階凌波函數。所謂許多凌波函數族類，包括正交（這本書的主題）、雙正交、半正交、區間中的凌波函數等等。這些族類的凌波函數各是一系列的函數，隨著一個所謂的階數 p 而變化。當 $p = 1$ 的時候它們都是 Haar 函數。

idx:wavelets-

* 爲了實用，我們希望 $\psi(x)$ 的函蓋是個有限長的區間。即使無法如願，退而求其次也希望 $\psi(x)$ 是指數消散 (*exponential decay*)，但是這本書並不打算包含這類課題。

** 除了前面所說的那一些冠冕堂皇的理由之外，我必須承認，凌波這個譯名，乃至於這本書的書名，有一大部分受到金庸武俠小說裡面一種優雅武功的啓發：凌波微步。

Haar

1.14 凌波係數的消散定理

以前我們曾說，當 $j \rightarrow \infty$ ，固然 V_j 是越來越靠近 V ，但是 ϕ_{jk} 的函蓋寬度 $\rightarrow 0$ ，所以實際上我們並不能使用這套基底。現在我們知道 V_j 有了另外一套正則基底 $\psi_{jk}(x)$ ，它就比較實用了。雖然當 $j \rightarrow \infty$ 的時候， ψ_{jk} 的函蓋寬度還是 $\rightarrow 0$ ，但現在我們是漸次使用 ψ_{jk} 基底。我們從寬的基底漸漸用到細的基底。換句話說，我們從大尺度漸漸逼近到小尺度，從低解析度漸漸逼近到高解析度。實際上我們不能真的做到無窮解析度，但是當解析的層次 j 漸漸提高的時候，我們漸漸添上寬度越來越窄的基底函數 ψ_{jk} ，使得整個的逼近程度漸趨於 0。

讓我們使用 ψ_{jk} 基底將 V_j 的逼近理論 (1.8) 改寫成

idx:app

$$\lim_{\ell \rightarrow \infty} \left\| f(x) - \left(\langle f, \phi_{00} \rangle \phi_{00}(x) + \sum_{j=0}^{\ell} \sum_{k=0}^{2^j-1} \langle f, \psi_{jk} \rangle \psi_{jk}(x) \right) \right\| = 0 \quad (1.39)$$

eq:xpsim

或是將含大 O 符號的 (1.9) 寫成

$$\langle f, \phi_{00} \rangle \phi_{00}(x) + \sum_{j=0}^{\ell} \sum_{k=0}^{2^j-1} \langle f, \psi_{jk} \rangle \psi_{jk}(x) = f(x) + O\left(\frac{1}{2^\ell}\right) \quad (1.40)$$

eq:xpsio

以前我們利用方盒函數的特性，求得 (1.40) 的逼近理論。那是一個特殊的情況，只對方盒函數有效。在這一節裡，我們要用另一個方法來引介 (1.40)。這個方法將利用到凌波函數 (就目前而言，就是 Haar 函數) 的消散性質。而這種方法將是比較一般性的，以後還可以再用。所謂凌波函數的消散性質，一般來說就是當 $f(x)$ 夠平滑的時候，它的凌波投影係數會隨著解析度的提高 (j 變大，尺度變小) 而逐漸變小，最後趨近於 0。下面是一個明確的敘述。

idx:vanish

Haar 凌波係數消散定理 令 $\psi(x)$ 為 Haar 函數。則若 $f(x) \in C^1([0, 1])$ idx: thmHaar

$$|\langle f(x), \psi_{jk}(x) \rangle| = O(2^{-\frac{3}{2}j})$$

證明：因為 $\text{supp } \psi_{jk} = [\frac{k}{2^j}, \frac{k+1}{2^j}]$ ，我們選擇將 $f(x)$ 以 $\frac{k}{2^j}$ 為參考點做一階泰勒展開：

$$f(x) = f\left(\frac{k}{2^j}\right) + f'(\xi)\left(x - \frac{k}{2^j}\right), \quad x \in \left[\frac{k}{2^j}, \frac{k+1}{2^j}\right]$$

其中 ξ 是某個在 $(\frac{k}{2^j}, \frac{k+1}{2^j})$ 內的數。因為 $f'(x)$ 是連續函數，我們令

$$M_{jk} = \max_{\xi \in (\frac{k}{2^j}, \frac{k+1}{2^j})} f'(\xi)$$

利用 $\int_0^1 \psi_{jk}(x) dx = 0$ 的特性，推導

$$\begin{aligned} & \langle f(x), \psi_{jk}(x) \rangle \\ &= \sqrt{2^j} \int_{\frac{k}{2^j}}^{\frac{k+1}{2^j}} \left(f\left(\frac{k}{2^j}\right) + f'(\xi)\left(x - \frac{k}{2^j}\right) \right) \psi(2^j x - k) dx \\ &= \sqrt{2^j} \int_{\frac{k}{2^j}}^{\frac{k+1}{2^j}} f'(\xi)\left(x - \frac{k}{2^j}\right) \psi(2^j x - k) dx \end{aligned} \tag{1.41}$$

eq: xpv

因此，

$$\begin{aligned} & |\langle f(x), \psi_{jk}(x) \rangle| \\ & \leq \sqrt{2^j} M_{jk} \int_{\frac{k}{2^j}}^{\frac{k+1}{2^j}} \left| x - \frac{k}{2^j} \right| |\psi(2^j x - k)| dx \\ & \leq \sqrt{2^j} M_{jk} \frac{1}{2^j} \int_{\frac{k}{2^j}}^{\frac{k+1}{2^j}} 1 dx \\ & = \sqrt{2^j} M_{jk} \frac{1}{2^j} \cdot \frac{1}{2^j} = 2^{-\frac{3}{2}j} M_{jk} \end{aligned} \quad \square$$

凌波消散定理有很重要的意義，幾乎所有凌波函數的應用都以這個性質為基礎。我們已經知道 W_j 空間就是低層解析度 V_j 與高層解析度 V_{j+1} 之間的差異。直覺上，如果 j 夠大的時候，即使是低層的 V_j 也有足夠高的解析度。因此 V_j 和 V_{j+1} 之間的差異就應該很小。而凌波消散定理告訴我們，這個直覺是正確的，

當 j 越來越大，則 V_j 與 V_{j+1} 之間相差的細節部分，就越來越小。

而且它明確地描述一個定量的關係。

在定理的證明過程中，我們還發現 $|\langle f(x), \psi_{jk}(x) \rangle|$ 的估計是局部性的。如果 $f(x)$ 在 $[\frac{k}{2^j}, \frac{k+1}{2^j}]$ 中有劇烈起伏， M_{jk} 比較大，所以 $|\langle f, \psi_{jk} \rangle|$ 就會比較大；當 $f(x)$ 在 $[\frac{k}{2^j}, \frac{k+1}{2^j}]$ 中平緩的時候， $|\langle f, \psi_{jk} \rangle|$ 就會比較小。明確地說，如果 $f(x)$ 在 $[\frac{k}{2^j}, \frac{k+1}{2^j}]$ 內的最大斜率是 M_{jk} ，則

$$v_{jk}\phi(2^j x - k) \quad \text{和} \quad v_{j+1, 2k}\phi(2^{j+1}x - 2k) + v_{j+1, 2k+1}\phi(2^{j+1}x - 2k - 1)$$

之間相差的細節就是

$$w_{jk}\psi(2^j x - k) \quad \text{其中} \quad |w_{jk}| = \sqrt{2^j} |\langle f, \psi_{jk} \rangle| \leq \frac{1}{2^j} M_{jk}$$

所以，從 w_{jk} 或是 $\langle f, \psi_{jk} \rangle$ 的大小，我們可以探測 $f(x)$ 的局部變化平緩或劇烈的程度。訊號處理中的影像壓縮 (image compression) 和邊界探測 (edge detection) 都是這個定理的應用。參照後面的圖 7。

現在，我們示範如何從凌波消散定理推導多層解析空間的逼近理論 (1.40)。根據 V_j 在 V 中的稠密性，知道

idx:dense

$$f(x) = \langle f, \phi_{00} \rangle \phi_{00}(x) + \sum_{j=0}^{\infty} \sum_{k=0}^{2^j-1} \langle f, \psi_{jk} \rangle \psi_{jk}(x)$$

所以

$$\begin{aligned} & \left\| f(x) - \left(\langle f, \phi_{00} \rangle \phi_{00}(x) + \sum_{j=0}^{\ell} \sum_{k=0}^{2^j-1} \langle f, \psi_{jk} \rangle \psi_{jk}(x) \right) \right\| \\ &= \left\| \sum_{j=\ell+1}^{\infty} \sum_{k=0}^{2^j-1} \langle f, \psi_{jk} \rangle \psi_{jk}(x) \right\| \end{aligned} \quad (1.42)$$

eq: xva

由於 $\|\cdot\|$ 符合三角不等式，所以

$$(1.42) \text{ 式} \leq \sum_{j=\ell+1}^{\infty} \left\| \sum_{k=0}^{2^j-1} \langle f, \psi_{jk} \rangle \psi_{jk}(x) \right\| \quad (1.43)$$

eq: xvb

對所有的 $x \in (0, 1)$ ，令 $M = \max f'(x)$ ，所以 $M_{jk} \leq M$ 。(這時候我們要推導的是在整個 $[0, 1]$ 內的逼近理論，所以只好放棄比較精細的局部估計 M_{jk} 了。) 因此

$$\left| \sum_{k=0}^{2^j-1} \langle f, \psi_{jk} \rangle \psi_{jk}(x) \right| \leq \frac{M}{\sqrt{2^j}} \sum_{k=0}^{2^j-1} |\psi_{jk}(x)| = \frac{M}{2^j} \sum_{k=0}^{2^j-1} |\psi(2^j x - k)|$$

所以

$$(1.43) \text{ 式} \leq \sum_{j=\ell+1}^{\infty} \frac{M}{2^j} \left\| \sum_{k=0}^{2^j-1} |\psi(2^j x - k)| \right\| = \sum_{j=\ell+1}^{\infty} \frac{M}{2^j} \quad (1.44)$$

eq: xvc

求上式中的等比級數和，就得到

$$(1.42) \text{ 式} \leq M \frac{1}{2^\ell}$$

並無意外，這個結果和以前的 (1.7) 相同。

由此可知，凌波函數的消散性質可以推導出多層解析空間 $\{V_j\}$ 的逼近理論。如同前面說過的，僅僅是 $O(2^{-j})$ 的逼近能力，其實不甚令人滿意。我們還希望能有 $O(2^{-2j})$ 甚至 $O(2^{-3j})$ 或更好的逼近能力。以後會看到，一般來說，如果凌波函數的消散性質可達 $O(2^{-(p+\frac{1}{2})j})$ ，則其對應的多層解析空間可達 $O(2^{-pj})$ 的逼近能力。而從 Haar 函數消散定理的證明中可以看出來，導至凌波投影係數會消散的原因是 (1) $f(x)$ 本身要足夠平滑，使得高階的泰勒展開式存在；(2) $\psi(x)$ 正交於某些階數的多項式，使得泰勒展開中的多項式會被消除掉，如同 (1.41) 中的作法。例如 Haar 函數 (一階凌波函數) 就是正交於常數函數 (零階多項式)。我們將在第三章中推廣這些理論。

習題 1.24 若 $\|f(x)\| < \infty$, $\alpha \in \mathbb{R}$, 證明

$$\| |f(x)| \| = \|f(x)\|, \quad \|\alpha f(x)\| = |\alpha| \|f(x)\|$$

習題 1.25 證明 (1.44) 中的

$$\left\| \sum_{k=0}^{2^j-1} |\psi(2^j x - k)| \right\| = 1$$

習題 1.26 在 (1.43) 中，我們也可以利用三角不等式得到

$$(1.42) \text{ 式} \leq \sum_{j=\ell+1}^{\infty} \sum_{k=0}^{2^j-1} |\langle f, \psi_{jk} \rangle| \|\psi_{jk}(x)\|$$

請討論，如果這樣做會有什麼結果？

1.15 數位訊號處理

當我們取得一筆數位訊號，假設是 N 個數字。我們不去討論這筆訊號是如何得到的，是聲波、雷達、地震波、股市記錄還是心電圖。總之這筆訊號已經輸入電腦，以一串數字的形態儲存；稱為數位訊號。而所謂數位訊號處理 (*DSP: Digital Signal Processing*) 就是針對這串數字玩些把戲。例如壓縮訊號，使得它雖然在合成之後產生些許的誤差，但是誤差不大而可以大量節省儲存的空間與傳輸的時間；例如在訊號之間安插外來的訊息，像暗藏密碼或作者的著作權簽名；例如從衛星掃射海面的雷達波偵測船隻甚至潛水艇的行蹤；例如企圖從微弱的地震波中獲得地震來臨的警訊、從股市記錄中預測下一次的漲跌、從心電圖中診斷心臟的病變。(在 1980 初期，最初研究凌波這種函數的是一群研究地震波的法國學者。)

idx:dsp

在數學符號上，我們將它寫成一個序列

$$\mathbf{v} = (v_1, v_2, \dots, v_N) \in \mathbb{R}^N$$

通常由儀器記錄的訊號，是按照時間的順序記錄振幅的大小。對應在數學上，假設 \mathbf{v} 乃是由自然基底所合成：

idx:Nbasis

$$\mathbf{v} = v_1 \vec{e}_1 + v_2 \vec{e}_2 + \dots + v_N \vec{e}_N$$

其中

$$\vec{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \vec{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \dots, \vec{e}_N = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

同一個向量，在不同的基底之下會有不同的係數。我們企圖從一組基底所表現的係數中，探知在其他基底中無法突顯出來的性質。所以一般來說，

訊號處理的工作，第一步總是做某種的基底變換。

在凌波之前，研究人員已經掌握了幾種基底變換的工具，例如傅立葉轉換和 Z 轉換。從大約 1988 年開始，凌波函數提供了一種新的工具。這個工具，因為其性質豐富而且合於物理原理，所以受到廣泛的重視。我們將會在後面幾章談論凌波的性質。現在，我們先定義凌波轉換 (*wavelet transform*)：

idx:dwt

將 V_j 中的函數從 $\{\phi_{jk}(x)\}$ 基底變換到 $V_{j-1} \oplus W_{j-1}$ 的基底 $\{\phi_{j-1,k}(x)\} \cup \{\psi_{j-1,k}(x)\}$ ，此過程稱為凌波轉換。

有時候，針對數位訊號處理，我們也稱它為離散凌波轉換 (*DWT: discrete wavelet transform*)。

idx:dwt

凌波函數是怎麼和訊號處理搭上線的呢？為了數學理論推演，我們常常需要在函數空間內討論。類似 Haar 逼近定理就需要在 $C^1([0, 1])$ 中討論。所以，我們要暗自假設數位訊號 $v = (v_k)$ 來自於某個函數 $f(x)$ 。這個 $f(x)$ 具有無窮解析度。而我們假定取得的數位訊號落在某個有限解析度空間 V_j 內。再假定 v_k 就是 $\mathcal{P}_j f(x)$ 的投影係數：

$$\sum_k v_k \phi(2^j x - k) = \sum_k \sqrt{2^j} \langle f, \phi_{jk} \rangle \phi(2^j x - k)$$

由於這個關聯，凌波函數就從這裡帶進來了。

通常，當我們談訊號 (*signal*)，指的是一維的訊號，例如聲波。其實這個名詞可以泛指一維和二維的訊號。二維訊號就是影像 (*image*)。

我們在電腦螢幕上看到的所有影像都是數位影像。它們是以二維序列的資料結構儲存。而數學上，我們將數位影像寫成一個矩陣。可參考附錄 A.3 的 Matlab image 指令。凌波轉換也可以應用到數位影像上。最簡單的方法，就是將矩陣的行與列（這就變成了一維訊號）分別做凌波轉換。還有其他二維凌波轉換的方法，但是暫且不談。

一個二維訊號在數學符號上以矩陣 $V = (v_{km})$ 表達。爲了數學理論的推演，我們也暗自假設數位影像來自某個雙變數函數 $f(x, y)$ 。這個 $f(x, y)$ 具有無窮解析度。而我們假定取得的數位訊號落在某個有限解析度空間 $V_j(x, y)$ 內。顯然我們現在應該談談二維的階梯函數空間 $V_j(x, y)$ 。在這本書裡，我們不會經常探討二維的理論問題；至於二維的演算法亦不見得需要二維理論做基礎。所以我們只打算略述二維的階梯函數空間。

我們採用笛卡兒坐標的想法定義 $V_j(x, y)$ 空間：它是正方形 $[0, 1] \times [0, 1]$ 區域內的階梯函數空間。令

$$V_j(x, y) = \text{span} \{ \phi(2^j x - k) \phi(2^j y - m) \mid 0 \leq k, m \leq 2^j - 1 \}$$

例如 $V_0(x, y)$ 就是由一片常數函數 $\phi(x)\phi(y)$ 生成的，而 $V_1(x, y)$ 是由四片常數函數

$$\phi(2x)\phi(2y), \phi(2x-1)\phi(2y), \phi(2x)\phi(2y-1), \phi(2x-1)\phi(2y-1)$$

生成的。若 $v(x, y) \in V_j(x, y)$ 則

$$\begin{aligned} v(x, y) = & v_{1,00}\phi(2x)\phi(2y) + v_{1,10}\phi(2x-1)\phi(2y) \\ & + v_{1,01}\phi(2x)\phi(2y-1) + v_{1,11}\phi(2x-1)\phi(2y-1) \end{aligned}$$

其中 $v_{1,00}, v_{1,10}, v_{1,01}, v_{1,11} \in \mathbb{R}$ 。像這樣定義的 $V_j(x, y)$ 又可以寫成

$$V_j(x) \times V_j(y)$$

一般的情況下， $V_j(x) \times V_j(y)$ 的一組正則基底是

$$\phi_{j,km} = 2^j \phi(2^j x - k) \phi(2^j y - m), \quad \text{for } 0 \leq k, m \leq 2^j - 1 \quad (1.45) \quad \text{eq:phitwod}$$

我們仍然假定 v_{km} 就是 $\mathcal{P}_j f(x, y)$ 的投影係數：

$$\sum_{k=0}^{2^j-1} \sum_{m=0}^{2^j-1} v_{km} \phi_{j,km}(x) = \sum_{k=0}^{2^j-1} \sum_{m=0}^{2^j-1} \langle f, \phi_{j,km} \rangle \phi_{j,km}(x)$$

習題 1.27 若

$$\phi(2^j x - k) \phi(2^j y - m) = \begin{cases} 1 & \text{if } (x, y) \in \mathcal{B} \\ 0 & \text{otherwise} \end{cases}$$

請決定 \mathcal{B} 區域。

習題 1.28 證明 (1.45) 中定義的 $\phi_{j,km}(x)$ ，當固定一個 j 的時候，對 k, m 而言是一組正則集合。

1.16 DWT: 離散凌波轉換

已知

$$V_j = W_{j-1} \oplus W_{j-2} \oplus \cdots \oplus W_0 \oplus V_0$$

但是只要我們知道如何在兩層之間做凌波轉換 (分解與合成)

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

其他的步驟就可以遞迴執行 (參照圖 6 的圖示)。

假設數位訊號是

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

爲了行文的方便，我們令 $N = 2^j$ (不足的部分補零)，則有

$$\begin{aligned} V_j &\ni \sum_{k=0}^{N-1} v_k \phi(2^j x - k) \\ &= \sum_{k=0}^{\frac{N}{2}-1} u_k \phi(2^{j-1} x - k) + \sum_{k=0}^{\frac{N}{2}-1} w_k \psi(2^{j-1} x - k) \in V_{j-1} \oplus W_{j-1} \end{aligned}$$

離散凌波轉換就是要將 $\{v_k\} \mapsto \{u_k, w_k\}$ 和 $\{u_k, w_k\} \mapsto \{v_k\}$ 。從 (1.37) 明顯可知，若 v_k 爲已知，則

$$\begin{aligned} u_k &= \frac{1}{2}(v_{2k} + v_{2k+1}) \quad (\text{求平均值}) \\ w_k &= \frac{1}{2}(v_{2k} - v_{2k+1}) \quad (\text{求平均差}) \end{aligned} \quad 0 \leq k \leq \frac{N}{2} - 1 \quad (1.46)$$

eq:haar

以上被視爲分解的過程。從 (1.38) 明顯可知，若 u_k 及 w_k 爲已知，則

$$v_k = \begin{cases} u_n + w_n & \text{if } k = 2n \\ u_n - w_n & \text{if } k = 2n + 1 \end{cases} \quad 0 \leq k \leq N - 1 \quad (1.47)$$

eq:haarc

以上被視爲合成的過程。

但是回顧 (1.37) 和 (1.38) 的獲得方法，乃是藉由方盒函數和 Haar 函數的特性 (分片常數函數)。所以並非一般性的作法。現在，我們往一般性思維邁進一步，介紹另一種作法。同樣也是得到 (1.46) 和 (1.47) 兩種公式，但是在想法上具有一般性，以後還可以再用。

不論分解還是合成，都是在以下等式之兩端作文章：

$$\begin{aligned} &\sum_{k=0}^{N-1} v_k \phi(2^j x - k) \\ &= \sum_{k=0}^{\frac{N}{2}-1} u_k \phi(2^{j-1} x - k) + \sum_{k=0}^{\frac{N}{2}-1} w_k \psi(2^{j-1} x - k) \end{aligned} \quad (1.48)$$

eq:chbasis

先假設 v_k 已知而求 u_k 與 w_k 。複習 ϕ 和 ψ 的正交性：

$$\begin{aligned} \langle \phi(2^j x - k), \phi(2^j x - n) \rangle &= \begin{cases} 0 & \text{if } k \neq n \\ \|\phi(2^j x - n)\|^2 = \frac{1}{2^j} & \text{if } k = n \end{cases} \\ \langle \psi(2^j x - k), \psi(2^j x - n) \rangle &= \begin{cases} 0 & \text{if } k \neq n \\ \|\psi(2^j x - n)\|^2 = \frac{1}{2^j} & \text{if } k = n \end{cases} \\ \langle \phi(2^j x - k), \psi(2^j x - n) \rangle &= 0 \end{aligned}$$

所以若將 (1.48) 之兩端同時與 $\phi(2^{j-1}x - n)$ 作內積 ($0 \leq n \leq \frac{N}{2} - 1$)，得到

$$\begin{aligned} & \left\langle \sum_{k=0}^{N-1} v_k \phi(2^j x - k), \phi(2^{j-1} x - n) \right\rangle \\ &= \sum_{k=0}^{\frac{N}{2}-1} \langle u_k \phi(2^{j-1} x - k), \phi(2^{j-1} x - n) \rangle \\ & \quad + \sum_{k=0}^{\frac{N}{2}-1} \langle w_k \psi(2^{j-1} x - k), \phi(2^{j-1} x - n) \rangle \\ &= \frac{1}{2^{j-1}} u_n \end{aligned}$$

所以

$$u_n = 2^{j-1} \sum_{k=0}^{N-1} v_k \langle \phi(2^j x - k), \phi(2^{j-1} x - n) \rangle$$

由於 $V_{j-1} \subset V_j$ ，故 $\phi(2^{j-1}x - n)$ 可以寫成 $\phi(2^j x - k)$ 的線性組合。如果我們代入這個線性組合，再利用正交性就可以求出內積。其實，根據脹縮平移的特性， V_{j-1} 與 V_j 的關係其實等價於 V_0 與 V_1 的關係。因此我們只須考慮， $\phi(x)$ 如何寫成 $\phi(2x - k)$ 的線性組合。根據 (1.20) 知道

$$\phi(x) = c_0 \phi(2x) + c_1 \phi(2x - 1) \quad (1.49) \quad \text{eq:coefbox}$$

其中 $c_0 = 1$, $c_1 = 1$ 。因此

$$\phi(2^{j-1}x - n) = c_0\phi(2^jx - 2n) + c_1\phi(2^jx - 2n - 1)$$

故根據正交性得到

$$\begin{aligned} u_n &= 2^{j-1} \left(c_0 v_{2n} \langle \phi(2^jx - 2n), \phi(2^jx - 2n) \rangle \right. \\ &\quad \left. + c_1 v_{2n+1} \langle \phi(2^jx - (2n + 1)), \phi(2^jx - (2n + 1)) \rangle \right) \quad (1.50) \\ &= \frac{1}{2} c_0 v_{2n} + \frac{1}{2} c_1 v_{2n+1} = \frac{v_{2n} + v_{2n+1}}{2} \end{aligned}$$

eq:haardc

除了足標從 k 變成 n 之外，與 (1.46) 的結論相同。此乃意料中事。

同樣的道理，若將 (1.48) 之兩端同時與 $\psi(2^{j-1}x - n)$ 作內積 ($0 \leq n \leq \frac{N}{2} - 1$)，得到

$$w_n = 2^{j-1} \sum_{k=0}^{N-1} v_k \langle \phi(2^jx - k), \psi(2^{j-1}x - n) \rangle$$

根據 (1.34) 知道

$$\psi(x) = d_0\phi(2x) + d_1\phi(2x - 1) \quad (1.51) \quad \text{eq:coefhaar}$$

其中 $d_0 = 1$, $d_1 = -1$ 。因此

$$\psi(2^{j-1}x - n) = d_0\phi(2^jx - 2n) + d_1\phi(2^jx - 2n - 1)$$

故根據正交性得到

$$\begin{aligned} w_n &= 2^{j-1} \left(d_0 v_{2n} \langle \phi(2^jx - 2n), \phi(2^jx - 2n) \rangle \right. \\ &\quad \left. + d_1 v_{2n+1} \langle \phi(2^jx - (2n + 1)), \phi(2^jx - (2n + 1)) \rangle \right) \quad (1.52) \\ &= \frac{1}{2} d_0 v_{2n} + \frac{1}{2} d_1 v_{2n+1} = \frac{v_{2n} - v_{2n+1}}{2} \end{aligned}$$

eq:haardd

除了足標從 k 變成 n 之外，也與 (1.46) 的結論相同。

至此，我們認識到， $\phi(x)$ 和 $\psi(x)$ 在 V_1 中的基底變換公式可以寫成以下的一般形式

$$\begin{aligned}\phi(x) &= \sum_k c_k \phi(2x - k) \\ \psi(x) &= \sum_k d_k \phi(2x - k)\end{aligned}\tag{1.53}$$

eq:decompcc

它們將會是凌波轉換中分解步驟的鑰匙。

現在假設 u_k 與 w_k 已知而求 v_n 。同樣是利用正交的性質，若將 (1.48) 之兩端同時與 $\phi(2^j x - n)$ 作內積 ($0 \leq n \leq N-1$)，得到

$$\begin{aligned}v_n \langle \phi(2^j x - n), \phi(2^j x - n) \rangle &= \frac{1}{2^j} v_n \\ &= \sum_{k=0}^{\frac{N}{2}-1} u_k \langle \phi(2^{j-1} x - k), \phi(2^j x - n) \rangle \\ &\quad + \sum_{k=0}^{\frac{N}{2}-1} w_k \langle \psi(2^{j-1} x - k), \phi(2^j x - n) \rangle\end{aligned}\tag{1.54}$$

eq:vnip

由於 $V_j = V_{j-1} \oplus W_{j-1}$ ，故 $\phi(2^j x - n)$ 可以寫成 $\phi(2^{j-1} x - k)$ 和 $\psi(2^{j-1} x - k)$ 的線性組合。如果我們代入這個線性組合，再利用正交性就可以求出內積。同樣地，我們只須考慮 $\phi(2x)$ 如何寫成 $\phi(x - k)$ 和 $\psi(x - k)$ 的線性組合，其他的就利用脹縮平移來變化。

假設

$$\phi(2x) = a_0 \phi(x) + b_0 \psi(x)\tag{1.55}$$

eq:phieven

則在兩端同時與 $\phi(x)$ 做內積得到

$$\langle \phi(2x), \phi(x) \rangle = a_0 \langle \phi(x), \phi(x) \rangle + b_0 \langle \psi(x), \phi(x) \rangle = a_0$$

這時候，利用 (1.49) 計算

$$\langle \phi(2x), \phi(x) \rangle = c_0 \langle \phi(2x), \phi(2x) \rangle + c_1 \langle \phi(2x), \phi(2x-1) \rangle = \frac{1}{2} c_0$$

故 $a_0 = \frac{1}{2} c_0 = \frac{1}{2}$ 。同理

$$b_0 = \langle \phi(2x), \psi(x) \rangle = \frac{1}{2} d_0 = \frac{1}{2}$$

但是我們發現僅只 (1.55) 是不夠用的。如果 $n = 2\ell$ 為偶數，可以利用 (1.55) 得到

$$\phi(2^j x - n) = \phi(2(2^{j-1} x - \ell)) = a_0 \phi(2^{j-1} x - \ell) + b_0 \psi(2^{j-1} x - \ell)$$

但是如果 $n = 2\ell + 1$ 為奇數，則無法利用 (1.55)。因為

$$\phi(2^j x - n) = \phi(2(2^{j-1} x - \ell) - 1)$$

所以我們發現，還需要知道

$$\phi(2x - 1) = g_0 \phi(x) + h_0 \psi(x) \quad (1.56) \quad \text{eq:phiodd}$$

我們留給讀者去驗證 $g_0 = \frac{1}{2} c_1 = \frac{1}{2}$ ， $h_0 = \frac{1}{2} d_1 = -\frac{1}{2}$ 。如此一來，則若 $n = 2\ell + 1$ ，

$$\phi(2^j x - n) = \phi(2(2^{j-1} x - \ell) - 1) = g_0 \phi(2^{j-1} x - \ell) + h_0 \psi(2^{j-1} x - \ell)$$

觀察 (1.55) 與 (1.56) 之中的係數 a_0 、 g_0 與 b_0 、 h_0 互不相關，所以我們使用一個技巧將它們合併成兩組係數：

$$a_0 = \frac{1}{2} p_0, g_0 = \frac{1}{2} p_1 \text{ 與 } b_0 = \frac{1}{2} q_0, h_0 = \frac{1}{2} q_1 \quad (1.57) \quad \text{eq:pqcoef}$$

也就是將 a_* 放入 $\frac{1}{2}p_k$ 的偶數項，將 g_* 放入 $\frac{1}{2}p_k$ 的奇數項；將 b_* 放入 $\frac{1}{2}q_k$ 的偶數項，將 h_* 放入 $\frac{1}{2}q_k$ 的奇數項。

所以，如果 $n = 2\ell$ 為偶數，則根據 (1.54) 可得

$$\begin{aligned} v_n &= 2^j \left(\sum_{k=0}^{\frac{N}{2}-1} u_k \langle \phi(2^{j-1}x - k), \frac{1}{2}p_0\phi(2^{j-1}x - \ell) + \frac{1}{2}q_0\psi(2^{j-1}x - \ell) \rangle \right) \\ &\quad + 2^j \left(\sum_{k=0}^{\frac{N}{2}-1} w_k \langle \psi(2^{j-1}x - k), \frac{1}{2}p_0\phi(2^{j-1}x - \ell) + \frac{1}{2}q_0\psi(2^{j-1}x - \ell) \rangle \right) \\ &= p_0 u_\ell + q_0 w_\ell \end{aligned} \tag{1.58} \quad \text{eq:vpqeven}$$

同理，如果 $n = 2\ell + 1$ 為奇數，則

$$v_n = p_1 u_\ell + q_1 w_\ell \tag{1.59} \quad \text{eq:vpqodd}$$

代入 p_i 、 q_i 的值，發現除了足標從 k 變成 n 之外，與 (1.47) 的結論相同。

至此，我們認識到， $\phi(2x)$ 和 $\phi(2x - 1)$ 在 $V_0 \oplus W_0$ 中的基底變換公式可以寫成以下的一般形式 (在第三章三節中會再談)

$$\begin{aligned} \phi(2x) &= \frac{1}{2} \sum_k p_{-2k} \phi(x - k) + \frac{1}{2} \sum_k q_{-2k} \psi(x - k) \\ \phi(2x - 1) &= \frac{1}{2} \sum_k p_{1-2k} \phi(x - k) + \frac{1}{2} \sum_k q_{1-2k} \psi(x - k) \end{aligned} \tag{1.60} \quad \text{eq:reconst}$$

它們將會是凌波轉換中合成步驟的鑰匙。

在這一節裡面，我們看到凌波轉換被四組係數 $\{c_k\}$ 、 $\{d_k\}$ 、 $\{p_k\}$ 和 $\{q_k\}$ 決定。其中 $\{c_k\}$ 的作用是将高解析度訊號取平均得到低解析度訊號，(c 可以記做 scale down)； $\{d_k\}$ 的作用是記錄高低解析度之間

差異的細節，(d 可以記做 **details**)。如果只從高解析度訊號中提取低解析度訊號，而不保留細節部分，則高解析度訊號中的某些資料就不見了，以後不可能從低解析度訊號中重建高解析度訊號。就是因為我們將細節部分保留在 W_{j-1} 裡面，以後才得以重建高解析度訊號。而且是完美地重建 (*perfect reconstruction*) 因為重建後的訊號沒有任何失真。 $\{p_k\}$ 和 $\{q_k\}$ 就是重建的係數，(p q 可以記做 **perfect reconstruction**)。

idx:pr

在第三章我們會知道，通常只需決定 c_k 和 d_k ，就可以推導 p_k 和 q_k 。讀者可以回顧從 (1.55) 到 (1.56) 之間的過程，一般的情形也差不多如此。而在第三章的正交凌波函數理論中，我們將看到就連 d_k 都可以由 c_k 造成，所以全部四組係數其實都由 c_k 產生。

有些同學 (尤其是數學系的同學) 會在這時候問，既然要完美地重建，那當初幹嘛要分解？殊不知訊號處理的工程師就是在分解與合成 (重建) 之間做手腳。如果在分解之後動了一些手腳才做合成 (比如說影像壓縮)，那麼合成後的訊號當然不會零失真。數學理論告訴我們，如果你在分解後什麼也沒做，那麼合成後就是原來的訊號。這叫做穩定性。更好的是，數學理論將會告訴我們，如果在分解後因為動手腳而產生誤差，那就可以預測重建之後的誤差。

習題 1.29 證明 (1.56) 中的 $g_0 = \frac{1}{2}c_1$, $h_0 = \frac{1}{2}d_1$ 。

習題 1.30 證明

$$\langle \phi(2^l x - m), \psi(2^j x - k) \rangle = 0, \quad \forall l \leq j, \forall m, \forall k$$

(提示：未必需要計算。)

習題 1.31 證明

$$\langle \psi(2^l x - m), \psi(2^j x - k) \rangle = 0, \quad \forall l \neq j, \forall m, \forall k$$

(提示：未必需要計算。)

習題 1.32 證明 (1.59)。

習題 1.33 在 (1.54) 中，我們也可以利用 (1.49) 及 (1.51) 將 $\phi(2^{j-1}x - k)$ 和 $\psi(2^{j-1}x - k)$ 變換成 $\phi(2^jx - *)$ 的形式，然後求內積。請討論這種做法和課文中的不同。

1.17 DWT 之矩陣表達

在這一節中，我們利用矩陣與向量的運算符號來表達離散凌波轉換。這些符號，將來對數學與算法的觀念性發展是有益的。真正實現在電腦程式中的演算法，倒不一定要透過矩陣運算。我們在下一節討論演算法。

先說分解過程。我們將輸入訊號 v 寫成一個 $N \times 1$ 的向量：

$$v = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{N-1} \end{pmatrix}_{N \times 1}$$

當我們將向量的陣列表達一律寫成直向量 (*column vector*)。稱橫向量為序列 (*array*)。根據 (1.50)，分解後的係數 $u = (u_k)$ 與 $w = (w_k)$ 可以寫成兩個 $\frac{N}{2} \times 1$ 的向量，而且

$$\begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{\frac{N}{2}-1} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} c_0 & c_1 & & & & & & & \\ & & c_0 & c_1 & & & & & \\ & & & & \ddots & & & & \\ & & & & & & c_0 & c_1 & \\ & & & & & & & & \end{pmatrix}_{\frac{N}{2} \times N} \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{\frac{N}{2}-1} \end{pmatrix}_{N \times 1}$$

令 \mathcal{L} 代表上式中的矩陣。則 $u = \mathcal{L}v$ 就是將高解析度的訊號降層到低解析度的訊號，一個術語稱此步驟為低頻濾波 (*low-pass filtering*)；因 `idx:lowpass`

此 \mathcal{L} 可以記作 **Low-pass** 矩陣。同理，令

$$\mathcal{H} = \frac{1}{2} \begin{pmatrix} d_0 & d_1 & & & \\ & d_0 & d_1 & & \\ & & \ddots & \ddots & \\ & & & d_0 & d_1 \end{pmatrix}_{\frac{N}{2} \times N}$$

(\mathcal{L} 和 \mathcal{H} 裡面的空白處表示 0) 則根據 (1.52) $\mathbf{w} = \mathcal{H}\mathbf{v}$ 就是將高低解析度之間的細節部分分解出來。如果 V_{j-1} 相對於 V_j 的低頻部分，那麼 W_{j-1} 就相對於 V_j 的高頻部分。所以術語稱此步驟為高頻濾波 (*high-pass filtering*)；因此 \mathcal{H} 可以記作 **High-pass** 矩陣。

idx:highpass

將 \mathcal{L} 和 \mathcal{H} 上下堆在一起，就成爲一個 $N \times N$ 的方陣：

$$\mathcal{W} = \begin{pmatrix} \mathcal{L} \\ \mathcal{H} \end{pmatrix}_{N \times N}$$

則 \mathcal{W} 可以被視爲 **Wavelet** 轉換矩陣：

idx:wtransmat

$$\mathcal{W}\mathbf{v} = \begin{pmatrix} \mathbf{u} \\ \mathbf{w} \end{pmatrix} \quad (1.61)$$

eq:Wvv

如果 \mathbf{v} 是 V_j 中的訊號，則 $\mathcal{W}\mathbf{v}$ 完成 $V_j \mapsto V_{j-1} \oplus W_{j-1}$ 的分解。在電腦程式中，我們將不再需要 \mathbf{v} (它可以被 \mathbf{u} 和 \mathbf{w} 重建)，所以可以將 \mathbf{u} 儲存在 \mathbf{v} 的上半段，而 \mathbf{w} 儲存在 \mathbf{v} 的下半段。如果還想做 $V_{j-1} \mapsto V_{j-2} \oplus W_{j-2}$ 的分解，就把 N 換成 $\frac{N}{2}$ (所以方陣 \mathcal{W} 的維度就縮小了一半)，把 \mathbf{v} 換成 \mathbf{u} ，再做一次分解步驟；依此類推。

以上說的是一維的訊號分解。二維的訊號分解可以重複利用一維的方式來做。將二維訊號寫成一個矩陣，爲了行文的方便，仍然假設它是個 $N \times N$ 的方陣，而 $N = 2^j$ 。如果輸入的方陣是

$$\mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_N \end{bmatrix}$$

其中 v_i 是 V 的 N 個行向量 (column vector)；每個 v_i 都是 $N \times 1$ 的向量。則 WV 就是對 V 的每一個行向量做分解：idx:colvec

$$WV = [Wv_1 \ Wv_2 \ \cdots \ Wv_N] = \begin{bmatrix} u_1 & u_2 & \cdots & u_N \\ w_1 & w_2 & \cdots & w_N \end{bmatrix}$$

如果 V 是一個平面區域內的影像，則 WV 完成了垂直方向上的高頻與低頻分解。以數學空間的符號寫來，就是完成了

$$\begin{aligned} V_j(x, y) &= V_j(x) \times V_j(y) \mapsto V_j(x) \times (V_{j-1}(y) \oplus W_{j-1}(y)) \\ &= V_j(x) \times V_{j-1}(y) \oplus V_j(x) \times W_{j-1}(y) \end{aligned}$$

的分解。

接著，我們應該繼續做水平方向上的高頻與低頻分解。也就是要將 WV 的列向量拿出來做分解。

給一個方陣 U ，如果將它寫成

$$U = \begin{bmatrix} u_1^T \\ u_2^T \\ \vdots \\ u_N^T \end{bmatrix}$$

其中每個 u_i 都是 $N \times 1$ 向量，則我們稱 u_i 是 U 的 N 個列向量 (row vector)。上標符號 T 是轉置 (transpose) 的意思，例如idx:transp

$$(1, 2, 3)^T = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}^T = (1, 2, 3)$$

而若

$$U = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

則

$$\mathbf{U}^T = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

且 \mathbf{U} 的三個列向量就是

$$\mathbf{u}_1 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad \mathbf{u}_2 = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} \quad \mathbf{u}_3 = \begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix}$$

我們留給讀者去驗證

$$\mathcal{W} \mathbf{V} \mathcal{W}^T \tag{1.62} \quad \text{eq:Wtwod}$$

就是將 $\mathcal{W} \mathbf{V}$ 的列向量分別做分解。結果可以寫成四塊矩陣：

$$\mathcal{W} \mathbf{V} \mathcal{W}^T = \begin{pmatrix} \mathbf{U} & \mathbf{W}_x \\ \mathbf{W}_y & \mathbf{W} \end{pmatrix} \tag{1.63} \quad \text{eq:UWWW}$$

以數學空間的符號寫來，就是完成了

$$\begin{aligned} & V_j(x) \times V_{j-1}(y) \oplus V_j(x) \times W_{j-1}(y) \\ \mapsto & V_{j-1}(x) \times V_{j-1}(y) \oplus W_{j-1}(x) \times V_{j-1}(y) \\ & \oplus V_{j-1}(x) \times W_{j-1}(y) \oplus W_{j-1}(x) \times W_{j-1}(y) \\ = & V_{j-1}(x, y) \oplus W_{j-1}(x) \times V_{j-1}(y) \oplus V_{j-1}(x) \times W_{j-1}(y) \oplus W_{j-1}(x, y) \end{aligned}$$

的分解。其中 \mathbf{W}_x 對應 $W_{j-1}(x) \times V_{j-1}(y)$ 空間內的訊號，它們是半高頻：只有 x 方向的高頻出現在這裡； \mathbf{W}_y 對應 $V_{j-1}(x) \times W_{j-1}(y)$ 空間內的訊號，它們也是半高頻：只有 y 方向的高頻出現在這裡。而 \mathbf{W} 就是對應 $W_{j-1}(x) \times W_{j-1}(y)$ 空間內的訊號，它們是全高頻： x 和 y 方向都有的高頻出現在這裡。參照後面的圖 9。

反過來，現在我們看合成的過程。這時候，輸入的是兩個 $\frac{N}{2} \times 1$ 的向量 \mathbf{u} 和 \mathbf{w} ，要合成一個 $N \times 1$ 的向量 \mathbf{v} ；也就是做 $V_{j-1} \oplus W_{j-1} \mapsto V_j$ 的基底轉換。根據 (1.61)，在線性代數上，就是

$$\mathbf{v} = \mathcal{W}^{-1} \begin{pmatrix} \mathbf{u} \\ \mathbf{w} \end{pmatrix}$$

通常，求一個矩陣的逆矩陣不是件簡單的事。但是現在卻異常簡單。

考慮 $2\mathcal{W}^T$ ，用次矩陣 (*sub-matrix*) 的符號寫，就是 $(2\mathcal{L}^T \ 2\mathcal{H}^T)$ ，因此

$$\mathcal{W}(2\mathcal{W}^T) = \begin{pmatrix} \mathcal{L} \\ \mathcal{H} \end{pmatrix} (2\mathcal{L}^T \ 2\mathcal{H}^T) = \begin{pmatrix} 2\mathcal{L}\mathcal{L}^T & 2\mathcal{L}\mathcal{H}^T \\ 2\mathcal{H}\mathcal{L}^T & 2\mathcal{H}\mathcal{H}^T \end{pmatrix}$$

請讀者自行驗證， $2\mathcal{L}\mathcal{L}^T = \mathbf{I}$ ， $\mathcal{L}\mathcal{H}^T = \mathcal{H}\mathcal{L}^T = \mathbf{0}$ ， $2\mathcal{H}\mathcal{H}^T = \mathbf{I}$ ，其中 \mathbf{I} 是 $\frac{N}{2} \times \frac{N}{2}$ 的單位矩陣， $\mathbf{0}$ 是 $\frac{N}{2} \times \frac{N}{2}$ 的零矩陣。所以， $\mathcal{W}(2\mathcal{W}^T)$ 就是 $N \times N$ 的單位矩陣，因此知道

$$\mathcal{W}^{-1} = 2\mathcal{W}^T$$

它的矩陣形式是

$$\begin{pmatrix} c_0 & & & & d_0 & & & & \\ c_1 & & & & d_1 & & & & \\ & c_0 & & & & d_0 & & & \\ & c_1 & & & & d_1 & & & \\ & & \dots & & & & \dots & & \\ & & & c_0 & & & & d_0 & \\ & & & c_1 & & & & d_1 & \end{pmatrix}_{N \times N}$$

另一方面，根據 (1.58) 和 (1.59)，我們知道合成矩陣應該是

$$\begin{pmatrix} p_0 & & & & q_0 \\ p_1 & & & & q_1 \\ & p_0 & & & q_0 \\ & p_1 & & & q_1 \\ & & \ddots & & \ddots \\ & & & p_0 & & q_0 \\ & & & p_1 & & q_1 \end{pmatrix}_{N \times N}$$

回顧 (1.57) 代入 p_k 和 q_k 的值。毫無意外地，我們發現這個矩陣也就是 $2W^T$ 。所以我們知道，其實在 (1.60) 中的 p_k 和 q_k 係數，就是 $\frac{1}{2}W^{-1}$ 的係數。

習題 1.34 若 V 的列向量是 v_1, v_2, \dots, v_n ，若 $Wv_i = u_i$ ，證明 u_i 是 VW^T 的列向量。(提示：也許可以利用 $(AB)^T = B^T A^T$ 這個關係。)

習題 1.35 我們利用這個題目來練習次矩陣的計算規則。若 A, B, C 和 D 為 $n \times n$ 的矩陣，

$$\begin{pmatrix} A \\ B \end{pmatrix} \text{ 和 } (C \ D)$$

分別是由它們堆成的 $2n \times n$ 與 $n \times 2n$ 矩陣。證明

$$(C \ D) \begin{pmatrix} A \\ B \end{pmatrix} = CA + DB, \quad \begin{pmatrix} A \\ B \end{pmatrix} (C \ D) = \begin{pmatrix} AC & AD \\ BC & BD \end{pmatrix}$$

並標明它們的維度。

習題 1.36 令 \mathcal{L} 和 \mathcal{H} 分別是本節中定義的 Low-pass 矩陣和 High-pass 矩陣。證明

$$2\mathcal{L}\mathcal{L}^T = I, \quad 2\mathcal{H}\mathcal{H}^T = I, \quad \mathcal{L}\mathcal{H}^T = 0$$

(回顧 $c_0 = c_1 = 1$, $d_0 = 1$, $d_1 = -1$ 。) 然後，再重排足標，把 v 的偶數項排在前半部，也就是 u ；奇數項排在後半部，也就是 w 。

我們探討這個演算法的計算複雜度 (*computational complexity*)。有些計算理論的書上，定義一個 flop 是一次 $a \times b + c$ 的浮點數計算。但是我們並不這樣定義；我們選擇比較簡單的定義：兩個浮點數做一次相加或相乘，稱為一個 *flop*。這個定義和 Matlab `flops` 指令的定義大致相同。根據這個定義，每產生一個 u_k 需要 3 flops，產生 w_k 亦同。故做一層分解總共的複雜度大約是 $4N$ flops。如果繼續做第二層分解，則因為向量的長度已經是 $\frac{N}{2}$ ，故複雜度是 $4\frac{N}{2}$ 。依此類推，如果做了 $\log_2 N$ 次的分解，總共的複雜度是

idx:cmplx
idx:flop

$$4\left(N + \frac{N}{2} + \frac{N}{4} + \cdots + 1\right) \approx 8N$$

用大 O 符號，我們說凌波轉換的分解過程是 $O(N)$ 的演算法。以後，即使是應用逼近效果更好 (也就是說消散得更快) 的凌波函數，其計算複雜度仍然是 $O(N)$ 。這是任何訊號分解的演算法所能達到的最低複雜度。與快速傅立葉轉換 (*FFT*) 相比，*FFT* 的複雜度是 $O(N \log_2 N)$ ，所以，在理論上，凌波轉換略勝一籌。但實際上呢， $\log_2 N$ 實在已經是個很小的數，而且 *FFT* 的計算程式比凌波轉換多了半個世紀的實際經驗，因此在計算速度方面，凌波轉換與快速傅立葉轉換應該沒有太大的差距。

前面的演算法必須用到迭代迴路 (`for-loop`)。用編譯語言 (例如 C 和 FORTRAN) 寫來不成問題，但是若用像 Matlab 的巨集指令來寫，就會變慢了 (請參考附錄 A.1)；除非您使用 Matlab `compiler`。

如果使用 Matlab 而空間不慮缺乏，最簡單也是最快的算法如下。

```

N = length(v);
u = (v(1:2:N) + v(2:2:N))/2;
w = (v(1:2:N) - v(2:2:N))/2;

```

以上指令中的 $v(1:2:N)$ 就是取出 v 向量的奇數項，而 $v(2:2:N)$ 就是取出 v 向量的偶數項。注意 Matlab 的向量編號一律從 1 而開始，而我們演算法中的足標記號卻是從 0 開始。所以 $v(k)$ 其實是 v_{k-1} 。

合成過程就是將分解過程反過來。如果輸入兩個長度為 $\frac{N}{2}$ 的向量 u 和 w 。先將它們交叉寫成一個長度為 N 的向量，然後以 $\widetilde{W}^{-1} = 2\widetilde{W}^T$ 乘之：

$$v = \begin{pmatrix} u_0 \\ w_0 \\ u_1 \\ w_1 \\ \vdots \\ u_{\frac{N}{2}-1} \\ w_{\frac{N}{2}-1} \end{pmatrix}, \quad 2\widetilde{W}^T v \mapsto v$$

在演算法方面，就是

$$\begin{aligned}
2(p_0 v_{2k} + q_0 v_{2k+1}) &\mapsto \text{tmp} \\
2(p_1 v_{2k} + q_1 v_{2k+1}) &\mapsto v_{2k+1} \\
\text{tmp} &\mapsto v_{2k}
\end{aligned}$$

(回顧 $2p_0 = 2p_1 = 1$, $2q_0 = 1$, $2q_1 = -1$ 。) 合成一次的計算複雜度也大約是 $4N$ ，合成 $\log_2 N$ 次的計算複雜度總共是 $O(N)$ 。同樣地，如果使用 Matlab 而空間不慮缺乏，則用簡單的指令即可解決。合成算法，

寫成向量形式，就是

$$v = \begin{pmatrix} u_0 \\ u_0 \\ u_1 \\ u_1 \\ \vdots \\ u_{\frac{N}{2}-1} \\ u_{\frac{N}{2}-1} \end{pmatrix} + \begin{pmatrix} w_0 \\ -w_0 \\ w_1 \\ -w_1 \\ \vdots \\ w_{\frac{N}{2}-1} \\ -w_{\frac{N}{2}-1} \end{pmatrix}_{N \times 1}$$

對應的 Matlab 指令如下。

```
N = length(u);
v(1:2:2*N) = u+w;
v(2:2:2*N) = u-w;
```

以上的指令，就是把 $u+w$ 寫入 v 的奇數項，而 $u-w$ 寫入 v 的偶數項。如同前面剛解釋過的，Matlab 的向量編號一律從 1 開始，而我們演算法中的足標記號卻是從 0 開始。所以 $v(k)$ 其實是 v_{k-1} 。

現在看一個例子。我們取圖 2 的那個 $f(x) = -\frac{16}{9}x^2 + \frac{8}{3}x$ 。在 $[0, 1]$ 中取節點 $x_i = \frac{i}{127}$ ， $0 \leq i \leq 127$ 。令 $v_7 = (v_0, v_1, \dots, v_{127})^T$ ，其中 $v_i = f(x_i)$ 。 v_7 的圖形就是圖 2 上方的圖形。令 v_6 是分解一次之後的低頻部分， w_6 是分解一次之後的細節部分。將 v_6 再分解，得到 v_5 和 w_5 。依此類推。注意 w_j 的長度就是 2^j 。圖 7 左上角的圖就是 v_6 和 w_6 的圖形。可見 v_6 的圖形和 v_7 很像，只是縮短了一半。而明顯地，細節的部分 w_6 幾乎全是 0。圖 7 的右上角就是 v_3 、 w_3 、 w_4 、 w_5 、 w_6 的圖形。可見 v_3 似乎還保持 v_7 的形狀，只是因為只有八個資料點，所以圖形很窄。而 w_j 是隨著 j 的降低而數值漸漸變大。這就是凌波消散定理的寫照。但是注意當 j 比較小的時候， w_j 中的資料點也比較少。

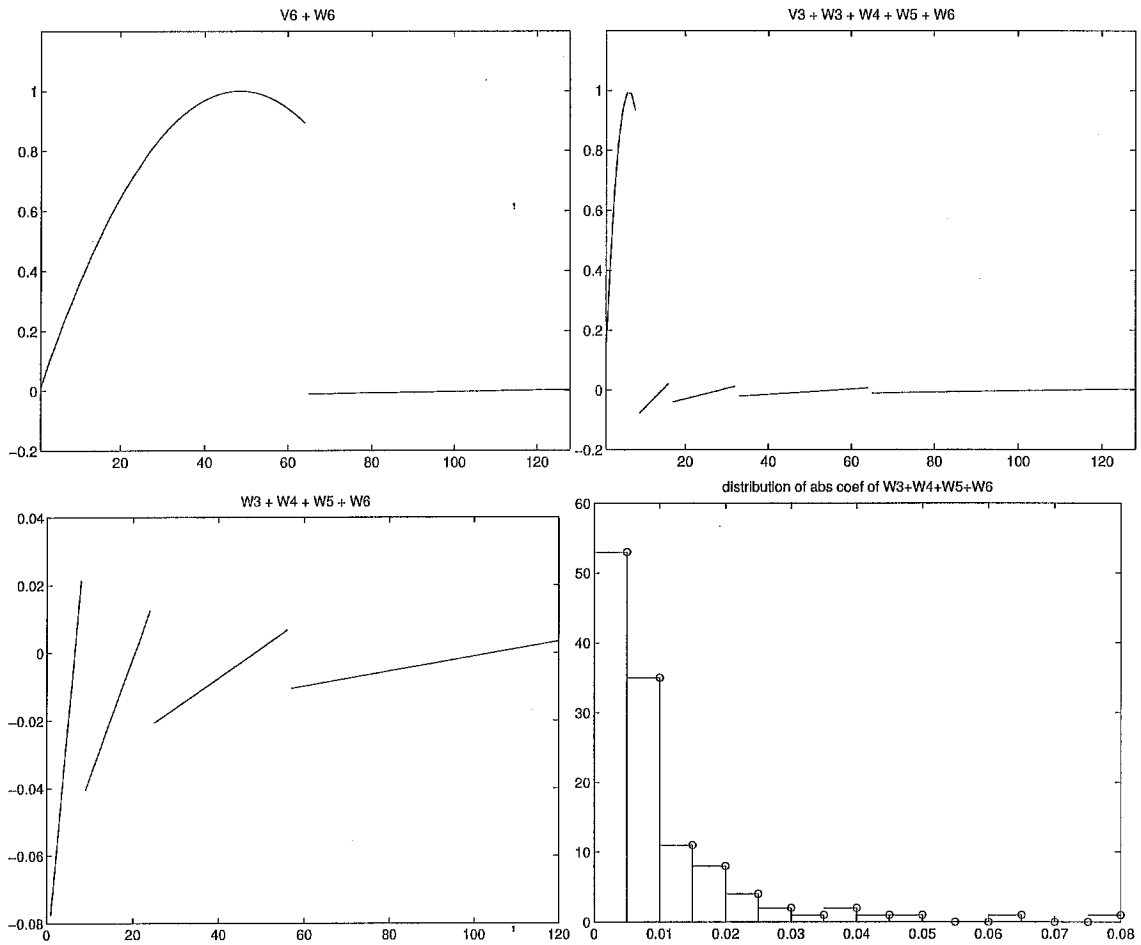


fig:haartestone 圖 7 一維 Haar 凌波分解的例子

爲了更明白地比較細節係數的情形，我們在圖 7 的左下角呈現 $w_3 \dots w_6$ 的圖形。這張圖比較看得出來， w_j 的絕對值隨著 j 的升高而變小的趨勢。同時，參照圖 2 最上方的 $f(x)$ 圖形 (其實就是 v_7 的圖形)，因爲 $f(x)$ 在 $x = 0$ 附近的切線斜率比較大，所以我們看到 w_j 的絕對值都是靠左邊比較大。而 $f(x)$ 在 $x = \frac{3}{4}$ 附近的切線斜率比較小，所以 w_j

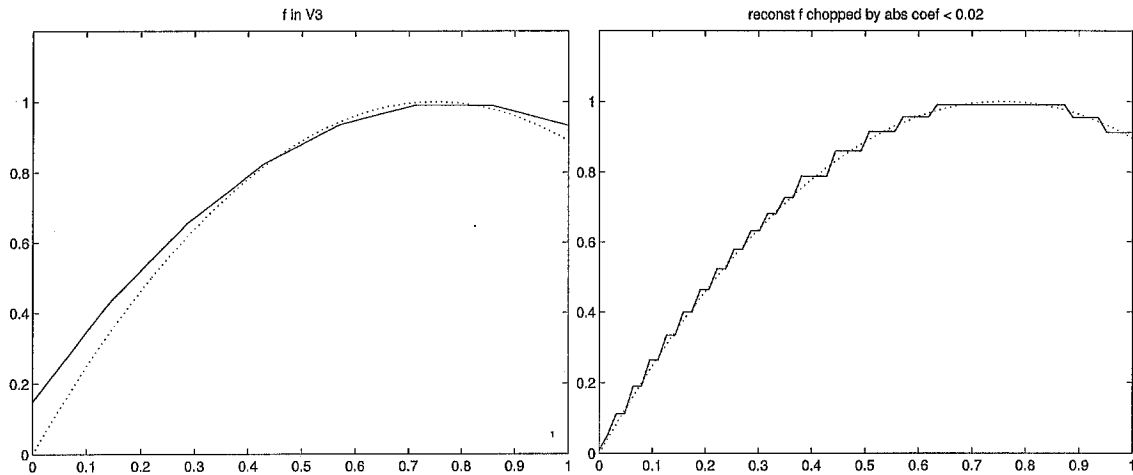


fig:haartestoneV 圖 8 一維 Haar 凌波合成的例子

的絕對值都在靠右邊比較小，而且在靠右端 $\frac{1}{4}$ 處最接近 0。而圖 7 的右下角是 w_j ($j = 3, 4, 5, 6$) 的係數絕對值分佈圖。共有 120 個係數，絕對值都在 $[0, 0.08]$ 之內。而絕大多數的係數的絕對值落在 $[0, 0.005]$ 之內。其實這 120 個數的中位數是 0.0058。

在圖 8 的左邊，我們將 v_3 放到和 v_7 一樣的尺度來比較。虛線是 v_7 。可見雖然只有八個資料點， v_3 的折線圖究竟還是保留了一點 v_7 的原貌。圖 8 的右邊，是將 w_j 中絕對值 < 0.02 的係數都剔除了之後，所做的還原圖形。以 0.02 為門檻，一共去除了 107 個係數，只用了 21 個係數。相當於只用了原來的 16% 的係數。像這樣的重建當然不會完美。但是我們發現重建回來的圖形與原圖 (虛線) 相去不遠，而相差的就是那些被我們去掉的細節部分。

如果是二維的訊號 (影像)，則假設輸入訊號是 $N \times N$ 矩陣。那麼在分解的時候，我們先對每個行向量做一次分解，然後將計算結果的每個列向量做一次分解，所以原來的矩陣在分解後被分成四塊矩陣。

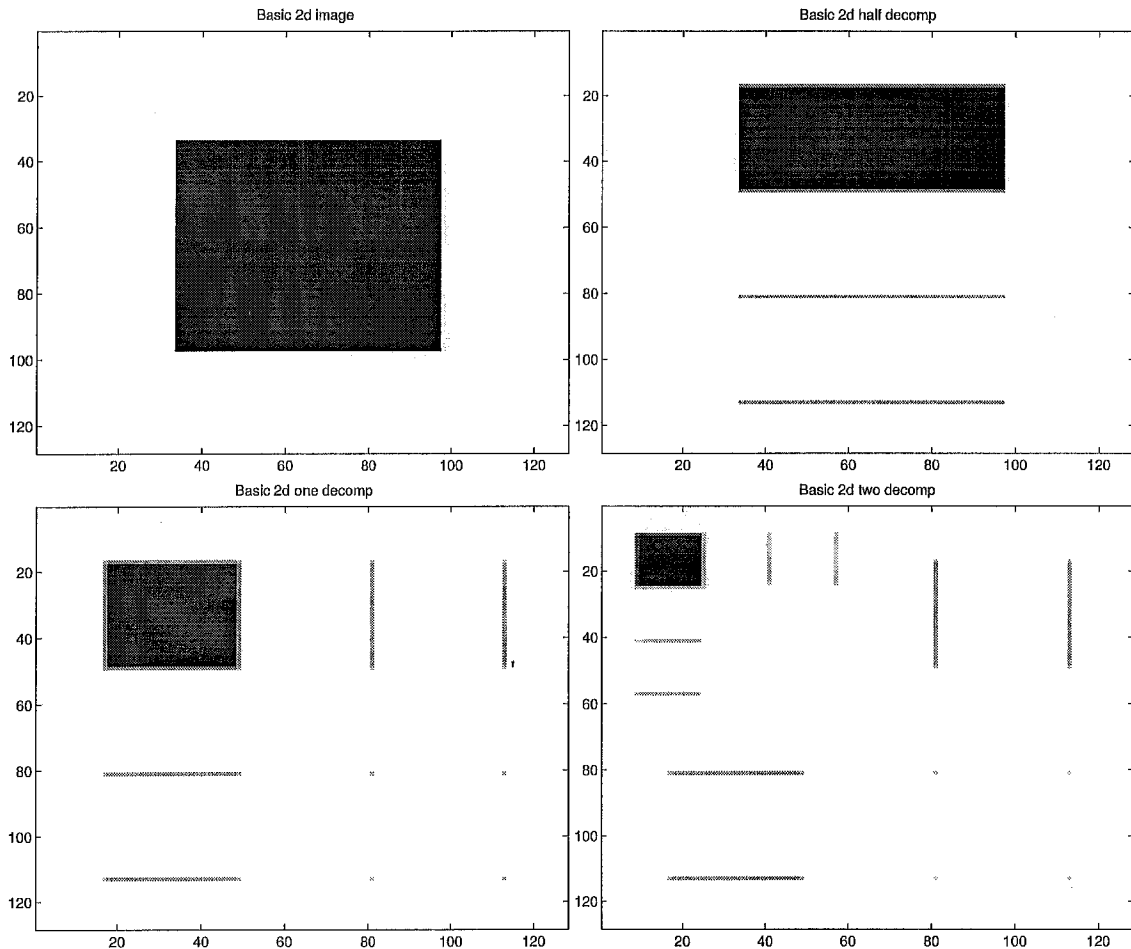


fig:haartesttwo 圖 9 二維 Haar 凌波分解的例子

我們用一種最簡單的圖來表現二維凌波轉換的特性。令 A 是一個 128×128 的影像。如圖 9 之左上角。圖中只有黑白兩色。但色盤定義了 64 段的灰階。爲了表現凌波係數的消散性質，我們定義大的數字對應深色，小的數字對應淺色。對 A 的行向量做一次分解，所得的影像放在圖 9 之右上角。影像上方是低解析度部分，而下方是細節處

(顯示的是係數的絕對值)。可見細節處只有當原圖在垂直方向有變化的時候才会有大的係數發生。然後我們再對每個列向量做分解，這就完成了一次分解過程。結果的影像放在左下角。從這張影像可以明顯地看到，二維凌波轉換產生四塊矩陣，一塊代表低解析度部分，三塊代表細節部分。參照 (1.63)，特別注意那細節的三塊：其中右上部分是 W_x ，只有當原圖在水平方向有變化的時候，它才有大的係數；左下部分是 W_y ，只有當原圖在垂直方向有變化的時候，它才有大的係數；右下部分是 W ，只有當原圖在垂直與水平方向都有變化的時候，它才有大的係數。這些現象，也都是凌波係數消散性質的寫照。而且這些現象是凌波函數可以應用在影像處理上的主要依據。在圖 9 之右下角，我們呈現一個分解了兩次的影像。

如果想知道顯現這些影像的程序，和完整二維 Haar 分解的程序，請參考附錄 A.2。

前面討論的都是假設輸入訊號長度為 $N = 2^j$ 的情形。這是為了行文的方便，因為每降一層資料長度 $\div 2$ ，所以如果 $N = 2^j$ ，則每層的資料長度都可以被 2 整除。在實用上，我們必然會面臨 $\log_2 N \notin \mathbb{Z}$ 的情形。一個最簡單的處理方式是，將此向量補零，使其長度變成 2^j 。這麼做的缺點有二。一個是增加資料量，最糟的時候幾乎增加一倍（例如從 513 增加到 1024）。第二是在銜接零的地方很可能有個不連續點，所以在那個點附近，訊號的背景函數 $f(x)$ 不是 C^1 ，因而會造成比較大的 ψ_{jk} 係數。其他的作法還有，做鏡射補足 2^j ，或是做週期擴充補足 2^j 筆資料。總之，這些都是將訊號資料補滿 2^j 的作法。最長被套用的是週期擴充方法，因為在這個方法下，分解與合成的程式都很好寫。

另一種想法是，若輸入向量為奇數時，自動遺留一筆資料不做分解，而將那筆資料接在 w 的後面。例如輸入長度是 370 時，分解一次得到高低頻各 185 筆資料。將低頻部分再分解時，就遺留一筆資料而

分解 184 筆，得到低頻 92 筆和高頻 92 + 1 筆資料。依此類推，我們用以下的形式表達資料的分配情形：

$$\begin{aligned}
 370 &= [185] + [185] = [185] + [92 + 1] + [92] = \dots \\
 &= [185] + [92 + 1] + [46] + [23] + [11 + 1] + [5 + 1] \\
 &\quad + [2 + 1] + [1] + [0 + 1]
 \end{aligned} \tag{1.64}$$

eq:notpower

合成的時候必須知道，在分解過程中，什麼時候遺留了一筆資料。其實這有幾種方法可以辨別。其中之一就是看 N 的二進位數字。比如說 $370 = 101110010$ ，從低位元 (右邊) 讀到高位元，就對應了從高層到低層的分解過程中有遺留 (1) 或是沒有遺留 (0) 任何數字。

當 $N = 2^j + 1$ 的時候第一種想法 (補足次方) 最沒效率，當 $N = 2^j - 1$ 的時候第二種想法 (逐次遺留) 最沒效率。所以，在程式設計上，可以兩者兼用而取比較有利的作法。總而言之，只要我們知道當 $N = 2^j$ 時的作法，對於其他的 N ，可以用程式設計的技巧來解決。

最後，讓我們將本書中陳述凌波轉換演算法的矩陣形式，對照訊號處理文獻中常見的疊積形式。 c_k 是凌波分解中製造低頻的係數。在訊號處理中稱它為低頻濾波器 (*low-pass filter*)。訊號處理的文獻中，喜歡用疊積為運算工具。若 $\mathbf{u} = (u_i)$ 與 $\mathbf{v} = (v_i)$ 為兩個序列，則它們的離散疊積是 $\mathbf{x} = \mathbf{u} * \mathbf{v}$ ，其中

idx:lowpass

idx:dconvol

$$x_i = \sum_n u_{i-n} v_n = \sum_{n+m=i} u_n v_m$$

例如

$$\begin{aligned}
 &(a_0, a_1) * (b_0, b_1, b_2) \\
 &= (a_0 b_0, a_0 b_1 + a_1 b_0, a_0 b_2 + a_1 b_1, a_1 b_2)
 \end{aligned}$$

注意疊積的長度是原序列長度之和減一。

對照 (1.50)，我們可以反轉 c_k 的順序，使得

$$\mathbf{c} = \left(\frac{1}{2}c_1, \frac{1}{2}c_0 \right)$$

是一組低頻濾波器。令 $\tilde{\mathbf{u}} = \mathbf{c} * \mathbf{v}$ 。若 \mathbf{v} 的長度是 N ，則 $\tilde{\mathbf{u}}$ 的長度是 $N + 1$ 。我們只需要 $\frac{N}{2}$ 筆低頻資料，所以還要經過 *down sampling* 的步驟。所謂 *down sampling*，就是從序列的第二個元素開始，每隔一個元素取一個出來，符號上記做 \downarrow 。例如 idx:down

$$\downarrow \begin{pmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

所以，用訊號處理的符號，凌波分解的低頻部分就是

$$\mathcal{L}v = \downarrow \mathbf{c} * v$$

而凌波分解的細節部分，在訊號處理的觀點中是一組搭配 \mathbf{c} 的高頻濾波器 (*high-pass filter*)。為什麼特別說是搭配 \mathbf{c} 的呢？因為如果只做低頻濾波而不做搭配的高頻濾波，則分解後的訊號不可能做完美的重建 (PR: Perfect Reconstruction)。因此一定要有一個高頻濾波器，而這個高頻濾波器必須爲了 \mathbf{c} 而特殊設計，其目的是這一對濾波器分解出來的訊號，將來可以被完美地重建。 idx:highpass

對照 (1.50)，我們可以反轉 d_k 的順序，使得

$$\mathbf{h} = \left(\frac{1}{2}d_1, \frac{1}{2}d_0 \right)$$

就是一組搭配 \mathbf{c} 的高頻濾波器。同樣地，用訊號處理的符號，凌波分解的細節部分 (或稱高頻部分) 就是

$$\mathcal{H}v = \downarrow \mathbf{h} * v$$

第二章

數學理論基礎

—— 複習與預習

這一章的目的是將這本書裡面所要用到的數學做個整理。這份整理當然不能細說從頭 (否則我們要從九九乘法表說起)，只能基於讀者具備大學二年級數學基礎的假設，擇要敘述。對於某些應該學過的知識，例如向量空間、基底、抽象的長度與內積、傅立葉級數，因為擔心各人以前學習的方式不同，所以還是簡短地複習。對於各種教材中比較一致的內容 (比如說矩陣與向量的乘法)，就不再重複。我們強調這些都是簡要複習，所以內容不會完備，也不會詳盡。

至於某些大三學生，或是工學院與專科同學，可能不曾接觸的數學，我們在此預習。這些主要包括了 L^2 空間、勒貝格積分和傅立葉轉換。這些課題各自在數學系都至少是一學期的課程。我們不可能在這裡說：「親愛的，我把數學變簡單了」。數學無帝王之路 (就是學習數學沒有捷徑的意思) 早在兩千年前就是至理名言。讀者若要認真學習上述知識，最好到數學系選修對應的課程 (實變函數論和富氏分析)，或是找一本專書自修 (我自己受教育的書是 Rudin [9] 和 Folland [4])。我在這一章裡的介紹方式，都是將抽象的觀念類比於比較初等、讀者

可能比較熟悉的觀念；比如說將 L^2 類比於 \mathbb{R}^2 (平面向量)。

某些非數學背景的讀者可能沒有機會真的去學習那些比較深的數學。對於這些讀者，我希望這一章有助於提高一點對數學的瞭解，並且有助於閱讀各人專業領域內的數學相關論文。對於這些讀者，只要您在瞭解了抽象化的目的和用處、以及其具體意義之後，以為自己懂了，那這一章的目的就算達到了。

2.1 基本符號與慣例

\mathbb{R} 代表實數所成的集合， \mathbb{Z} 代表整數， \mathbb{C} 代表複數。通常 $[a, b]$ 代表實數線上的一個閉區間，

$$[a, b] = \{x \mid a \leq x \leq b, x \in \mathbb{R}\}$$

(a, b) 代表開區間。 $x \in A$ 表示 x 屬於集合 A ，我們稱 x 是 A 的元素；而 $A \subset B$ 代表集合 A 包含於集合 B ，或稱集合 B 包含集合 A 。亦即

$$\text{if } x \in A \text{ then } x \in B$$

例如 $2 \in \mathbb{Z}$ ， $\mathbb{Z} \subset \mathbb{R}$ 。

$C^0([a, b])$ 代表定義域在 $[a, b]$ 內的連續函數所成的集合；如果不必強調定義域，或者定義域是 \mathbb{R} ，我們就簡記成 C^0 。而 $C^1([a, b])$ 是定義域在 $[a, b]$ 內而且其一次導函數連續的函數所成的集合，簡稱為一次可微連續。通常我們只在開區間 (a, b) 中定義導函數。所謂 $[a, b]$ 內的導函數，意思是在 (a, b) 內取導函數 $f'(x)$ ，而 $\lim_{x \rightarrow a^+} f'(x)$ 與 $\lim_{x \rightarrow b^-} f'(x)$ 存在。依此類推， $C^n([a, b])$ 代表定義域在 $[a, b]$ 內而且第 n 次導函數連續的函數所成的集合，簡稱為 n 次可微連續。

$\text{supp } f(x)$ 表示使得 $f(x) \neq 0$ 的最小閉集，稱為 $f(x)$ 的函蓋 (*support*)。如果 $\text{supp } f(x)$ 是有限長的區間，稱 $f(x)$ 為有限函蓋 (*compactly supported*)。

在本書內，不標明積分範圍的積分符號 \int 就代表在整個 \mathbb{R} 上積分，也就是 $\int_{-\infty}^{\infty}$ 的意思。它不是一般微積分課本中不定積分的意思。同樣地，不標明足標範圍的連加符號 \sum_k 和連乘符號 \prod_k 就代表在所有整數 \mathbb{Z} 上求和或求積，也就是 $\sum_{k=-\infty}^{\infty}$ 和 $\prod_{k=-\infty}^{\infty}$ 的意思。而 $\int_{-\infty}^{\infty}$ 是依照廣義積分 (或譯取積分，原文是 *improper integral*) 而定義。同樣的道理，事實上沒有任何整數可以 $= \pm\infty$ 。所以 $\sum_{k=-\infty}^{\infty}$ 的實際意思是

$$\lim_{N \rightarrow \infty} \sum_{k=-N}^N$$

當連加或連乘式中的足標明顯只有一個的時候，我們甚至可以省略足標符號；例如 $\sum c_k \phi(2x - k)$ 即表示對參數 k 求和。

連加式中的足標變換是本書中經常使用的基本動作，讀者必須有所掌握。例如

$$\phi(x - n) = \sum_k c_k \phi(2x - 2n - k) = \sum_k c_{k-2n} \phi(2x - k) \quad (2.1)$$

習題 2.1 證明 (2.1)。

習題 2.2 證明

$$\begin{aligned} & \sum_m \sum_n a_m b_n \int u(\xi) v(\xi - 2x + m - n) d\xi \\ &= \sum_k \sum_n a_{k+n} b_n \int u(\xi) v(\xi - 2x + k) d\xi \end{aligned}$$

2.2 泰勒展開

泰勒展開 (*Taylor expansion*) 是微積分提供給我們的一個超級工具。微積分課程中，必定不會忽略掉這個最重要的課題。但是當時未必適當地強調了這個工具的重要性。所以我們在這裡做簡單的複習。

泰勒是牛頓的最後一位弟子。他當初遇到一個多項式換參考點的問題。當他以微積分解決了這個問題之後，發現可以推廣到一般的用途。許多數學家跟著做了後續的發展；例如等一下我們要看到的誤差項，是 Lagrange 的寫法。讓我們追蹤泰勒的足跡，尋覓泰勒展開的動機。

idx:taylor

考慮一個多項式，例如

$$p(x) = a_3x^3 + a_2x^2 + a_1x + a_0$$

它記錄了一種 $x \mapsto y$ 的變化規律。這種記錄的方式，可以想像成相對於參考點 0 所寫成的數學式子。因為

$$a_3x^3 + a_2x^2 + a_1x + a_0 = a_3(x-0)^3 + a_2(x-0)^2 + a_1(x-0)^1 + a_0(x-0)^0$$

如果我們換成以 c 為參考點，那麼 $p(x)$ 還是可以寫成一個多項式形狀的數學式子，

$$p(x) = p_3(x-c)^3 + p_2(x-c)^2 + p_1(x-c) + p_0$$

如果 a_i 是已知，有沒有方便的辦法讓我們知道 p_i 呢？泰勒應該很容易地發現了 $p(c) = p_0$ (因為 a_i 是已知，所以 $p(c)$ 是可以計算的)。而泰勒身為牛頓的高足，想必是當時世界上最瞭解微積分的人之一。所以他想到了，因為

$$p'(x) = 3p_3(x-c)^2 + 2p_2(x-c) + p_1$$

所以 $p'(c) = p_1$ 。依此類推，因為 $p''(x) = (3 \times 2)p_3(x - c) + 2p_2$ ， $p'''(x) = (3 \times 2)p_3$ ，所以

$$p_2 = \frac{1}{2}p''(c), \quad p_3 = \frac{1}{3 \times 2}p'''(c)$$

用階乘符號寫出來，就是

$$p(x) = p(c) + \frac{1}{1!}p'(c)(x - c) + \frac{1}{2!}p''(c)(x - c)^2 + \frac{1}{3!}p'''(c)(x - c)^3$$

這就是我們在微積分課本裡面看到的泰勒展開的樣子。

如果 $f(x)$ 具有 n 次導函數，則

$$f(c) + \frac{1}{1!}f'(c)(x - c) + \frac{1}{2!}f''(c)(x - c)^2 + \cdots + \frac{1}{n!}f^{(n)}(c)(x - c)^n$$

稱為 $f(x)$ 以 c 為參考點的泰勒多項式。其中 $f^{(n)}(c)$ 是 $f(x)$ 在 c 的 n 次導數。 idx:taylor1

如果對所有的正整數 n ，對所有的 $x \in (a, b)$ ， $f^{(n)}(x)$ 都存在，我們稱 $f(x)$ 在 (a, b) 內無窮可微。例如多項式、 $\sin x$ 和 e^x 都是無窮可微，但是 $x^2 \sin \frac{1}{x}$ 在 $x = 0$ 處只有一次可微。其實還有許多不是無窮可微的例子，在這一章的最後要介紹的 p 階基本樣條函數，只有 $p - 2$ 次可微。

如果 $f(x)$ 在 (a, b) 內無窮可微，而 $c \in (a, b)$ ，則

$$\sum_{k=0}^{\infty} \frac{1}{k!}f^{(k)}(c)(x - c)^k$$

稱為泰勒級數。其中 $f^{(0)}(c) = f(c)$ ， $0! = 1$ 。如果泰勒級數在 (a, b) 內可以收斂到 $f(x)$ ，則它就稱為 $f(x)$ 在 (a, b) 內以 c 為參考點的泰勒展 idx:zero

開。這裡所謂的收斂是古典意義的逐點收斂，也是我們在微積分課程可能會學過的收斂意義。

泰勒級數可以和實數的十進位表達法做類比。我們已經熟悉實數的小數表達法，例如

$$1.41421356237309504880\dots$$

任取一個實數 x ，必定存在唯一的整數 n 使得 $x \in [n, n+1)$ 。讓我們只考慮 x 的非整數部分 \tilde{x} ，也就是 $\tilde{x} \in [0, 1)$ ， $x = n + \tilde{x}$ 。在十進位計數法中，當我們寫

$$\tilde{x} = 0.d_1d_2d_3\dots \quad \text{for } d_k \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

意思是

$$\tilde{x} = d_1 \frac{1}{10} + d_2 \frac{1}{100} + d_3 \frac{1}{1000} + \dots = \sum_{k=1}^{\infty} d_k \left(\frac{1}{10}\right)^k \quad (2.2)$$

eq:ten

也就是說，任何實數可以寫成 $n + \sum d_k \left(\frac{1}{10}\right)^k$ ($n \in \mathbb{Z}$) 的無窮級數形式 (未必有無窮多個非零的 d_k ，例如 $\frac{1}{2}$ 只有 $d_1 = 5$ 其他的 $d_k = 0$)。而且這個形式是唯一的。所謂的實數完備性保證了這些級數的收斂。或許比較懇切的說法是：實數完備性規定了這些級數必須收斂。

泰勒級數之於無窮可微函數就類似 (2.2) 之於實數。基本多項式 x 、 x^2 、 x^3 就像 (2.2) 中的 $\left(\frac{1}{10}\right)$ 、 $\left(\frac{1}{10}\right)^2$ 、 $\left(\frac{1}{10}\right)^3$ 。換了參考點的多項式 $(x-c)$ 、 $(x-c)^2$ 、 $(x-c)^3$ 就像是換成 N 進位實數表達中的 $\left(\frac{1}{N}\right)$ 、 $\left(\frac{1}{N}\right)^2$ 、 $\left(\frac{1}{N}\right)^3$ 。其中 $N \geq 2$ 是個整數。

例如 $\sin x$ 和 e^x 在 \mathbb{R} 上無窮可微，而且它們以任何 c 為參考點

的泰勒級數都收斂。[✓]* 但是並非所有的泰勒級數都收斂。而且，如果 $f(x)$ 在 (a_0, b_0) 中無窮可微，它的泰勒級數可能只在一個比較小的區間 (a, b) 內收斂。例如 $\ln x$ 在 $(0, \infty)$ 無窮可微，但是它以 1 為參考點的泰勒級數只有在 $x \in (0, 2]$ 之中收斂。

泰勒級數究竟收不收斂，必須要知道級數部分和的誤差項

$$f(x) - \sum_{k=0}^n \frac{1}{k!} f^{(k)}(c)(x-c)^k$$

是否隨著 $n \rightarrow \infty$ 而收斂到 0？而誤差項是什麼？這個問題在泰勒身故二十年之後才塵埃落定。現在常用的有兩種形式的誤差項表達方式，我們只用一種：如果 $f(x) \in C^n([a, b])$ ， $c \in (a, b)$ ，則以 c 為參考點的 $n-1$ 階泰勒多項式與 $f(x)$ 之間的誤差為

$$f(x) - \sum_{k=0}^{n-1} \frac{1}{k!} f^{(k)}(c)(x-c)^k = \frac{1}{n!} f^{(n)}(\xi)(x-c)^n \quad (2.3)$$

eq:Terr

其中 ξ 是 (a, b) 之間的某一個點。我們不知道 ξ 是誰，只知道存在某一個 ξ ，使得這個誤差等式成立。

常有初學者懷疑這個不確定的 ξ 是怎麼來的。這個誤差公式其實是微分均值定理 (*mean value theorem*) 的一個推廣。這個在直覺上非常簡單的定理，也該是微積分課程中的重要課題之一，只是它在當時也未必受到適當的重視。所謂微分均值定理就是，若 $f(x) \in C^1([a, b])$

* 其實， $\sin x$ 的級數展開在牛頓年輕的時候就知道了，那時候泰勒還沒有出生；只是牛頓的做法並不像後來的泰勒展開那樣的系統化。而 e^x 的級數展開是歐拉 (Euler) 的傑作，他也沒有使用泰勒展開這個工具，他是從 $e^x = \lim_{n \rightarrow \infty} (1 + \frac{x}{n})^n$ 這個定義下手做二項式展開得到的。

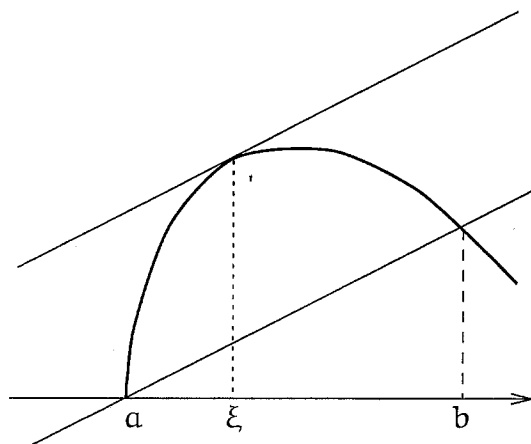


fig:mean 圖 10 微分均值定理圖示

則在 (a, b) 內必有一點的切線斜率等於 $f(x)$ 在 $[a, b]$ 內的割線斜率。用速度的觀念來說，運動過程中必有一剎那的瞬間速度等於平均速度。在直覺上，請看圖 10。從圖上可知，那個切線斜率等於割線斜率的一個點必定存在，但是我們理所當然不能確定它在哪裡？寫成數學表達式，就是

$$f'(\xi) = \frac{f(b) - f(a)}{b - a}, \quad \text{for some } \xi \in (a, b)$$

或者改寫成

$$f(b) = f(a) + f'(\xi)(b - a), \quad \text{for some } \xi \in (a, b)$$

在 (2.3) 中代入 $n = 1$ 、 $c = a$ 、 $x = b$ 就是微分均值定理。(代入 $c = a$ 這個假設是不合法的，因為 $c \in (a, b)$ ，但是我們只要假設 $f(x)$ 在 a 的左邊一點也是可微的就可以了。不去理會這個小節。)

我們對泰勒展開的介紹就到此為止。至於詳細而嚴謹的證明，請參考微積分課本，或是許多數值分析的課本也都會記載；例如 [6]。在

第一、三、五章中有許多 (2.3) 的應用，其實在 Haar 函數的凌波消散定理 (第一章十四節) 的證明中，我們已經用過一次。

2.3 向量空間

向量空間 (*vector space*) 被抽象化之後，其元素未必是我們在中學時代學習的那種平面或空間中的向量，所以又稱為線性空間 (*linear space*) 以免混淆視聽。無論如何，所謂抽象化的向量空間只是將平面向量之最純粹的特性萃取出來，定義其基本結構。因為是從平面向量的特性萃取出來的，所以即使是抽象化的向量空間，還是可以類比於平面向量，做具象化的圖示或思考。

在這本書裡，我們使用粗體斜字代表抽象化的、一般性的向量，像 \mathbf{v} 。而我們特別用類似中學生使用的 \vec{v} 符號代表平面向量。平面向量寫成 $\vec{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ 的時候， v_1 代表其 x -軸坐標分量， v_2 代表其 y -坐標分量。

所謂向量空間 \mathcal{V} 是一個數學結構，它包含一對集合 $(\mathcal{V}, \mathcal{F})$ 和它們彼此的關係。在下一頁的表格中，我們列出 \mathcal{V} 和 \mathcal{F} 的抽象性質，以及它們對應平面向量的具象意義。

表中的 (V.2) 稱作結合律，它的意思是，運算的順序不要緊。如果沒有 (V.6)-(V.9) 這四種關係，則 \mathcal{V} 和 \mathcal{F} 只是風馬牛不相及的兩個數學結構，是這四種關係使它們結合在一起成為向量空間。我們稱 \mathcal{V} 中的元素為向量；相對地我們稱 \mathcal{F} 中的元素為純量，但是經常我們還是說常數。因為 \mathcal{F} 總是 \mathbb{R} 或 \mathbb{C} ，所以常常不提。因此常將 $(\mathcal{V}, \mathcal{F})$ 簡記做 \mathcal{V} ，稱為向量空間。除非特別說明，我們總假設 $\mathcal{F} = \mathbb{R}$ 。

任何一個集合 \mathcal{V} ，只要符合了 (V.1)-(V.9) 這些條件，就叫做向量空間。最具體的向量空間是平面向量所成的集合，配合我們所熟知的

1. \mathcal{V} 是一個代數結構：交換群 (平面向量 \mathbb{R}^2 有加減運算)

	抽象性質	平面向量性質
	任意 $u, v, w \in \mathcal{V}$	任意 $\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \in \mathbb{R}^2$
(V.1)	$u + v \in \mathcal{V}$	$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} u_1 + v_1 \\ u_2 + v_2 \end{bmatrix} \in \mathbb{R}^2$
(V.2)	$(u + v) + w = u + (v + w)$	$\left(\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}\right) + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} u_1 + v_1 + w_1 \\ u_2 + v_2 + w_2 \end{bmatrix}$
(V.3)	$u + v = v + u$	$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$
(V.4)	$\exists! \mathbf{0} \in \mathcal{V}, v + \mathbf{0} = v$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \mathbb{R}^2, \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$
(V.5)	$\exists! (-v)$ 使得 $v + (-v) = \mathbf{0}$	$\exists! \begin{bmatrix} -v_1 \\ -v_2 \end{bmatrix}$ 使得 $\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} -v_1 \\ -v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

2. \mathcal{F} 是一個代數結構：體 (\mathcal{F} 就是實數 \mathbb{R} 或複數 \mathbb{C})

idx:1space

我們不必覆述 \mathbb{R} 或 \mathbb{C} 的運算結構。3. \mathcal{V} 和 \mathcal{F} 的交互作用。

	抽象性質	平面向量與實數性質
	任意 $v, w \in \mathcal{V}, \alpha, \beta \in \mathcal{F}$	任意 $\begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \in \mathbb{R}^2, \alpha, \beta \in \mathbb{R}$
(V.6)	$\alpha v \in \mathcal{V}; 1v = v; 0v = \mathbf{0}$	$\alpha \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \alpha v_1 \\ \alpha v_2 \end{bmatrix} \in \mathbb{R}^2$
(V.7)	$\alpha(\beta v) = (\alpha\beta)v$	$\alpha\left(\beta \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}\right) = (\alpha\beta) \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \alpha\beta v_1 \\ \alpha\beta v_2 \end{bmatrix}$
(V.8)	$\alpha(v + w) = \alpha v + \alpha w$	$\alpha\left(\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}\right) = \alpha \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \alpha \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$
(V.9)	$(\alpha + \beta)v = \alpha v + \beta v$	$(\alpha + \beta) \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \alpha \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \beta \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$

向量加法運算。我們用 \mathbb{R}^2 代表這個熟悉的空間。

idx:R2

若 $v \in \mathcal{V}$ ，不管它長得多麼不像個向量，我們都可以叫它向量。例如第一章中的 j 次階梯函數空間 V_j 就是個向量空間。例如

idx:Pn

$$\begin{aligned} P_n &= \{ \text{階數} \leq n \text{ 的多項式} \} \\ &= \left\{ \sum_{k=0}^n a_k x^k \mid a_k \in \mathbb{R} \right\} \end{aligned} \quad (2.4)$$

eq:Pn

也是一個向量空間。

如果 S 和 \mathcal{V} 各自是向量空間，而 $S \subset \mathcal{V}$ ，稱 S 是 \mathcal{V} 的一個向量子空間，或簡稱子空間。

習題 2.3 證明 P_n 是一個向量空間。

習題 2.4 若 $v_1, \dots, v_n \in \mathcal{V}$ 定義

$$S = \text{span} \{v_1, \dots, v_n\}$$

證明 S 是 \mathcal{V} 的一個向量子空間。

習題 2.5 證明任何通過原點的直線是 \mathbb{R}^2 的子空間。

習題 2.6 若 $n \leq m$ ，證明 P_n 是 P_m 的子空間。

習題 2.7 證明 (1.2) 定義的 V 是一個向量空間。

2.4 基底

任意 n 個向量 v_1, \dots, v_n 和 n 個純量 $\alpha_1, \dots, \alpha_n$ ，

$$\alpha_1 v_1 + \dots + \alpha_n v_n = \sum_{i=1}^n \alpha_i v_i$$

稱為 v_1, \dots, v_n 的一個線性組合 (*linear combination*)。如果 $x = \alpha_1 v_1 + \dots + \alpha_n v_n$ ，我們說 x 可以被 v_1, \dots, v_n 展開，而展開的係數是 $\alpha_1, \dots, \alpha_n$ 。

如果 $\{v_1, \dots, v_n\}$ 是一組向量，則它們所有的線性組合所成的集合為

$$\text{span}\{v_1, \dots, v_n\} = \left\{ \sum_{i=1}^n \alpha_i v_i \mid \alpha_i \in \mathbb{R} \right\}$$

令 $S = \text{span}\{v_1, \dots, v_n\}$ ，我們稱 S 為 v_1, \dots, v_n 所展成的空間。或者說 v_1, \dots, v_n 展開了空間 S 。很顯然地， S 仍是一個向量空間。

對於一組向量 $\{v_1, \dots, v_n\}$ ，如果找得到一組不全為零的純量 $\alpha_1, \dots, \alpha_n$ ，使得它們的線性組合是 $\mathbf{0}$ ：

$$\alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n = \mathbf{0}, \quad \text{with } \alpha_1^2 + \alpha_2^2 + \dots + \alpha_n^2 \neq 0$$

則我們說 $\{v_1, v_2, \dots, v_n\}$ 是線性相關 (*linearly dependent*)。線性相關的另外一個意思就是 v_1, \dots, v_n 之中的一個可以被其他 $n-1$ 個展開來。例如若 $\alpha_1 \neq 0$ 則

$$v_1 = \frac{\alpha_2}{\alpha_1} v_2 + \dots + \frac{\alpha_n}{\alpha_1} v_n$$

如果一組向量不是線性相關，就叫做線性無關或是線性獨立 (*linearly independent*)。也就是說，

$$\alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n = \mathbf{0} \quad \Rightarrow \quad \alpha_1 = \alpha_2 = \dots = \alpha_n = 0$$

在平面上，平行的向量就是線性相關，不平行的非零向量就是線性獨立。在三度空間中，共面的向量就是線性相關，不共面的非零向

量就是線性獨立。在 P_2 中， $\{1, x, x^2\}$ 是線性獨立， $\{1, x, x^2, (x-1)^2\}$ 是線性相關。

若 v_1, \dots, v_n 展開 S ，而且它們是展開 S 最精簡的可能：意思就是這 n 個向量只要少了一個就無法展開 S 。則稱 $\{v_1, \dots, v_n\}$ 為 S 的一組基底 (*basis*)。如果 $\{v_1, \dots, v_n\}$ 是一組線性相關的向量，它們就不可能最精簡，因為它們之中的其中一個可以被其他的向量展開。所以我們可以去掉那一個，而仍然展開 S 。可見能夠成為基底的一組向量必為線性獨立。

idx:basis

若向量空間 V 有一組 n 個向量的基底，我們稱 V 的維度是 n ，或者 V 是 n 維向量空間。若向量空間 V 有一組無窮多個向量的基底，我們稱 V 為無窮維空間。一般在線性代數課程中不考慮無窮維空間，因為無窮多個向量的線性組合好似無窮級數，就必須牽涉到級數的收斂問題。而收斂問題是數學分析的課題。

idx:dim

一個向量空間的基底並不唯一，但是維度是一定的。任何一組個數與維度相同的線性獨立向量，都可以成為一組基底。例如在平面上任何兩個不平行的向量都可以成為一組基底，在 P_2 上 $\{1, x, x^2\}$ 和 $\{1, (x-2), (x-2)^2\}$ 也都可以成為一組基底。但只要 $B = \{v_1, \dots, v_n\}$ 是 V 的一組基底，則對任意一個 $v \in V$ ，存在唯一的一組係數 $\alpha_1, \dots, \alpha_n$ 使得

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n$$

此時我們稱

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix}_B$$

是 v 在 B 基底下的表達陣列。當我們不強調其對應基底的時候，就把

idx:vecpre

足標 B 省略掉。就好像我們寫數字的時候，例如 11，我們總是心照不宣地假定是以十為底。但是其實它也可以是以二為底或以十六為底的數字。在這種特殊情況下我們總是會說明它的底數。

同樣的向量，在不同的基底下，會有不同的表達陣列。就好像同一個整數在不同的進位底數之下，會有不同的表達字串。將一個向量的表達陣列，從一組基底換到另一組基底，叫做基底轉換 (*change basis*)。基底轉換有可能使我們更清楚地瞭解向量的性質。基底轉換是線性代數發展出來的最重要工具之一。我們在第一章中說，幾乎所有的數位訊號處理，都要經過基底轉換的過程；這本書的主題—凌波函數，其實也是提供一種基底轉換的可能。讀者在數值分析或矩陣計算這些課題裡面學習的矩陣分解、矩陣對角化、矩陣的 Jordan 形式，以及解一個線性聯立方程組的條件數潤子 (preconditioner)，都可以看做是基底的轉換的應用。

idx:chbasis

例如

$$P_2 = \text{span}\{x^2, x, 1\} = \text{span}\{(x-1)^2, x-1, 1\}$$

則 $p(x) = x^2 - 2x + 1$ 在兩組基底下的表達陣列分別是

$$\begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} \quad \text{和} \quad \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

當我們把 $p(x)$ 寫成表達陣列的時候，它看起來就像一個 \mathbb{R}^3 中的向量。在這本書裡，我們總是把向量寫成行向量。我們所熟悉的 \mathbb{R}^3 中的向量，其實就是在自然基底之下的表達陣列。所謂 \mathbb{R}^3 的自然基底就是

$$\vec{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad \vec{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \vec{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

以後我們並不刻意去分別一個 n 維向量空間的表達陣列，和一個 \mathbb{R}^n 中的向量。只是要留意，抽象 n 維向量空間內的幾何意義（例如長度和角度）和 \mathbb{R}^n 中的向量，看起來並不相同（但在抽象的意義下還是一樣的）。我們在下面兩節介紹抽象化的長度和內積。

習題 2.8 證明若 $\{v_1, v_2, \dots, v_n\}$ 之中有一個向量是 $\mathbf{0}$ ，則它們是線性相關。

習題 2.9 證明以下三個向量是線性相關的：

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} \quad \begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix}$$

寫出第三個向量被前兩個向量展開的係數。

習題 2.10 證明以下的多項式是線性獨立：

$$1 \quad x \quad x^2 \quad x^3$$

習題 2.11 若 $p(x) = x^3 - 6x^2 + 4x - 9$ ，則 $p(x)$ 是 P_3 內的一個向量。試針對以下基底寫出 $p(x)$ 的表達陣列：

$$B_1 = \{1, x, x^2, x^3\} \quad B_2 = \{1, (x-2), (x-2)^2, (x-2)^3\}$$

(提示：泰勒展開。)

習題 2.12 令 V 為 (1.2) 所定義的向量空間。請討論它的維度是多少？

2.5 向量的長度

回顧第三節中向量空間的定義，並沒有提到什麼是向量的長度。所以抽象向量空間內的向量，未必具有長度的性質。長度這個觀念是額外

加在基本向量空間性質之上的。例如 \mathbb{R}^2 中已經有了長度的觀念，但是 P_2 裡面卻沒有直覺的長度觀念。這一節的主題，就是介紹一個抽象的長度觀念。

抽象的長度觀念是從平面向量的長度性質中萃取出來的。在平面向量之長度的所有性質中，有哪些是最精要的呢？若 $\vec{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ 為一平面向量，令

$$\|\vec{v}\| = \sqrt{v_1^2 + v_2^2} \quad (2.5) \quad \text{eq: Eunorm}$$

表示 \vec{v} 的長度，這樣定義的長度又稱為歐幾里得長度。數學家從歐幾里得長度裡面萃取出來的最精要性質是

$$(N.1) \quad 0 \leq \|\vec{v}\| < \infty, \forall \vec{v} \in \mathbb{R}^2$$

$$(N.2) \quad \|\vec{v}\| = 0 \iff \vec{v} = \vec{0}$$

$$(N.3) \quad \text{若 } \alpha \in \mathbb{C}, \|\alpha\vec{v}\| = |\alpha| \|\vec{v}\|$$

$$(N.4) \quad \|\vec{u} + \vec{v}\| \leq \|\vec{u}\| + \|\vec{v}\|, \forall \vec{u}, \vec{v} \in \mathbb{R}^2$$

其中 (N.1)–(N.3) 都是很淺顯明白的，只有 (N.4) 看起來不那麼基本。若考慮 \vec{u} 與 \vec{v} 是三角形任兩個邊的向量，則 $\vec{u} + \vec{v}$ 就是第三邊向量。所以 (N.4) 的意思就是說，三角形的兩邊之和不小於第三邊。這叫做三角不等式。在千錘百練之後，數學家發現這個不等式是長度性質的最精要部分之一。注意 (N.1) 是說長度必須是實值函數，它不可以有複數值。這是因為複數不能比大小（複數的絕對值可以比大小，但是複數不可以）。有了 (N.1)–(N.4) 這些最基本性質之後，所有的長度性質都可以被證明出來。我們在習題中練習幾個。

所以我們就用這四個最基本性質來定義抽象的長度。若 \mathcal{V} 是一個向量空間，只要有一個函數 $v \in \mathcal{V} \mapsto \|v\|$ ，而且 $\|v\|$ 符合 (N.1)–(N.4) 的性質（把 \vec{v} 換成 v ，把 \mathbb{R}^2 換成 \mathcal{V} ），我們就稱 $\|\cdot\|$ 是 \mathcal{V} 上面的一個長度，而 \mathcal{V} 就稱為一個具有長度性質的向量空間 (*normed linear*

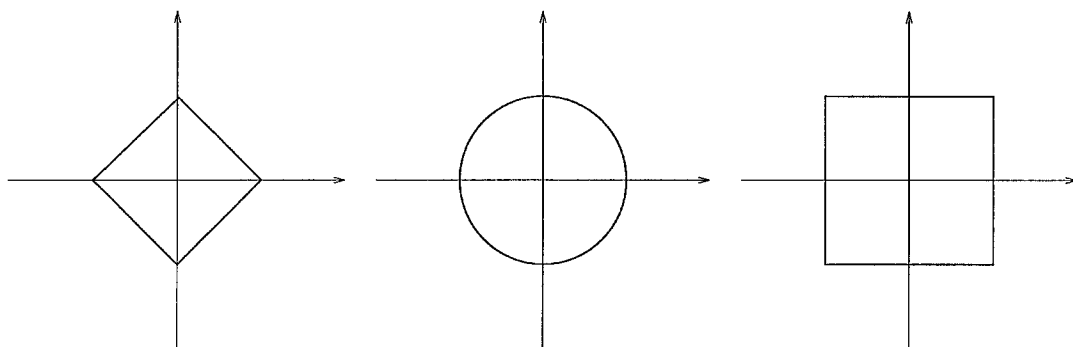


fig:circls 圖 11 l^1 、 l^2 、 l^∞ 長度意義之下的單位圓

$space$)。抽象的長度函數又稱為模 ($norm$)。

idx:norm

在 \mathbb{R}^2 上，除了歐幾里得長度，我們還可以定義其他種類的長度。例如 l^1 長度和 l^∞ 長度是如此定義：

$$\|\vec{v}\|_{l^1} = |v_1| + |v_2|, \quad \|\vec{v}\|_{l^\infty} = \max\{|v_1|, |v_2|\}$$

歐幾里得長度又記做 l^2 長度。

在抽象的長度定義之下，平常我們熟悉的幾何就會不一樣了。有時候會出現奇異而有趣的現象。例如，我們定義平面上的單位圓 (unit circle) 是

$$S = \{\vec{v} \in \mathbb{R}^2 \mid \|\vec{v}\| = 1\}$$

當 $\|\cdot\|$ 是 l^2 長度意義的時候， S 在平面上的圖形的確是我們所熟悉的圓。但是如果 $\|\cdot\|$ 是 l^1 長度意義，這個圓看起來像菱形，如果 $\|\cdot\|$ 是 l^∞ 長度意義，這個圓看起來像正方形。請看圖 11。請讀者檢查這張圖的正確性。

如果我們在 $[a, b]$ 區間內考慮多項式空間 P_n ，則可以仿照 (1.16) 定義 P_n 的長度：

$$\|p(x)\| = \left(\int_a^b |p(x)|^2 dx \right)^{\frac{1}{2}} \text{ for } p(x) \in P_n \quad (2.6) \quad \text{eq:pnorm}$$

因爲 $p(x) \in P_n$ ，所以 $p(x)$ 必然是 $[a, b]$ 內的連續函數，所以 (N.1) 成立了。如果 $\|p(x)\| = 0$ ，則 $\int_a^b |p(x)|^2 dx = 0$ 。但 $|p(x)|^2$ 是非負連續函數。如果有任意一個點 ξ 使得 $|p(\xi)|^2 = \eta > 0$ ，就會有一小段區間 $(\xi - h, \xi + h)$ 使得 $|p(x)|^2$ 在整個區間上 $> (\eta/2)$ ，因此 $\int_a^b |p(x)|^2 dx > \eta h > 0$ 。這就產生了矛盾，可見不會有任何點 ξ 使得 $|p(\xi)|^2 > 0$ 。也就是說，所有的 $x \in [a, b]$ 都滿足 $|p(x)|^2 = 0$ 。所以 $p(x) = 0$ 。這樣就證明了 (N.2) 成立。(N.3) 是顯然成立的，底下討論 (N.4)。

對任何 $p(x), q(x) \in P_n$ ，

$$\begin{aligned} \|p + q\|^2 &= \int_a^b |p + q|^2 dx \\ &= \int_a^b |p|^2 dx + 2 \int_a^b |p q| dx + \int_a^b |q|^2 dx \\ &= \|p\|^2 + 2 \int_a^b |p| |q| dx + \|q\|^2 \end{aligned}$$

只要 $\int_a^b |p| |q| dx \leq \|p\| \|q\|$ ，上面的式子就可以繼續寫成

$$\|p + q\|^2 \leq \|p\|^2 + 2\|p\| \|q\| + \|q\|^2 = (\|p\| + \|q\|)^2$$

我們就得到了三角不等式，(N.4) 也就成立了。

性質 2.1 若 $\int_a^b f(x)g(x) dx$ 、 $\int_a^b f(x)^2 dx$ 與 $\int_a^b g(x)^2 dx$ 都存在，則

prop:CSineq

$$\int_a^b f(x)g(x) dx \leq \sqrt{\int_a^b f(x)^2 dx} \sqrt{\int_a^b g(x)^2 dx}$$

證明：爲了方便，令 $A = \int_a^b f g dx$ 、 $B = \int_a^b f^2 dx$ 與 $C = \int_a^b g^2 dx$ 。對任意 $\alpha \in \mathbb{R}$ ，

$$0 \leq \int_a^b (f - \alpha g)^2 dx = B - 2\alpha A + \alpha^2 C, \text{ for all } \alpha \in \mathbb{R}$$

若 $B = 0$ 或 $C = 0$ 則 $f(x) = 0$ 或 $g(x) = 0$ ，欲證的不等式自然成立。若 B 與 C 皆不為 0，則特別取 $\alpha = A/C$ 上式依然成立，亦即

$$0 \leq B - 2\frac{A}{C}A + \left(\frac{A}{C}\right)^2 C = B - \frac{A^2}{C}$$

所以 $A^2 \leq BC$ ，得證。 \square

在性質 2.1 中代入 $f(x) = |p(x)|$ ， $g(x) = |q(x)|$ ，就得到我們想要的_{不等式}，因此證明了 (2.6) 所定義的 $\|\cdot\|$ 的確是 P_n 上面的一個長度，所以 P_n 是一個具有長度性質的向量空間。但是讀者們請注意，在整個驗證 (N.1)–(N.4) 的過程中我們其實沒有用到 $p(x)$ 和 $q(x)$ 的任何多項式性質。我們需要的只是它們的長度積分是可積的。所以，同樣的定義還可以用在所有閉區間的連續函數上 (如同第一章的 V 和 (1.16))。

就如同 \mathbb{R}^2 中有不同定義的長度， P_n 中也可以有不同定義的長度。像 (1.16) 和 (2.6) 之中定義的函數長度，我們稱為 L^2 長度。對應 l^1 和 l^∞ 還有 L^1 和 L^∞ 長度：

$$\|f(x)\|_{L^1} = \int_a^b |f(x)| dx, \quad \|f(x)\|_{L^\infty} = \max_{x \in [a,b]} |f(x)|$$

但是在這本書裡，我們不用其他的長度。在下一節中會述及原因。

最後，我們介紹方陣的長度。在第五章六節中，我們將會用到方陣的長度。一個 $n \times n$ 方陣形成一個 $\mathbb{R}^n \rightarrow \mathbb{R}^n$ 的線性映射。如果它的值域還是 \mathbb{R}^n ，我們稱該方陣為非奇異 (*non-singular*) 或是可逆 (*invertable*)。如果它的值域是維度 $< n$ 的空間，我們稱該方陣奇異或不可逆。奇異矩陣的長度定為 ∞ 。可逆矩陣的長度必有為有限，定義為

$$\begin{aligned} \|A\| &= \max\{\|Ax\| \mid \text{for all } \|x\| = 1\} \\ &= \max\left(\frac{\|Ax\|}{\|x\|}\right) \end{aligned} \tag{2.7}$$

eq:matnorm

其中的向量長度 $\|x\|$ 可以是任何符合 (N.1)–(N.4) 的長度。但是通常我們取 ℓ^2 長度 (歐幾里得長度)。在平面與空間幾何上， $\|x\| = 1$ 的向量構成單位圓或單位球，被映射後的 Ax 構成一個橢圓或橢球。橢圓或橢球上最長的半徑就是 $\|A\|$ 。根據矩陣長度的定義，

$$\|Ax\| \leq \|A\| \|x\|$$

所以如果 A 是可逆的，則 $x \mapsto Ax$ 的函數是連續函數。

習題 2.13 若 \mathcal{V} 是一個具有長度性質的向量空間，若 $x, y \in \mathcal{V}$ ，證明

$$\|x - y\| = \|y - x\|$$

習題 2.14 若 \mathcal{V} 是一個具有長度性質的向量空間，若 $u, v, w \in \mathcal{V}$ ，證明

$$\|u - v\| \leq \|u - w\| + \|v - w\|$$

習題 2.15 若 \mathcal{V} 是一個具有長度性質的向量空間，若 $x, y \in \mathcal{V}$ ，證明

$$\left| \|x\| - \|y\| \right| \leq \|x - y\|$$

這個性質對應平面上三角形的什麼性質？

習題 2.16 請檢查 $\|\cdot\|_{\ell^1}$ 符合 (N.1)–(N.4) 的性質。

習題 2.17 請問在 ℓ^1 長度和 ℓ^∞ 長度的意義之下，平面上單位圓的周長是多少？

習題 2.18 請驗證 (2.7) 中的可逆方陣之長度定義符合 (N.1)–(N.4) 四個條件。(提示： Ax 是一個向量，它的長度已經符合那四個條件。)

2.6 向量的內積

回顧第三節中向量空間的定義，並沒有提到什麼是兩個向量的夾角，因此也沒有什麼垂直或平行的觀念（只有線性相關或獨立的觀念）。可見角度這個觀念也是額外加在基本向量空間性質之上的。根據我們從平面向量得到的經驗，角度可以從內積得到。用中學生的符號，就是

$$\theta = \cos^{-1} \left(\frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \|\vec{v}\|} \right)$$

其中 θ 是 \vec{u} 與 \vec{v} 的夾角。如果 $\vec{u} = \vec{v}$ ，則 $\theta = 0$ 故

$$\|\vec{u}\| = \sqrt{\vec{u} \cdot \vec{u}}$$

這一節的主題，就是介紹一個抽象的內積觀念。

從平面向量的內積觀念，我們發現角度與長度都可以從內積定義出來。所以內積是個比較強的性質。具有長度性質的向量空間，未必具有內積性質。但是具有內積性質的向量空間，必定具有長度性質。我們的抽象內積，也希望保持這個特性。我們用 $\langle \cdot, \cdot \rangle$ 作為抽象內積的符號。

在平面向量的內積性質中，萃取最精要的部分出來，就成為抽象內積的條件。令 $(\mathcal{V}, \mathbb{C})$ 是一個向量空間， $u, v, w \in \mathcal{V}$ 。注意我們現在暫時假設純量是複數。如果 $\langle \cdot, \cdot \rangle : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{C}$ 符合以下性質：

$$(I.1) \quad \langle u, v \rangle = \overline{\langle v, u \rangle}$$

$$(I.2) \quad \langle u + v, w \rangle = \langle u, w \rangle + \langle v, w \rangle$$

$$(I.3) \quad \langle \alpha u, v \rangle = \alpha \langle u, v \rangle$$

$$(I.4) \quad \langle u, u \rangle \in \mathbb{R}^+ \cup \{0\}, \text{ 而且若 } \langle u, u \rangle = 0 \text{ 必定 } u = 0。$$

則稱 $\langle \cdot, \cdot \rangle$ 是 \mathcal{V} 上的一個內積。若 $\alpha \in \mathbb{C}$ 則 $\bar{\alpha}$ 代表 α 的共軛複數。

以上是一般的教科書裡面的定義。我們解釋那個共軛複數是怎麼進來的。舉一個簡單的例子：令 $\mathcal{V} = \mathbb{C}$ 。則 $u \in \mathcal{V}$ 可以寫成 $u = u + iv$ 這種形式，其中 $i = \sqrt{-1}$ 、 $u, v \in \mathbb{R}$ 。由於 (I.4) 這項規定，我們不能將 $\langle u, u \rangle$ 定義成 u^2 ；因為 $u^2 = (u^2 - v^2) + i(2uv) \in \mathbb{C}$ 。我們必須定義 $\langle u, u \rangle = u \bar{u}$ ；因為 $u \bar{u} = (u + iv)(u - iv) = u^2 + v^2 \in \mathbb{R}^+ \cup \{0\}$ 。所以，一般的情況下，我們定義複數的內積為

$$\langle u, v \rangle = u \bar{v}$$

既然如此，則 $\langle u, v \rangle = u \bar{v} = \overline{v \bar{u}} = \overline{\langle v, u \rangle}$ 。這就解釋了為什麼 (I.1) 中會出現共軛複數符號。

由 (I.1) 和 (I.2) 可知 $\langle u, v + w \rangle = \langle u, v \rangle + \langle u, w \rangle$ 。這個性質和 (I.2) 合稱 $\langle \cdot, \cdot \rangle$ 是雙線性算子 (*bilinear form*)。由 (I.1) 和 (I.3) 知 $\langle u, \alpha v \rangle = \bar{\alpha} \langle u, v \rangle$ 。(I.4) 是爲了使得 $\langle v, v \rangle = \|v\|^2$ 能夠成立，而量身訂作的一條規定。

仿照 (1.17) 中連續函數的內積定義，若 $f(x)$ 與 $g(x)$ 是 $[a, b] \rightarrow \mathbb{C}$ 的函數，定義其內積為

$$\langle f(x), g(x) \rangle = \int_a^b f(x) \overline{g(x)} dx \quad (2.8) \quad \text{eq:ip}$$

根據性質 2.1，只要 f 與 g 是平方可積的函數，則它們的內積存在。而

$$\langle f(x), f(x) \rangle = \int_a^b f(x) \overline{f(x)} dx = \int_a^b |f(x)|^2 dx = \|f(x)\|^2 \quad (2.9) \quad \text{eq:norm}$$

席瓦茲不等式 (*Schwarz inequality*) 若 $\langle \cdot, \cdot \rangle$ 是 \mathcal{V} 的內積。定義 $\|v\|^2 = \langle v, v \rangle$ ，則

$$|\langle u, v \rangle| \leq \|u\| \|v\|, \quad \forall u, v \in \mathcal{V}$$

這個不等式的證明，只需要 $\langle \cdot, \cdot \rangle$ 符合 (I.1)–(I.4) 四個條件，不必訴諸於向量或積分的運算形式。而它的證明方法與性質 2.1 的證明幾乎完全相同。所以留作練習。

如果我們所考慮的函數都是實數值的，而且不使用複數純量，則一切計算都在實數中進行。所以就不需要那些共軛複數的符號。如同第一章的 (1.17)。

如果向量空間 \mathcal{V} 內可以定義一個內積，我們稱 \mathcal{V} 是一個具有內積性質的向量空間 (*inner product linear space*)。如果 \mathcal{V} 具有內積性質，則它必定具有長度性質。反過來未必成立。例如我們在前一節定義的 L^1 長度與 L^∞ 長度，就無法找到對應的內積 $\langle \cdot, \cdot \rangle_1$ 與 $\langle \cdot, \cdot \rangle_\infty$ ，使得 $\langle v, v \rangle_1 = \|v\|_{L^1}^2$ 與 $\langle v, v \rangle_\infty = \|v\|_{L^\infty}^2$ 。所以，具有 L^1 長度或 L^∞ 長度性質的向量空間，就不會具有內積性質。這就是為什麼在前一節之末，我們聲明這本書裡面只考慮 L^2 長度。

我們還可以從內積定義 u 與 v 之間的夾角是

$$\theta = \cos^{-1} \frac{\langle v, w \rangle}{\|v\| \cdot \|w\|}$$

但是在抽象數學中很少需要角度，我們通常只需要一種角度：垂直的角度。當 $\langle u, v \rangle = 0$ ，我們就說 u 和 v 垂直或正交，記做 $u \perp v$ 。在前一節我們看到，改變了長度定義之後，單位圓的外形完全走了樣。現在也應該理解，改變了內積定義之後，所謂垂直的兩個向量，外觀上也未必是在平面幾何上所呈現的垂直。但是在性質上卻保有垂直的特性。(舉例說明於後。)

性質 2.2 若 \mathcal{V} 是具有內積性質的向量空間， $\mathcal{S} \subset \mathcal{V}$ 是一個子空間。則對每個 $v \in \mathcal{V}$ 存在唯一在 \mathcal{S} 上的垂直投影 (*orthogonal projection*) $\mathcal{P}v$ 使得

$$\|v - \mathcal{P}v\| \leq \|v - x\|, \quad \forall x \in \mathcal{S}$$

idx:perp

prop:orthpr

證明：所謂在 S 上的垂直投影 $\mathcal{P}v$ 就是

$$\langle v - \mathcal{P}v, x \rangle = 0, \quad \forall x \in S$$

所以

$$\begin{aligned} \|v - \mathcal{P}v\|^2 &= \langle v - \mathcal{P}v, v - \mathcal{P}v \rangle = \langle v - \mathcal{P}v, v \rangle - \langle v - \mathcal{P}v, \mathcal{P}v \rangle \\ &= \langle v - \mathcal{P}v, v \rangle = \langle v - \mathcal{P}v, v \rangle - \langle v - \mathcal{P}v, x \rangle \\ &= \langle v - \mathcal{P}v, v - x \rangle \leq \|v - \mathcal{P}v\| \|v - x\| \quad \forall x \in S \end{aligned}$$

其中的 \leq 是因為席瓦茲不等式。約掉一個 $\|v - \mathcal{P}v\|$ ，就得證前述的不等式。而唯一性已經在第一章七節中證明過了。 \square

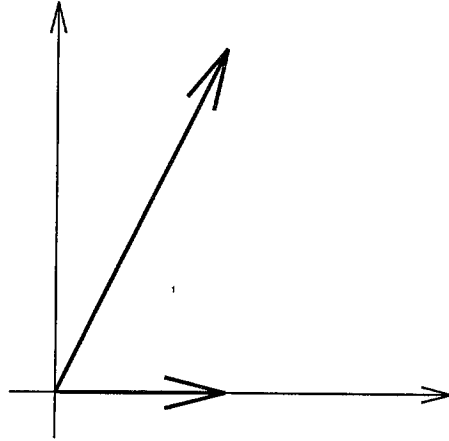
性質 2.2 就是說 v 在子空間中的垂直投影是唯一的，而且它是 v 在那個子空間中的最佳估計。回顧第一章九節中的闡述，對那個子空間中的生物而言，根本無法分辨 v 和 $\mathcal{P}v$ 的不同。一般來說，若有 u 與 v 符合

$$\langle u, x \rangle = \langle v, x \rangle, \quad \forall x \in S$$

則我們稱 u 與 v 在 S 中弱相等 (*equivalence in the weak sense*)。這是數學分析中常用的一個觀念，但是這本書還用不到它。

到目前為止我們還沒有看過不是 $u_1v_1 + u_2v_2$ 形式的內積定義。在這節的最後，讓我們舉一個例子，看看其他形式的向量內積。我們還是在 \mathbb{R}^2 上舉例。令 A 是 2×2 矩陣， $\vec{v} \in \mathbb{R}^2$ 。如果 A 是一個對稱正定 (*SPD: Symmetric and Positive Definite*) 的矩陣。意思是說 $A^T = A$ ，而且只要 $\vec{v} \neq \vec{0}$ ，必有 $\vec{v}^T A \vec{v} > 0$ 。若定義

$$\langle \vec{u}, \vec{v} \rangle_A = \vec{u}^T A \vec{v} \tag{2.10} \quad \text{eq:Aip}$$

fig:Aorth 圖 12 $\langle \cdot, \cdot \rangle_A$ 內積意義之下的正交向量

則讀者可以檢查 $\langle \cdot, \cdot \rangle_A$ 符合 (I.1)-(I.4) 是 \mathbb{R}^2 上的另外一個內積。一般的內積其實就是當 $\mathbf{A} = \mathbf{I}$ 的時候 (\mathbf{I} 是 2×2 單位矩陣)。

例如

$$\mathbf{A} = \begin{pmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 2 \end{pmatrix} \quad (2.11) \quad \text{eq:spdA}$$

則若 $\vec{v} \neq \vec{0}$,

$$\begin{aligned} \vec{v}^T \mathbf{A} \vec{v} &= (x \ y) \begin{pmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \\ &= x^2 - xy + 2y^2 = \left(\frac{1}{\sqrt{2}}x - \frac{1}{\sqrt{2}}y\right)^2 + \frac{1}{2}x^2 + \frac{3}{2}y^2 > 0 \end{aligned}$$

所以 \mathbf{A} 是對稱正定。因為

$$\begin{aligned} \left\langle \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 2 \end{bmatrix} \right\rangle_A &= (1 \ 0) \begin{pmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\ &= (1 \ -\frac{1}{2}) \begin{pmatrix} 1 \\ 2 \end{pmatrix} = 0 \end{aligned}$$

所以 $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ 與 $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ 在 $\langle \cdot, \cdot \rangle_A$ 內積的意義之下正交。在坐標平面上畫出 $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ 與 $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ ，它們顯然不同於我們所熟悉的垂直外觀。參照圖 12。

習題 2.19 只用內積的 (I.1)–(I.4) 四個性質，定義 $\|v\|^2 = \langle v, v \rangle$ ，證明席瓦茲不等式。

習題 2.20 證明只要內積的 (I.1)–(I.4) 四個性質，定義 $\|v\|^2 = \langle v, v \rangle$ ，則三角不等式 (N.4) 就會成立。

習題 2.21 證明 I 是對稱正定矩陣。

習題 2.22 若 A 是對稱正定矩陣，請驗證 (2.10) 中定義的內積真的符合 (I.1)–(I.4) 四個條件。

2.7 再談基底

我們在第一章已經談過正交基底和正則基底，例如方盒函數的脹縮平移 $\{\phi(2^j x - k)\}$ 可以形成 V_j 的一組正交基底，因為它們的長度不是 1 (是 $\frac{1}{2^j}$) 所以不成為正則基底。但是我們也知道，只要有一組正交基底，那麼正則基底也就垂手可得了：只要把每個基底向量的長度變成 1 就好了。(既然已經是正交基底，當然不包含 0 向量，所以它們的長度原不為 0 。) 在這一節中，我們再談談有關基底的課題。我們先討論無窮維線性空間的基底意義，再討論非正交基底的問題。

在這本書裡，我們不須討論無長度性質的無窮維向量空間。

若 \mathcal{V} 是一個具有長度性質的無窮維向量空間。我們說一組無窮多個向量 $B = \{v_1, v_2, \dots\}$ 是線性獨立的意思，就是在 B 中任選有限多個向量出來，這有限多個向量是線性獨立的。如果 B 是一組無窮多的線性獨立向量，對所有的 $v \in \mathcal{V}$ 和任意小的正實數 ϵ ，我們都找得到一

個正整數 N 和一組係數 $\alpha_1, \alpha_2, \dots, \alpha_N$ 使得

$$\|v - \sum_{k=1}^N \alpha_k v_k\| < \epsilon \quad (2.12)$$

eq:basiseps

(這就是為什麼我們需要長度觀念) 則稱 B 是 \mathcal{V} 的一組基底。

從 (2.12) 我們可以推論，若 B 是無窮維空間 \mathcal{V} 的一組基底，則對任意 $v \in \mathcal{V}$ ，找得到一個數列 (α_n) 使得

$$v = \sum_{n=1}^{\infty} \alpha_n v_n$$

其中那個等於符號的意義是 $\|v - \sum \alpha_n v_n\| \rightarrow 0$ 。

無窮級數的相加順序如果被改變了，未必會保持它的和。例如

$$\begin{aligned} \ln 2 &= 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots \\ &\neq (1 + \frac{1}{3} + \frac{1}{5} + \dots) + (\frac{1}{2} + \frac{1}{4} + \frac{1}{6} + \dots) = \infty - \infty \end{aligned}$$

所以，一般來說，無窮多的基底向量，其編號順序或許會造成不同。如果有一組無窮多的基底向量，不論如何重排其編號順序，結果都還是一組基底，而且，將對應的合成係數跟著重排，其級數仍然收斂到同一個向量，則我們稱之為無條件基底 (*unconditional basis*)。有限維空間的基底都是無條件基底。正則基底都是無條件基底。

idx:uncond

對於無窮多個向量的一組基底 B ，所謂 B 是正則基底的意思是，在 B 中任選兩個向量 v_i, v_j 都符合

$$\langle v_i, v_j \rangle = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

其中 δ_{ij} 稱為 Dirac δ 符號，我們會常見到它。

idx:delta

如果有一組無窮多的向量 $F = \{w_1, w_2, \dots\}$ ，它們並非線性獨立，但是它們可以展開 \mathcal{V} 。則稱 F 是 \mathcal{V} 的一組框架 (*frame*)。顯然框架不是展開 \mathcal{V} 的最精簡可能。所以對框架做分解，會出現多餘的資料 (*redundancy*)。多餘的資料雖然不經濟，但是在作訊號處理的時候，多餘的資料或許會使得處理的過程比較穩定。

idx:frame

如果 B 是正則基底， (α_k) 是一組係數。

$$\left\| \sum_{k=1}^{\infty} \alpha_k v_k \right\|^2 = \left\langle \sum_k \alpha_k v_k, \sum_k \alpha_k v_k \right\rangle = \sum_k |\alpha_k|^2$$

我們記 $\|(\alpha_k)\|_{\ell^2} = (\sum |\alpha_k|^2)^{\frac{1}{2}}$ ，所以 $\|\sum \alpha_k v_k\| = \|(\alpha_k)\|_{\ell^2}$ 。如果 F 是一組框架，對所有 $\|(\alpha_k)\|_{\ell^2}$ 有限的係數數列，若存在兩個實數 $0 < F_A \leq F_B < \infty$ 使得

$$F_A \|(\alpha_k)\|_{\ell^2} \leq \left\| \sum_k \alpha_k w_k \right\| \leq F_B \|(\alpha_k)\|_{\ell^2}$$

則 F_A 、 F_B 稱為框架 F 的上下界 (*frame bounds*)。正則基底也算是框架，它的上下界都 = 1。

idx:fbd

數學背景的讀者應該知道，所謂的無窮多可以分成可數的和不可數的無窮多。顯然我們在這裡討論的都是可數的無窮維向量空間。這些就夠用了。

我們已經知道正則基底的特殊好處就是，如果 v_1, \dots, v_n 是一組正則基底，任意 $v \in \mathcal{V}$ 可以很方便地分解與合成：

$$v = \sum_k \langle v, v_k \rangle v_k$$

其中從 v 到係數 $\langle v, v_1 \rangle, \dots, \langle v, v_n \rangle$ 的過程叫做分解，從係數做線性組合回到 v 的過程叫做合成。即使是無窮多維的正則基底，也同樣具有這個方便的功能。

idx:decomp

如果 v_1, \dots, v_n 不是一組正則基底，事情就沒有那麼簡單。所以我們引進對偶基底 (*dual basis*)。爲了行文方便，我們只描述有限維空間。無窮維空間的處理方式亦同。給定一組基底 v_1, \dots, v_n ，對任意的 $v \in \mathcal{V}$ ，存在唯一的一組純量 $\alpha_1, \dots, \alpha_n$ 使得 idx:dual

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_n v_n \quad (2.13) \quad \text{eq:valpha}$$

在一般的情況下，我們只有從一個線性系統求解 α_k 。將 (2.13) 的兩端分別與 v_1, \dots, v_n 做內積，得到 n 個線性聯立方程式

$$\sum_{i=1}^n \alpha_i \langle v_i, v_j \rangle = \langle v, v_j \rangle, \quad \text{for } j = 1, 2, \dots, n$$

若令

$$x = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} \quad b = \begin{pmatrix} \langle v, v_1 \rangle \\ \langle v, v_2 \rangle \\ \vdots \\ \langle v, v_n \rangle \end{pmatrix}$$

且 $M = (m_{ij})$ 爲一 $n \times n$ 矩陣： $m_{ij} = \langle v_i, v_j \rangle$ 。那麼 (2.13) 就可以用

$$Mx = b$$

求解 α_k 。可證 M 爲一對稱正定矩陣，所以它是非奇異矩陣，也就是可逆矩陣。這個 M 在有限元素法的術語中叫做質量矩陣 (*mass matrix*)。

其次的辦法是將基底向量 v_1, \dots, v_n 做正交化 (*orthogonalization*) 處理。一般線性代數教科書中都會列出所謂的 Gram-Schmidt 處理過程。它是一個將任何一組線性獨立向量正交化的方法。現在我們知道了質量矩陣，所以其實有另一個做法。因爲 M 是對稱正定，所以

存在一個矩陣，記做 $M^{\frac{1}{2}}$ ，它的每個元素都是實數而且它自己也是對稱正定，使得

$$M^{\frac{1}{2}}M^{\frac{1}{2}} = M$$

(對不起，不證明了； $M^{\frac{1}{2}}$ 就好像將 M 開根號一樣。) 我們取 $M^{\frac{1}{2}}$ 的逆矩陣 $M^{-\frac{1}{2}}$ 。定義一組新的向量 w_1, \dots, w_n ：

$$\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = M^{-\frac{1}{2}} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{pmatrix}$$

我們請讀者自己驗證，若 \widetilde{M} 是 w_1, \dots, w_n 的質量矩陣，亦即 $\widetilde{m}_{ij} = \langle w_i, w_j \rangle$ ，則

$$\widetilde{M} = M^{-\frac{1}{2}} M M^{-\frac{1}{2}} = I$$

這也就是說 $\langle w_i, w_j \rangle = \delta_{ij}$ ，所以 w_1, \dots, w_n 是一組正則基底。

第三種方法，就是替 v_1, \dots, v_n 找到它們的對偶基底 (*dual basis*)。所謂對偶基底是另外一組特別的基底，記做 $\tilde{v}_1, \dots, \tilde{v}_n$ 。而 v_k 與 \tilde{v}_k 之間存在交互正交的關係：

$$\langle v_i, \tilde{v}_j \rangle = \delta_{ij} \tag{2.14} \quad \text{eq:dualbasis}$$

可見所謂的對偶關係是平等相對的。我們也可以說 v_k 是 \tilde{v}_k 的對偶基底。如果找得到對偶基底，則 (2.13) 垂手可得：

$$\alpha_i = \langle v, \tilde{v}_i \rangle \tag{2.15} \quad \text{eq:alphadual}$$

一般來說，要找到對偶基底並不容易。但是某些非正交族類的凌波函數，就有一套方法求得對偶基底。這一節的最後，我們在 \mathbb{R}^2 上舉

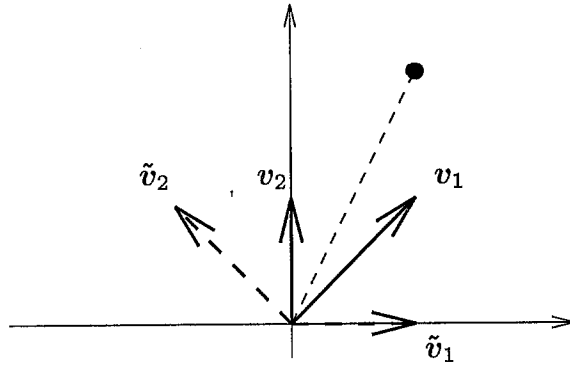


fig:dual 圖 13 平面上的一對對偶基底

個對偶基底的例子。讓我們取 $v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $v_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$; 這是在自然基底之下的表達陣列，通常我們是寫成 $\vec{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ 但是現在不方便這樣寫 (因為等一下要加 \sim)。令 $B = \{v_1, v_2\}$ 。明顯地 B 是 \mathbb{R}^2 的一組非正交基底。現在，我們要找 $\tilde{B} = \{\tilde{v}_1, \tilde{v}_2\}$ ，使得對任意的 $v \in \mathbb{R}^2$

$$v = \langle v, \tilde{v}_1 \rangle v_1 + \langle v, \tilde{v}_2 \rangle v_2$$

我們要找 \tilde{v}_1 ，它必須垂直於 v_2 而在 v_1 方向的投影是 1：

$$\langle \tilde{v}_1, v_1 \rangle = 1, \quad \langle \tilde{v}_1, v_2 \rangle = 0$$

我們要找 \tilde{v}_2 ，它必須垂直於 v_1 而在 v_2 方向的投影是 1：

$$\langle \tilde{v}_2, v_2 \rangle = 1, \quad \langle \tilde{v}_2, v_1 \rangle = 0$$

這時的內積是 \mathbb{R}^2 的標準內積，所以我們得到

$$\tilde{v}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \tilde{v}_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

讓我們試試 $v = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$:

$$\begin{aligned} v &= \langle v, \tilde{v}_1 \rangle v_1 + \langle v, \tilde{v}_2 \rangle v_2 \\ &= v_1 + v_2 \\ &= \langle v, v_1 \rangle \tilde{v}_1 + \langle v, v_2 \rangle \tilde{v}_2 \\ &= 3\tilde{v}_1 + 2\tilde{v}_2 \end{aligned}$$

用陣列表達來寫，就是

$$v = \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}_B = \begin{pmatrix} 3 \\ 2 \end{pmatrix}_{\tilde{B}}$$

參照圖 13。

習題 2.23 證明 (2.15)。

2.8 L^2 空間

在第一章中，我們留下一個非常微妙難解的問題。這個問題其實不太惹眼。那就是，我們當初在 (1.2) 中定義 V 是 $[0, 1]$ 中連續函數所成的集合。然後我們定義了多層解析空間 V_j 。當我們寫出它們層層相套的關係

$$V_0 \subset V_1 \subset V_2 \subset \dots$$

總是很心虛的在後面留下一個空白。究竟這些層層相套的空間最終是什麼空間呢？我們知道 $\cup V_j$ 在 V 裡面稠密，但是卻不能在 \dots 的右邊寫下 V ，因為 V_j 裡面的函數根本不是連續函數， $V_j \not\subset V$ 。此外，我們在第一章十四節內推導了數學理論，知道 $\phi(x)$ 和 $\psi_{jk}(x)$ 可以逼近任何 V 中的函數。運用前一節所揭示的無窮維空間之基底觀念，我們很

想說 $\phi(x)$ 和 $\psi_{jk}(x)$ 形成 V 的一組正則基底。但是我們不能，因為就差一件事： $\psi_{jk}(x)$ 根本不在 V 裡面，怎麼能說是 V 的基底呢？以前我們不知如何處理這個問題，而且一直逃避它。現在，我們要正視這個問題。

回顧第一章所推導的理論，我們究竟為什麼要假設 V 是閉區間內的連續函數？首先，要在 V_j 中找到最靠近 $f(x)$ 的階梯函數，必須 $f(x)$ 在每一段區間 $[\frac{k}{2^j}, \frac{k+1}{2^j}]$ 內是可積的，因此才有積分平均值。所謂可積就是積分值存在且 $< \infty$ 的意思。然後，為了談稠密性，必須計算長度，所以長度必須是有限的。根據長度的定義 (1.3) 或 (1.16)，我們需要 $|f(x)|^2$ 是可積的。再來，我們引入投影的觀念，必須計算內積，所以內積必須是有限的。所以，看起來我們需要平均值、長度和內積這三種積分是可積的。

但是在第一章十節中我們看到，抽象化之後，我們不需要經過像第一章三節那樣的討論。也毋須刻意去算積分平均值。因為所謂的積分平均值已經出現在正交投影的計算之中。回顧

$$\begin{aligned} \langle f(x), \phi_{jk}(x) \rangle \phi_{jk}(x) &= \int_0^1 f(x) \sqrt{2^j} \phi(2^j x - k) dx \sqrt{2^j} \phi(2^j x - k) \\ &= 2^j \int_{\frac{k}{2^j}}^{\frac{k+1}{2^j}} f(x) dx \phi(2^j x - k) = v_{jk} \phi(2^j x - k) \end{aligned}$$

所以，只要內積是可積的，平均值就可積了；因此現在我們只需要長度和內積這兩種積分是可積的。

最後，根據席瓦茲不等式 $|\langle f, g \rangle| \leq \|f\| \|g\|$ ，我們發現只要長度是可積的，內積也自然就是可積的了。其中的內積如 (2.8) 所定義，長度如 (2.9) 所定義。

在第一章中，我們為了簡化討論的對象，所以假設 $f(x)$ 是連續函數（這是我們比較熟悉的對象）。由於閉區間內的連續函數必定是可積

函數，所以它的長度有限。但是在抽象化之後，我們知道：任何一個函數空間 V ，只要所有 $f(x) \in V$ 的長度都是有限的，第一章的理論就可以推導出來，根本不需要 $f(x)$ 的連續性。所以關鍵不在於連續，而在於長度積分的可積性。

好了，現在我們知道誰是真命天子。我們真正需要的空間，是由所有長度有限的函數所組成的空間（長度乃是由 (2.9) 所定義）。雖然第一章中的階梯函數不是連續函數，但它們都是長度有限的。所以長度有限的空間包含了階梯函數。定義

$$L^2([a, b]) = \left\{ f(x) : [a, b] \rightarrow \mathbb{R} \mid \int_a^b |f(x)|^2 dx < \infty \right\} \quad (2.16) \quad \text{eq:Ltwo}$$

現在我們可以重述第一章的理論。令 V_j 為階梯函數空間，則

$$V_0 \subset V_1 \subset V_2 \subset \cdots \subset L^2([0, 1])$$

而且

$$\bigcup_{j=0}^{\infty} V_j \text{ 稠密於 } L^2([0, 1])$$

由於

$$V_j = W_{j-1} \oplus W_{j-2} \oplus \cdots \oplus W_0 \oplus V_0$$

所以

$$L^2([0, 1]) = V_0 \oplus \left(\bigoplus_{j=0}^{\infty} W_j \right)$$

而

$$\{\phi(x)\} \cup \{\psi_{jk}(x) \mid 0 \leq j < \infty, 0 \leq k \leq 2^j - 1\} \quad (2.17) \quad \text{eq:LtwoON}$$

形成 $L^2([0, 1])$ 的一組正則基底。對任意的 $f(x) \in L^2([0, 1])$ ，令 $\mathcal{P}_j f(x)$ 是它在 V_j 中的正交投影，則

$$\mathcal{P}_j f(x) = \sum_{k=0}^{2^j-1} \langle f, \phi_{jk} \rangle \phi_{jk}(x) = \langle f, \phi \rangle \phi(x) + \sum_{\ell=0}^{j-1} \sum_{k=0}^{2^\ell-1} \langle f, \psi_{\ell k} \rangle \psi_{\ell k}(x)$$

由於 Haar 函數的凌波消散定理

$$\langle f, \psi_{\ell k} \rangle = O(2^{-(1+\frac{1}{2})j})$$

所以可證

$$\|f(x) - \mathcal{P}_j f(x)\| = O(2^{-j}) \quad (2.18) \quad \text{eq:Lta}$$

類似於 (2.16) 我們可以定義

$$L^2(\mathbb{R}) = \left\{ f(x) : \mathbb{R} \rightarrow \mathbb{R} \mid \int |f(x)|^2 dx < \infty \right\} \quad (2.19) \quad \text{eq:LtwoR}$$

$L^2(\mathbb{R})$ 和 $L^2([a, b])$ 的性質稍有不同。若 $u(x) \in L^2([a, b])$ ，根據席瓦茲不等式

$$\int_a^b u(x) dx \leq \left(\int_a^b 1 dx \right)^{\frac{1}{2}} \left(\int_a^b u(x) dx \right)^{\frac{1}{2}} = \sqrt{b-a} \|u\| < \infty$$

所以 $u(x)$ 本身必定可以積分。但是如果 $u(x) \in L^2(\mathbb{R})$ 就不能這樣做了。其實只要在 $L^2([a, \infty))$ 或 $L^2((-\infty, b])$ 這樣的空間裡都不能這樣做。例如考慮 $g(x) = \frac{1}{x}$ ， $x \in [1, \infty)$ ，則

$$\int_1^\infty g^2 dx = \int_1^\infty \frac{1}{x^2} dx = \lim_{A \rightarrow \infty} -\frac{1}{x} \Big|_{x=1}^A = 1$$

所以 $\frac{1}{x} \in L^2([1, \infty))$ 。但是 $g(x)$ 在 $[1, \infty)$ 中卻不能積分。

當 $L^2(*)$ 的區間不要緊的時候，我們可以只寫 L^2 。

順便介紹一個常用的術語。若將 $f(x)$ 想做一個波或是一個訊號，則它的平方積分 $\int |f|^2 dx$ 在許多場合又稱為 $f(x)$ 的能量 (energy)。所以，在 L^2 中的函數又稱為能量有限函數。

從 (2.18) 得知 $\lim \|f(x) - \mathcal{P}_j f(x)\| = 0$ 。亦請參閱 (1.39)。這種收斂的意義，稱為 L^2 收斂。我們稱 $f(x)$ 是 $\mathcal{P}_j f(x)$ 在 L^2 意義下的極限 (*limit in the L^2 sense*)。 L^2 的收斂意義和古典的逐點收斂有些不同。所謂逐點收斂，是針對每一個定義域中的 x ，定義一個實數數列 $y_n = f_n(x)$ (每一個 x 產生一個 y_n 數列)，然後取 y_n 的極限。如果每一個 x 所製造的 y_n 數列都收斂，則我們稱 $f_n(x)$ 收斂。然後定義一個新的函數 $f(x)$ ，它把定義域中的 x 映射到 y_n 的極限，這個函數就叫做 f_n 的極限函數。記做

$$\lim_{n \rightarrow \infty} f_n(x) = f(x)$$

而所謂 L^2 收斂，則是說 $f_n(x) \in L^2$ ，存在某個 $f(x) \in L^2$ 使得 $f_n(x)$ 和 $f(x)$ 之間的距離趨近於 0。記做

$$\lim_{n \rightarrow \infty} \|f_n(x) - f(x)\| = 0$$

在 L^2 收斂的意義之下，有可能存在某個 \hat{x} ，使得 $\lim f_n(\hat{x}) \neq f(\hat{x})$ 。

例如，考慮函數數列 $f_n(x) = x^n$ ，顯然 $f_n(x) \in L^2([0, 1])$ 。因為

$$\lim_{n \rightarrow \infty} x^n = \begin{cases} 0 & \text{if } 0 \leq x < 1 \\ 1 & \text{if } x = 1 \end{cases}$$

如果我們令

$$\eta = \begin{cases} 0 & \text{if } 0 \leq x < 1 \\ 1 & \text{if } x = 1 \end{cases} \quad (2.20) \quad \text{eq:Leta}$$

顯然 $\eta(x) \in L^2([0, 1])$ 而且 $f_n(x)$ 在 $[0, 1]$ 內 (逐點) 收斂到 $\eta(x)$ 。所以 $\eta(x)$ 當然也是 $f_n(x)$ 在 L^2 意義下的極限。但是，在 L^2 意義下 $f_n(x)$ 還

有其他的極限。因為

$$\lim_{n \rightarrow \infty} \|f_n(x) - 0(x)\| = 0$$

其中 $0(x)$ 代表在 $[0, 1]$ 中全是 0 的函數，它是個連續函數。所以 $0(x)$ 也是 $f_n(x)$ 在 L^2 意義下的極限。甚至如果我們定義

$$\tilde{\eta}(x) = \begin{cases} 1 & \text{if } x \in \{0, \frac{1}{2}, 1\} \\ 0 & \text{otherwise} \end{cases}$$

則 $\tilde{\eta}(x)$ 又是另外一個 $f_n(x)$ 在 L^2 意義下的極限。依此要領，讀者可以製造無窮多種 $f_n(x)$ 在 L^2 意義下的極限。這豈不是說 $f_n(x)$ 在 L^2 意義下的極限不唯一嗎？這就違背了我們當初對極限的認識。其實並沒有違背，而其極限仍然是唯一的。既然要談在 L^2 意義下的極限，就要談在 L^2 意義下的相等。因為 $\|\eta(x) - 0(x)\| = 0$ ， $\|\tilde{\eta}(x) - \eta(x)\| = 0$ ，所以 $0(x)$ 、 $\eta(x)$ 和 $\tilde{\eta}(x)$ 其實在 L^2 的意義之下是相等的。它們都是 $f_n(x)$ 在 L^2 意義下的極限。

泰勒級數的收斂意義是逐點收斂。稍後要介紹傅立葉級數，它的收斂意義是 L^2 收斂。在這本書裡，除非特別說明，否則所謂的極限仍然是逐點收斂意義之下的極限。

idx:limit

在 (2.17) 中的那組正則基底，基本上是由一個函數 (Haar 函數) 的脹縮平移所組成的。從 1910 到 1985 之間，像這樣的正則基底只知道這麼一組 (Haar 獨領風騷七十五年)。由於 Haar 函數是不連續的，在 1985 年左右法國的學者 Yves Meyer 本來想要證明，如果一個函數的脹縮平移可以組成 L^2 的正則基底，則它必定不是連續函數。但是他證明不出來，卻找到了一組反例；也就是說，他意外地發現了一族連續函數 $\psi(x)$ ，這些 $\psi(x)$ 的脹縮平移也可以成為 L^2 的正則基底。凌波函數就這樣誕生了。

至此，有些讀者可能認為已經懂得 L^2 空間了：就是那些長度有限的函數所成的向量空間，它還具有內積性質。其實我們還缺了一個很緊要的細節。在這一節剩下的篇幅、並且延伸到下一節裡面，我們要進入一個純數學的領域去探討這個細節。礙於這本書的篇幅（和目的）限制，我只能很淺地介紹這個命題和它的動機。其實以下、以及下一節的內容，與本書的其他部分並沒有明顯關係；對於本書主要內容的瞭解，也未必有直接幫助。但是身為一個數學老師，我認為有義務向讀者們* 介紹這一段細節。

我們不斷地以平面向量空間 \mathbb{R}^2 作為抽象向量空間的類比。這一節定義了 L^2 函數空間，它將是這本書和許多應用數學所討論的空間。 L^2 空間和 \mathbb{R}^2 空間都是具有內積觀念的向量空間；只是前者的維度是無限大，後者的維度只有二。我們已經探討過， L^2 和 \mathbb{R}^2 在代數與幾何上的確有許多性質可以類比。現在我們要探討一項非常基本的分析性質：完備性 (*completeness*)。數學家如果希望 L^2 和 \mathbb{R}^2 能夠有完全的類比，必定不能失去這個性質。否則，數學分析中的許多課題，根本不能討論下去。

完備性要從實數說起。什麼特性使得實數與有理數不同？為什麼微積分建構在實數上而不是有理數？就是因為完備性。在實數上敘述完備性的方式之一是

漸增有上界的實數數列必收斂

舉例說明。定義數列 x_n ：

$$x_n = \max\{x^2 < 2 \mid x = 1 + d_1 \frac{1}{10} + d_2 \left(\frac{1}{10}\right)^2 + \cdots + d_n \left(\frac{1}{10}\right)^n\}$$

其中 $d_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

* 我假設大部分的讀者是大學二年級程度以上的工程師和科學家。對您來說，數學只是工具。但是您永遠不確定您的工具夠用了沒有？

說白話一點，其實就是

$$x_n = 1.b_1 b_2 \dots b_n \quad \text{其中 } b_n \text{ 是 } \sqrt{2} \text{ 的小數點下第 } n \text{ 位數字}$$

亦即 $x_1 = 1.4$, $x_2 = 1.41$, $x_3 = 1.414$, $x_4 = 1.4142$, \dots 。這個數列是漸增的： $x_1 \leq x_2 \leq x_3 \leq \dots$ ，而且有共同的上界： $x_n \leq 2$ 。這個數列中的每一項 x_n 在小數點下只有 n 項，所以是有理數。但是這個數列在有理數中不能收斂，因為很顯然 x_n 將要趨近於 $\sqrt{2}$ ，而 $\sqrt{2}$ 不是有理數。所謂完備性，就是規定像這樣的數列必須收斂。我們定義實數具有完備性；它真正想說的意義是，漸增有上界的數列一定有個極限存在於實數裡面。就是有了這個性質，我們寫 $\sqrt{2} = 1.4142\dots$ 的時候才有意義。因為當我們這樣寫的時候，其實是寫了一個級數

$$1 + 4\left(\frac{1}{10}\right) + \left(\frac{1}{10}\right)^2 + 4\left(\frac{1}{10}\right)^3 + 2\left(\frac{1}{10}\right)^4 + \dots$$

而 x_n 就是這個級數的 n 項部分和。實數的完備性保證 x_n 收斂，也就是保證級數 $1.4142\dots$ 收斂。所以 $1.4142\dots \in \mathbb{R}$ 而且 $\sqrt{2} \in \mathbb{R}^*$ 。

若 x_n 是個漸減有下界的實數數列：

$$x_1 \geq x_2 \geq x_3 \geq \dots \geq \xi$$

則 $-x_1 \leq -x_2 \leq \dots \leq -\xi$ ，所以 $-x_n$ 是漸增有上界的數列。根據完備性知道 $-x_n$ 會收斂。若 $\lim(-x_n) = -\eta$ ，則 $\lim x_n = \eta$ 。所以，如果空間具有完備性，漸減有下界的實數數列也是會收斂的。

* 有很長的一段時間，古代的西方數學家不承認 $\sqrt{2}$ 是個數（東方人比較實際一點，比較早接受 $\sqrt{2}$ 這個數），後來漸漸承認它是個很奇怪的東西使得 $x^2 = 2$ 有一個根。

連續函數空間 $C^0([0, 1])$ 中的函數都是有限長度的。所以它也可以成爲一個具有內積性質的向量空間。但是這個空間不具有完備性。例如考慮前面提過的函數數列 $f_n(x) = x^n$ ，很顯然地

$$f_1(x) \geq f_2(x) \geq f_3(x) \geq \cdots \geq 0, \quad \text{for all } x \in [0, 1]$$

可見 $f_n(x)$ 是個漸減有下界的函數數列，其極限是 (2.20) 中的 $\eta(x)$ 。但 $\eta \notin C^0([0, 1])$ 。所以 $f_n(x)$ 在 $C^0([0, 1])$ 之內不能有極限。也就是說， $C^0([0, 1])$ 不是一個完備的向量空間。在這個意義上，它並不能完全地與 \mathbb{R}^2 做類比。

既然 \mathbb{R} 具有完備性， \mathbb{R}^2 甚至 \mathbb{R}^n 當然也具有。但是實數集合與 $f: \mathbb{R}^n \rightarrow \mathbb{R}$ 的函數集合才能定義數列的漸增或漸減。這個觀念若要應用到更一般性的空間，例如 \mathbb{R}^2 吧，就需要將漸增有上界的觀念抽象化。我們觀察，若 x_k 是個漸增收斂到 ξ 的實數數列，則不管要求一個多小的正數 $\epsilon > 0$ ，必存在一個門檻 K 使得凡是足標 $n \geq K$ 之 x_n 必符合 $|x_n - \xi| < \epsilon$ 。根據這個道理，一定也找得到另一個 (可能更高的) 門檻 N 使得凡是足標 $n \geq N$ 之 x_n 必符合 $|x_n - \xi| < \frac{\epsilon}{2}$ 。那麼，任選兩個足標 $n \geq N$ ， $m \geq N$ ，則

$$|x_n - x_m| \leq |x_n - \xi| + |x_m - \xi| < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon$$

這就是一個可以萃取出來的抽象特性了。

在一個具有長度性質的向量空間 \mathcal{V} 中，考慮一個數列 $v_k \in \mathcal{V}$ 。如果不管要求一個多小的正數 $\epsilon > 0$ ，都找得到一個門檻 N 使得凡是足標 $n \geq N$ ， $m \geq N$ 之 v_n 與 v_m 必符合

$$\|v_n - v_m\| < \epsilon$$

則我們稱 v_k 是 \mathcal{V} 中的一個柯西數列 (*Cauchy sequence*)。在前一段中，我們已經利用三角不等式知道：舉凡收斂的數列都是柯西數列。但是一般而言柯西數列未必收斂。如果 \mathcal{V} 中的所有柯西數列都收斂 (都在 \mathcal{V} 中有極限)，則我們稱 \mathcal{V} 是一個完備向量空間。 idx:cauchy

從定義中得知，要談完備性必須要有長度性質。但是未必要有內積性質。數學術語稱具有長度性質的完備向量空間為 *Banach* 空間，稱具有內積性質的完備向量空間為希爾伯特空間 (*Hilbert space*)。 \mathbb{R}^2 是一個希爾伯特空間。凡是希爾伯特空間，其抽象性質常常可以與 \mathbb{R}^2 的性質做類比。 idx:BHspace

那麼， L^2 是不是希爾伯特空間呢？我們要它是。

數學家在求證的過程中發現，決定 L^2 是否具有完備性者，既不是長度的定義方式也不是收斂的定義方式；而是積分的定義方式！當我們算內積或是長度的時候需要積分，在微積分課程中學習的黎曼積分定義，不足以證明 L^2 是完備的。我們需要一個更廣泛的積分定義，使得 L^2 成為希爾伯特空間。這個積分的定義在二十世紀初由勒貝格 (*Lebesgue*) 獲得。其實 L^2 裡面的 L 就是紀念 *Lebesgue* 用的。

什麼是勒貝格的積分定義？且待下回分解。

習題 2.24 若 $\vec{v}_k \in \mathbb{R}^2$ 是一個柯西數列。證明它在 \mathbb{R}^2 中有極限。

習題 2.25 請討論 \mathbb{C} 是否具有完備性？

習題 2.26 若 $f_n(x) = x^n$ ， $x \in [0, 1]$ ， $n = 1, 2, \dots$ 。證明 $f_n(x)$ 是一個柯西數列。

習題 2.27 令 $\eta(x)$ 定義於 (2.20)。證明任何一個 $\tilde{\eta}(x) \in L^2([0, 1])$ ，只要 $\|\tilde{\eta}(x) - \eta(x)\| = 0$ ，則 $\tilde{\eta}(x)$ 是數列函數 $f_n(x) = x^n$ 在 $L^2([0, 1])$ 意義下的極限。

習題 2.28 找一個函數 $u(x)$ ，使得 $\int_0^1 u \, dx$ 存在，但是 $u \notin L^2([0, 1])$ 。

2.9 勒貝格積分

非數學背景的讀者，或許納悶：積分就是求函數曲線下的面積嘛，為什麼數學家要分黎曼 (Riemann) 積分還是勒貝格 (Lebesgue) 積分？難道在不同的定義之下，積分有不同的意義？會算出來不同的數字？不是的。不論我們採用什麼積分的定義，它的直覺意義仍然是求函數曲線下的面積。如果根據兩個定義去算某函數曲線下的面積，都算得出結果，那兩個結果必定相等。

問題是，有些函數，它所謂的函數曲線極端不合常理，那時候，可能根據黎曼的積分定義，卻算不出曲線下的面積，因此我們說它不存在。而我們的直覺卻說，那曲線下的面積必定是有限的，怎麼能不存在呢？積分定義的聖杯，應該要為每一個直覺上含括有限面積的區域決定一個數字。當然，這個數字還要符合我們對面積的直覺感受。

我們首先要看一個黎曼積分失敗的例子。

令 $Q = [0, 1] \cap \mathbb{Q}$ 是 $[0, 1]$ 之中所有有理數的集合。我們將 Q 的元素編號成爲*

$$Q = \{r_1, r_2, r_3, \dots\}$$

定義一個函數數列

$$g_0(x) = 0$$

$$g_1(x) = \begin{cases} 1 & \text{if } x = r_1 \\ 0 & \text{otherwise} \end{cases}$$

* 我們實在不能再進入細節了。請讀者接受這個事實：有理數可以一個一個地編號，但是實數不行。意思是說，如果一個一個將有理數編號，當號碼 $\rightarrow \infty$ 的時候終究會編完。但是無論用什麼方法去將實數編號，即使號碼 $\rightarrow \infty$ 也編不完。這通常是集合論的課題。

$$g_2(x) = \begin{cases} 1 & \text{if } x \in \{r_1, r_2\} \\ 0 & \text{otherwise} \end{cases}$$

...

$$g_n(x) = \begin{cases} 1 & \text{if } x \in \{r_1, r_2, \dots, r_n\} \\ 0 & \text{otherwise} \end{cases}$$

很明顯地 $g_n(x) \in L^2([0, 1])$ 而且 $g_1(x) \leq g_2(x) \leq g_3(x) \leq \dots \leq 1$ 。所以這是個漸增有上界的函數數列。而 $g_n(x)$ 的極限函數顯然是

$$g(x) = \begin{cases} 1 & \text{if } x \in \mathcal{Q} \\ 0 & \text{otherwise} \end{cases} \quad (2.21) \quad \text{eq:NR}$$

我們不能畫出 $g(x)$ 的函數圖形，只能憑想像了。黎曼的積分定義，將無法決定 $\int_0^1 g(x) dx$ 的值。

爲了觀察黎曼積分爲何失敗，我們只好很快地複習黎曼積分的定義。我們只考慮 $f(x)$ 定義在 $[a, b]$ 上的情形。對每個正整數 n ，令 $\Pi_n = \{x_0, x_1, \dots, x_n\}$ 其中 idx:Rieman

$$a = x_0 < x_1 < x_2 < \dots < x_n = b$$

則 Π_n 稱爲 $[a, b]$ 的一個 n 段 (未必是均勻的) 分割。顯然對每個 n 有無窮多種 n 段分割的方法。定義

$$|\Pi_n| = \max_k \{x_k - x_{k-1}\}$$

$\Pi_n, n = 1, 2, \dots$ 稱爲一個分割系列。顯然 $[a, b]$ 存在無窮多種分割系列。如果 $\lim |\Pi_n| = 0$ ，我們稱 Π_n 爲一個次均勻分割系列。對應 Π_n 的每一段分割 $X_k = [x_{k-1}, x_k]$ ，定義 Y_k 是 $f(X_k)$ 的值域。由於 $f(x)$ 未必是連續函數，所以 Y_k 未必是一個區間。在每個 Y_k 裡面任選一個數 η_k ，定義

$$R_n = \sum_{k=1}^n \eta_k |X_k| \in \mathbb{R}$$

其中 $|X_k|$ 就是 X_k 的長度 $x_k - x_{k-1}$ 。顯然一般來說對每個 X_k 有無窮多種選擇 η_k 的方法。如果對每一種可能的次均勻分割系列 Π_n ， $n = 1, 2, 3, \dots$ ，和 Π_n 裡面每一個 η_k 的可能選擇，所造成的所有數列 R_n 都收斂，而且收斂到同一個數 A ，則稱 $f(x)$ 在 $[a, b]$ 內 (黎曼) 可積，其積分值為 A 。記做

$$\int_a^b f(x) dx = A$$

現在，對於 (2.21) 中的 $g(x)$ ，我們只要選一個系列的 Π_n 就夠看出來不可積了。令 Π_n 是 $[0, 1]$ 的 n 段均勻分割。亦即 $|X_k| = \frac{b-a}{n}$ 。因為每個 X_k 裡面必定有有理數也有無理數，所以 $Y_k = \{0, 1\}$ 。當 n 是偶數的時候，我們在每個 Y_k 裡面選 $\eta_k = 0$ ，所以 $R_n = 0$ 。當 n 是奇數的時候，我們在每個 Y_k 裡面選 $\eta_k = 1$ ，所以 $R_n = \sum |X_k| = 1$ 。對應這一套選法，得到的 R_n 數列是

$$1, 0, 1, 0, 1, \dots$$

顯然這個數列不收斂。

根據黎曼的積分定義，因為無法決定 $\int_0^1 g(x) dx$ 的值，所以說它的積分不存在。但是，直覺上，整個 $g(x) \leq 1$ ，所以 $g(x)$ 和 x -軸在 $[0, 1]$ 內圍成的區域全部落在正方形 $[0, 1] \times [0, 1]$ 裡面。整個正方形的面積是 1，所以 $g(x)$ 所圍的面積最大也不過是 1，怎麼能說它的面積不存在呢？

接著，我們介紹勒貝格的想法。參照圖 14，黎曼積分的想法，基本上是在 x -軸上做切割；而勒貝格積分的想法，基本上是在 y -軸上做切割。令 $f: [a, b] \rightarrow Y$ 其中 $Y \subset \mathbb{R}$ 是 $f(x)$ 的值域。因為 $f(x)$ 未必是連續函數所以 Y 未必是一個區間；例如 $g(x)$ 的值域只有兩個點

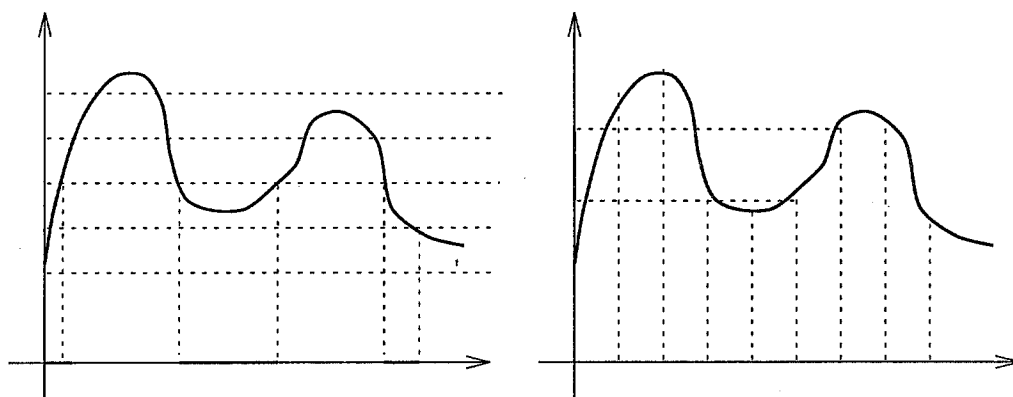


fig:Lebeg 圖 14 勒貝格積分 (L) 與黎曼 (R) 積分圖示

$\{0, 1\}$ 。我們只考慮 $Y \subset [c, d]$ 的情形。在 $[c, d]$ 上面取 n 段分割 $\Pi_n = \{\eta_1, \dots, \eta_n\}$ 其中

$$c = \eta_0 < \eta_1 < \eta_2 < \dots < \eta_n = d$$

對應 Π_n 的每一段分割 $Y_k = [\eta_{k-1}, \eta_k]$ 定義 X_k 是 Y_k 的反值域，亦即

$$X_k = f^{-1}(Y_k) = \{x \in [a, b] \mid f(x) \in Y_k\}$$

因為 $f(x)$ 未必是連續函數，所以 X_k 有可能是空集合。例如若 $Y_k = [\frac{3}{4}, \frac{5}{6}]$ ，則 $g^{-1}(Y_k) = \emptyset$ 。依照直覺定義 $|\emptyset| = 0$ 。令

$$E_n = \sum_{k=1}^n \eta_k |X_k|$$

如果對每一種可能的次均勻分割系列 Π_n ， $n = 1, 2, 3, \dots$ ，所造成的每一個數列 E_n 都收斂，而且收斂到同一個數 A ，則稱 $f(x)$ 在 $[a, b]$ 內 (勒貝格) 可積，其積分值為 A 。記做

$$\int_a^b f(x) dx = A$$

根據這個定義，我們可以推得

$$\int_0^1 g(x) dx = |Q|$$

現在，大傻推二傻，新的問題是： Q 的長度是多少？所以我們看到，勒貝格積分理論的關鍵，乃是在於一個實數子集合的長度如何定義。像 $[a, b]$ 這樣的子集合，長度顯然應該是 $b - a$ 。但是像 Q 這樣的集合呢？這一套理論稱為測度論 (measure theory)，而剛才所謂的長度，正確的術語是測度。若 $X \subset \mathbb{R}$ ， X 的測度記做 $m(X)$ 。 $m(\emptyset) = 0$ ，如果 X 只有有限多個點則亦 $m(X) = 0$ 。如果 X 是有限多個區間的聯集，則 $m(X)$ 就是這些區間長度的和。但是測度論可以處理更奇異的實數子集合，例如 $m(Q)$ 。測度論是實變函數論和機率論的基礎，屬於數學系高年級或研究所的課程。我們不能再深入細節了。

使得勒貝格的積分理論，比黎曼的積分理論更為成功的關鍵，就在於下面這個定理。

勒貝格收斂定理 若 $f_n(x)$ 與 $f(x)$ 都是在 $[a, b]$ 上的可積函數。如果 $|f_n(x)| \leq f(x)$ 而且 $f_n(x)$ 逐點收斂，則極限函數亦可積，並且

$$\int_a^b \lim_n f_n(x) dx = \lim_n \int_a^b f_n(x) dx$$

此處所說的可積都是指勒貝格可積。

因為我們沒有深入太多細節，所以上面定理的敘述比一般實變課本上的粗糙一點。但是這個定理的重點已經呈現。它的意思是說，如果有一個正值的函數 $f(x)$ ，它所圍的面積有限，而 $|f_n(x)| \leq f(x)$ 表示 $f_n(x)$ 被圍在 $\pm f(x)$ 之間，這塊區域的面積還是有限。那麼如果 $f_n(x)$ 收斂，其極限函數還是被圍在 $\pm f(x)$ 之間。在直覺上它應該是可積的，

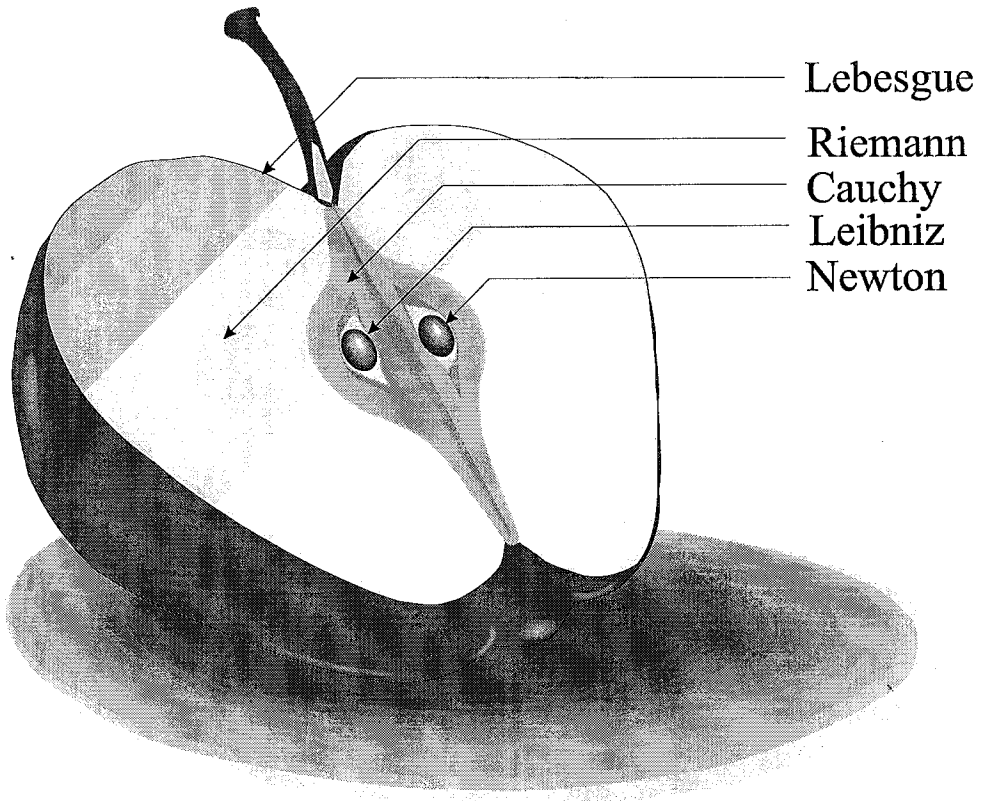


fig:apple 圖 15 積分理論發展之示意圖

而勒貝格理論也說它的確是可積的。這個定理說得更多的是，在此定理的假設條件下，極限與積分的步驟可以交換。

我們不能在此證明勒貝格收斂定理，也不能認真證明，在勒貝格積分的意義之下，所有 L^2 中的柯西數列必定收斂（在 L^2 的意義下收斂）。我們在此驟下結論：如果 L^2 空間中的內積與長度之積分都是勒貝格積分，則 L^2 空間是一個完備的向量空間，亦即 L^2 是希爾伯特空間。

提到微積分，就一定有牛頓。提到牛頓，少不了聯想到蘋果。如果我們將所有“應該”可積的函數，視為一個蘋果。那麼回顧積分學的發展史，在牛頓與萊布尼茲的時代，找得到反導函數的函數在有限區間內是可積的。這些可積函數相當於蘋果的種子。柯西的積分理論，使得所有閉區間內的連續函數都可積，即使找不到反導函數也行。例如 $\int_0^1 \sqrt{x^3+1} dx$ 。柯西理論的可積函數相當於蘋果的果核，它包含了種子。黎曼的積分理論，使得所有閉區間中的分片連續函數都可積，也可以討論在有限區間內發散的函數積分，例如 $\int_0^1 \frac{1}{\sqrt{x}} dx$ ，還可以討論無限區間內的積分，例如 $\int_1^\infty \frac{1}{x^2} dx$ 。黎曼理論的可積函數相當於蘋果的果肉，它包含了果核和種子。至此，大約 1860 年代，我們有了絕大部分的可積函數，就缺一層皮。勒貝格的積分理論，就包上了那層皮，它使得整個可積函數是完備的。

凡是黎曼可積，都是勒貝格可積，而且兩者的積分值相同。黎曼可積，卻找不到反導函數的函數，可以用數值積分來估計（通常是數值分析的課題，我們在第五章五節中稍有介紹）。而黎曼不可積，但勒貝格可積的函數呢？它就長得像 (2.21) 中的 $g(x)$ 。它應該不會出現在一般人的“例行工作”中。

您或許要問，數學家花了那麼大把勁，只爲了那一層皮嗎？嗯……如果沒有那層皮，蘋果就不完整了。數學家的審美眼光受不了這種不完整。我們如果不能瞭解他們，至少應該諒解他們必須這麼做。而且，比較有建設性的講法是，數學家花了那麼大把勁，才使得 L^2 中的任何柯西數列（或漸增有上界數列）必定有極限。

從現在開始，當我們說到積分的時候，都假設是根據勒貝格定義的積分。

習題 2.29 根據勒貝格積分的定義，證明 $\int_0^1 g(x) dx = m(Q)$ 。其中

$g(x)$ 如 (2.21) 所定義， $Q = [0, 1] \cap \mathbb{Q}$ 。

習題 2.30 利用勒貝格收斂定理求 (2.21) 中的 $\int_0^1 g(x) dx$ 。因此，可以推論 $m(Q)$ 是多少？

2.10 疊積

在這一節裡面，我們介紹兩種運算：離散和連續疊積。疊積之使用，有長久的歷史。例如 Weierstrass 提出他那著名的多項式逼近定理*時，就在證明步驟中使用了疊積。疊積運算有非常豐富的性質，而且也引發了許多數學上的問題。對這本書而言，疊積本身的理論並非目的，我們可以單純地視它為工具。

我們已經在第一章十八節見過離散疊積，現在要補充它的定義動機。離散疊積是依照多項式乘積的係數關係定義的。考慮兩個多項式

$$\begin{aligned} p(x) &= a_m x^m + a_{m-1} x^{m-1} + \cdots + a_1 x + a_0, \\ q(x) &= b_n x^n + b_{n-1} x^{n-1} + \cdots + b_1 x + b_0 \end{aligned}$$

若 $a_m \neq 0$ ， $b_n \neq 0$ ，則 $h(x) = p(x)q(x)$ 是一個 $m+n$ 階多項式。令

$$h(x) = c_{m+n} x^{m+n} + c_{m+n-1} x^{m+n-1} + \cdots + c_1 x + c_0$$

則係數 c_k 符合

$$c_k = \sum_{i+j=k} a_i b_j = \sum_{i=0}^k a_i b_{k-i} = \sum_{i=0}^k a_{k-i} b_i$$

* 若 $f(x)$ 是閉區間 $[a, b]$ 中的連續函數，任給 $\epsilon > 0$ ，存在一個多項式 $p(x)$ 使得 $\max |f(x) - p(x)| < \epsilon$ 。 亦即。 韋爾施特拉斯。

根據這個觀察，若 $u = (u_1, \dots, u_m)$ 、 $v = (v_1, \dots, v_n)$ 為兩個序列，我們定義 u 與 v 的離散疊積 (*discrete convolution*) 為 $x = u * v$ ，其中 $x = (x_k)$ 的元素為

$$x_k = \sum_i u_{k-i} v_i \quad (2.22)$$

只要經過簡單的足標變換，就可以證明離散疊積是可交換的。上面的式子裡面並沒有明述足標 n 的上下界。我們可以想像它是所有的整數。遇到 $i < 1$ 或 $i > n$ 的時候，我們就假設 $v_i = 0$ ；同理遇到 $i < 1$ 或 $i > m$ 的時候，我們就假設 $u_i = 0$ 。這種很自然的擴展方式，可以將所有有限多元素的序列，擴展成無窮多元素的數列。我們經常會使用這種方法，將一個有限長的數列擴展成有限個非零項的無窮數列。

而離散疊積的定義，對無窮數列也可以適用。只是 (2.22) 中的 \sum 就要做無窮級數的和了，要小心它未必收斂。比較簡單而確定的情況是，如果 (u_n) 與 (v_n) 是 ℓ^2 數列，意思是說

$$\|(u_n)\|_{\ell^2} < \infty \quad \text{其中} \quad \|(u_n)\|_{\ell^2} = \sqrt{\sum_n |u_n|^2}$$

則 $(u_n) * (v_n)$ 必定還是一個 ℓ^2 數列。而且 $(u_n) * (v_n) = (v_n) * (u_n)$ 。

離散疊積的定義可以很自然地推廣到函數的疊積。若 $f(x)$ 與 $g(x)$ 是 L^2 函數，定義它們的疊積 (*convolution*) 為

$$[f * g](x) = \int f(x-y)g(y) dy \quad (2.23)$$

由席瓦茲不等式，知道上式的積分存在。注意疊積的結果仍然是一個函數，可以簡記為 $[f * g](x)$ 或更簡為 $f * g$ 。經過變數變換，還是可以證明 $f * g = g * f$ 。因為經過一次積分，所以 $f * g$ 的函數圖形比 f 和

g 更平滑一點。一般來說，我們可以肯定 $\int [f * g](x) dx$ 是可積的，但是 $[f * g](x)$ 未必是個 L^2 函數。在最後一節，我們用疊積定義基本樣條函數，而每疊積一次，樣條函數的階數就提高一次。

做訊號處理的人，常常假設雜訊是以疊積的方式參雜在原始訊號裡面。比如說外太空訊號經過大氣層到達遙測雷達的時候，就受到大氣層的干擾。研究這類問題的其中一種數學模型是，假設原始訊號是 $f(x)$ ，假設雜訊是 $n(x)$ ，則接收到的訊號是 $r(x) = [f * n](x)$ 。我們的目標是從 $r(x)$ 和假設的 $n(x)$ 去求 $f(x)$ 。這種做法叫做反疊積 (*deconvolution*)。

但是凌波函數的雜訊處理有其他的模型和方法。在這本書裡，您並不會看到凌波與反疊積的關係。不過，有些人研究這方面的問題。疊積的性質就暫時說到這裡。在後面的三節中，它還會出現。

習題 2.31 證明疊積符合交換律： $[f * g](x) = [g * f](x)$ 。

習題 2.32 證明疊積符合分配律 $f * (\alpha g + \beta h) = \alpha(f * g) + \beta(f * h)$ 和結合律 $f * (g * h) = (f * g) * h$ (假設所需要的積分都存在)。再配合交換律，可見疊積可以被視為一種函數的乘法。在本節之初，我們曾說，疊積引發了一些數學問題。其中一個問題就是，對疊積這種乘法而言，有沒有單位元素？亦即，是否有一個函數 $\delta(x)$ ，使得 $f * \delta = \delta * f = f$ ？讀者可以自己想想這個問題，但是這本書並不會提供答案。

習題 2.33 計算以下的離散疊積： $(1, 1, 1, 1) * (1, 2, 3)$

習題 2.34 令 b_n 是 n 階二項式係數，亦即楊輝三角上面的第 $n + 1$ 列：

$$\begin{aligned} b_1 &= (1, 1) \\ b_2 &= (1, 2, 1) \\ &\vdots \\ b_n &= \left(\binom{n}{0}, \binom{n}{1}, \dots, \binom{n}{n} \right) \end{aligned}$$

idx:choose

證明對所有的 $n \geq 2$, $b_n = b_1 * b_{n-1}$ 。其中 $*$ 表示離散疊積。

2.11 傅立葉級數

傳統的應用數學，指的是將數學應用在物理方面。其實這也是牛頓、歐拉、高斯這些早期的人物所關心的應用問題（或者說這是他們那個年代所關心的問題）。這一類問題導引出來的數學模型，大致上是微分方程。基本上可以說，二十世紀前的應用數學都在處理自然科學的問題。在二十世紀的中段，大致上就是二次世界大戰之後，統計學與計算機科學的快速發展，帶動了像數學規劃、離散數學與圖論這些數學分支的發展。也將數學的應用範圍擴展到了社會科學的領域。而電子計算機的應用，必然產生離散的，數位化的新資料形態。可以說，數位化的聲音與影像，是最近三十年才發生的。這些新的資料形態產生了新的問題，它們需要新的方法來解決。這一門新的科學（或技術），統稱為訊號處理（signal processing）。而這門科學的核心理論與工具之一，就是傅立葉級數。

一百八十年前傅立葉級數的發展，還是為了解決物理問題：熱傳導微分方程。但是，沒有人規定我們只能按照發明人的意圖去運用他發明的東西，對不對？例如高斯導出的最小平方差方法，在今天可以用來處理統計和經濟學方面的問題。但是當初他是為了推測一個天文學上的現象（慧星的軌跡），才發明這套方法（高斯在解決這個問題的同時，還順便發明了今天所謂的高斯消去法）。

Joseph Fourier (1768–1830) 是一位與拿破侖同期的法國人，曾經是拿破侖的御用科學家，隨軍遠征埃及，並對古埃及文化的研究有所貢獻。他所發掘的一件著名契形文字泥版，Rosetta stone，在他被英國海軍俘虜的時候給沒收了，現在展示於大英博物館。他穩定的科學家生活始於拿破侖被流放南大西洋的小島（1814）。但是他在 1807 就已經提

到過這一類的級數。

所謂傅立葉級數 (*Fourier series*) 是指以下這種形式的級數。

idx:Fseries

$$a_0 + \sum_{k=1}^{\infty} a_k \cos kx + b_k \sin kx \quad (2.24)$$

eq:Fseries

如果只取有限多項的部分和，則稱為三角多項式 (*trigonometric polynomial*)。其中 a_k, b_k 都是實數，稱為傅立葉係數。它首次出現於歐拉 (Euler) 的一個等式

idx:trigpol

$$\frac{x}{2} = \sin x - \frac{1}{2} \sin 2x + \frac{1}{3} \sin 3x - \dots, \quad x \in (-\pi, \pi) \quad (2.25)$$

eq:EuFs

(此級數並沒有在 $\pm\pi$ 這兩個點收斂到 $\pm\frac{\pi}{2}$ 。) 但是數學史上並沒有以歐拉來命名這一類的級數。這或許是因為傅立葉發現這類級數的原因具有比較深的數學影響。

前面提到，傅立葉當初想要研究的物理問題是熱傳導現象。簡單地說，根據熱的擴散定律，我們可以推導一維熱傳導偏微分方程式：

$$\frac{\partial}{\partial t} u = c \frac{\partial^2}{\partial x^2} u$$

其中 $u(t, x)$ 是溫度， t 代表時間變數， x 代表空間變數， $c > 0$ 是熱傳導係數。顯然

$$e^{-ct} \sin x \quad \text{和} \quad e^{-ct} \cos x$$

都會符合上述微分方程。但是，爲了要符合這個微分方程所附帶的初始條件和邊界條件，我們還需要一些處理。傅立葉級數的研究，差不多就是這麼開始的。

由所謂的歐拉等式

$$e^{i\theta} = \cos \theta + i \sin \theta$$

我們可以改寫

$$\cos kx = \frac{e^{ikx} + e^{-ikx}}{2}, \quad \sin kx = \frac{e^{ikx} - e^{-ikx}}{2i}$$

其中 $i = \sqrt{-1}$ 是單位虛數。所以，傅立葉級數 (2.24) 又可以寫成

$$\sum_{k=-\infty}^{\infty} c_k e^{ikx} \quad (2.26)$$

eq:Fcomplex

這時 $x \in \mathbb{R}$ 而 $c_k \in \mathbb{C}$ 。(2.24) 和 (2.26) 是常見的兩種傅立葉級數的表達方式。這兩種方式很容易互相轉換：

$$\begin{aligned} c_0 &= a_0, & c_k &= \frac{a_k - ib_k}{2}, & c_{-k} &= \frac{a_k + ib_k}{2} & \text{for } k \in \mathbb{N} \\ a_0 &= c_0, & a_k &= (c_k + c_{-k}), & b_k &= i(c_k - c_{-k}) & \text{for } k \in \mathbb{N} \end{aligned} \quad (2.27)$$

eq:Fchange

泰勒級數和傅立葉級數的功用，都是將一個比較複雜的函數，拆成無窮多項比較單純的函數級數。這些無窮多項的低次項總是比較重要，而高次項的數值越來越小。只要項次夠高，即使將所有高次項省略了也不會造成太大的誤差。

泰勒級數把函數分解成多項式，傅立葉級數把函數分解成不同頻率的三角函數。泰勒級數通常在某個參考點的附近逼近函數，傅立葉級數通常在某個區間內平均地逼近函數。泰勒級數的極限函數 (如果存在)，一定等於原函數，傅立葉級數的極限函數 (如果存在)，不見得在每個點都等於原函數。

若存在一個常數 T ，使得 $f(x) = f(x+T)$ ， $\forall x \in \mathbb{R}$ ，則 $f(x)$ 稱為週期性函數 (*periodic function*)。而 T 稱為 $f(x)$ 的週期。如果要特別表明其週期，我們也可以說 $f(x)$ 是 T 週期函數。例如 $\sin x$ 就是一個 2π 週期函數。以圖形來看，就是 $f(x)$ 的函數圖形每隔 T 就重複一次。既

idx:period

然重複，我們可以只看 $f(x)$ 在一段週期內的函數曲線。例如通常我們在 $[0, 2\pi]$ 或 $[-\pi, \pi]$ 內呈現 $\sin x$ 的圖形。

若 $f(x)$ 是一個 T 週期函數，而且 $\int_0^T f(x) dx = c \neq 0$ ，則它在 \mathbb{R} 上的廣義積分必不存在，因為

$$\begin{aligned} \int_{-\infty}^{\infty} f(x) dx &= \sum_{k=-\infty}^{\infty} \int_{kT}^{(k+1)T} f(x) dx \\ &= \sum_{k=-\infty}^{\infty} \int_0^T f(x) dx = \sum_{k=-\infty}^{\infty} c = \infty \end{aligned}$$

所以我們不討論 T 週期函數在 \mathbb{R} 上的積分，也不討論 T 週期函數在 \mathbb{R} 上的長度。

當我們說 $f(x)$ 是 T 週期函數，言下之意似乎 T 是 $f(x)$ 的最小週期。但是在數學的定義上，並沒有這麼說。例如 $\sin 2x$ 是 $\sin x$ 縮窄一倍，它的週期應該恰好是 π ，我們還是可以說它是 2π -週期函數，只是在 $[0, 2\pi]$ 之間其實 $\sin 2x$ 重複了兩遍。同理， $\sin kx$ ($\forall k \in \mathbb{Z}$) 也都是 2π -週期函數。但是 $\sin \frac{x}{2}$ 是 $\sin x$ 脹大一倍，它的週期是 4π ，因此不能說是 2π 週期函數。而 $\sin \frac{7}{2}x$ 的週期是 $\frac{4}{7}\pi$ ，雖然它 $< 2\pi$ ，但是它在 $[0, 2\pi]$ 之間並不能重複整數次，所以它不能說是一個 2π 週期函數。

從 (2.24) 可以看出，傅立葉級數必定是一個 2π 週期函數。通常我們只看它在 $[-\pi, \pi]$ 之間的圖形。我們用 $L_p^2([-T, T])$ 代表在 $[-T, T]$ 之內長度有限的 $2T$ 週期函數所成的集合。若定義

idx:L2p

$$\langle f(x), g(x) \rangle_p = \int_{-T}^T f(x) \overline{g(x)} dx$$

請檢查 $\langle \cdot, \cdot \rangle_p$ 是 $L_p^2([-T, T])$ 上面的一個內積。所以 $L_p^2([-T, T])$ 是一個具有內積觀念的向量空間。以後，除非特別聲明，否則我們討論的都是 $L_p^2([-\pi, \pi])$ ；爲了行文的方便，將它簡記做 L_p^2 。

在這個觀點之下， $\frac{1}{\sqrt{2\pi}}e^{inx} \in L_p^2$ 。而且

$$\begin{aligned} \left\langle \frac{1}{\sqrt{2\pi}}e^{imx}, \frac{1}{\sqrt{2\pi}}e^{inx} \right\rangle_p &= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{imx} \overline{e^{inx}} dx \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i(m-n)x} dx \\ &= \begin{cases} \frac{1}{2\pi} \frac{1}{i(m-n)} e^{i(m-n)x} \Big|_{x=-\pi}^{\pi} = 0 & \text{if } m \neq n \\ \frac{1}{2\pi} \int_{-\pi}^{\pi} dx = 1 & \text{if } m = n \end{cases} \\ &= \delta_{mn} \end{aligned}$$

請讀者自行檢查 $e^{i\theta} = \cos\theta + i\sin\theta$ 。可見 $\{\frac{1}{\sqrt{2\pi}}e^{inx} \mid n \in \mathbb{Z}\}$ 在 L_p^2 上形成一組正則集合。傅立葉級數理論告訴我們， $\{\frac{1}{\sqrt{2\pi}}e^{inx}\}$ 形成 L_p^2 的一組正則基底。它的意思是，任取一個 $f(x) \in L_p^2$ ，它可以被 $\{\frac{1}{\sqrt{2\pi}}e^{inx}\}$ 展開，亦即

$$f(x) = \sum_k \left\langle f(x), \frac{1}{\sqrt{2\pi}}e^{ikx} \right\rangle_p \frac{1}{\sqrt{2\pi}}e^{ikx}$$

但是爲了符號上的方便，我們通常寫成

$$f(x) = \frac{1}{2\pi} \sum_k f_k e^{ikx} \tag{2.28} \quad \text{eq:Freconst}$$

其中

$$f_k = \left\langle f(x), e^{ikx} \right\rangle_p = \int_{-\pi}^{\pi} f(x) e^{-ikx} dx \tag{2.29} \quad \text{eq:Fdecomp}$$

可見 (2.26) 中的

$$c_k = \frac{1}{2\pi} f_k$$

以訊號處理的眼光來看，若 $f(x)$ 是一個 2π 週期訊號，則從 $f(x)$ 變換到傅立葉係數 f_k 的步驟是為分解，而從係數數列 f_k 變換到 $f(x)$ 的步驟是為合成。可見 (2.29) 是分解公式，(2.28) 是合成公式。

前面說過，傅立葉級數把函數分解成不同頻率的三角函數。由於 $\sin kx$ 在 $[-\pi, \pi]$ 中振動 k 次，我們可以說 $\sin kx$ 的頻率是 k 。所以在 (2.24) 中的 a_k 與 b_k 都是頻率 k 的振幅 (或是分量)。其實正弦 (\sin) 與餘弦 (\cos) 函數可以看做是同一個函數，只是相差了一個 $\frac{\pi}{2}$ 的相位差 ($\cos x = \sin(x + \frac{\pi}{2})$)。而 e^{ix} 簡潔地將正弦與餘弦函數合併記在一起，並將相位差改以複數值表示。所以，合併之後，我們可以說 f_k 就是 $f(x)$ 在頻率 k 的振幅。想像 $f(x)$ 是一組聲波，它記錄空氣壓力與時間的關係。所以我們稱 $f(x)$ 是一組訊號在時間域 (time domain) 的表象。而傅立葉級數將 $f(x)$ 轉換 (或分解) 成整數頻率的振幅 f_k ，所以我們稱 $\{f_k\}$ 是同一組訊號在頻率域 (frequency domain) 的表象。傅立葉級數的分解與合成步驟，簡單地說，就是將一組週期性的訊號，在時間域與頻率域之間做轉換。

傅立葉級數如何收斂？也就是說 (2.28) 裡面的等號在什麼意義下成立？這可以說是十九世紀的主要數學問題之一。也是帶動數學分析發展的火車頭之一*。我們只說結論。(2.28) 的意義就如同 (1.39) 是 L^2 收斂的意思，亦即

$$\lim_{N \rightarrow \infty} \left\| f(x) - \frac{1}{2\pi} \sum_{n=-N}^N f_n e^{inx} \right\| = 0$$

其中 f_n 如 (2.29) 所定義。這個理論只保證了 $f(x)$ 與它的傅立葉級數之間差異的長度是 0；它們之間未必逐點相等。所以 (2.28) 並非古典的逐

* 例如我們在第 9 節中介紹的那個奇怪的函數 (2.21)，就是 Dirichlet 在研究傅立葉級數的收斂性和唯一性的時候發明的。

點收斂。由於 $\{e^{inx}\}$ 的正交性，我們還可以推導出

$$\|f(x)\|_p^2 = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} |f_k|^2 = \frac{1}{2\pi} \|(f_k)\|_{\ell^2}^2 \quad (2.30)$$

eq:Pnd

其中 $\|f\|_p^2 = \langle f, f \rangle_p$ 。而且，如果 $g(x) \in L_p^2$ 而 $g(x) = \frac{1}{2\pi} \sum g_k e^{ikx}$ ，其中 $g_k = \langle g(x), e^{ikx} \rangle_p$ ，則

$$\langle f(x), g(x) \rangle_p = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} f_k \bar{g}_k \quad (2.31)$$

eq:Pid

如果 $f(x)$ 是分片平滑函數，我們可以更明確地刻劃收斂的情形：若存在有限多個點

$$-\pi \leq x_1 < x_2 < \cdots < x_m < \pi$$

使得 $f(x)$ 在每一段小區間 (x_{k-1}, x_k) 中是一次可微的函數，而且

$$f(x_{k\pm}) = \lim_{t \rightarrow x_k^\pm} f(t)$$

都存在 ($f(x)$ 在不連續處的左右極限都存在)，則

$$\lim_{N \rightarrow \infty} \sum_{n=-N}^N c_n e^{inx} = \begin{cases} \frac{f(x-) + f(x+)}{2} & \text{if } x \in \{x_1, x_2, \dots, x_m\} \\ f(x) & \text{otherwise} \end{cases}$$

可見傅立葉級數不會逐點收斂到不連續的函數。這是無可厚非的事。因為 $\sin x$ 與 $\cos x$ 都是非常平滑的 C^∞ 函數，所以傅立葉級數的部分和永遠是平滑的函數。拿這麼平滑的函數去逼近不連續的函數，理所當然是不合適的。除此之外，當我們用傅立葉級數去逼近一個不

連續函數的時候，還會出現所謂的 Gibbs 現象：傅立葉級數在不連續點的兩側都會出現超射的現象。 idx:Gibbs

雖然事後發現，這個現象曾經在 1848 年由劍橋三一學院 Wilbraham 所討論。但今人一般還是把這個現象的發現與解釋歸功於兩篇分別於 1898 和 1899 年刊在美國 Nature 雜誌上的文章。

在 1898 年提出這個觀察現象的是美國的物理學家 Michelson (1852–1931)。他在 1892 年出任剛成立的芝加哥大學物理系的第一位系主任。因為他測定光速以及證明以太 (ether) 不存在，使他獲得 1907 年的諾貝爾物理獎。當時，在自動計算機的發展歷史上，處於機械類比型的時代。在 1880 左右，英國的物理學家 Lord Kelvin 利用類比積分器發明了一種稱作調和分析儀 (Harmonic Analyzer) 的計算機。它的功能是可以依據輸入的 $f(x)$ 圖形計算其三角多項式係數；也就是，輸入圖形而輸出 (2.24) 中前幾項的 a_k 和 b_k 。當時被用在海潮的研究上，所以又有個名字叫潮汐調和分析儀 (tidal harmonic analyzer)。這種機器一直到二次大戰期間還在使用。在 1897 年間，Michelson 設計了一些技術，使得這種調和分析儀可以算出更高項的傅立葉係數；起先是 20 項，最後是 80 項。他帶著這個機器參加 1900 年在巴黎舉辦的世界博覽會，得了首獎。

從這個計算機的輸出，Michelson 觀察到以下這個現象。令 $f(x)$ 是一個 2π 週期函數，而 $f(x)$ 在 $[-\pi, \pi)$ 內是 $f(x) = x$ 。很明顯地 $f \in L^2_p$ 而且它在 $x = (2k+1)\pi$ 處不連續。如圖 16。我們只看 $-\pi$ 和 π 兩點就可以了。根據歐拉早已知道的等式 (2.25)，

$$f(x) = 2\left(\sin x - \frac{1}{2} \sin 2x + \frac{1}{3} \sin 3x - \dots\right) = 2 \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} \sin nx$$

($f(x)$ 是個奇函數。) 但是，不論是取二十項還是四十項，級數的部分和看起來好像都會射過頭一點點 (超射)，而且過頭的程度好像不會隨

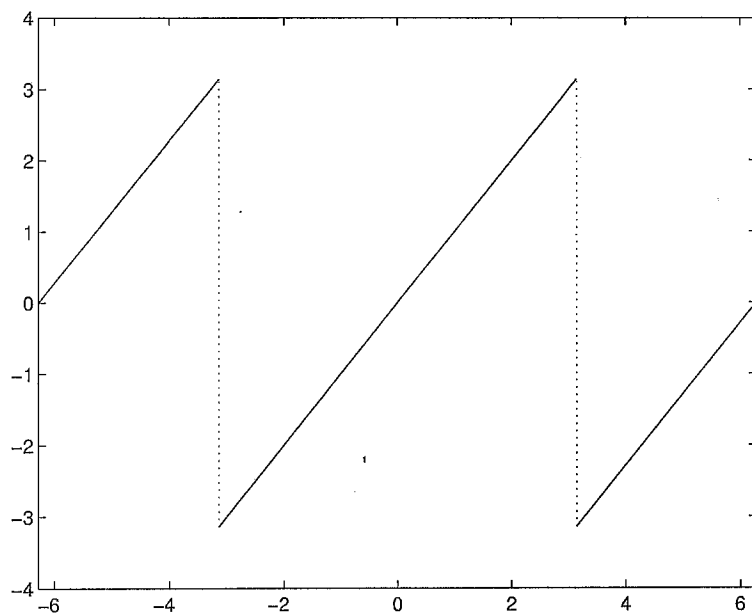


fig:gibbf 圖 16 發現 Gibbs 現象的不連續週期函數

著項數的增加而改善。我們用 Matlab 的浮點計算重造這兩個例子，如圖 17。但是當時 Michelson 的計算機的精確度只有三五個有效數字，製圖的功能也很原始。他觀察這個超射的部份大約是常數 0.56。

1899 年，Gibbs (1839–1903) 回應了 Michelson 的發現。Gibbs 被譽為十九世紀美國最偉大的數學物理學家。他在一篇幾乎只有一頁的文章中闡述了上述的現象，並且證明，在前述的例子中，傅立葉級數在 $(2k + 1)\pi$ 附近的最大值減最小值趨近於

$$4 \int_0^{\pi} \frac{\sin x}{x} dx$$

而上下超射的部份各是

$$2 \int_0^{\pi} \frac{\sin x}{x} dx - \pi \approx 0.562281.$$

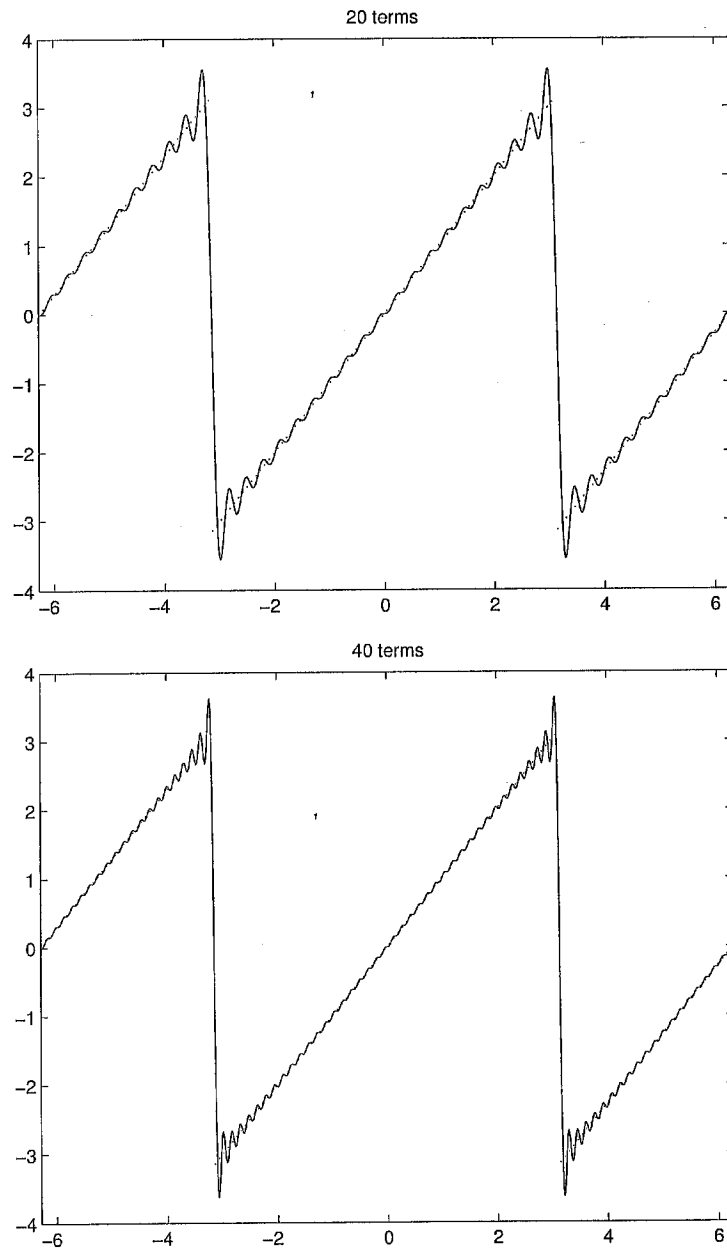


fig:Gibbs 圖 17 Gibbs 現象，取二十項 (上圖) 或四十項 (下圖) 部分和

他並且說，函數圖形的極限看起來並非函數極限的圖形。Strang 稱讚這篇論文是訊雜比 (S/N ratio) 最高的一篇論文。

雖然 Gibbs 現象是由此特例發現的，但它是一個普遍性的現象。只要是以連續函數去逼近不連續函數，都可能出現這個現象。即使 (連續的) 凌波函數也不能避免。我們將在第四章看到例子。

前面我們看到傅立葉級數和泰勒級數之間的一個不同處：泰勒級數會在參考點 c 附近的某個區間內逐點收斂，傅立葉級數則會在整個 \mathbb{R} 中以 L^2 的意義收斂。接著，我們要提出傅立葉級數和泰勒級數之間的另一個不同處：傅立葉級數具有局部性。即使兩個傅立葉級數 $\sum c_k e^{ikx}$ 和 $\sum h_k e^{ikx}$ 在某一段區間 (a, b) 內收斂到同一個函數，它們在 (a, b) 之外仍可能收斂到不同的函數。反觀泰勒級數，若 $\sum \alpha_n (x - c)^n$ 和 $\sum \beta_n (x - c)^n$ 在某一段區間 (a, b) 內收斂到同一個函數，其中 $c \in (a, b)$ ，則

$$u(x) = \sum (\alpha_n - \beta_n)(x - c)^n = 0, \quad \forall x \in (a, b)$$

代入 $u(c) = 0$ 得到 $\alpha_0 = \beta_0$ 。代入 $u'(c) = 0$ 得到 $\alpha_1 = \beta_1$ 。依此類推，可見 $\alpha_n = \beta_n, \forall n$ 。所以 $\sum \alpha_n x^n$ 和 $\sum \beta_n x^n$ 全然相等 (只要收斂了)。所以，兩個泰勒級數如果在某一段區間 (a, b) 內相等，則它們在整個會收斂的區間內全都相等。一個簡單的結論：

泰勒級數如果在某一段區間 (a, b) 內等於零，則它在整個會收斂的區間內都等於零。傅立葉級數沒有這個現象。

爲了表現傅立葉級數的局部性，我們考慮以下這個例子。令

$$f(x) = \begin{cases} \frac{1}{4} - |x| & \text{if } |x| < \frac{1}{4} \\ 0 & \text{otherwise} \end{cases} \quad x \in [-\pi, \pi] \quad (2.32) \quad \text{eq:fxa}$$

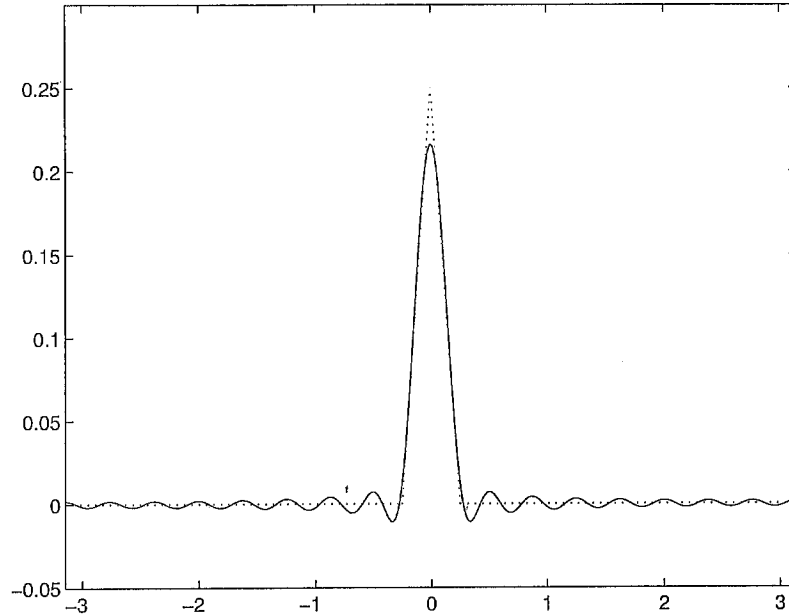


fig:Fsixtn 圖 18 十六項傅立葉級數部分和

然後將 $f(x)$ 週期化成 2π 週期函數。注意 $f(x)$ 在 $[\frac{1}{4}, \pi]$ 內等於零。因為 $f(x)$ 是一個偶函數，所以 (2.24) 中的 b_n 都是零。計算 $a_0 = \frac{1}{32\pi}$ ， a_1 到 a_{16} 的數值分別如下

0.0099	0.0097	0.0095	0.0091	0.0087	0.0082	0.0077	0.0070
0.0064	0.0057	0.0051	0.0044	0.0038	0.0031	0.0026	0.0021

此部分和的圖形顯示在圖 18。讀者應該可以直覺地相信，當部分和的項數提高的時候，傅立葉級數會收斂到 $f(x)$ 。

如果 $f(x)$ 是一個定義在 $[-\pi, \pi]$ 中的函數，所謂將它週期化，就是 idx:periodi:把 $f(x)$ 在區間 $[-\pi, \pi]$ 內的圖形，分片抄到以下的區間去

$$[(2k-1)\pi, (2k+1)\pi], \quad k \in \mathbb{Z}$$

也就是硬把它變成 2π 週期函數的意思。如果當 $|x| > \pi$ 時 $f(x) = 0$ ，令 $f_p(x)$ 代表被週期化的函數，則

$$f_p(x) = \sum_k f(x - 2k\pi) \quad (2.33)$$

eq:periodize

除非原函數之左右端點相等： $f(-\pi) = f(\pi)$ ，否則週期化之後的函數不是連續函數。

傅立葉級數本身，無論在理論與應用上都是很重要的數學命題。目前，它的主要目的是爲了下一節的傅立葉轉換鋪路。在第三章中，還會闡述它與訊號處理的關係。所以現在就此打住。這一節的內容，部分取自作者刊登於數學傳播季刊的文章 [11]。在此省略了某些細節，特別是 Gibbs 現象的推導與收斂定理的詳細討論。意欲深究的讀者，請參考季刊，或至以下網址取得全文。

<http://www.math.ncu.edu.tw/~shann/Teach/article.html>

習題 2.35 下列哪些級數是傅立葉級數？

- (1) $\frac{1}{2} - \sin x + \sin 2x - \sin 3x + \dots$
- (2) $1 + \cos x + \cos^2 x + \cos^3 x + \dots$
- (3) $\sin x + \sin(x+1) + \sin(x+2) + \sin(x+3) + \dots$
- (4) $\cos x + \sin x - \cos 2x - \frac{1}{2} \sin 2x + \cos 3x + \frac{1}{3} \sin 3x + \dots$

習題 2.36 計算以下方塊波 ($\in L_p^2([- \pi, \pi])$) 的傅立葉係數：

$$f(x) = \begin{cases} -1 & \text{for } -\pi \leq x < 0 \\ 1 & \text{for } 0 \leq x < \pi \end{cases}$$

如果不使用電腦，應該至少算出前三項。如果使用電腦軟體，應該至少算出前二十項並且製圖。

習題 2.37 針對以下函數，再做上一題。

$$f(x) = \begin{cases} -x & \text{for } -\pi \leq x < 0 \\ x & \text{for } 0 \leq x < \pi \end{cases}$$

習題 2.38 令 $p(x) = \sum a_n e^{inx}$ 和 $q(x) = \sum b_n e^{inx}$ 為兩個三角多項式。若 $h(x) = p(x)q(x)$ ，證明 $h(x)$ 也是一個三角多項式，而且 $h(x)$ 的係數就是 a_n 與 b_n 的離散疊積。

習題 2.39 證明 $\{\frac{1}{\sqrt{2\pi}}, \frac{1}{\sqrt{\pi}} \cos nx, \frac{1}{\sqrt{\pi}} \sin nx \mid n > 0\}$ 在 L^2_p 上形成一組正則集合。

習題 2.40 證明 (2.27)

習題 2.41 證明 (2.33) 中的 $f_p(x)$ 是 2π 週期函數。亦即，證明 $f_p(x) = f_p(x + 2\pi)$ ， $\forall x \in \mathbb{R}$ 。

習題 2.42 定義

$$f(x) = \begin{cases} \pi + x & \text{for } -\pi \leq x \leq 0 \\ \pi - x & \text{for } 0 \leq x \leq \pi \end{cases}$$

並將之週期化爲 2π 週期函數。證明 $c_k \in \mathbb{R}$ 而且 $c_k = c_{-k}$ 。

習題 2.43 若 $f(x)$ 是 1 週期函數，且 $f(x) \in L^2([0, 1])$ ，證明 $f(x)$ 之傅立葉級數可寫成

$$f(x) = \sum_k f_k e^{i2k\pi x} \quad (2.34)$$

eq:perione

其中的傅立葉係數是

$$f_k = \int_0^1 f(x) e^{-i2k\pi x} dx$$

2.12 傅立葉轉換

傅立葉級數只處理了 2π 週期函數： 2π 週期函數可以寫成整數頻率的正弦餘弦波。那麼對於週期比 2π 大的週期函數呢？對於非週期性的函數呢？傅立葉級數就要被推廣為傅立葉轉換。傅立葉轉換和傅立葉級數一樣，可以被理解成一個從時間域 (time domain) 轉換到頻率域 (frequency domain) 的運算。以後所要討論的凌波函數，都不是週期函數。所以需要傅立葉轉換來探討它們的頻率特性。我們在這一節裡面介紹傅立葉轉換，至於它所啟發的深刻看法，以及它所闡述的意義，則出現在第三章。

如果 $f(x)$ 是個 4π 週期函數，則 $f(2x)$ 縮窄了一倍，所以是個 2π 週期函數。可見 $f(2x)$ 可以寫成傅立葉級數 $f(2x) = \sum c_k e^{ikx}$ ，其中

$$c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(2x) e^{-ikx} dx = \frac{1}{2\pi} \frac{1}{2} \int_{-2\pi}^{2\pi} f(x) e^{-i\frac{k}{2}x} dx$$

而且經過變數變換，也可以寫做 $f(x) = \sum c_k e^{i\frac{k}{2}x}$ 。我們可以想像， 2π 週期函數可以分解成整數頻率上的分量，而 4π 週期函數可以分解成二分之整數頻率上的分量。依此類推，若 $f(x)$ 是一個 $2n\pi$ 週期函數，則它可以分解成 n 分之整數頻率上的分量。也就是

$$f(x) = \frac{1}{2\pi} \frac{1}{n} \sum_k f_k e^{i\frac{k}{n}x}, \quad \text{where } f_k = \int_{-n\pi}^{n\pi} f(x) e^{-i\frac{k}{n}x} dx \quad (2.35)$$

eq:Fmotiv

想像當 $n \rightarrow \infty$ 的時候，我們取樣的頻率 $\frac{k}{n}$ 幾乎佈滿整個實數。我們何不對任何一個實數頻率取其分量？亦即，將 (2.35) 中的 f_k 變成*

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x) e^{-i\omega x} dx, \quad \forall \omega \in \mathbb{R} \quad (2.36)$$

eq:Ftrans

* 有些讀者或許會注意到，不管 n 多麼大， $\frac{k}{n}$ 仍是有理數。再怎麼說它們也頂多佈滿有理數而已。是怎樣擴充到實數的呢？而且原來 f_k

上面的 $\hat{f}(\omega)$ 就稱做 $f(x)$ 的傅立葉轉換 (*Fourier transform*)。以訊號處理的眼光來看，它相當於一個分解的過程。當 $f(x)$ 是 2π 週期函數時，它可以分解到整數頻率上；當 $f(x)$ 是 $2n\pi$ 週期函數時，它可以分解到離散的有理數頻率上；但是當 $f(x)$ 根本不是週期函數時，它可以分解到連續的實數頻率上。這個分解就是傅立葉轉換。讀者請對照離散情況的分解過程 (2.29)。

再看 (2.35)，當 n 很大的時候，

$$f(x) = \frac{1}{2\pi} \frac{1}{n} \sum_k f_k e^{i\frac{k}{n}x} \approx \frac{1}{2\pi} \frac{1}{n} \sum_k \hat{f}\left(\frac{k}{n}\right) e^{i\frac{k}{n}x}$$

讀者請觀察，上式之右端可以看做以 $\frac{1}{n}$ 為均勻分割的黎曼和。所以當 $n \rightarrow \infty$ 的時候就得到

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i\omega x} d\omega, \quad \forall x \in \mathbb{R} \quad (2.37)$$

這就是所謂的傅立葉逆轉換 (*Fourier inverse transform*)。以訊號處理的眼光來看，它相當於一個合成或還原的過程。讀者請對照離散情況的合成過程 (2.28)。

當被轉換的函數比較複雜，以至於 $\hat{\cdot}$ 符號不適用的時候，我們以 $\mathcal{F}[f(x)](\omega)$ 表示傅立葉轉換。並以 $\mathcal{F}^{-1}[\hat{f}(\omega)](x)$ 表示傅立葉逆轉換。讀者很容易可以檢查

$$\mathcal{F}[cf(x) + g(x)](\omega) = c\mathcal{F}[f(x)](\omega) + \mathcal{F}[g(x)](\omega)$$

的 k 只是整數，怎麼變成實數的 ω 呢？我們在此只是提供一個動機，並非嚴格地推導傅立葉轉換的由來。如果您對這一類的問題感到目眩神迷，您或許適合閱讀集合論的書籍，特別注意 Cantor 的基數 (cardinal numbers) 理論。

(如果它們存在的話) 所以傅立葉轉換是線性映射。

仿照傅立葉級數的講法，我們想像 $f(x)$ 是一組聲波，它記錄空氣壓力與時間的關係。我們稱 $f(x)$ 是一組訊號在時間域 (time domain) 的表象。這時候， $f(x)$ 不必是個週期函數，亦即這組訊號不必是週期性的訊號。而傅立葉轉換將 $f(x)$ 轉換成各個實數頻率的振幅 $\hat{f}(\omega)$ ，所以我們稱 $\hat{f}(\omega)$ 是同一組訊號在頻率域 (frequency domain) 的表象。傅立葉轉換的分解與合成步驟，簡單地說，就是將一組訊號，在時間域與頻率域之間做轉換。

這裡，我們面臨的數學問題是，什麼樣的條件可以保證 $\hat{f}(\omega)$ 存在，也就是 (2.36) 中的積分值有限？再者，什麼情況下 (2.37) 成立？因為

$$|e^{-i\omega x}| = |\cos \omega x + i \sin \omega x| = 1$$

(順便一提： $e^{-i\omega x} \notin L^2(\mathbb{R})$) 所以

$$|\hat{f}(\omega)| \leq \int |f(x)e^{-i\omega x}| dx = \int |f(x)| dx$$

可見只要 $|f(x)|$ 是可積的，則 $\hat{f}(\omega)$ 存在。我們稱 $\int |f| dx < \infty$ 的函數為絕對值可積。所有這些函數所成的集合以 $L^1(\mathbb{R})$ 表示。讀者可以檢查 $L^1(\mathbb{R})$ 形成一個抽象的向量空間。

同樣的道理，只要 $\hat{f}(\omega)$ 是絕對值可積，則 $\frac{1}{2\pi} \int \hat{f}(\omega) e^{i\omega x} d\omega$ 存在 (但其積分值未必等於 $f(x)$ ，見後面的 L^1 傅立葉轉換定理)。可是我們並不願意將傅立葉轉換的理論，建立在 $L^1(\mathbb{R})$ 空間上。原因有二。第一，有些絕對值可積的函數，其傅立葉轉換不再是絕對值可積。所以傅立葉轉換不是一個從 L^1 映射到自己的線性映射。第二， $L^1(\mathbb{R})$ 只能形成具有長度性質的完備向量空間 (Banach 空間)，它不能形成具有內積性質的完備向量空間 (Hilbert 空間)；而我們經常需要內積性質。

希爾伯特

針對第一點，可以舉一個例子。令

$$u(x) = \begin{cases} e^{-x} & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (2.38) \quad \text{eq:Fexmp}$$

很明顯地 $\int |u| dx = 1 < \infty$ 故 $u \in L^1(\mathbb{R})$ 。則

$$\hat{u}(\omega) = \int_0^{\infty} e^{-x} \cos \omega x dx - i \int_0^{\infty} e^{-x} \sin \omega x dx$$

請讀者驗證 $\int_0^{\infty} e^{-x} \cos \omega x dx = \frac{1}{1+\omega^2}$ ， $\int_0^{\infty} e^{-x} \sin \omega x dx = \frac{\omega}{1+\omega^2}$ 。所以

$$\hat{u}(\omega) = \frac{1-i\omega}{1+\omega^2} = \frac{1}{1+i\omega} \quad (2.39) \quad \text{eq:Fexmphot}$$

檢查

$$\begin{aligned} \int |\hat{u}(\omega)| d\omega &= \int \frac{1}{\sqrt{1+\omega^2}} d\omega \\ &= \int_{-1}^1 \frac{1}{\sqrt{1+\omega^2}} d\omega + 2 \int_1^{\infty} \frac{1}{\sqrt{1+\omega^2}} d\omega \\ &\geq 2\sqrt{2} + 2 \int_1^{\infty} \frac{1}{\sqrt{2\omega^2}} d\omega \\ &= 2\sqrt{2} + \sqrt{2} \int_1^{\infty} \frac{1}{\omega} d\omega \\ &= 2\sqrt{2} + \sqrt{2} \ln' \omega \Big|_1^{\infty} = \infty \end{aligned}$$

可見 $\hat{u}(\omega)$ 不是絕對值可積的函數。

L^1 傅立葉轉換定理 若 $f(x) \in L^1(\mathbb{R})$ 則 $\hat{f}(\omega)$ 存在。若 $\hat{f}(\omega)$ 也在 $L^1(\mathbb{R})$ 裡面，且 $f(x)$ 在 x_0 連續，則

idx:thmFLor

$$f(x_0) = \mathcal{F}^{-1}[\hat{f}(\omega)](x_0) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{i\omega x_0} d\omega$$

前面說過，我們經常需要內積性質。所以我們希望在 $L^2(\mathbb{R})$ 中建構傅立葉轉換的理論。但是我們不能直接這麼做。因為有些在 $L^2(\mathbb{R})$ 內的函數不是絕對值可積。一個簡單的例子是

$$g(x) = \begin{cases} \frac{1}{x} & \text{if } x \geq 1 \\ 0 & \text{if } x < 1 \end{cases} \quad (2.40)$$

eq: notLone

則 $g(x) \in L^2(\mathbb{R})$ 但它不是絕對值可積。可見 $L^2(\mathbb{R}) \not\subset L^1(\mathbb{R})$ 。順便，我們應該提醒讀者，也有一些在 $L^1(\mathbb{R})$ 內的函數不在 $L^2(\mathbb{R})$ 裡面。一個簡單的例子是

$$h(x) = \begin{cases} \frac{1}{\sqrt{x}} & \text{if } 0 < x < 1 \\ 0 & \text{otherwise} \end{cases}$$

則

$$\int |h| dx = \int_0^1 \frac{1}{\sqrt{x}} dx = 2\sqrt{x} \Big|_0^1 = 2$$

所以 $h(x) \in L^1(\mathbb{R})$ 。但是

$$\int |h|^2 dx = \int_0^1 \frac{1}{x} dx = \ln x \Big|_0^1 = \infty$$

所以 $h(x) \notin L^2(\mathbb{R})$ 。從這兩個例子看來， $L^1(\mathbb{R})$ 和 $L^2(\mathbb{R})$ 是互不包含的兩個向量空間。而它們亦不互相排斥；因為顯然 $L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ 並非空集合。

回顧 (2.38) 中定義的 $u(x)$ ，很明顯 $\int |u|^2 dx = \frac{1}{2} < \infty$ ，故 $u(x)$ 也是 L^2 函數；也就是說 $u(x) \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ 。我們已經知道在 (2.39) 中的 $\hat{u}(\omega) \notin L^1(\mathbb{R})$ 。但是

$$\begin{aligned} \int |\hat{u}(\omega)|^2 d\omega &= \int \frac{1}{1+\omega^2} d\omega \\ &= \int_{-1}^1 \frac{1}{1+\omega^2} d\omega + 2 \int_1^{\infty} \frac{1}{1+\omega^2} d\omega \\ &< 2 + 2 \int_1^{\infty} \frac{1}{\omega^2} d\omega = 4 < \infty \end{aligned}$$

所以 $\hat{u}(\omega) \in L^2(\mathbb{R})$ 。

其實這是個普遍的現象：對任何的 $f(x) \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ ， $\hat{f}(\omega)$ 存在而且屬於 $L^2(\mathbb{R})$ 。更好的是，

$$\|f(x)\|^2 = \frac{1}{2\pi} \|\hat{f}(\omega)\|^2 \quad (2.41) \quad \text{eq:Fnd}$$

我們不便在此證明上面這個等式，請讀者對照傅立葉級數的一個類似等式 (2.30)。

如果 $f(x)$ 是一個在 $L^2(\mathbb{R})$ 卻不在 $L^1(\mathbb{R})$ 裡面的函數，我們不見得能直接用 (2.36) 計算 $f(x)$ 的傅立葉轉換。但是我們可以透過以下的數學方法，來定義 $f(x)$ 的傅立葉轉換。令 n 是一個正整數，定義

$$f_n(x) = \begin{cases} f(x) & \text{if } |x| \leq n \\ 0 & \text{otherwise} \end{cases}$$

則

$$\int |f_n|^2 dx \leq \int |f|^2 dx < \infty$$

而且，根據席瓦茲不等式，

$$\int |f_n| dx = \int_{-n}^n |f| dx \leq \sqrt{2n} \|f\| < \infty$$

所以 $f_n(x) \in L^1(\mathbb{R}) \cap L^2(\mathbb{R})$ 。因此 $\hat{f}_n(\omega)$ 存在而且屬於 $L^2(\mathbb{R})$ 。很明顯地 $\lim f_n(x) = f(x)$ ，所以 $f_n(x)$ 是 L^2 中的一個柯西數列。因為

$$\|\hat{f}_n(\omega) - \hat{f}_m(\omega)\| = \|\mathcal{F}[f_n(x) - f_m(x)](\omega)\| = \sqrt{2\pi} \|f_n(x) - f_m(x)\|$$

可見 $\hat{f}_n(\omega)$ 也是 L^2 中的一個柯西數列。根據 L^2 的完備性，這個數列必定在 L^2 的意義之下收斂。也就是說，存在一個極限函數。我們就令這個極限函數是 $\hat{f}(\omega)$ ：

$$\lim_{n \rightarrow \infty} \|\hat{f}_n(\omega) - \hat{f}(\omega)\| = 0 \quad (2.42) \quad \text{eq:FLtwo}$$

至此，我們知道 L^2 中的每一個函數 $f(x)$ 都有一個對應的傅立葉轉換函數 $\hat{f}(\omega)$ 。如果 $f \in L^2 \cap L^1$ 則 $\hat{f}(\omega)$ 由 (2.36) 定義，否則 $\hat{f}(\omega)$ 由 (2.42) 定義。但不論如何， $\hat{f}(\omega) \in L^2$ 。所以傅立葉轉換 $\mathcal{F}[\cdot]$ 是一個從 $L^2(\mathbb{R})$ 映射到自己的線性映射。在下面的定理中，我們綜合介紹 L^2 空間中的傅立葉理論。這個定理又稱做 *Plancherel* 定理。

L^2 傅立葉轉換定理 (Plancherel 定理) 傅立葉轉換 $\mathcal{F}: L^2(\mathbb{R}) \rightarrow L^2(\mathbb{R})$ 是一對一且映成的線性映射。所以對任意 $g(\omega) \in L^2$ 存在唯一的 $f(x) \in L^2$ 使得 $\mathcal{F}[f](\omega) = g(\omega)$ 而且 $\mathcal{F}^{-1}[g](x) = f(x)$ 。此外，對任意 $f, g \in L^2$

$$\langle f(x), g(x) \rangle = \frac{1}{2\pi} \langle \hat{f}(\omega), \hat{g}(\omega) \rangle \quad (2.43)$$

idx:thmFLtwo

eq:Parseval

idx:Parseval

前面的 (2.43) 又稱作 *Parseval* 等式。請讀者對照傅立葉級數的一個類似等式 (2.31)。有了這個等式，可見其實 (2.41) 對任意 L^2 中的函數都成立。讀者或許已經察覺，上面那個定理，只說明了 \mathcal{F}^{-1} 的存在性，並沒有告訴我們， $\mathcal{F}^{-1}[\hat{f}]$ 要怎麼計算？當 $\hat{f} \in L^1 \cap L^2$ 時，可以用 (2.37) 來計算 $\mathcal{F}^{-1}[\hat{f}]$ ；否則我們必須透過極限的程序。在爾後的應用中，並不會出現這樣的情況，所以就不深入研究了。

在第一章九節中，我們提出做內積 $\langle f, g \rangle$ 就是以 g 去探測 f 的看法。*Parseval* 等式告訴我們，用 g 去探測 f 的結果，同時也就（幾乎）等於用 \hat{g} 去探測 \hat{f} 的結果（只差 2π 倍）。這個基本的性質，就是傅立葉轉換應用在訊號處理、影像處理與訊息論 (*information theory*) 這些方面的利器之一。

至此，我們確立了 L^2 空間中的傅立葉轉換理論。在此後三章之中，我們將會更加闡述它的意義，並且用它來探討凌波函數的性質。到時候，還會就實際需要，開發一些傅立葉轉換的性質與工具。但是在本節中，我們還想要介紹以下三個課題：與疊積的關係，*Poisson* 求和公式，和 *Paley-Wiener* 定理。

我們首先介紹，傅立葉轉換與疊積之間的一個有趣關係：

$$\mathcal{F}[(f * g)(x)] = \hat{f} \hat{g} \quad (2.44) \quad \text{eq:fconv}$$

這個關係可以由交換重積分的順序（數學上稱為 Fubini 定理）推導出來，如下。

$$\begin{aligned} \mathcal{F}[(f * g)(x)](\omega) &= \int \int f(x-y)g(y) dy e^{-i\omega x} dx \\ &= \int \int f(x-y) e^{-i\omega x} dx g(y) dy \\ &= \int \int f(x) e^{-i\omega(x+y)} dx g(y) dy \\ &= \int \hat{f}(\omega) g(y) e^{-i\omega y} dy \\ &= \hat{f}(\omega) \hat{g}(\omega) \end{aligned}$$

就本書的需求而言，傅立葉轉換的介紹到此已經足夠。以下的課題，在本書中未受重用（那是因為這本書沒有進入那麼深的細節）。雖然如此，它們經常出現於凌波函數的學術論文之中。讀者應該要有所涉獵。

在這一節的開始，我們將週期性函數的週期從 2π 放大到 $2n\pi$ ，然後令 $n \rightarrow \infty$ 得到非週期性函數 $f(x)$ ，藉此我們將傅立葉級數的觀念推廣到傅立葉（積分）轉換。現在，反過來，考慮一個非週期性的函數 $f(x)$ ，如何將它週期化成爲 2π 週期函數，以便使用傅立葉級數？如果 $f(x)$ 是在 $[-\pi, \pi]$ 中的有限函蓋函數，我們已經在前一節中討論其週期化的方法，就是利用 (2.33)。但是，仔細思考一下，我們發現只要 (2.33) 之中的 $\sum f(x-2k\pi)$ 會收斂，則 $f_p(x)$ 當然就是 2π 週期函數，這個事實與 $f(x)$ 是否爲有限函蓋，根本沒有關係。所以我們就用 (2.33) 來做週期化。

idx:periodi

如果 $\text{supp } f = [a, b]$ 。若 $[a, b] \subset [-N\pi, N\pi]$ ，則對任一 $x \in \mathbb{R}$ ，級數 $\sum f(x + 2k\pi)$ 只有當 $|k| \leq \frac{N+1}{2}$ 出現非零項。所以它其實是有限和，當然收斂。因此 $f_p(x)$ 是有定義的。而且若 $\|f\| < \infty$ ，則 $\|f_p(x)\| \leq N\|f\| < \infty$ 。所以，如果 $f \in L^2([a, b])$ 則 $f_p \in L^2_p([-\pi, \pi])$ 。故 $f_p(x)$ 可以做傅立葉級數展開。又因為 $L^2([a, b]) \subset L^1([a, b])$ ，所以 $f \in L^2 \cap L^1$ 故 f 的傅立葉轉換 $\hat{f}(\omega)$ 可以用 (2.36) 求得。有了這些準備之後，回顧 (2.28) 和 (2.29)，我們知道

$$f_p(x) = \frac{1}{2\pi} \sum_n f_n e^{inx} \quad (2.45)$$

eq:Pta

其中

$$\begin{aligned} f_n &= \langle f_p, e^{inx} \rangle_p = \sum_k \int_{-\pi}^{\pi} f(x - 2k\pi) e^{-inx} dx \\ &= \sum_k \int_{(2k-1)\pi}^{(2k+1)\pi} f(x) e^{-in(x+2k\pi)} dx = \int f(x) e^{-inx} dx = \hat{f}(n) \end{aligned}$$

前面我們利用了 e^{-inx} 是 2π 週期函數的特性。將 $f_n = \hat{f}(n)$ 代回 (2.45) 並做足標變換 $n \mapsto k$ ，得到

$$\sum_k f(x - 2k\pi) = \frac{1}{2\pi} \sum_k \hat{f}(k) e^{ikx} \quad (2.46)$$

eq:Poisson

這就是所謂的 Poisson* 求和公式 (*Poisson Summation Formula*)。

idx:Poisson

* Poisson 是法國人，所以請不要將他的名字唸成英文的 poison。我找不到適切的音譯，所以乾脆不翻了。法文的 pois 發音與英文的 bwa 相近，而 son 的發音與中文的“頌”相近。

如果 $f(x)$ 不是有限函蓋，那麼級數 $\sum f(x - 2k\pi)$ 可能發散，所以 $f_p(x)$ 可能沒有定義。即使 $f_p(x)$ 有定義，也未必 $f_p(x) \in L^2_p([-\pi, \pi])$ 。我們充其量只能根據

$$\begin{aligned} \int_{-\pi}^{\pi} |f_p(x)| dx &\leq \sum_k \int_{-\pi}^{\pi} |f(x - 2k\pi)| dx \\ &= \sum_k \int_{(2k-1)\pi}^{(2k+1)\pi} |f(x)| dx = \int |f(x)| dx \end{aligned}$$

得知：如果 $f(x) \in L^1(\mathbb{R})$ ，則 $f_p(x) \in L^1([-\pi, \pi])$ 。如果 $f_p(x)$ 不在 L^2_p 裡面，則它的傅立葉級數未必收斂。我們束手無策，只好加強條件上的要求。加強條件的方式之一是，除了要求 $f \in L^1(\mathbb{R})$ 之外，另外要求 $\hat{f}(\omega)$ 的消散速率

$$|\hat{f}(\omega)| = O\left(\frac{1}{(1 + |\omega|)^\alpha}\right), \quad \text{for some } \alpha > 1$$

在上式的分母加 1 是爲了避免 $\omega \rightarrow 0$ 所可能引起的麻煩。而當 $|\omega| \rightarrow \infty$ 時， $|\hat{f}(\omega)|$ 的大小差不多就是 $\frac{1}{|\omega|^\alpha}$ 。如果 $\hat{f}(\omega)$ 符合上述的消散條件，則 Poisson 求和公式 (2.46) 成立。另一種更強的條件是，令 $f(x)$ 屬於所謂的 Wiener class，則 Poisson 求和公式亦成立。我們就不再深究了。

Poisson 求和公式有幾個常見的變形。例如

$$\sum_k f(x - k) = \sum_k \hat{f}(2k\pi) e^{i(2k\pi)x} \quad (2.47) \quad \text{eq:Pois a}$$

代入 $x = 0$ 得到

$$\sum_k f(k) = \sum_k \hat{f}(2k\pi) \quad (2.48) \quad \text{eq:Pois b}$$

將 (2.46) 應用到 $|\hat{f}(\omega)|^2$ ，得到

$$\sum_k |\hat{f}(\omega - 2k\pi)|^2 = \sum_k f_k e^{-ik\omega} \quad (2.49)$$

$$\text{where } f_k = \langle f(x), f(x-k) \rangle$$

eq:Poisc

上式中我們要求 $f \in L^2(\mathbb{R})$ 而且 \hat{f} 的消散速率是

$$|\hat{f}(\omega)| = \frac{1}{(1+|\omega|)^\alpha}, \quad \text{for some } \alpha > \frac{1}{2}$$

最後，我們介紹 Paley-Wiener 定理。這是一個從 $\hat{f}(\omega)$ 判斷 $f(x)$ 是否為有限函蓋 (並估計函蓋的上下界) 的方法。首先，令 $f(x)$ 是個有限函蓋的 L^2 函數。假設 $\text{supp } f \subset [-A, A]$ ，所以

$$\hat{f}(\omega) = \int_{-A}^A f(x) e^{-i\omega x} dx$$

其中 $\omega \in \mathbb{R}$ 是一個實變數。現在我們將 $\hat{f}(\omega)$ 拓展到複變函數；考慮

$$\hat{f}(\zeta) = \int_{-A}^A f(x) e^{-i\zeta x} dx \quad \zeta \in \mathbb{C}$$

那麼 $\hat{f}(\zeta)$ 將會是 \mathbb{C} 上的一個全函數 (entire function)，不說細節了。如果 $\zeta = u + iv$ ($u, v \in \mathbb{R}$)，則

$$|e^{-i\zeta x}| = |e^{vx - iux}| = |e^{vx}| |e^{-iux}| = |e^{vx}| \leq e^{|v||x|} \leq e^{|\zeta||x|}$$

所以

$$|\hat{f}(\zeta)| \leq \int_{-A}^A |f(x)| e^{|\zeta||x|} dx \leq \int_{-A}^A |f(x)| dx \cdot e^{A|\zeta|}$$

令 $\int |f(x)| dx = C$ ，我們可以下結論：

$$\text{若 } \text{supp } f \subset [-A, A] \text{ 則 } |\hat{f}(\zeta)| \leq Ce^{A|\zeta|} \quad (2.50) \quad \text{eq:PWeasy}$$

以上的推論是很基本的*，但是其反向命題就沒那麼基本了。以下的定理，就是說 (2.50) 的反向命題也是正確的。

* 福爾摩斯常說的：“*It's elementary*”。

Paley-Wiener 定理 令 $f(\zeta)$ 是 \mathbb{C} 上的全函數，而它被限制在實數上時 $f(x) \in L^2(\mathbb{R})$ 。若存在正常數 A 和 C 使得

idx:PW

$$|f(\zeta)| \leq Ce^{A|\zeta|}, \quad \forall \zeta \in \mathbb{C}$$

則存在一個有限函蓋函數 $\check{f}(x) \in L^2([-A, A])$ 使得

$$f(\zeta) = \int \check{f}(x)e^{-i\zeta x} dx, \quad \forall \zeta \in \mathbb{C}$$

當我們取 $\zeta = \omega \in \mathbb{R}$ 的時候， $f(\omega)$ 就是 $\check{f}(x)$ 的傅立葉轉換。Paley-Wiener 還有一個更明確的形式：令 $\zeta = u + iv$ ($u, v \in \mathbb{R}$)，則 $\text{supp } f = [a, b]$ 的充分必要條件是存在正常數 C_1, C_2, M_1 及 M_2 ，使得

$$|\hat{f}(\zeta)| \leq \begin{cases} C_1(1 + |\zeta|)^{M_1} e^{b|v|} & \text{if } v \geq 0 \\ C_2(1 + |\zeta|)^{M_2} e^{a|v|} & \text{if } v < 0 \end{cases} \quad \forall \zeta \in \mathbb{C}$$

這個定理，時常出現於有關有限函蓋的凌波函數論文中。

習題 2.44 證明

$$\int_0^{\infty} e^{-x} \cos \omega x dx = \frac{1}{1 + \omega^2}$$

習題 2.45 證明 (2.40) 中的 $g(x)$ 是 $L^2(\mathbb{R})$ 函數，但不是絕對值可積函數。

習題 2.46 對任意 $f(x) \in L^2([a, b])$ ，證明

$$\int_a^b |f(x)| dx \leq \sqrt{b-a} \|f(x)\|$$

所以 $f(x)$ 必定是絕對值可積。因此 $L^2([a, b]) \subset L^1([a, b])$ 。

習題 2.47 請討論 $L^1([a, b]) \subset L^2([a, b])$ 是否正確？

習題 2.48 若 $\text{supp } f(x) = [a, b]$ ，請討論 $\sum f(x + 2k\pi)$ 之中有哪些 k 對應的是非零項？(亦即 $f(x + 2k\pi) \neq 0$ 。)

習題 2.49 若 $\sum_k |\hat{f}(\omega + 2k\pi)|^2 = \frac{1}{2\pi}$ (是個常數函數)，證明

$$\langle f(x - k), f(x - m) \rangle = 0 \quad \forall k \neq m$$

那麼 $\|f(x)\|$ 是多少？

2.13 基本樣條函數

凌波函數的初期發現，來自於做調和分析的數學家。但是很快地，逼近論學者發現凌波函數與基本樣條函數 (*basic spline*) 有著密切的對應關係。從逼近論的眼光出發，也可以發展一套凌波函數的理論。這本書只探討正交凌波函數 (見第三章)，所以未必會用到基本樣條函數。但是許多其他類型的凌波函數，都與基本樣條函數有密切關係。而且基本樣條函數本身，也是應用數學中最常見的工具之一。所以我們在這章的最後一節簡單地介紹基本樣條函數，以備不時之需。順便，我們也從樣條函數的傅立葉轉換中，觀察一些普遍性的現象。

所謂 p 階基本樣條函數，就是

idx:Bspline

$$B_p(x) = B_1(x) * B_{p-1}(x), \quad \text{for } p > 1 \quad (2.51) \quad \text{eq:Bspline}$$

其中 $*$ 是疊積的意思。這個遞迴定義需要明確標定 $B_1(x)$ 。定義

$$B_1(x) = \begin{cases} 1, & \text{for } x \in [0, 1) \\ 0 & \text{otherwise} \end{cases} \quad (2.52) \quad \text{eq:Bone}$$

參照 (1.1)，我們發現 $B_1(x)$ 其實就是第一章的方盒函數。

idx:box

注意，為了以後符號上的方便，我們的樣條函數之階數編號與一般的書上不同。常見的編號方式是從 0 開始，所以我們的 $B_1(x)$ 就是

常見的零階樣條。以後讀者會看到 p 階自格函數和 p 階凌波函數之類的名詞，爲了對比的方便，我們統統用 p 代表它們的階數。但是， p 階樣條函數與 p 階自格函數未必是相同的（有時候相同）。請不要弄糊塗了。

樣條函數 (spline) 的理論與想法，首次出現於 1946 年 Schoenberg 的論文 [10]。樣條函數的應用，大約始於 1950 年代後期的汽車公司。工程師發明這個工具來做汽車設計。主要的人物有雪鐵龍 (Citroën) 公司的 de Casteljau、雷諾 (Renault) 公司的 Bezier 和通用公司 (GM) 的 de Boor。de Boor 奠定了基本樣條函數 (B-splines) 的基礎。基本樣條函數有非常豐富而美麗的性質，同時也是應用數學中的常見工具。在此我們僅列出它的幾個基本的性質。但是都不證明了。

性質 2.3 令 $p \geq 1$ 爲整數，而 $B_p(x)$ 是 p 階樣條函數，如 (2.51) 所定義。則 $\text{supp } B_p = [0, p]$ ，而且它對稱於 $\frac{p}{2}$ 。 $B_p(x)$ 在 \mathbb{R} 中是 $p-2$ 次的可微函數。而 $B_p(x)$ 在每一個整數區間 $[n, n+1]$ 內是一個 $p-1$ 階多項式，我們稱之爲分片多項式。 prop:Bsplir

根據 (2.51)，我們得到二階基本樣條 $B_2(x)$ 是

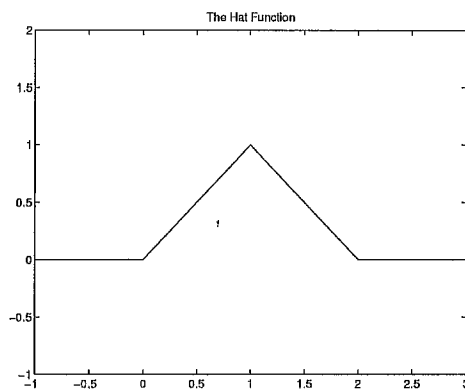
$$B_2(x) = \begin{cases} x & \text{for } x \in [0, 1) \\ 2-x & \text{for } x \in [1, 2) \\ 0 & \text{otherwise} \end{cases} \quad (2.53) \quad \text{eq:hat}$$

請讀者檢查它符合性質 2.3 中提到的每個性質。而 $B_2(x)$ 的圖形像是頂三角帽，所以它常被暱稱爲三角帽函數 (*hat function*)。 idx:hat

利用基本樣條函數的遞迴定義 (2.51)，我們發現其函數點值也符合一個遞迴關係。

性質 2.4 令 $B_p(x)$ 爲 p 階基本樣條函數，則 prop:Brecu

$$B_p(x) = \frac{x}{p-1} B_{p-1}(x) + \frac{p-x}{p-1} B_{p-1}(x-1)$$

fig:hat 圖 19 $B_2(x)$ (三角帽函數)

考慮

$$F_p(x) = \sum_k B_p(x - k)$$

意思是把 $B_p(x)$ 平移 k 個單位，然後相加起來。所以結果還是一個函數。又由於

$$\begin{aligned} F_p(x+1) &= \sum_{k=-\infty}^{\infty} B_p(x+1-k) = \sum_{n+1=-\infty}^{\infty} B_p(x-n) \\ &= \sum_{n=-\infty}^{\infty} B_p(x-n) = F_p(x) \end{aligned}$$

發現 $F_p(x)$ 是個以一單位為週期的週期函數 (記作 1-週期函數)。根據 `idx:one-per` 性質 2.4，其實 $F_p(x)$ 是個常數函數：

$$F_p(x) = 1 \tag{2.54} \quad \text{eq:Bsumone}$$

根據 (2.51) 和 (2.44)，

$$\mathcal{F}[B_p(x)](\omega) = \hat{B}_1(\omega)^p$$

而

$$\begin{aligned}\hat{B}_1(\omega) &= \int B_1(x)e^{-i\omega x} dx = \int_0^1 e^{-i\omega x} dx \\ &= \frac{1}{-i\omega} e^{-i\omega x} \Big|_{x=0}^{x=1} = \frac{i}{\omega} (e^{-i\omega} - 1)\end{aligned}$$

提出 $e^{-i\frac{\omega}{2}}$ ，得到

$$\begin{aligned}\hat{B}_1(\omega) &= \frac{i}{\omega} e^{-i\frac{\omega}{2}} (e^{-i\frac{\omega}{2}} - e^{i\frac{\omega}{2}}) = \frac{i}{\omega} e^{-i\frac{\omega}{2}} (-2i \sin \frac{\omega}{2}) \\ &= \frac{2}{\omega} e^{-i\frac{\omega}{2}} \sin \frac{\omega}{2} = \frac{\sin \frac{\omega}{2}}{\frac{\omega}{2}} e^{-i\frac{\omega}{2}}\end{aligned}$$

我們定義一個新的函數 $\text{sinc } x^*$ 為

idx:sinc

$$\text{sinc } x = \frac{\sin x}{x} \quad (2.55) \quad \text{eq:sinc}$$

因此

$$\hat{B}_1(\omega) = e^{-i\frac{\omega}{2}} \text{sinc } \frac{\omega}{2}$$

所以

$$\mathcal{F}[B_p(x)](\omega) = e^{-ip\frac{\omega}{2}} \left(\text{sinc } \frac{\omega}{2} \right)^p$$

$B_p(x)$ 都是有限函蓋函數。觀察 $\text{sinc } \omega$ 不是有限函蓋函數，所以 $\mathcal{F}[B_p(x)](\omega)$ 都不是有限函蓋函數。其實，這是一個普遍的現象：如果 $f(x)$ 是有限函蓋函數，則 $\hat{f}(\omega)$ 必定不是。反過來亦同：如果 $\hat{f}(\omega)$ 是有限函蓋函數，則 $f(x)$ 必定不是。也就是

$$\left. \begin{array}{l} f(x) \text{ 是有限函蓋} \\ \hat{f}(\omega) \text{ 是有限函蓋} \end{array} \right\} \text{ 勢不兩立}$$

* sinc 的讀音像是 sink 。

因為 $f(x)$ 與 $\hat{f}(\omega)$ 可以被認為是同一組訊號分別在時間域與頻率域的表象。所以說，任何函數都無法同時在時間域與頻率域為有限函蓋。在第三章，我們將有一個更明確的描述：海森堡測不準理論（數學版）。

在頻率域中的有限函蓋函數，又稱為有限頻寬 (*band-limited*) 函數。根據前面的論述，有限頻寬的訊號，在時間域中必為無限函蓋。這個現象可以用單農取樣定理*做更明確的描述。如果 $f(x) \in L^2$ 是有限頻寬函數，亦即存在 $N \in \mathbb{N}$ 使得當 $|\omega| > N\pi$ 時 $\hat{f}(\omega) = 0$ ，則整個 $f(x)$ 可以用離散的樣本點 (samples) 重建：

idx:band
idx:shannon

$$f(x) = \sum_k f\left(\frac{k}{N}\right) \operatorname{sinc}(N\pi x - k\pi)$$

其中 sinc 如 (2.55) 所定義。這個定理中的樣本點是 $\frac{k}{N}$ ，也就是每單位長中取 N 個樣本。我們稱其取樣頻率為 N 。又因為當 $-N\pi \leq \omega \leq N\pi$ 時 $\hat{f}(\omega)$ 可能不為 0，所以稱 $f(x)$ 的頻寬為 $2N$ 。用這套術語，單農取樣定理就是說

取樣頻率必須是頻寬的至少兩倍，才能從樣本重建原來的訊號

在我們常聽的 CD 雷射唱盤中，就有這個定理的應用。

從 $B_p(x)$ 和 $\mathcal{F}[B_p(x)](\omega)$ 中，我們還可以闡述另一個普遍性的現象。根據性質 2.3， $B_p(x)$ 是 $p-2$ 次可微，也就是說當 p 越大 $B_p(x)$ 就越平滑。觀察

$$|\mathcal{F}[B_p(x)](\omega)| = \left| \operatorname{sinc} \frac{\omega}{2} \right|^2 \leq \frac{1}{|\omega|^p}$$

* Shannon Sampling Theorem，這是訊息論 (information theory) 的奠基理論之一。因為我自己的姓氏 (單) 翻譯做 Shann，所以跟他攀點關係，翻譯做單農。

我們稱 $\mathcal{F}[B_p(x)](\omega)$ 的消散速率 (decay rate) 是 $\frac{1}{|\omega|^p}$ ，另一種記述的方法是

idx:decay

$$\mathcal{F}[B_p(x)](\omega) = O\left(\frac{1}{(1+|\omega|)^p}\right)$$

可見 $B_p(x)$ 越平滑則 $\mathcal{F}[B_p(x)](\omega)$ 就消散得越快。我們可以想像，當 p 小的時候， $\mathcal{F}[B_p](\omega)$ 要餘波盪漾很長一段距離才趨於平靜；當 p 大的時候， $\mathcal{F}[B_p](\omega)$ 很快就趨於平靜。其中 $B_1(x)$ 一點也不平滑（它甚至不是連續函數），而 $\hat{B}_1(\omega) \notin L^1(\mathbb{R})$ 。普遍性的現象是：

$f(x)$ 越平滑（也就是越多次可微的意思），則 $\hat{f}(\omega)$ 的消散速率越快；如果 $\hat{f}(\omega)$ 的消散速率越快，則 $f(x)$ 越平滑。亦可將的 $f(x)$ 與 $\hat{f}(\omega)$ 的角色互換。

在角色互換之後，前面的敘述是：如果 $f(x)$ 的消散速率越快，則 $\hat{f}(\omega)$ 越平滑。例如 $B_p(x)$ 都是有限函蓋，所以消散得比任意 $\frac{1}{|x|^n}$ 都快。可見 $\mathcal{F}[B_p](\omega)$ 應該非常平滑。其實 $\mathcal{F}[B_p](\omega)$ 是無窮次可微的函數。

習題 2.50 請寫出 $B_3(x)$ 和 $B_4(x)$ 在每個整數區間 $[n, n+1]$ 內之多項式函數。

習題 2.51 證明 (2.54)。(建議：可能利用性質 2.4 比較好做。)

習題 2.52 觀察 $B_1(x + \frac{1}{2})$ 是（幾乎）對稱於 $x = 0$ 的函數。求 $\mathcal{F}[B_1(x + \frac{1}{2})](\omega)$ 。

習題 2.53 證明

$$\mathcal{F}[f(x - a)](\omega) = e^{-ia\omega} \hat{f}(\omega)$$

所以 $f(x)$ 在時間域的平移，造成頻率域的相位差。以此現象驗證前一題中計算的 $\mathcal{F}[B_1(x + \frac{1}{2})](\omega)$ 。

習題 2.54 計算

$$\mathcal{F}[B_2(x+1)](\omega)$$

習題 2.55 計算

$$\int_{-\infty}^{\infty} \left| \frac{\sin x}{x} \right|^2 dx$$

(建議：可能利用 Parseval 等式 (2.43) 比較簡單。)

習題 2.56 只要 $\operatorname{sinc} x$ 是無窮次可微的函數，就不難看出 $\mathcal{F}[B_p](\omega)$ 是無窮次可微的函數。請嘗試證明 $\operatorname{sinc} x$ 無窮次可微。如果有困難，請利用電腦畫 $\operatorname{sinc} x$ 的圖形，並寫出 $\operatorname{sinc} x$ 的泰勒展開式，以便做直覺上的觀察。

第三章

正交凌波理論概述

在 1985 年前後，凌波函數還在數學家的襁褓之中，當時的傳播工具已經發展得很有效率，於是消息就傳了出去。地震波學者、量子物理學家、電磁波學者、數位訊號處理工程師，都在很早期的時候進入了這個剛被發現的礦場。多層解析空間的觀念大約在 1987 年由 Mallat 和 Meyer 發展出來，伴之而來的是一套製造各種不同凌波函數的系統化方法。Daubechies 運用這套方法製造出來一族正交有限函蓋的凌波函數，發表於 1988 年 [2]。數學、科學與工程界對於凌波函數的熱烈反應大約就這麼開始了。在她的發現之前，已經有 Meyer、Battle 和 Lemarié 的凌波函數族 (但都不是有限函蓋)。在她的論文之後，出現了更多族類的凌波函數。挑幾個名詞說說，有 Coiflets, 凌波包 (wavelet packets), 崔錦泰與王建中的崔王 (Chui-Wang) 半正交 (semiorthogonal) 凌波函數, Cohen、Daubechies、Feauveau 的 CDF 雙正交 (biorthogonal) 凌波函數, 分別由 Meyer、Daubechies、Jawerth、崔錦泰領導的四種有限區間中的凌波函數 (wavelets on finite intervals), Harten 的插值凌波函數, 陳宜良的差分凌波函數 (difference wavelets), 以及複合凌波 (multi-wavelets)。有趣的是，這些族類的凌波函數，大多有一個共同的最低階級，那就是第一章的 Haar 函數。

在這本書裡，我們僅以 Daubechies 的正交有限函蓋凌波函數族為例，介紹凌波函數的理論與應用。這一章的主題是理論的推演。雖然我們沒有提到其他凌波函數的理論與應用，但是以正交理論為核心，讀者可以發展其他種類凌波函數的理論。至於應用上的演算法，更是大同小異。

說給想要跳過這章的讀者們：這一章的理論將盡量處處與第一章對應。當您覺得太抽象的時候，可以對照第一章的情況，那可能會比較具體。雖然，要瞭解第四、五章的應用方法，並不是非瞭解這章的理論不可。但是在應用的章節裡，我們只說了如何做；在理論的章節裡，我們說了為什麼要這樣做。讀者諸君在各自的專業領域中，會遇到與書本裡完全不同的問題。對理論的掌握，必將成為將來發展更有創意的應用方法之鑰匙。何況，我相信只要是從第一章讀過來的人，必定可以瞭解這一章的內容。讓我們開始吧。

3.1 自格函數

在這一節裡，我們要推廣第一章的方盒函數，以及第一章的階梯函數空間 V_j 。若 $\phi(x) = B_1(x)$ 是方盒函數，則 $\phi(x - k)$ 可以合成常數函數。簡單地說， $\phi(x - k)$ 可以展開常數多項式。我們想要推廣 $\phi(x)$ ，使得新的 $\phi(x - k)$ 能夠展開更高階的多項式。（這是一個簡約的說法，後面將以嚴格的方式說明其意義。）而展開更高階多項式的目的又是什麼呢？這是為了使得 V_j 更為豐富。在第一章三節中，我們知道階梯函數空間 V_j 的逼近能力是 $O(\frac{1}{2^j})$ ，如果 V_j 更為豐富，也就是能夠包含更高階的多項式，則我們可以提高 V_j 的逼近能力。我們將要定義一個族群的自格函數，每一個正整數 $p \geq 1$ 將會對應一個自格函數，而第一章的方盒函數 $\phi(x)$ 對應 $p = 1$ 的狀況。

我們希望 $\phi(x)$ 保持哪些方盒函數的特色呢？首先希望 $\phi(x)$ 是有

限函蓋函數，並且 $\phi \in L^2$ 。其次，希望 $\{\phi(x - k)\}$ 是正交集，其中 $k \in \mathbb{Z}$ 。如果我們取 $\|\phi(x)\| = 1$ ，則

$$\int |\phi(2^j x - k)|^2 dx = \int |\phi(x)|^2 \frac{1}{2^j} dx = \frac{1}{2^j} \|\phi\|^2 = \frac{1}{2^j}$$

所以 $\|\phi(2^j x - k)\| = \frac{1}{\sqrt{2^j}}$ 。因此

$$\{\sqrt{2^j} \phi(2^j x - k) \mid k \in \mathbb{Z}\} \text{ 是一個正則集合} \quad (3.1) \quad \text{eq:phi0N}$$

令

$$V_j = \text{span} \{ \sqrt{2^j} \phi(2^j x - k) \mid k \in \mathbb{Z} \} \quad (3.2) \quad \text{eq:Vj}$$

我們希望 $\{V_j \mid j \in \mathbb{Z}\}$ 形成 L^2 的一組多層解析空間。亦即 $V_j \subset V_{j+1}$ 而且 $\cup V_j$ 稠密於 $L^2(\mathbb{R})$ 。

讓我們從 $V_0 \subset V_1$ 開始。由於 $\phi \in V_0$ ，所以 $\phi \in V_1$ 。而 $\{\phi(2x - k)\}$ 是 V_1 的基底，所以應該存在係數 c_k 使得

$$\phi(x) = \sum c_k \phi(2x - k)$$

注意，這其實就是推廣了 (1.49)。

一個寫成以下這種形式的等式

$$f(x) = \sum_k c_k f(2x - k), \quad c_k \in \mathbb{R}, k \in \mathbb{Z} \quad (3.3) \quad \text{eq:sceq}$$

稱作自格等式 (*scaling equation*)。這裡 \mathbb{R} 代表實數所成的集合， \mathbb{Z} 代表整數所成的集合。如果 $c_k \in \mathbb{R}$ 是一組係數，使得 (3.3) 有解；意即，存在某個函數 $f(x)$ 符合 (3.3)。那麼稱 $\{c_k\}$ 是一組自格係數 (*scaling coefficients*)，而 $f(x)$ 是一個自格函數 (*scaling function*)。 idx:scaling

它們的譯名如何而來？根據函數脹縮平移之特性，可見等式 (3.3) 的意義，就是 $f(x)$ 縮窄之後的幾個平移，可以經過某種線性組合而將自己重新拼湊出來。因此，我建議將符合 (3.3) 的函數稱為自格函數。所以，沿用這個名字， $\{c_k\}$ 是一組自格係數，(3.3) 是一個自格等式。

最基本的自格函數族群就是基本樣條函數 (*B-splines*)。讓我們取整數節點的一階和二階基本樣條為例。 $B_1(x)$ 和 $B_2(x)$ 分別符合以下的自格等式：

$$\begin{aligned} B_1(x) &= B_1(2x) + B_1(2x - 1) \\ B_2(x) &= \frac{1}{2}B_2(2x) + B_2(2x - 1) + \frac{1}{2}B_2(2x - 2) \end{aligned}$$

其實所有 p 階樣條函數都符合一個自格等式。

性質 3.1 令 $B_p(x)$ 是 p 階基本樣條函數，其中 p 是個正整數，則

$$B_p(x) = \sum_{k=0}^p \frac{1}{2^{p-1}} \binom{p}{k} B_p(2x - k)$$

其中 $\binom{p}{k} = \frac{p!}{k!(p-k)!}$ 是二項式係數，或楊輝三角之係數。

我們把這個性質的證明拖延到第十二節。雖然 $B_p(x)$ 是有限函蓋函數，可惜 $\{B_p(x - k)\}$ 並非正交集，所以它不是我們想要的*。

我們想要找到自格函數 $\phi(x) \in L^2(\mathbb{R})$ ，符合

$$\phi(x) = \sum_k c_k \phi(2x - k) \quad (3.4)$$

如果 $\int \phi dx = 0$ ，那代表 $\langle 1, \phi \rangle = 0$ ，所以 ϕ 垂直於常數函數。令 V_j 是由 (3.2) 所定義。所以， $1 \notin V_0$ 。其實，如果 $\int \phi dx = 0$ ，則

* 有許多其他族類的凌波函數，以基本樣條函數為其自格函數。

idx:Bspline

pdxpcRsoaëing

eq:scaling

$\int \phi(2^j x - k) dx = 0$ 。因此 $1 \notin V_j, \forall j \in \mathbb{Z}$ 。但是，根據第一章的探討，我們需要 $1 \in V_j$ 。所以我們要求

$$\int \phi(x) dx \neq 0$$

觀察 (3.4)。如果 $\phi(x)$ 是一個符合 (3.4) 的函數，則對任意的常數 $\alpha \neq 0$ ， $\alpha\phi(x)$ 也是一個符合 (3.4) 的函數。可見固定一組自格係數，若存在對應的自格函數，則存在無窮多個。既然前面已經要求 $\int \phi dx \neq 0$ ，我們可以規定

$$\int \phi(x) dx = 1 \tag{3.5} \quad \text{eq:phiunit}$$

這樣就唯一決定了一個符合 (3.4) 的自格函數。

其實， $\int \phi dx \neq 0$ 是本質上必要的條件。沒有了這個條件， $\phi(x)$ 就不會產生一個低頻濾波器，而我們也會失去一些多層解析空間的好性質。但是 $\int \phi dx = 1$ 並非本質的條件。只要規定 $\int \phi dx = c \neq 0$ ，在本質上並不會改變整套自格函數、多層解析空間、以及凌波函數的理論。只是在實用上不太方便，在理論的記述上不太漂亮，如此而已。話雖如此，我們還是從此規定 $\phi(x)$ 符合 (3.5) 這個條件。

如果 $\phi(x)$ 是個有限函蓋函數，我們可以平移這個函數，故可假設 $\text{supp } \phi(x) = [0, A]$ 。從性質 3.1 我們觀察，若 ϕ 是有限函蓋，則只需要有限多個非零的 c_k 即可符合 (3.4)。例如 $\text{supp } B_p = [0, p]$ ，而 B_p 的非零自格係數只有 $p + 1$ 個。

假設當 $k < M$ 和 $k > N$ 時， $c_k = 0$ 。所以

$$\phi(x) = \sum_{k=M}^N c_k \phi(2x - k)$$

由於

$$\text{supp } \phi(2x - k) = \left[\frac{k}{2}, \frac{k+A}{2} \right]$$

所以

$$\text{supp} \left(\sum_{k=M}^N c_k \phi(2x - k) \right) = \left[\frac{M}{2}, \frac{N+A}{2} \right]$$

如果我們令

$$\frac{M}{2} = 0, \quad \frac{N+A}{2} = A$$

也就是 $M = 0$ 而且 $A = N$ (我們令 $\text{supp } \phi$ 的長度一定是個整數 N)，就使得 (3.4) 簡化成

$$\phi(x) = \sum_{k=0}^N c_k \phi(2x - k)$$

爲了行文的方便，我們還是假設有無窮多個 c_k ，只是當 $k < 0$ 和 $k > N$ 時 $c_k = 0$ 。

最後，我們要提醒讀者，只有有限多個 $c_k \neq 0$ 的假設只是充分，而非必要。我們的確可以找到例子，使得 $\text{supp } \phi$ 是有限區間，但卻有無窮多個非零的自格係數。它們必須將超出 $\text{supp } \phi$ 的 $\phi(2x - k)$ 漸次相消成零。但是我們不去自找麻煩討論這種情況。

習題 3.1 證明 (3.1)。

3.2 自格係數

在前一節中，我們列出了希望 $\phi(x)$ 保有的性質。並且確定了目標是 $\text{supp } \phi = [0, N]$ 而非零的自格係數只有 c_0, c_1, \dots, c_N ，其中 N 是一個正整數。現在，我們從所希望的 $\phi(x)$ 的性質中，找到一些 c_k 的必要條件。

假設 $\{\phi(x-k)\}$ 是個正則集合。則 $\int \phi(x)\phi(x-k) dx = \delta_{0k}$ 。但是根據 (3.4)，

$$\begin{aligned} \int \phi(x)\phi(x-k) dx &= \sum_{\ell} \sum_{m} c_{\ell} c_m \int \phi(2x-\ell)\phi(2x-(2k+m)) dx \\ &= \frac{1}{2} \sum_{\ell} \sum_{m} c_{\ell} c_m \int \phi(x)\phi(x-(2k+m-\ell)) dx \\ &= \frac{1}{2} \sum_{m} c_{2k+m} c_m \end{aligned}$$

所以 c_k 必須符合

$$\sum_k c_k c_{k+2m} = 2\delta_{0m}, \quad \forall m \in \mathbb{Z} \quad (3.6) \quad \text{eq:ckorth}$$

如果 N 是個偶數，則可在 (3.6) 中取 $2m = N$ 。得到 $c_0 c_N = 0$ 。也就是 c_0 或 $c_N = 0$ 。不論何者，我們都可以把非零係數的個數減一。可見我們只需要考慮 N 是奇數的情形。

令 $p \geq 1$ 是個正整數，取 $N = 2p - 1$ 。所以，現在我們有 $2p$ 個未知的自格係數 c_k 。也就是

$$c_k = 0 \quad \text{for } k \leq -1 \text{ or } k \geq 2p \quad (3.7) \quad \text{eq:ckzero}$$

而且

$$\text{supp } \phi = [0, 2p - 1] \quad (3.8) \quad \text{eq:phisupp}$$

而 (3.6) 中只有當 $1-p \leq m \leq p-1$ 的時候才是有用的 (否則它們都是 $0=0$ 的恆等式)。再考慮足標變換：

$$\sum_k c_k c_{k+2m} = \sum_k c_k c_{k-2m} \quad (3.9) \quad \text{eq:ckm}$$

可見 m 和 $-m$ 在 (3.6) 中產生同樣的等式。因此 (3.6) 一共提供了 p 個不同的等式，分別是 $0 \leq m \leq p-1$ 的情形。這些是 c_k 所必須符合的 p 個條件。

再者，將 (3.4) 左右都積分，得到

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

因為 $\int \phi dx = 1$ ，故得

$$\sum_k c_k = 2 \tag{3.10}$$

eq:cktwo

(其實只要 $\int \phi dx \neq 0$ ，就可以得到 $\sum c_k = 2$ 。) 讓我們綜合這兩節之所得，作成以下結論：

給定一正整數 p ，若存在 $\phi(x) \in L^2(\mathbb{R})$ 使得

- (1) $\text{supp } \phi = [0, 2p-1]$ ， $\int \phi dx = 1$ ，
- (2) $\{\sqrt{2^j} \phi(2^j x - k) \mid k \in \mathbb{Z}\}$ 是一個正則集合，
- (3) 若 V_j 如 (3.2) 所定義，則 $\bigcup V_j$ 在 $L^2(\mathbb{R})$ 中稠密。

則存在一組自格係數 c_k ，符合 (3.7)、(3.6) 和 (3.10)，使得 $\phi(x)$ 是符合 (3.4) 的自格函數。

讓我們先接受，對任意的 $p \in \mathbb{N}$ ，符合上述三條件的 $\phi(x)$ 必定存在。(而且未必是唯一的。)

性質 3.2 若 c_k ， $0 \leq k \leq 2p-1$ ，符合 (3.6) 和 (3.10)，則

prop:ckone

$$\sum_k c_{2k} = \sum_k c_{2k+1} = 1$$

上面那條式子的意思是， c_k 的偶數項和與奇數項和都是 1。所以

$$\sum_k c_{2k} - \sum_k c_{2k+1} = \sum_k (-1)^k c_k = 0 \quad (3.11)$$

eq: ckappon

證明：令 $X = \sum c_{2k}$ ， $Y = \sum c_{2k+1}$ 。則

$$\begin{aligned} X^2 + Y^2 &= \left(\sum_k c_{2k}^2 + 2 \sum_k \sum_{m \geq 1} c_{2k} c_{2k+2m} \right) \\ &\quad + \left(\sum_k c_{2k+1}^2 + 2 \sum_k \sum_{m \geq 1} c_{2k+1} c_{2k+2m+1} \right) \quad (3.12) \\ &= \sum_k c_k^2 + 2 \sum_{m \geq 1} \sum_k c_k c_{k+2m} = 2 \end{aligned}$$

eq: XYsq

前面的最後一個等式利用了 (3.6)。由 (3.10)， $X + Y = 2$ ，所以

$$4 = (X + Y)^2 = X^2 + Y^2 + 2XY = 2 + 2XY$$

得到 $XY = 1$ 。因此 $(X - Y)^2 = 2 - 2XY = 0$ 。故 $X = Y = 1$ 。□

回顧這一節的討論，我們有 $2p$ 個未知數 c_k 。由 (3.6) 和 (3.10) 可以決定 $p + 1$ 個條件。當 $p = 1$ 時，條件數與未知數已經等量。兩個條件是

$$c_0^2 + c_1^2 = 2, \quad c_0 + c_1 = 2$$

唯一的一組解就是

$$c_0 = 1, \quad c_1 = 1 \quad (p = 1)$$

再加上 $\text{supp } \phi = [0, 1]$ 和 $\int \phi dx = 1$ 這兩個要求， $\phi(x) = B_1(x)$ 就是唯一符合這組自格係數的自格函數。所以，對應 $p > 1$ 的 $\phi(x)$ 就是方

盒函數的推廣，而那個 ϕ 所產生的多層解析空間 V_j 也就是階梯函數空間的推廣。

當 $p > 1$ 時，這些條件就不夠唯一決定一組 c_k 了。我們需要 $2p$ 個線性無關的條件。所以還缺 $p - 1$ 個。換個角度來看，也可以說自格係數 c_k 還有 $p - 1$ 個自由度可資利用。我們將可以利用這 $p - 1$ 個自由度，設計額外的 $p - 1$ 個條件，使得 $\phi(x)$ 增添某些性質。在後面，我們將所有的 $p - 1$ 個條件，全部用來增加 V_j 的逼近能力。讀者應該瞭解，除此之外還有別的可能。

最後，我們交代一些術語。爲了行文的方便，當 $\{\phi(x - k)\}$ 形成一個正則集合的時候，我們稱 $\phi(x)$ 是一個正則自格函數。因爲非零的正交集很容易就可以變化成正則集合，所以我們也會說 $\phi(x)$ 是一個正交自格函數。

idx:0Nphi

習題 3.2 證明當 $m \geq p$ 時 (3.6) 是恆等式。

習題 3.3 寫出當 $p = 3$ 時 (3.6) 所決定的 p 個等式。

習題 3.4 證明 (3.9)。

習題 3.5 請以 $p = 3$ 爲例驗證 (verify) (3.12)。

3.3 充分條件

綜合前兩節討論，給定一個正整數 p ，如果 $\phi(x)$ 是一個函蓋區間在 $[0, 2p - 1]$ 之內的正交自格函數，則 $\phi(x)$ 符合一個自格等式 (3.4)。爲了理論推演及應用上的方便，而且又不失其一般性，我們額外地要求 $\int \phi dx = 1$ ，而且自格係數 c_k 中只有 $2p$ 個非零項，(見 (3.7))。在這些假設之下，我們發現 c_k 必須符合 $p + 1$ 個條件 (3.6) 及 (3.10)。這些條件是 ϕ 之成爲正交自格函數的必要條件，在這一節中，我們簡短地探討，什麼是使得 ϕ 成爲正交自格函數的充分條件？

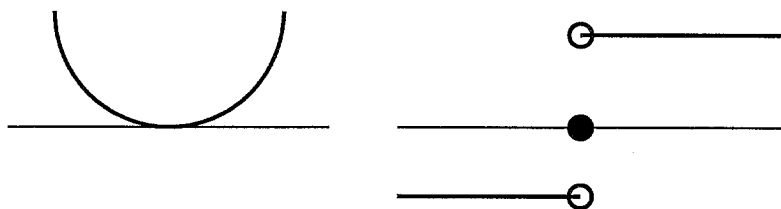


fig:notsuff 圖 20 中間值定理不成立之範例

首先，我們簡單地複習所謂的充分與必要條件。令 P 和 Q 是兩個數學命題，如果它們符合若 P 則 Q 的邏輯關係，簡記做 $P \Rightarrow Q$ 。我們稱 P 是 Q 的充分條件 (*sufficient condition*)， Q 是 P 的必要條件 (*necessary condition*)。比如說，令

idx:suff

$P = \text{“}f(x) \text{ 在 } [a, b] \text{ 內是連續函數，而且 } f(a)f(b) < 0\text{”}$

$Q = \text{“存在一個 } c \in (a, b) \text{ 使得 } f(c) = 0\text{”}$

那麼 $P \Rightarrow Q$ 。在數學中，這個事實稱做中間值定理。讀者必須注意， $P \Rightarrow Q$ 的意義是，如果 P 成立，則 Q 必成立。反之未必：如果 Q 成立， P 未必成立。例如圖 20 中所示的兩種情況，都是 Q 成立但 P 不成立的例子。

由此可見， c_k 符合 (3.6) 和 (3.10) 只是 ϕ 成為正交自格函數的必要條件，一般而言它們並不是充分條件。也就是說，它們並不能保證 ϕ 是正交自格函數。在此我們介紹 Lawton [7] 提出的充分條件。在第十二節中，利用傅立葉技巧還會得到一些其他形式的充分或必要條件。

令

$$T_n = \int \phi(x)\phi(x - n) dx$$

則根據自格等式得到

$$\begin{aligned} T_n &= \frac{1}{2} \sum_k \sum_m c_k c_m \int \phi(x) \phi(x - (2n + m - k)) dx \\ &= \frac{1}{2} \sum_k \sum_m c_m c_{m+k-2n} T_k \end{aligned} \quad (3.13)$$

eq:Tnp

如果我們令

$$t_{ij} = \frac{1}{2} \sum_m c_m c_{m+j-2i} \quad (3.14)$$

eq:matT

則 (3.13) 可寫成

$$T_n = \sum_k t_{nk} T_k$$

所以，如果令 $\mathbf{T} = (t_{ij})$ 為一矩陣，則

$$\mathbf{T} \begin{pmatrix} \vdots \\ T_{-1} \\ T_0 \\ T_1 \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ T_{-1} \\ T_0 \\ T_1 \\ \vdots \end{pmatrix}$$

我們稱 \mathbf{T} 為 *Lawton* 矩陣。考慮 *Lawton* 矩陣之維度。因為 $\text{supp } \phi = [0, 2p - 1]$ ，所以當 $|n| \geq 2p - 1$ 時 $T_n = 0$ 。因此，可能非零的 T_n 只有 $|n| \leq 2p - 2$ 的情形，共 $4p - 3$ 個。所以前面的向量長度應是 $4p - 3$ ，而 *Lawton* 矩陣的維度應是 $(4p - 3) \times (4p - 3)$ 。

如果 ϕ 是正交自格函數，則 $T_k = \delta_{0k}$ 。令

$$e = (\cdots 0 0 1 0 0 \cdots)^T$$

是長度 $4p - 3$ 的單位向量。 e 的中間位置元素是 1，其他元素都是 0。很明顯地，如果 ϕ 是正交自格函數，則以下等式成立：

$$\mathbf{T}e = e \quad (3.15) \quad \text{eq:Lawton}$$

所以它是一個必要條件。 Lawton 發現，它也是一個充分條件。亦即， idx:Lawton
如果 c_k 符合 (3.7)、(3.6)、(3.10) 和 (3.15)，而且 $\phi \in L^2$ 是滿足 (3.4) 的自格函數，則它是一個正交自格函數。

在 (3.15) 中，我們稱 e 是 \mathbf{T} 的一個特徵向量。在第五章三節與七節中，我們還會再回到這個課題。

習題 3.6 詳細推導 (3.13)。

3.4 凌波函數

到目前為止，我們有了 $\phi(x)$ ，它可以在 $L^2(\mathbb{R})$ 中形成一組多層解析空間 V_j 。雖然當 $j \rightarrow \infty$ 的時候， V_j 中的函數可以逼近 L^2 中的函數。但是，當 $j \rightarrow \infty$ 的時候， $\phi(2^j x - k)$ 的函蓋寬度 $\rightarrow 0$ ；我們很難有效地使用這些基底函數。簡單地說， V_j 無法提供給我們一個有用的逼近過程。什麼是“有用”的逼近過程呢？我們希望它是由粗到細、由近到遠的漸進過程。因此，雖然實際上我們不能做到無窮多個步驟，即使只做了前面有限多項，也可以得到一個合理的逼近效果。例如泰勒級數和傅立葉級數，都是有用的逼近過程。泰勒級數的基底是 $(x - c)^n$ ，隨著 n 的提高而漸次逼近。即使不能做到 $n \rightarrow \infty$ 的級數和，只做前面 N 項的和也差不多了。傅立葉級數的基底是 e^{-ikx} ，隨著 k 的提高而漸次逼近。即使不能做到 $k \rightarrow \pm\infty$ 的級數和，只做 $|k| \leq K$ 有限多項的和也差不多了。但是 V_j 的基底 $\phi(2^j x - k)$ 卻沒有這樣的性質。在 V_j 的逼近理論當中，我們無法讓 j 從小到大漸次逼近。

根據我們從第一章獲得的經驗：如果將 V_j 分解成 $V_{j-1} \oplus W_{j-1}$ ，如果 W_0 是由 $\psi(x-k)$ 所組成的，那麼 $V_0 \oplus W_0 \oplus W_1 \oplus W_2 \oplus \dots$ 就可以是一個有用的逼近過程了。這些空間的基底 $\phi(x-k)$ 、 $\psi(x-k)$ 、 $\psi(2x-k)$ 、 $\psi(2^2x-k)$ 、 \dots 提供了一個由粗到細、由近到遠的漸進過程。

現在，我們要找 $\psi(x)$ 。根據第一章的經驗，我們要求 $\{\psi(x-k)\}$ 形成正則集合；而且，若令 $W_0 = \text{span}\{\psi(x-k)\}$ ，則 $V_1 = V_0 \oplus W_0$ 。也就是說， W_0 是在 V_1 中正交於 V_0 的函數集合。令 \mathcal{P}_0 是從 L^2 到 V_0 的正交投影算子，則對任意的 $f \in V_1$ ，

idx:orthproj

$$\langle f - \mathcal{P}_0 f, u \rangle = 0, \quad \forall u \in V_0$$

所以 $f - \mathcal{P}_0 f \in W_0$ 。

定義

$$\phi^\perp(2x-k) := \phi(2x-k) - \mathcal{P}_0 \phi(2x-k)$$

則 $\phi^\perp(2x-k) \in W_0$ ，所以

$$\text{span}\{\phi^\perp(2x-k) \mid k \in \mathbb{Z}\} \subset W_0$$

反過來，若 $f \in W_0$ ，則 $\langle f, \phi(x-k) \rangle = 0$ 所以 $\mathcal{P}_0 f = 0$ 或 $f = f - \mathcal{P}_0 f$ 。由於 $\{\phi(2x-k)\}$ 是 V_1 的基底，故存在 v_k 使得

$$f = \sum_k v_k \phi(2x-k) - \mathcal{P}_0 \left(\sum_k v_k \phi(2x-k) \right) = \sum_k v_k \phi^\perp(2x-k)$$

所以

$$W_0 \subset \text{span}\{\phi^\perp(2x-k) \mid k \in \mathbb{Z}\}$$

因此我們得到

$$W_0 = \text{span} \{ \phi^\perp(2x - k) \mid k \in \mathbb{Z} \}$$

現在，我們先觀察 $\mathcal{P}_0\phi(2x - \ell)$ 。由於 $\{\phi(x - k)\}$ 是 V_0 的正則基底，所以

$$\mathcal{P}_0\phi(2x - \ell) = \sum_k \langle \phi(2x - \ell), \phi(x - k) \rangle \phi(x - k)$$

而根據 (3.4) 和 $\int |\phi(x)|^2 dx = 1$ ，得到

$$\begin{aligned} \langle \phi(2x - \ell), \phi(x - k) \rangle &= \sum_n c_n \langle \phi(2x - \ell), \phi(2x - 2k - n) \rangle \\ &= \frac{1}{2} c_{\ell - 2k} \end{aligned} \tag{3.16}$$

eq: Pell

所以

$$\begin{aligned} \phi^\perp(2x - \ell) &= \phi(2x - \ell) - \mathcal{P}_0\phi(2x - \ell) \\ &= \phi(2x - \ell) - \frac{1}{2} \sum_k c_{\ell - 2k} \phi(x - k) \end{aligned}$$

讓我列出幾個例子：

$$\begin{aligned} \phi^\perp(2x) &= \phi(2x) - \frac{1}{2} [c_{2p-2}\phi(x+p-1) \\ &\quad + c_{2p-4}\phi(x+p-2) + \cdots + c_0\phi(x)] \\ \phi^\perp(2x-1) &= \phi(2x-1) - \frac{1}{2} [c_{2p-1}\phi(x+p-1) \\ &\quad + c_{2p-3}\phi(x+p-2) + \cdots + c_1\phi(x)] \\ \phi^\perp(2x-2) &= \phi(2x-2) - \frac{1}{2} [c_{2p-2}\phi(x+p-2) + \cdots \\ &\quad + c_2\phi(x-2) + c_0\phi(x-1)] \\ \phi^\perp(2x-3) &= \phi(2x-3) - \frac{1}{2} [c_{2p-1}\phi(x+p-2) + \cdots \\ &\quad + c_3\phi(x-2) + c_1\phi(x-1)] \end{aligned}$$

亦即

$$d_k = (-1)^k c_{2p-1-k} \quad (3.17) \quad \text{eq:dk}$$

則 $\mathbf{d}^T \mathbf{A} = \mathbf{0}$ ($\mathbf{0}$ 表示對應維度的零矩陣)。所以

$$\mathbf{d}^T \begin{pmatrix} \phi^\perp(2x) \\ \phi^\perp(2x-1) \\ \phi^\perp(2x-2) \\ \phi^\perp(2x-3) \\ \vdots \\ \phi^\perp(2x-2p+4) \\ \phi^\perp(2x-2p+3) \\ \phi^\perp(2x-2p+2) \\ \phi^\perp(2x-2p+1) \end{pmatrix} = \mathbf{d}^T \begin{pmatrix} \phi(2x) \\ \phi(2x-1) \\ \phi(2x-2) \\ \phi(2x-3) \\ \vdots \\ \phi(2x-2p+4) \\ \phi(2x-2p+3) \\ \phi(2x-2p+2) \\ \phi(2x-2p+1) \end{pmatrix} \quad (3.18)$$

eq:perpA

爲了行文的方便，我們假設 d_k 對所有的 $k \in \mathbb{Z}$ 均存在，只是當 $k \leq -1$ 或 $k \geq 2p$ 時 $d_k = 0$ 。因此 (3.18) 可以寫成

$$\sum_k d_k \phi^\perp(2x+k) = \sum_k d_k \phi(2x-k)$$

所以

$$\mathcal{P}_0\left(\sum_k d_k \phi(2x-k)\right) = \sum_k d_k \phi(2x-k) - \sum_k d_k \phi^\perp(2x-k) = 0$$

可見 $\sum_k d_k \phi(2x-k) \in W_0$ 。令

$$\psi(x) = \sum_k d_k \phi(2x-k) = \sum_{k=0}^{2p-1} (-1)^k c_{2p-1-k} \phi(2x-k) \quad (3.19)$$

eq:psishift

讀者可以檢查，因爲 $\text{supp } \phi = [0, 2p-1]$ ， $\text{supp } \psi$ 也是 $[0, 2p-1]$ 。而且，因爲 $\phi \in L^2$ ， ψ 也是 L^2 函數。我們要提醒讀者，一般文獻中常見

的定義是

$$\psi(x) = \sum_{k=2-2p}^1 (-1)^k c_{1-k} \phi(2x - k) \quad (3.20)$$

eq:psi

我們發現若 $\text{supp } \phi = [0, 2p - 1]$ ，則由上式定義的 $\psi(x)$ 符合

$$\text{supp } \psi = \left[\frac{1}{2}(2 - 2p), \frac{1}{2}(2p - 1) + \frac{1}{2} \right] = [1 - p, p]$$

有時候我們會覺得這個函蓋的範圍不方便，因為我們希望將 $\psi(x)$ 與 $\phi(x)$ 對齊。比方說拿 ϕ 與 ψ 來解微分或積分方程的時候，如果 $\phi_{jk}(x)$ 的函蓋與 $\psi_{jk}(x)$ 一致，會方便一些。所以我們選擇用 (3.19) 來定義 $\psi(x)$ 。這時候的 $\text{supp } \psi(x) = [0, 2p - 1]$ 就與 $\text{supp } \phi(x)$ 一樣了。而這個經過平移的 $\psi(x)$ 的定義，完全不影響 $\psi(x)$ 原有的性質。由 (3.19) 定義的 ψ 與 (3.20) 中定義的 ψ 彼此互為對方的平移。

因為

$$\begin{aligned} \int \psi(x)\psi(x - k) dx &= \sum_{\ell} \sum_m d_{\ell} d_m \int \phi(2x - \ell)\phi(2x - (2k + m)) dx \\ &= \frac{1}{2} \sum_{\ell} \sum_m d_{\ell} d_m \int \phi(x)\phi(x - (2k + m - \ell)) dx \\ &= \frac{1}{2} \sum_m d_{2k+m} d_m \end{aligned}$$

由於 $2k + m$ 和 m 同為奇數或同為偶數，所以，根據 d_k 的定義 (3.17) 得到

$$\sum_m d_{2k+m} d_m = \sum_m c_m c_{m+2k}$$

根據 (3.6)，我們發現 $\int \psi(x)\psi(x - k) dx = \delta_{0k}$ 。經過變數變換，讀者很容易可以檢查

$$\langle \psi(x - m), \psi(x - n) \rangle = \delta_{mn}$$

所以

$$\{\sqrt{2^j}\psi(2^j x - k) \mid k \in \mathbb{Z}\} \text{ 是一個正則集合} \quad (3.21) \quad \text{eq:psi0N}$$

我們定義

$$W_j = \{\sqrt{2^j}\psi(2^j x - k) \mid k \in \mathbb{Z}\} \quad (3.22) \quad \text{eq:Wj}$$

接著，檢驗

$$\begin{aligned} \int \psi(x)\phi(x-k) dx &= \sum_{\ell} \sum_m d_{\ell} c_m \int \phi(2x-\ell)\phi(2x-(2k+m)) dx \\ &= \frac{1}{2} \sum_{\ell} \sum_m d_{\ell} c_m \int \phi(x)\phi(x-(2k+m-\ell)) dx \\ &= \frac{1}{2} \sum_m (-1)^m c_{2p-2k-(m+1)} c_m \end{aligned}$$

因為

$$\begin{aligned} &\sum_m (-1)^m c_{2p-2k-(m+1)} c_m \\ &= \sum_{m=0}^{p-1} (-1)^m c_{2p-2k-(m+1)} c_m + (-1)^{2p-2k-(m+1)} c_m c_{2p-2k-(m+1)} \\ &= \sum_{m=0}^{p-1} (-1)^m c_{2p-2k-(m+1)} c_m + (-1)^{m+1} c_{2p-2k-(m+1)} c_m = 0 \end{aligned} \quad (3.23) \quad \text{eq:summ}$$

所以 $\int \psi(x)\phi(x-k) dx = 0$ ；經過變數變換，讀者很容易可以檢查

$$\langle \psi(x-m), \phi(x-n) \rangle = 0, \quad \forall m, n \in \mathbb{Z}$$

更進一步，

$$\langle \psi(2^j x - m), \phi(2^j x - n) \rangle = 0, \quad \forall j, m, n \in \mathbb{Z}$$

但是因爲 $V_{j-1} \subset V_j$ ，所以

$$\langle \psi(2^j x - m), \phi(2^l x - n) \rangle = 0, \quad \forall l \leq j \in \mathbb{Z}, m, n \in \mathbb{Z}$$

簡單地說，就是對所有的 $l \leq j$ ， $W_j \perp V_l$ 。

最後，我們驗證 $V_{j+1} = V_j \oplus W_j$ 。經過變數變換，其實只需要驗證 $V_1 = V_0 \oplus W_0$ 。我們已經知道 $V_0 \perp W_0$ ，而且 $V_0 + W_0 \subset V_1$ （因爲 (3.4) 和 (3.20)），所以只需要驗證 $V_1 \subset V_0 \oplus W_0$ 。由於 $\{\phi(2x - k)\}$ 是 V_1 的基底，只要我們證明 $\phi(2x - k)$ 可以被 $\{\phi(x - k)\}$ 和 $\{\psi(x - k)\}$ 展開，就完成了這段論述。

回顧在第一章十六節所獲得的經驗。經過平移，我們只需要證明 $\phi(2x)$ 和 $\phi(2x - 1)$ 可以被 $\{\phi(x - k)\}$ 和 $\{\psi(x - k)\}$ 展開即可。利用 (1.60) 的符號系統，我們要找兩組係數 p_k 和 q_k 使得以下關係成立。

$$\begin{aligned} \phi(2x) &= \frac{1}{2} \sum_k p_{-2k} \phi(x - k) + \frac{1}{2} \sum_k q_{-2k} \psi(x - k) \\ \phi(2x - 1) &= \frac{1}{2} \sum_k p_{1-2k} \phi(x - k) + \frac{1}{2} \sum_k q_{1-2k} \psi(x - k) \end{aligned} \quad (3.24)$$

eq: pqagain

我們只推導 $\phi(2x)$ 的等式，另一個等式可以用類似的步驟求得。

如果 (3.24) 中 $\phi(2x)$ 的等式成立，將等式之左右分別與 $\phi(x - n)$ 作內積。則得

$$\int \phi(2x) \phi(x - n) dx = \frac{1}{2} p_{-2n}$$

但是 $\langle \phi(2x), \phi(x - n) \rangle = \frac{1}{2} c_{-2n}$ ，因此 $p_{2k} = c_{2k}$ 。同理 $q_{2k} = d_{2k}$ 。以上是 $\phi(2x)$ 等式成立的必要條件。現在我們直接將 p_{-2k} 和 q_{-2k} 的值代入 (3.24)，來驗證 $\phi(2x)$ 的等式成立：

$$\frac{1}{2} \sum_k p_{-2k} \phi(x - k) + q_{-2k} \psi(x - k)$$

$$\begin{aligned}
 &= \frac{1}{2} \sum_{k=0}^{p-1} c_{2k} \phi(x+k) + d_{2k} \psi(x+k) \\
 &= \frac{1}{2} \sum_k \sum_n (c_{2k} c_n + d_{2k} d_n) \phi(2x+2k-n) \\
 &= \frac{1}{2} \sum_m \left(\sum_k c_{2k} c_{2k+m} + (-1)^m c_{2p-1-2k} c_{2p-1-2k-m} \right) \phi(2x-m)
 \end{aligned}$$

最後一條式子，是因為足標變換 $2k-n=-m$ 。我們將上式中的 m 分成奇數項與偶數項來討論。若 m 為偶數，則括號內為

$$\begin{aligned}
 &\sum_k c_{2k} c_{2k+2m} + c_{2p-1-2k} c_{2p-1-2k-2m} \\
 &= \sum_k c_{2k} c_{2k+2m} + \sum_n c_{2n-1} c_{2n-1+2m} \quad (\text{let } 2p-2k-2m=2n) \\
 &= \sum_k c_k c_{k+2m} = 2\delta_{0m}
 \end{aligned}$$

若 m 為奇數，則括號內為

$$\begin{aligned}
 &\sum_k c_{2k} c_{2k+2m-1} - c_{2p-1-2k} c_{2p-2k-2m} \\
 &= \sum_k c_{2k} c_{2k+2m-1} - \sum_n c_{2n} c_{2n+2m-1} \quad (\text{let } 2p-2k-2m=2n) \\
 &= 0
 \end{aligned}$$

可見

$$\frac{1}{2} \sum_k p_{-2k} \phi(x-k) + q_{-2k} \psi(x-k) = \frac{1}{2} \sum_m 2\delta_{0m} \phi(2x-m) = \phi(2x)$$

因此，我們證明了，當 $p_{2k} = c_{2k}$ 且 $q_{2k} = d_{2k}$ 時，(3.24) 中 $\phi(2x)$ 的等式成立。

用完全一樣的做法，可證明當 $p_{2k-1} = c_{2k-1}$ 且 $q_{2k-1} = d_{2k-1}$ 時，(3.24) 中 $\phi(2x-1)$ 的等式成立。綜合以上所得，就是

$$p_k = c_k, \quad q_k = d_k \quad (3.25) \quad \text{eq:pqk}$$

現在我們看的出來，為什麼 (3.24) 中的足標要寫得那麼奇怪 ($-2k$ 和 $1-2k$)。這是因為只有當 $\phi(x-k)$ 將 $\phi(2x)$ 和 $\phi(2x-1)$ 整個函蓋住的時候，才會有係數 $p_k \neq 0$ 。當 $k > 0$ 時，顯然 $\phi(x-k)$ 不能將 $\phi(2x)$ 和 $\phi(2x-1)$ 整個函蓋住。只有當

$$1-p \leq k \leq 0$$

的時候， $\phi(x-k)$ (向左平移) 才能將 $\phi(2x)$ 和 $\phi(2x-1)$ 整個函蓋住。因此，如果按照 (1.60) 的形式，則非零的 p_k 與 q_k 係數出現於

$$0 \leq k \leq 2p-1$$

其足標與 c_k 相對應。

至此，我們發現，四組係數 c_k 、 d_k 、 p_k 和 q_k 其實全部由 c_k 所決定。這個現象只發生在 $\{\phi(x-k)\}$ 是正交集合的情況下。否則未必有這個現象。但是，即使在一般的情況下，這四組係數總是會具備某種關係。

因為我們假設 $\cup V_j$ 在 $L^2(\mathbb{R})$ 中稠密，而

$$V_j = W_{j-1} \oplus W_{j-2} \oplus \cdots \oplus W_0 \oplus V_0$$

若我們推廣 (3.22) 中 W_j 的定義到 $j < 0$ (ψ 膨脹而非縮窄)，則上式可以繼續往下寫：

$$V_j = W_{j-1} \oplus W_{j-2} \oplus \cdots \oplus W_0 \oplus W_{-1} \oplus \cdots$$

當 $j \rightarrow \infty$ 時得到

$$L^2(\mathbb{R}) = \bigoplus_{j=-\infty}^{\infty} W_j \quad (3.26)$$

eq:LtwoWj

因此 (2.17) 可以改寫成

$\{\psi_{jk}(x) \mid j \in \mathbb{Z}, k \in \mathbb{Z}\}$ 是 $L^2(\mathbb{R})$ 的一組正則基底

其中

$$\psi_{jk}(x) = \sqrt{2^j} \psi(2^j x - k) \quad (3.27)$$

eq:psijkaga

順便要在此闡述，雖然 (3.26) 成立，但是在應用上，我們通常只能處理有限解析度的資訊，因此我們考慮的函數空間是某個 V_j 而非 $L^2(\mathbb{R})$ 。同樣地我們無法真的處理 $j \rightarrow -\infty$ 情況下的 W_j 空間，所以我們會停在某個 V_j 空間。因此 V_j 將是實際上的最高解析度，而 V_j 將是實際上的最低解析度。又因為通常我們只考慮一個閉區間內的函數，所以 k 不會佈滿整個 \mathbb{Z} ，而是 $N \leq k \leq M$ 。

根據 (3.11)，我們可以證明

$$\int \psi(x) dx = 0 \quad (3.28)$$

eq:psiappor

也就是說 $1 \perp \psi$ ，故 $1 \perp W_0$ ，其實 $1 \perp W_j$ 。

因為 (3.28)， $\psi(x)$ 的函數圖形必定上下起伏（相對於 0），像個波浪。又因為 $\|\psi(x)\| = 1$ ，所以使其振盪的能量並不大（只是 1）。所以 $\psi(x)$ 的函數圖形是個小小的波浪。我們稱 $\psi(x)$ 是一個凌波函數。有關這個名詞的由來和翻譯，請回顧第一章十三節的闡述。

一般而言，若存在一個函數 $\psi(x)$ ，符合

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

$$(2) \|\psi\| = 1$$

(3) $\{\psi(2^j x - k) \mid j, k \in \mathbb{Z}\}$ 形成 $L^2(\mathbb{R})$ 的一組基底

則我們稱 $\psi(x)$ 為一個凌波函數 (*wavelet*) 有些文獻上會說 $\psi(2^j x - k)$ 都是凌波函數，而特別稱 $\psi(x)$ 是一個凌波母函數 (*mother wavelet*)。相對地，有些人稱 $\phi(x)$ 是凌波父函數* (*father wavelet*)。

最後，我們再交代一些術語。當 $\{\psi(x - k)\}$ 形成一個正則集合的時候，我們說 $\psi(x)$ 是一個正則凌波函數 或 正交凌波函數。為了與其他種類的凌波函數區分，在此定義的 $\psi(x)$ ，又特別稱為 Daubechies 之 p 階有限函蓋正交凌波函數；它們對應的 $\phi(x)$ 又特別稱為 Daubechies 之 p 階有限函蓋正交自格函數。

習題 3.7 證明 (3.16)。

習題 3.8 請以 $p = 3$ 驗證 (3.23)。

習題 3.9 證明 (3.28)。

習題 3.10 證明 (3.24) 中 $\phi(2x - 1)$ 的等式。

習題 3.11 令 $\mathcal{P}_0 : L^2 \rightarrow V_0$ 和 $\mathcal{Q}_0 : L^2 \rightarrow W_0$ 分別是映射到 V_0 和 W_0 的正交投影算子。請問 $\mathcal{P}_0 \phi(2x)$ 和 $\mathcal{Q}_0 \phi(2x)$ 是什麼？而 $(\mathcal{P}_0 + \mathcal{Q}_0)\phi(2x)$ 是什麼？

* 我個人並不喜歡這套說法。“母函數”還可以接受，但是“父函數”就太牽強了。因為 $\phi \perp \psi$ ，為什麼“父”要和“母”垂直呢？何況 $\psi(x)$ 還是從 $\phi(2x - k)$ 製造出來的（你可想到了亞當和他那根肋骨的故事？）而且 $\psi(x)$ 和 $\psi(2^j x - k)$ 之間，只是相對的脹縮平移之關係， j 與 k 只是相對的參數，並非絕對的。只要換一下參數，任何 $\psi(2^j x - k)$ 都可以被當做“母”函數來看。

3.5 波係數消散定理

回顧第一章十四節中的 Haar 波係數消散定理，其證明過程中的關鍵步驟乃是在 (1.41) 中利用 $\int \psi dx = 0$ 的性質，使得 $\psi(x)$ 可以消去 $f(x)$ 的零階泰勒多項式 (常數項)。對任何 $p > 1$ 的 $\psi(x)$ ，由於 (3.28)，使得 ψ 總是正交於常數函數。所以，我們可以依樣畫葫蘆地證明，對任意 $f \in C^1$ ，

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

其中 $\langle f, \psi_{jk} \rangle$ 就是 $f(x)$ 在 $\psi_{jk}(x)$ 的投影係數，簡稱為波係數。在這一節裡，我們將要做更詳細的分析；並且探討在什麼條件下，可以加快波係數的消散速度。

idx:wavcoef

首先，我們定義 Hölder 連續指數。藉由這個指數，我們將連續而不可微的函數做更詳細的分類。一個在 x_0 處連續的函數 $f(x)$ 和某個定數 $0 < \sigma \leq 1$ ，如果存在某常數 $0 \leq C < \infty$ 和 $h > 0$ 使得

$$|f(x) - f(x_0)| \leq C|x - x_0|^\sigma, \quad \forall x \in (x_0 - h, x_0 + h)$$

則稱 $f(x)$ 在 x_0 的 Hölder 連續指數是 σ ，或稱 $f(x)$ 在 x_0 處是 σ 階 Hölder 連續。如果對所有的 $x, y \in [a, b]$ ，都符合

idx:Holder

$$|f(x) - f(y)| \leq C|x - y|^\sigma, \quad \forall x, y \in [a, b] \quad (3.29) \quad \text{eq:holder}$$

則稱 $f(x)$ 在 $[a, b]$ 內是 σ 階 Hölder 連續，計作

$$f(x) \in C^{0,\sigma}([a, b])$$

$\frac{1}{\sigma}$

另兩種記述方法是

$$\lim_{y \rightarrow x} \frac{|f(y) - f(x)|}{|y - x|^\sigma} < \infty \quad \text{或} \quad |f(x) - f(y)| = O(|x - y|^\sigma)$$

\leq
 $=$

注意，此處我們稍微放寬了大 O 記號的意義。以前我們規定 $0 < C < \infty$ ，現在我們容許 $0 \leq C < \infty$ 。

在微積分中，我們把函數分類為不連續函數、連續函數 (C^0)、可微函數、可微而且一次導函數連續之函數 (C^1)，等等。試想，如果將所有連續但不可微的函數全部被歸類在一起，那是很粗糙的分類。因為這些函數當中，有連續但每一個點都不可微的函數，也有除了一個點之外其他都可微的函數*。Hölder 連續指數就是把所有連續而不可微的函數作更細的分類。 σ 比較小的函數，可以想像是鋸齒狀比較嚴重，距離可微比較遠的函數。反過來， σ 比較大的函數，可以想像是鋸齒狀比較不嚴重，幾乎可微的函數。當 $\sigma = 1$ 時， $C^{0,1}$ 就是 Lipschitz 連續函數空間，這種函數是分片可微的連續函數。例如 $|x| \in C^{0,1}(\mathbb{R})$ 。在第五章四節中，我們會看到具體的 Hölder 連續指數之範例。

注意，我們只考慮 $0 < \sigma \leq 1$ 的 Hölder 指數。(當 $\sigma = 0$ 的時候 (3.29) 沒有意義。當 $\sigma < 0$ 時， $f(x)$ 未必是連續函數。例如取

$$f(x) = \begin{cases} \frac{1}{\sqrt{|x|}} & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases}$$

顯然 $f(x)$ 在 $x = 0$ 處不連續。若取 $\sigma = -0.5$ ，則

$$\lim_{x \rightarrow 0} \frac{|f(x) - f(0)|}{|x - 0|^\sigma} = \lim_{x \rightarrow 0} \frac{|x|^{0.5}}{\sqrt{|x|}} = 1 < \infty$$

當 $\sigma > 1$ 時，

$$|f'(x)| = \lim_{y \rightarrow x} \frac{|f(y) - f(x)|}{|y - x|} = \lim_{y \rightarrow x} \frac{|f(y) - f(x)|}{|y - x|^\sigma} \lim_{y \rightarrow x} |y - x|^{\sigma-1} = 0$$

* 前者較難舉例，通常在數學系的高等微積分中，會舉出這樣的例子。它不是一個簡單的基本函數，必須透過函數級數的和來定義。讀者可以想像它有無限多鋸齒的連續函數。後者的一個例子是 $y = |x|$ 。

可見這些函數都符合 $f'(x) = 0$ ；也就是說，符合 $\sigma > 1$ 的 Hölder 連續函數都是常數函數。所以沒有什麼值得討論的。

以下的性質告訴我們，只要 $f(x)$ 的 Hölder 連續指數 $\sigma > 0$ ，則其凌波係數就會消散。

性質 3.3 對任意凌波函數 $\psi(x)$ ， ψ_{jk} 如 (3.27) 所定義。若

prop:psisig

$$f(x) \in C^{0,\sigma}(\text{supp } \psi_{jk})$$

則

$$|\langle f(x), \psi_{jk}(x) \rangle| = O(2^{-(\sigma+\frac{1}{2})j}) \quad (3.30) \quad \text{eq:psisigma}$$

證明：因為 (3.28)，所以

$$\int f(x)\psi(2^jx - k) dx = \int (f(x) - f(\frac{k}{2^j}))\psi(2^jx - k) dx$$

上式的積分範圍是 $\text{supp } \psi_{jk}$ 。所以利用 (3.29) 得到

$$\begin{aligned} |\langle f(x), \psi_{jk}(x) \rangle| &\leq \sqrt{2^j} \int |f(x) - f(\frac{k}{2^j})| |\psi(2^jx - k)| dx \\ &\leq \sqrt{2^j} C \int |x - \frac{k}{2^j}|^\sigma |\psi(2^jx - k)| dx \\ &= \frac{C}{\sqrt{2^j}} \int |\frac{x}{2^j}|^\sigma |\psi(x)| dx \\ &= C \int |x|^\sigma |\psi(x)| dx 2^{-(\sigma+\frac{1}{2})j} \end{aligned}$$

上面的等式乃是利用了變數變換： $x \mapsto \frac{x+k}{2^j}$ 。因為 ψ 是有限函蓋函數，所以 $\int |x|^\sigma |\psi| dx < \infty$ ，將 $C \int |x|^\sigma |\psi| dx$ 視為一個常數，故得到 (3.30) 中的大 O 符號。 □

請注意，當 $f \in C^{0,1}$ 時，(3.30) 的結果即與 Haar 凌波係數消散定理中的 $f \in C^1$ 結果相同。但是 $C^1 \subset C^{0,1}$ ，所以性質 3.3 的應用範圍比較廣。何況性質 3.3 還描述了更多的狀況（當 $0 < \sigma < 1$ 的狀況）。可見性質 3.3 是 Haar 凌波係數消散定理的推廣。亦請注意，性質 3.3 中對於 $f(x)$ 的 Hölder 指數之要求，乃是局部的一只在 $\text{supp } \psi_{jk}$ 之中。

idx:Haarshm

在性質 3.3 的證明過程當中，我們需要 $\int |x|^\sigma |\psi| dx < \infty$ 。這對有限函蓋函數來說，是簡單的。但如果 $\psi(x)$ 不是有限函蓋，則這個額外的條件就等於是要求 $|\psi|$ 的消散速率。例如，若 $\sigma = 0.5$ 而 $|\psi|$ 的消散速率是 $O(\frac{1}{|x|^{1.5}})$ ，則當 $|x|$ 夠大時 $|x|^\sigma |\psi| \approx \frac{1}{|x|}$ ，所以 $|x|^\sigma |\psi|$ 是不可積的（積分值 $\rightarrow \infty$ ）。一般而言，若

idx:decay

$$|\psi(x)| = O\left(\frac{1}{(1+|x|)^{\sigma+1+\epsilon}}\right), \quad \text{for some } \epsilon > 0$$

則其消散速率夠快，足以使得 $\int |x|^\sigma |\psi| dx < \infty$ 。而我們在第二章十三節中曾經探討， $f(x)$ 的消散速率等價於 $\hat{f}(\omega)$ 的平滑程度。所以，如果 $\psi(x)$ 不是有限函蓋，則性質 3.3 還需要對 $\psi(x)$ 的消散速率或 $\hat{\psi}(\omega)$ 的平滑程度做額外的假設，才能成立。本書中不探討非有限函蓋的凌波函數，所以不再多說了。

接下來，我們探討，什麼時候能夠使得凌波係數消散得更快呢？以下的定理就做了明確的刻劃。它告訴我們，如果 $\psi(x)$ 正交於越高階的多項式，而且 $f(x)$ 本身越平滑，則凌波係數 $\langle f, \psi_{jk} \rangle$ 就會消散得越快。先介紹一個符號： $f \in C^{n,\sigma}$ 的意思是 $f(x)$ 為 n 次可微，而其第 n 階導函數 $f^{(n)} \in C^{0,\sigma}$ 。

凌波係數消散定理 令 $\psi(x)$ 為一凌波函數，其正交於階數 $\leq n$ 之多項式；亦即

$$\int x^m \psi(x) dx = 0, \quad \text{for } 0 \leq m \leq n$$

令 ψ_{jk} 如 (3.27) 所定義。若

$$f(x) \in C^{n,\sigma}(\text{supp } \psi_{jk})$$

則

$$|\langle f(x), \psi_{jk}(x) \rangle| = O(2^{-(n+\sigma+\frac{1}{2})j}) \quad (3.31)$$

idx:psithm
eq:psisigma

證明：由於 $f \in C^n(\text{supp } \psi_{jk})$ ，且 $\frac{k}{2^j}$ 為 $\text{supp } \psi_{jk}$ 之左端點，故存在以 $\frac{k}{2^j}$ 為參考點的 n 階泰勒展開：

$$f(x) = \sum_{m=0}^{n-1} \frac{1}{m!} f^{(m)}\left(\frac{k}{2^j}\right) \left(x - \frac{k}{2^j}\right)^m + \frac{1}{n!} f^{(n)}(\xi_x) \left(x - \frac{k}{2^j}\right)^n, \quad \forall x \in \text{supp } \psi_{jk}$$

其中 ξ_x 是介於 x 與 $\frac{k}{2^j}$ 之間的某個數。由於 ψ_{jk} 正交於 $\leq n$ 階的多項式，所以

$$\langle f, \psi_{jk} \rangle = \int \frac{1}{n!} f^{(n)}(\xi_x) \left(x - \frac{k}{2^j}\right)^n \psi_{jk} dx$$

因為 $f^{(n)}\left(\frac{k}{2^j}\right) \left(x - \frac{k}{2^j}\right)^n$ 是 n 階多項式，所以

$$\langle f, \psi_{jk} \rangle = \frac{1}{n!} \int \left[f^{(n)}(\xi_x) - f^{(n)}\left(\frac{k}{2^j}\right) \right] \left(x - \frac{k}{2^j}\right)^n \psi_{jk} dx$$

由於 $f^{(n)} \in C^{0,\sigma}(\text{supp } \psi_{jk})$ ，而且 $|\xi_x - \frac{k}{2^j}| \leq |x - \frac{k}{2^j}|$ ，得到

$$|\langle f, \psi_{jk} \rangle| \leq \frac{\sqrt{2^j}}{n!} C \int |x - \frac{k}{2^j}|^{n+\sigma} |\psi(2^j x - k)| dx$$

再以 $x \mapsto \frac{x+k}{2^j}$ 之變數變換，得到

$$|\langle f, \psi_{jk} \rangle| \leq C \frac{1}{n!} \int |x|^{n+\sigma} |\psi(x)| dx 2^{-(n+\sigma+\frac{1}{2})j}$$

因為 $\psi(x)$ 為有限函蓋，故上式中之積分可積。將它們合併成一個常數，就得到了 (3.31) 中的大 O 估計。 \square

前面的定理，是 Haar 凌波係數消散定理與性質 3.3 的共同推廣。如果 $f \in C^{n+1}$ ，則其結論與 $f \in C^{n,1}$ 相同。我們還要特別申論一點：定理的假設條件中，有 ψ 正交於多項式的要求，也有 f 的平滑要求。這兩個條件必須互相配合。如果 f 本身不夠平滑，那麼 ψ 正交於再高階的多項式也沒有用。比如說，若 $f \in C^{1,\sigma}$ 而 ψ 正交於 ≤ 3 階的多項式，則凌波係數的消散速率只能達到

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

其效果亦如同 ψ 只與 ≤ 1 階的多項式正交。反過來，如果正交於 ψ 的多項式階數不夠高，那麼 f 再平滑也沒有用。比如說，若 ψ 正交於 ≤ 2 階的多項式，則即使 $f \in C^\infty$ ，其凌波係數的消散速率也只能達到

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

其效果亦如同 $f \in C^2$ 或 $f \in C^{1,1}$ 。

凌波係數消散定理是凌波函數的核心理論。幾乎所有的應用都直接或間接與它有關。我們現在提出一個立即的應用：凌波係數消散得越快，多層解析空間 V_j 的逼近能力就越高。明確地說，如果 $f \in L^2(\mathbb{R})$ 且 $f \in C^{n,\sigma}(\mathbb{R})$ ，則

$$\|f(x) - \mathcal{P}_j f(x)\| = O(2^{-(n+\sigma)j}) \quad (3.32) \quad \text{eq:Pjapp}$$

其中 \mathcal{P}_j 就是 L^2 到 V_j 空間的正交投影，也就是 V_j 中最靠近 f 的函數，或稱為 f 在 V_j 中的最佳估計 (*best approximation*)。這其實是階梯函數空間的推廣。參見 (2.18)。如果 $f(x)$ 不是有限函蓋，那麼 (3.32) 的證明有點麻煩。但是如果 $\text{supp } f = [a, b]$ ，則 (3.32) 的證明大致上與 idx:app

第一章十四節中所示範的相同。現在

$$\begin{aligned} \mathcal{P}_j f(x) &= \sum_k \langle f, \phi_{jk} \rangle \phi_{jk}(x) \\ &= \sum_k \langle f(x), \phi(x-k) \rangle \phi(x-k) + \sum_{\ell=0}^{j-1} \sum_k \langle f, \psi_{\ell k} \rangle \psi_{\ell k}(x) \end{aligned}$$

故

$$\|f - \mathcal{P}_j f\| = \left\| \sum_{\ell=j}^{\infty} \sum_k \langle f, \psi_{\ell k} \rangle \psi_{\ell k}(x) \right\|$$

根據 (3.31)，

$$|\langle f, \psi_{\ell k} \rangle| \leq C_1 \frac{1}{\sqrt{2^\ell}} 2^{-(n+\sigma)\ell}$$

故

$$\left| \sum_k \langle f, \psi_{\ell k} \rangle \psi_{\ell k}(x) \right| \leq C_1 2^{-(n+\sigma)\ell} \sum_k |\psi(2^\ell x - k)|$$

這裡是一個關鍵，請參考 (1.44) 的想法。如果 $\text{supp } f = [a, b]$ ，則對每個 ℓ 只有有限多個 k 使得 $\langle f, \psi_{\ell k} \rangle \neq 0$ ；比如說 $N_\ell \leq k \leq M_\ell$ 。則我們只需考慮

$$\sum_{k=N_\ell}^{M_\ell} |\psi(2^\ell x - k)|$$

若 $\text{supp } \psi = [0, 2p-1]$ ，一般的情況下

$$\left\| \sum_{k=N_\ell}^{M_\ell} |\psi(2^\ell x - k)| \right\| \leq C_2 (b-a)$$

在後面的習題中，我們將請讀者在一個更簡單的情況下練習上式的證明。總之，現在我們有

$$\begin{aligned} \|f - \mathcal{P}_j f\| &\leq C_1 C_2 \sqrt{b-a} \sum_{\ell=j}^{\infty} 2^{-(n+\sigma)\ell} \\ &= \left(C_1 C_2 (b-a) \frac{2^{n+\sigma}}{2^{n+\sigma}-1} \right) 2^{-(n+\sigma)j} \end{aligned}$$

合併前面的常數，則得到 (3.32) 中的大 O 估計。

逐點來看，對每一個 $x \in \mathbb{R}$ 和 $j \in \mathbb{Z}$ ，最多只有 $2p$ 個 ψ_{jk} 函蓋了 x 。令

$$I_j(x) = \{k \in \mathbb{Z} \mid x \in \text{supp } \psi_{jk}(x)\} \quad (3.33) \quad \text{eq:Ij}$$

我們很容易檢查，如果 x 不是 $\frac{x}{2^j}$ 這種形式的數，則 $I_j(x)$ 中恰有 $2p-1$ 個整數，否則 $I_j(x)$ 中有 $2p$ 個整數。所以，若 f 在 x 附近是 $C^{n,\sigma}$ 的函數，而且 ψ 正交於 $\leq n$ 階的多項式，則

$$\begin{aligned} |f(x) - \mathcal{P}_j f(x)| &= \left| \sum_{\ell=j}^{\infty} \sum_{k \in I_j(x)} \langle f, \psi_{\ell k} \rangle \psi_{\ell k}(x) \right| \\ &\leq C \sum_{\ell=j}^{\infty} 2^{-(n+\sigma+\frac{1}{2})\ell} \sqrt{2^j} \sum_{k \in I_j(x)} |\psi(2^j x - k)| \\ &\leq 2pSC \sum_{\ell=j}^{\infty} 2^{-(n+\sigma)\ell} \end{aligned}$$

其中 S 是 $\psi(x)$ 的極大值*。將常數合併在一起，我們得到

$$|f(x) - \mathcal{P}_j f(x)| = O(2^{-(n+\sigma)j}), \quad \forall x \in \mathbb{R} \quad (3.34) \quad \text{eq:Pjapt}$$

* 我們並沒有證明 $\psi(x)$ 是有界函數。但是將會看到，除了 $p=1$ 的凌波函數，也就是 Haar 函數，是不連續之外，其他 $p>1$ 的凌波函數都是連續函數。所以 $|\psi(x)|$ 的確會有極大值。

(3.32) 是平均的逼近估計，而 (3.34) 是逐點的逼近估計。也就是說，(3.32) 是在整個 $\text{supp } f$ 的區間內討論 $\|f - \mathcal{P}_j f\|$ 與 j 的關係，(3.34) 是針對每個點 $x \in \text{supp } f$ ，討論 $|f(x) - \mathcal{P}_j f(x)|$ 與 j 的關係。這兩種估計的結果基本上是一樣的，所以不論是在古典的逐點意義之下，或是 L^2 的意義之下， $\mathcal{P}_j f(x)$ 都會收斂到 $f(x)$ ，而且收斂的速率都是 $2^{-(n+\sigma)j}$ 。

根據 (3.34)，我們稱 $\mathcal{P}_j f(x)$ 對 $f(x)$ 的逼近階數 (*approximation order*) 是 $n + \sigma$ ，或者說 $\mathcal{P}_j f(x)$ 是 $f(x)$ 的 $n + \sigma$ 階逼近。在數學上，我們不關心對個別例子而言， $\mathcal{P}_j f(x)$ 和 $f(x)$ 的誤差是多少；因為這個誤差與 $f(x)$ 的個別性質 (例如平滑程度) 有關。我們要討論的是，一般而言，如果 $f(x)$ 本身有某種程度的平滑，那麼這個逼近的誤差變化率如何？在這裡的意思是，如果提高一層解析度，也就是 $j \mapsto j + 1$ ，那麼新的逼近會比舊的逼近更近多少？換句話說，新的誤差會比舊的誤差變小幾倍？這就是逼近階數告訴我們的資訊。所謂 $n + \sigma$ 階逼近，就是當我們提高一層解析度時，新的逼近比舊的逼近“好” $n + \sigma$ 倍。這句話的意義是，新誤差是舊誤差的 $2^{-(n+\sigma)}$ 倍。

在這一節的最後，我們要將 V_j 的逼近論與多項式的逼近論做一個比較。給定一個函數 $f(x)$ ，取一個實數 $h > 0$ ，以 h 為間隔取均勻分佈的節點 x_0, x_1, \dots, x_n 。意思是說 $x_i = x_0 + ih$ 。那麼，通過 $n + 1$ 個點 $(x_i, f(x_i))$ 有一條 (惟一的) n 階多項式 $\pi_n(x)$ 。顯然 $\pi_n(x)$ 與 $f(x)$ 在節點上相等： $\pi_n(x_i) = f(x_i)$ 。我們稱 $\pi_n(x)$ 為 $f(x)$ 的內插多項式 (*interpolation polynomial*)。可見 $f(x) \approx \pi_n(x)$ ，故 $\pi_n(x)$ 是 $f(x)$ 的一個多項式逼近。如果 $f(x)$ 在區間 $[x_0, x_n]$ 內是 C^{n+1} 的函數，此逼近的誤差是

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

這種形式的理論證明，出現於許多數值分析教科書上，例如 [6]。如果

我們將 $\max |f^{(n+1)}|$ 合併於常數項，則 $|f(x) - \pi_n(x)| = O(h^{n+1})$ 。若取 $h = 2^{-j}$ ，則這個結果和 $f \in C^{n,1}$ 而且 ψ 正交於 $\leq n$ 階多項式的結果相同。可見以 $\mathcal{P}_j f(x)$ 逼近 $f(x)$ 的效果，類似於以 2^{-j} 為間隔的 n 階內插多項式逼近 $f(x)$ 的效果。這就好像 V_j 空間中包含了這些內插多項式一樣。

事實上就是如此。因為 $\psi \perp V_0$ ，所以如果 ψ 正交於 $\leq n$ 階的多項式，我們可以想像，這些多項式應該在 V_0 裡面。在第十三節中我們會詳加討論這個事實，現在請先接受這個粗略的描述：

$$\int x^m \psi(x) dx = 0, \quad 0 \leq m \leq n \quad \Leftrightarrow \quad \{1, x, \dots, x^n\} \subset V_0$$

但是因為對所有的 $j > 0$ ， $V_0 \subset V_j$ ，所以這些多項式也都在 V_j 裡面。

習題 3.12 假設 $f \in L^2(\mathbb{R})$ 且 $\text{supp } f = [0, 2p - 1]$ ，其中 $p \geq 1$ 是個正整數。選擇 p 階的凌波函數 $\psi(x)$ ，亦即 $\text{supp } \psi = [0, 2p - 1]$ 。此處我們只考慮 $L^2([0, 2p - 1])$ 的長度，亦即

$$\|f\|^2 = \int_0^{2p-1} |f(x)|^2 dx$$

證明 $\|\psi(2^j x)\|^2 = \frac{1}{2^j}$ ，而且

$$\left\| \sum_k |\psi(2^j x - k(2p - 1))| \right\| = 1, \quad \forall j \geq 0$$

再證明

$$\left\| \sum_k |\psi(2^j x - k)| \right\| \leq 2p - 1, \quad \forall j \geq 0$$

3.6 再談自格係數

在前一節中，我們看到，如果 $\psi(x)$ 能夠正交於 $\leq n$ 階的多項式，則如果 $f(x)$ 足夠平滑，就有消散較快的波係數 $|\langle f, \psi_{jk} \rangle|$ ，也因此就可以提高多層解析空間 V_j 的逼近能力。由 (3.28) 已經知道 $\int \psi dx = 0$ ，所以 $\psi(x)$ 至少正交於 0 階 (常數) 多項式。在這一節，我們要討論，如何使得 ψ 正交於更高階的多項式呢？

回顧 $\psi(x) = \sum d_k \phi(2x - k)$ 及 (3.11)， $\sum d_k = \sum (-1)^k c_k = 0$ 。我們已經知道，就是因為 $\sum d_k = 0$ 的這個性質，使得 $\int \psi dx = 0$ 。在第二節的闡述中，我們知道只要 (3.6) 和 (3.10) 這 $p + 1$ 個條件，就可以推得 $\sum d_k = 0$ 。而針對 $2p$ 個自格係數 c_0, \dots, c_{2p-1} ，我們還有 $p - 1$ 個自由度。現在，我們要將這些自由度全部用來提高 V_j 的逼近能力，亦即加快 $|\langle f, \psi_{jk} \rangle|$ 的消散速率。根據波係數消散定理，我們需要 ψ 正交於越高階的多項式越好。這 $p - 1$ 個自由度可以用來製造

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

其中 $m = 0$ 的部分已經成立了。

首先，讓我們看看 $\int x\psi dx = 0$ 和 c_k 有什麼關係？

$$\begin{aligned} 0 &= \int x\psi(x) dx \\ &= \sum_k d_k \int x\phi(2x - k) dx \\ &= \frac{1}{4} \sum_k d_k \int (x + k)\phi(x) dx \end{aligned}$$

定義 $\phi(x)$ 的 m 次動量 (m -th moment) 為

idx:moment

$$M_m = \int x^m \phi(x) dx \tag{3.35}$$

eq:Mmoment

則 M_m 都是與 p 有關的常數，其中 $M_0 = 1$ 。所以我們得到

$$M_0 \left(\sum_k k d_k \right) + M_1 \left(\sum_k d_k \right) = 0$$

由於 $\sum_k d_k = 0$ ，故得

$$\sum_k k d_k = 0$$

根據 (3.17)，上式可以寫成 c_k 的條件：

$$\sum_k (-1)^k k c_{2p-1-k} = 0$$

經過足標變換 $k \mapsto 2p-1-k$ ，我們得到一個比較簡潔的敘述：

$$0 = \sum_k (-1)^{2p-1-k} (2p-1-k) c_k = -(2p-1) \sum_k (-1)^k c_k + \sum_k (-1)^k k c_k$$

根據 (3.11) 得到

$$\sum_k (-1)^k k c_k = 0$$

現在，假設當 $1 \leq \ell \leq m-1$ 時， $\int x^\ell \psi dx = 0$ 可以推得 $\sum k^\ell d_k = 0$ ，那麼

$$\int x^m \psi(x) dx = \frac{1}{2^{m+1}} \left(\sum_{\ell=0}^{m-1} \binom{m}{\ell} M_{m-\ell} \sum_k k^\ell d_k + \sum_k k^m d_k M_0 \right)$$

由歸納法假設，得知若 $\int x^m \psi dx = 0$ ，則

$$\sum_k k^m d_k = 0$$

同樣地，根據 (3.17) 和足標變換的處理，可以得到

$$\sum_k (-1)^k k^m c_k = 0$$

因此，我們可以利用剩下的 $p - 1$ 個自由度，要求

$$\int x^m \psi(x) dx = 0, \quad m = 1, 2, \dots, p - 1$$

這些要求等價於有關 c_k 的 $p - 1$ 個條件：

$$\sum_k (-1)^k k^m c_k = 0, \quad m = 1, 2, \dots, p - 1$$

與 (3.11) 合併，我們不妨說

$$\sum_k (-1)^k k^m c_k = 0, \quad \text{for } 0 \leq m \leq p - 1 \quad (3.36)$$

eq: ckapp

上式中，定義 $0^0 = 1$ 。以 $p = 2$ 為例，即

$$\begin{aligned} c_0 - c_1 + c_2 - c_3 &= 0 & (\text{for } m = 0) \\ -c_1 + 2c_2 - 3c_3 &= 0 & (\text{for } m = 1) \end{aligned} \quad (3.37)$$

eq: ckappexr

由 (3.6)、(3.10)、(3.36) 所組成的 $2p$ 個聯立方程式有一組非零解。其實它們至少有兩組解。如果 $c_0, c_1, \dots, c_{2p-1}$ 是一組解，那麼將它們的順序反過來 $c_{2p-1}, \dots, c_1, c_0$ 也是一組解。這兩組解所對應的自格函數 $\phi(x)$ 只是互為鏡射，在本質上並無不同。所以此後我們並不考慮順序的翻轉。當 $p = 1$ 時只有一組解

$$1, \quad 1;$$

當 $p = 2$ 時，只有一組解：

$$\frac{1 + \sqrt{3}}{4}, \frac{3 + \sqrt{3}}{4}, \frac{3 - \sqrt{3}}{4}, \frac{1 - \sqrt{3}}{4};$$

當 $p = 3$ 時，也只有一組解：

$$\frac{1 + \sqrt{10} + \theta}{16}, \frac{5 + \sqrt{10} + 3\theta}{16}, \frac{5 - \sqrt{10} + \theta}{8},$$

$$\frac{5 - \sqrt{10} - \theta}{8}, \frac{5 + \sqrt{10} - 3\theta}{16}, \frac{1 + \sqrt{10} - \theta}{16}$$

其中 $\theta = \sqrt{5 + 2\sqrt{10}}$ 。當 $p \geq 4$ 時，就可能出現第二組解。一般而言，會有

$$2^{\lfloor \frac{p}{2} \rfloor - 1}$$

組不同的解。這些不同的 c_k 所造成的自格函數或凌波函數，其逼近能力、計算複雜度與函數平滑度均相同，只是 $\phi(x)$ 和 $\psi(x)$ 的函數圖形稍有不同。詳細的討論必須使用傅立葉轉換為工具。有意深入的讀者，請參考 Daubechies 的凌波十講 [3] 或是嚴健彰先生與作者的一篇短文 [12]。

當 $p \geq 4$ 時我們用牛頓法求解。陳永康使用四倍精度的 FORTRAN 程式，計算 $2 \leq p \leq 14$ 的自格係數 c_k 。我們並沒有解得所有可能的解，這些只是其中一組解。其他解在理論上所造成的性質完全相同，所以沒有太大的必要去計算所有的解。這些係數的數值以雙精度儲存在 Matlab 的資料檔中：

daub_coef.mat

此外，我們還附了一個小程式 getdaubc.m，幫您從這個資料檔中讀取正確的係數。例如，您只要說

`c=getdaubc(3);`

就會得到一個向量 c ，其長度為 6，其數值就是 $p = 3$ 情況下的 $2p$ 個自格係數 c_0, \dots, c_5 。

習題 3.13 仿照 (3.37) 寫出當 $p = 3$ 時的三條等式。

習題 3.14 證明 $\sum k^m d_k = 0, 0 \leq m \leq N$ ，和 $\sum (-1)^k k^m c_k = 0, 0 \leq m \leq N$ ，是等價的。也就是說，若前者成立，則後者也成立，反之亦然。

習題 3.15 由 (3.36)，證明對任意的整數 ℓ 都符合

$$\sum_k (-1)^k (k - \ell)^m c_k = 0, \text{ for } 0 \leq m \leq p - 1$$

3.7 凌波轉換

讓我們回顧前面五節所得到的結果。對任一正整數 p ，存在一個正交自格函數 $\phi(x)$ ，使得 $\int \phi dx = 1$ 而且 $\text{supp } \phi = [0, 2p - 1]$ 。 $\{\phi_{jk}(x) \mid k \in \mathbb{Z}\}$ 形成 V_j 空間的正則基底，而且

$$\dots \subset V_{-1} \subset V_0 \subset V_1 \subset \dots$$

$$\bigcap V_j = 0, \quad \bigcup V_j \text{ 稠密於 } L^2(\mathbb{R})$$

亦即 $\{V_j\}$ 形成 $L^2(\mathbb{R})$ 的一組多層解析空間，而我們可以將 $\phi(x)$ 視為 $\{V_j\}$ 的生成元。

從 $\phi(2x - k)$ 我們可以製造一個凌波函數 $\psi(x)$ ，使得 $\text{supp } \psi = [0, 2p - 1]$ 而且當 $0 \leq m \leq p - 1$ 時 $\int x^m \psi dx = 0$ 。 $\{\psi_{jk}(x) \mid k \in \mathbb{Z}\}$ 形成 W_j 空間的正則基底，而且

$$V_j = W_{j-1} \oplus V_{j-1}, \quad L^2(\mathbb{R}) = \bigoplus_{j=-\infty}^{\infty} W_j$$

所以

$$\{\psi_{jk}(x) \mid j, k \in \mathbb{Z}\}$$

就是 $L^2(\mathbb{R})$ 的一組正則基底。這裡所說的基底，可以允許無窮和的情形：

$$f(x) = \sum_j \sum_k \langle f, \psi_{jk} \rangle \psi_{jk}(x)$$

其中的 $=$ 是 L^2 意義之下的收斂：

$$\lim_{J \rightarrow \infty} \lim_{K \rightarrow \infty} \left\| f(x) - \sum_{j=-J}^J \sum_{k=-K}^K \langle f, \psi_{jk} \rangle \psi_{jk}(x) \right\| = 0$$

正則基底是一種無條件基底 (*unconditional basis*)。

idx:uncond-ba

$f(x) \in L^2(\mathbb{R})$ 在每一個 V_j 中有唯一的正交投影 $\mathcal{P}_j f(x)$ 。 $\mathcal{P}_j f$ 是 f 在第 j 層解析度的最佳逼近。若 $f \in C^{m, \sigma}$ ，其中 $0 \leq m \leq p-1$ 、 $0 < \sigma \leq 1$ ，則其逼近階數為 $m + \sigma$ 。相對而言， $\mathcal{P}_j f(x)$ 是高解析度訊號，而 $\mathcal{P}_{j-1} f(x)$ 是低解析度訊號。在高低解析度之間，有所謂的細節部分。這些細節部分落在 W_{j-1} 空間內。總括來說，

V_j 是層層相套的空間，它們全部由一個自格函數 $\phi(x)$ 的脹縮平移所造成； W_j 是兩兩正交的空間，它是 V_j 中層與層之間的細節部分，它們全部由一個凌波函數 $\psi(x)$ 的脹縮平移所造成。

所謂凌波轉換 (*wavelets transform*)，有連續與離散兩種。我們只呈現連續凌波轉換的計算形式，不再多作描述。因為本書只討論離散形態的凌波轉換。但是我們要先提醒讀者，連續形態的凌波轉換並非沒有應用。在 *pattern recognition*、*edge detection*、*image deformation* 方面，都可見到連續凌波轉換的應用。所謂連續凌波轉換 (*continuous*

wavelets transform)乃是將一個 L^2 中的變數函數 $f(x)$ 轉換成雙變數的函數 idx:cwt

$$f^w(a, b) = \frac{1}{\sqrt{|a|}} \int f(x) \psi\left(\frac{x-b}{a}\right) dx, \quad a, b \in \mathbb{R} \quad (3.38) \quad \text{eq:cwt}$$

而其逆轉換即為還原運算：

$$f(x) = \frac{1}{C_\psi} \iint f^w(a, b) \psi\left(\frac{x-b}{a}\right) \frac{1}{a^2} da db \quad (3.39) \quad \text{eq:cwtinv}$$

其中

$$C_\psi = \int \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega \quad (3.40) \quad \text{eq:Cpsi}$$

使 (3.39) 成立的關鍵條件是 $C_\psi < \infty$ 。若 $C_\psi < \infty$ 則必有 $\hat{\psi}(0) = 0$ ，否則 (3.40) 中的被積分函數在 $\omega = 0$ 附近就差不多是 $\frac{1}{|\omega|}$ ，其積分值不存在。而

$$\hat{\psi}(0) = \int \psi(x) e^{-i0x} dx = \int \psi(x) dx$$

所以也就是 $\int \psi dx = 0$ 。我們又可以看到 $\psi(x)$ 的圖形上下起伏如波的特性。

所謂離散波轉換 (*DWT: Discrete Wavelet Transform*)，是將一個有限解析度的函數 $v(x) \in V_j$ ，在 V_j 與 $V_{j-1} \oplus W_{j-1}$ 之間做基底轉換。我們稱

$$\text{從 } \{\phi(2^j x - k) \mid k \in \mathbb{Z}\} \text{ 轉換到 } \{\phi(2^{j-1} x - k), \psi(2^{j-1} x - k) \mid k \in \mathbb{Z}\}$$

的轉換過程為分解 (*decomposition* 或 *analysis*)，而

$$\text{從 } \{\phi(2^{j-1} x - k), \psi(2^{j-1} x - k) \mid k \in \mathbb{Z}\} \text{ 轉換到 } \{\phi(2^j x - k) \mid k \in \mathbb{Z}\}$$

的轉換過程為合成或還原 (*synthesis* 或 *reconstruction*)。不論如何，都是在處理以下等式：

$$\sum_k v_k \phi(2^j x - k) = \sum_k u_k \phi(2^{j-1} x - k) + \sum_k w_k \psi(2^{j-1} x - k) \quad (3.41)$$

eq:wtbasis

根據我們在第一章十六節所獲得的經驗，等式 (3.4) 及 (3.19) (其實也就是 (1.53) 所列出的關係式) 和它們之中的係數 c_k 與 d_k 將是分解步驟的鑰匙。再回顧 (1.60)，係數 p_k 和 q_k 將是合成步驟的鑰匙。對正交凌波函數而言，根據 (3.17) 和 (3.25)，我們知道 d_k 、 p_k 和 q_k 全部是由 c_k 決定的。但一般而言它們可以是不同的係數。

經過變數變換，我們可以只考慮 V_1 與 $V_0 \oplus W_0$ 之間的凌波離散凌波轉換。在 (3.41) 中令 $j = 1$ ，也就是我們要處理

$$\sum_k v_k \phi(2x - k) = \sum_k u_k \phi(x - k) + \sum_k w_k \psi(x - k) \quad (3.42)$$

eq:basisone

先看分解步驟。假設已知 v_k ，要求 u_k 與 w_k 。將 (3.42) 之兩端同與 $\phi(x - n)$ 作內積，則由於正交性，右端的結果只剩 u_n 。故

$$\begin{aligned} u_n &= \sum_k v_k \langle \phi(2x - k), \phi(x - n) \rangle \\ &= \sum_k v_k \sum_\ell c_\ell \langle \phi(2x - k), \phi(2x - (2n + \ell)) \rangle \\ &= \sum_k \frac{1}{2} c_{k-2n} v_k = \frac{1}{2} \sum_k c_k v_{k+2n} \end{aligned} \quad (3.43)$$

eq:un

同樣地，如果將 (3.42) 之兩端同與 $\psi(x - n)$ 作內積，則得到

$$w_n = \sum_k v_k \langle \phi(2x - k), \psi(x - n) \rangle = \frac{1}{2} \sum_k d_k v_{k+2n} \quad (3.44)$$

eq:wn

(3.43) 與 (3.44) 就是分解步驟的演算法。因為只有有限多個 c_k 是非零的，所以上式中的級數都是有限項的和。

再看合成步驟。假設在 (3.42) 中已知 u_k 和 w_k ，要求 v_n 。將 (3.42) 之兩端同與 $\phi(2x - n)$ 作內積，則由於正交性，左端的結果只剩 $\frac{1}{2}v_n$ 。故

$$\begin{aligned} \frac{1}{2}v_n &= \sum_k u_k \langle \phi(x - k), \phi(2x - n) \rangle \\ &\quad + \sum_k w_k \langle \psi(x - k), \phi(2x - n) \rangle \end{aligned} \quad (3.45)$$

eq:vnuk

先作右端的第一項，若 $n = 2\ell$ 為一偶數，根據 (1.60) 推導

$$\begin{aligned} \sum_k u_k \langle \phi(x - k), \phi(2x - n) \rangle &= \sum_k u_k \langle \phi(x - k), \phi(2(x - \ell)) \rangle \\ &= \sum_k u_k \sum_m \frac{1}{2} p_{-2m} \langle \phi(x - k), \phi(x - \ell - m) \rangle \\ &= \frac{1}{2} \sum_k u_k p_{-2(k-\ell)} = \frac{1}{2} \sum_k u_k p_{n-2k} \end{aligned}$$

若 $n = 2\ell + 1$ 為一奇數，亦可根據 (1.60) 推導

$$\begin{aligned} \sum_k u_k \langle \phi(x - k), \phi(2x - n) \rangle &= \sum_k u_k \langle \phi(x - k), \phi(2(x - \ell) - 1) \rangle \\ &= \sum_k u_k \sum_m \frac{1}{2} p_{1-2m} \langle \phi(x - k), \phi(x - \ell - m) \rangle \\ &= \frac{1}{2} \sum_k u_k p_{1-2(k-\ell)} = \frac{1}{2} \sum_k u_k p_{n-2k} \end{aligned}$$

以上我們用到了 $\phi \perp \psi$ 的性質。同理可對 (3.45) 右端的第二項做同樣

的處理，不論 n 是奇數還是偶數都得到

$$\sum_k w_k \langle \psi(x-k), \phi(2x-n) \rangle = \frac{1}{2} \sum_k w_k q_{n-2k}$$

合併一起，就得到

$$v_n = \sum_k u_k p_{n-2k} + w_k q_{n-2k} \quad (3.46)$$

eq:vn

或是

$$\begin{cases} v_{2n} = \sum_k p_{2k} u_{n-k} + q_{2k} w_{n-k} \\ v_{2n+1} = \sum_k p_{2k+1} u_{n-k} + q_{2k+1} w_{n-k} \end{cases} \quad (3.47)$$

eq:vntwo

由 (3.43)、(3.44) 及 (3.47) 我們已經明確知道離散凌波轉換的算法。亦請參考圖 6。有關離散凌波轉換之矩陣表達、計算法及應用，留待第四章再討論。在本章的後面幾節中，我們繼續深入對凌波理論的探討。

習題 3.16 在合成步驟中，當我們面臨

$$\sum_k u_k \langle \phi(x-k), \phi(2x-n) \rangle$$

的時候，課文中選擇將 $\phi(2x-n)$ 根據 (1.60) 作變換。其實也可以選擇將 $\phi(x-k)$ 根據 (3.4) 作變換。這樣做的話，就不需要 p_k 與 q_k 係數。在課文中如此做的目的，只是為了介紹另一種看法。現在，請讀者嘗試將 (3.45) 中右端的 $\phi(x-k)$ 與 $\psi(x-k)$ 代換成 $\phi(2x-k)$ 的基底，應該可以求得與 (3.46) 同樣的結果。

3.8 傅立葉分析與頻譜

在訊號處理中，傅立葉分析是一個頻譜分析的工具。如果 $f(x)$ 是一個 2π 週期函數，或者 $f(x)$ 是一個在 $[-\pi, \pi]$ 中有限函蓋而後被週期化的函數，則分析的工具是離散形式的傅立葉級數 $\sum c_k e^{ik\omega}$ 。如果 $f(x) \in L^2$ ，則分析的工具是積分形式的傅立葉轉換 $\hat{f}(\omega)$ 。不論如何，我們稱 $f(x)$ 是一個訊號在物理域 (physical domain) 或時間域 (time domain) 的表現，可以想像它是空氣壓力隨時間的變化；我們稱 c_k 或 $\hat{f}(\omega)$ 是一個訊號在頻率域 (frequency domain) 的表現，可以想像它是在頻率 k 或 ω 處的振幅。

idx:domains

回顧傅立葉級數 (2.24)，當 k 小的時候， $\cos kx$ 和 $\sin kx$ 是低頻的波動，所以我們稱 a_k 和 b_k 為低頻係數；當 k 大的時候， $\cos kx$ 和 $\sin kx$ 是高頻的波動，所以我們稱 a_k 和 b_k 為高頻係數。從 (2.32) 的範例 (參見圖 18)，觀察 a_1 到 a_{16} 的數值，我們發現低頻係數比較大，而高頻係數比較小。其實這是個一般性的現象：任何 L^2 函數或有限能量訊號，其低頻部分的振幅比較大 (所以其資訊量比較多)，而高頻部分的振幅則漸漸衰減成零 (所以其資訊量漸漸變少)。而且，所謂的雜訊通常出現在高頻部分。

對於這個現象，我們可以從能量的角度做直覺的理解：造成高頻所需要的能量比較大，但是總共只有有限多的能量，當然能量會集中在低頻的部分。也可以從數學的角度理解：若 $f \in L_p^2$ ，則根據 (2.30) 下列無窮級數必須收斂。

$$\sum_{k=1}^{\infty} |a_k|^2 + |b_k|^2$$

所以當 $k \rightarrow \infty$ 時 $|a_k|^2 + |b_k|^2 \rightarrow 0$ 。也就是高頻係數要趨近於零。還可以從圖形上觀察：假想一個 2π 週期函數 $f(x)$ ，其圖形如圖 21 之上

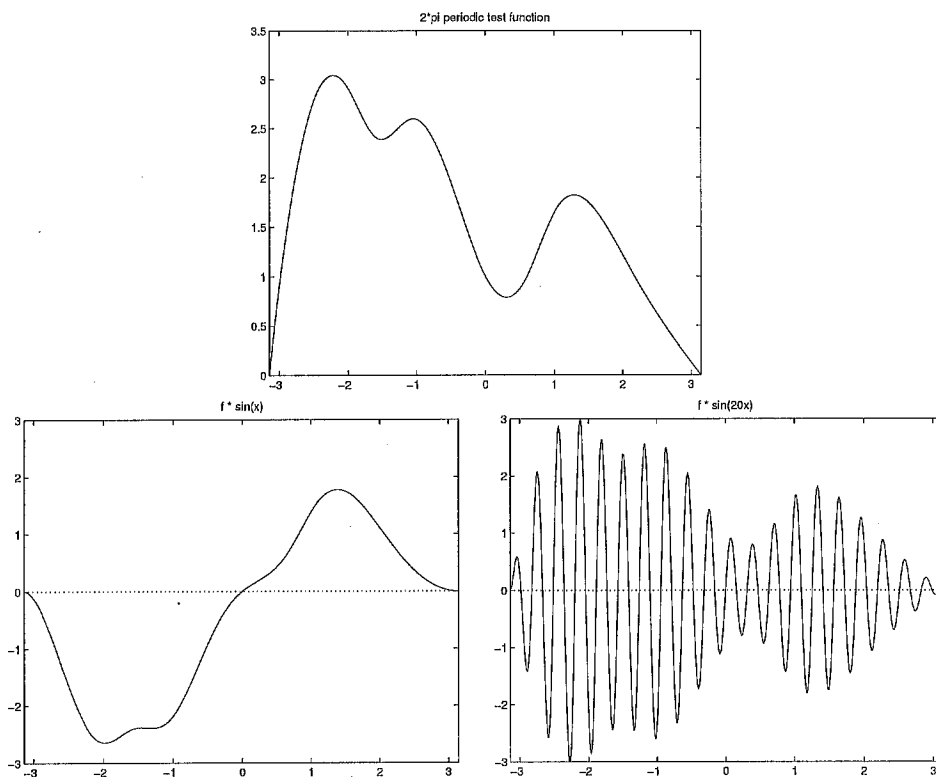


fig:Ftest 圖 21 高低頻係數測試函數

圖所示。這個 $f(x)$ 在 $[-\pi, \pi]$ 內是 C^2 函數*。

而圖 21 之左下與右下圖分別是 $f(x) \sin x$ 和 $f(x) \sin 20x$ 的圖形。

$$\int_{-\pi}^{\pi} f(x) \sin x \, dx$$

是將 $f(x) \sin x$ 在 x 軸上方的面積減去 x 軸下方的面積。

$$\int_{-\pi}^{\pi} f(x) \sin 20x \, dx$$

* 但是，以一個週期函數而言， $f(x)$ 並非 C^2 函數，它在 $n\pi$ 處只有連續而已，並不可微。

則是將 $f(x) \sin 20x$ 在 x 軸上方的面積減去 x 軸下方的面積。參照它們的圖形，可見 $f(x) \sin x$ 在 x 軸上下部分的面積差距較大，其積分值大約是 -2.4513 。而 $f(x) \sin 20x$ 在 x 軸上下部份的面積幾乎互相消去，其積分值大約是 0.0014 。所以，一般來說，當 k 越大，則 $\sin kx$ 和 $\cos kx$ 的波峰與波谷就越靠近，因此 $f(x) \sin kx$ 在 x 軸上下部份的面積就幾乎會互相消去，所以 $|a_k|$ 與 $|b_k|$ 應該會越小。

以上是針對 2π 週期函數在整數頻率之振幅所做的闡述。對一般的 L^2 函數而言， $\hat{f}(\omega)$ 亦有同樣的性質。當 $|\omega|$ 小的時候，我們稱 $\hat{f}(\omega)$ 是低頻係數；當 $|\omega|$ 大的時候，我們稱 $\hat{f}(\omega)$ 是高頻係數。根據 (2.41)，若 $f(x) \in L^2$ 則 $\hat{f}(\omega) \in L^2$ 故

$$\int |\hat{f}(\omega)|^2 d\omega < \infty$$

所以當 $|\omega| \rightarrow \infty$ 時 $|\hat{f}(\omega)| \rightarrow 0$ 。同樣地，低頻係數 (的絕對值) 通常比較大，高頻係數 (的絕對值) 必須越來越小，最後衰退到零。

所謂的高頻與低頻觀念，也可以類比到二維的訊號—影像。以一張人像近照為例，能夠從五官髮膚外形認出那個人是誰，屬於低頻部分。要數數他眼角有幾絲魚尾紋，頭上有幾莖白髮，兩頰的雀斑或腮旁的鬍渣，屬於高頻部分。通常需要比較高的能量 (比較貴的照相機，比較貴的沖洗程序)，才能顯現這些高頻。但即使沒有這些高頻，也並不影響我們辨識這張照片裡面的人像。

一個 L^2 中的 2π 週期函數 $f(x)$ 若寫成 (2.24) 形式的傅立葉級數，則

$$\sqrt{2}|c_0|, 2|c_1|, 2|c_2|, 2|c_3|, \dots$$

分別是 $f(x)$ 在各頻率波段的能量，這些數值 (或它們的平方) 稱為 $f(x)$ 的能量頻譜 (*energy spectrum*) 或簡稱頻譜。 $f(x)$ 的頻譜 就是它在頻率 idx:spectru

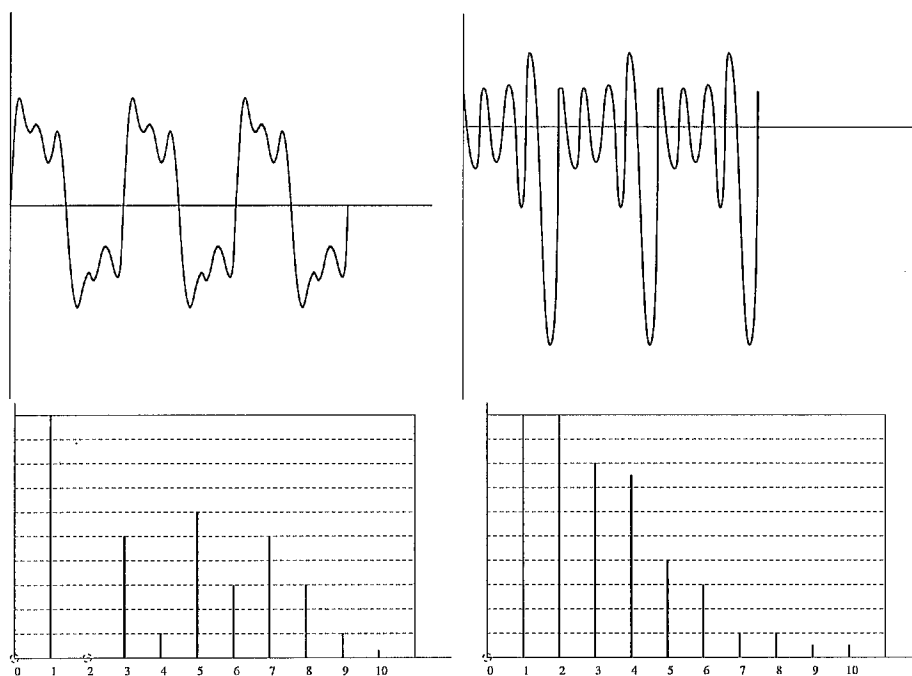


fig:Spectest 圖 22 豎笛與小喇叭的頻譜分析

域內的表象，訊號處理的目的之一，就是希望能在頻率域內探測到在時間域內不容易探測到的訊息。

例如在圖 22 中我們示範如何用頻譜來分辨豎笛 (clarinet) 和小喇叭 (trumpet) 的音色。在上方的左圖和右圖分別是豎笛和小喇叭吹奏同一個音階的所產生的聲波。那是空氣壓力對時間的變化情形。而對應在下方的圖形則是它們的頻譜。雖然用人的眼睛可以明顯感受兩種聲波的不同，但是卻很難明確描述它們到底如何的不同。難以描述也就難以量化，難以量化也就難以使其自動化。反觀兩種聲波的頻譜，則第三和第五個頻率的能量顯然不同。這就可以明確的量化了。讀者應可想像，許多的訊號處理，例如從語音自動辨識字詞，從心律波、腦波或胃的蠕動協助診斷器官的病變，從地震波探測地底的礦藏或地質

結構等等，都可以利用頻譜來當做一種工具。

從時間域到頻率域的變換，以數學眼光看來，無非就是基底的轉換。訊號處理的主要工具之一，就是企圖找到某種基底，使得訊號在那種特殊基底的表達之下，特別能夠突顯某種現象，以便人類從中探測到某種訊息。波轉換提供了一種新的基底變換之可能。在下面的兩節中，我們探討波函數在頻率域上的性質。

習題 3.17 證明當 $k \geq 1$ 時 $2|c_k| = \sqrt{a_k^2 + b_k^2}$ 而且 $|c_k| = |c_{-k}|$ 。

3.9 頻譜與尺度

在第二章十三節，我們知道

$$\mathcal{F}[f(x - k)](\omega) = e^{-ik\omega} \hat{f}(\omega)$$

在工程術語中， $\hat{f}(\omega)$ 與 $e^{-ik\omega} \hat{f}(\omega)$ 除了相位差之外兩者相等。所以，訊號在時間域的平移僅造成在頻率域的相位差，也就是說兩者的頻率變化是完全一樣的，只是發生的時間順延或提前而已。由於 $|e^{-ik\omega}| = 1$ ，故 $|e^{-ik\omega} \hat{f}(\omega)| = |\hat{f}(\omega)|$ 。簡單地說，函數 $f(x)$ 在時間域的平移在本質上並不會影響它在頻率域的性質。 idx:phase

在直覺上，函數 $f(x)$ 的脹縮就應該會影響它的頻率反應。比如說 $f(2x)$ 將函數縮窄了一倍，所以直覺上應該將頻率提高了一倍。只要以 $f(x) = \sin x$ 為例，就看得很清楚了。經過數學計算，看得更清楚：

$$\mathcal{F}[f(2^j x)](\omega) = \int f(2^j x) e^{-i\omega x} dx = \frac{1}{2^j} \hat{f}\left(\frac{\omega}{2^j}\right)$$

可見 $f(x)$ 的脹縮除了造成頻率的升降之外，還須乘上一個倍數。我們且不顧那個倍數，定性而言，當 j 越大，也就是將 $f(x)$ 縮得越窄，則將它的頻率升得越高。

接下來，讓我們探討自格函數 $\phi(x)$ 在時間域的不同尺度，與其在頻率域的對應關係。一言以蔽之，由於

$$\text{supp } \phi(2^j x - k) = \left[\frac{k}{2^j}, \frac{k + 2^j p - 1}{2^j} \right]$$

所以 $\phi(2^j x - k)$ 的寬度是 $\phi(x)$ 的 $\frac{1}{2^j}$ 倍，而且 $\phi(2^j x - k)$ 是以 $\frac{1}{2^j}$ 為跨距做左右的平移。因為每單位長內大約有 $2^j(2p - 1)$ 個 $\phi(2^j x - k)$ ，所以，相對 $\phi(x - k)$ 而言，當 $j > 0$ ， $\phi(2^j x - k)$ 是一群又窄又密的函數族。當 j 小的時候，我們稱 $\phi(2^j x - k)$ 是大尺度的函數；而當 j 大的時候，我們稱 $\phi(2^j x - k)$ 是小尺度的函數。同時， $\phi(2^j x - k)$ 會以它的尺度來自動調適其密度。當 ϕ 的在大尺度之下，它的密度自然較低；當 ϕ 的在小尺度之下，它的密度自然較高。參見圖 23，兩圖都是 $p = 3$ 的 $\phi(2^j x - k)$ ，上方是 $j = 0$ ，下方是 $j = 2$ 。

換一個角度來闡述。我們可以將 $\langle f(x), \phi(2^j x - k) \rangle$ 看成“用 $\phi(2^j x - k)$ 去探測 $f(x)$ ”的數學模型。這時候的 $\phi(2^j x - k)$ 就像顯微鏡的接物鏡，或是望遠鏡，當 j 越大的時候，觀看的尺度越小，放大的倍率也就越高。但因為高倍率的觀看範圍很窄，所以我們需要在單位長度內觀看許多鏡頭，才能看完全部的樣本。所以小尺度（高倍率）的探測函數 ϕ 需要小的平移跨距（高的探測密度），才能探測整個函數 f 。

在頻率域上，根據前面一般性的探討，已經知道 $\phi(x - k)$ 的傅立葉轉換是

$$\mathcal{F}[\phi(x - k)](\omega) = e^{-ik\omega} \hat{\phi}(\omega) \quad (3.48) \quad \text{eq:ftrans}$$

$\phi(2x)$ 的傅立葉轉換是

$$\mathcal{F}[\phi(2x)](\omega) = \frac{1}{2} \hat{\phi}\left(\frac{\omega}{2}\right) \quad (3.49) \quad \text{eq:fdilate}$$

合併得到

$$\mathcal{F}[\phi(2^j x - k)](\omega) = \frac{1}{2^j} e^{-i\frac{k}{2^j}\omega} \hat{\phi}\left(\frac{\omega}{2^j}\right)$$

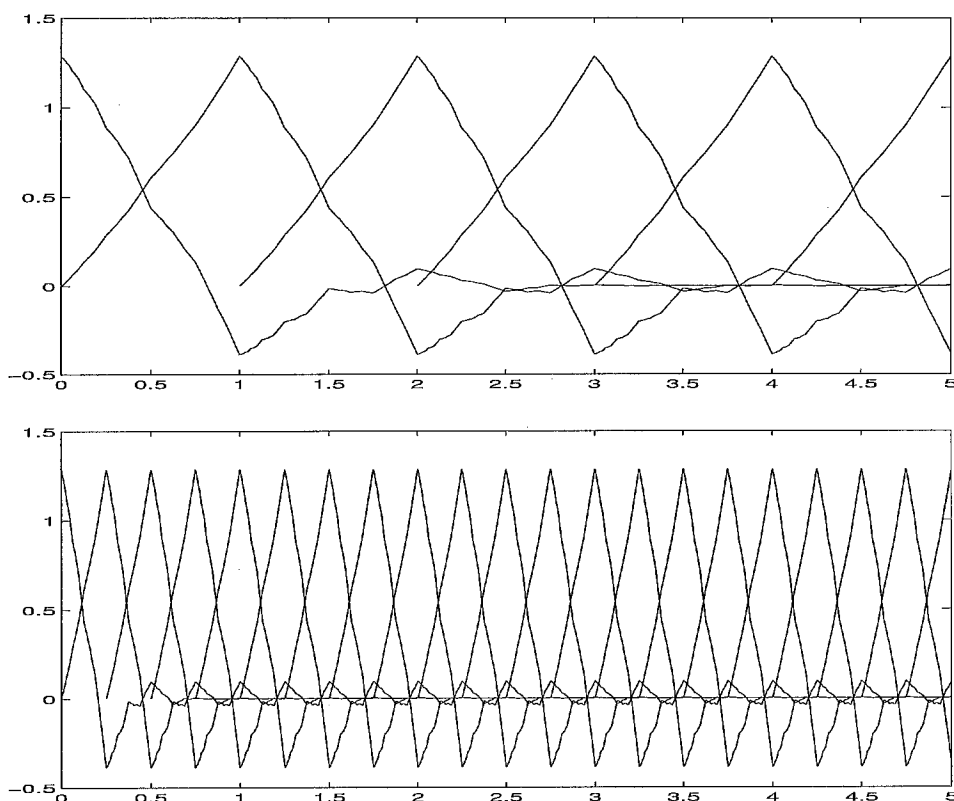


fig:phiscale 圖 23 兩種尺度的 $\phi(2^j x - k)$

我們稱 $\text{supp } \hat{\phi}$ 是 ϕ 的頻率反應範圍。不顧倍數 $\frac{1}{2^j}$ 和相位差 $e^{-i\frac{k}{2^j}\omega}$ ，idx:band
 定性來說，若 $\phi(x)$ 的頻率反應範圍是 $[\omega_a, \omega_b]$ ，則 $\phi(2^j x - k)$ 的頻率
 反應範圍是 $[2^j \omega_a, 2^j \omega_b]^*$ 。值得注意的是，當 $j > 0$ ，不但頻率反應的
 範圍提高了，同時也拓寬了（從 $\omega_b - \omega_a$ 放寬到 $2^j(\omega_b - \omega_a)$ ）。

根據 Parseval 等式 (2.43)， $2\pi \langle f, \phi \rangle = \langle \hat{f}, \hat{\phi} \rangle$ 。不顧倍數 2π ，定性 idx:Parseval

* 因為 ϕ 是有限函蓋函數，所以 $\hat{\phi}$ 必為無限函蓋。因此 $\phi(x)$ 的頻率
 反應範圍必定不是有限區間。但我們此時只是在做粗略的定性描述。

來說，以 $\phi(2^j x - k)$ 探測 $f(x)$ 的結果，同時也就是以 $\mathcal{F}[\phi(2^j x - k)](\omega)$ 探測 $\hat{f}(\omega)$ 的結果。這個一箭雙鵰的好結果，因為 Parseval 等式而得來全不費工夫。因此我們知道，當 j 比較大的時候， $\langle f(x), \phi(2^j x - k) \rangle$ 同時也在比較高的頻率範圍內做了探測。綜合來說，

當 j 較大時， $\phi(2^j x - k)$ 的探測密度較高，它在較小的尺度下探測較高亦較寬的頻率範圍。

由於 $\int \phi dx = 1$ ，故 $\hat{\phi}(0) = 1$ 。而 $\hat{\phi}(\omega)$ 是連續函數（因為 ϕ 是有限函蓋，所以 $\hat{\phi}$ 其實是無限次可微；請回顧第二章十三節的闡述）。故 $\hat{\phi}(\omega)$ 在 $\omega = 0$ 的附近差不多是 1。當 j 越大時， $\mathcal{F}[\phi(2^j x - k)](\omega)$ 在 $\omega = 0$ 的附近就有越大的範圍差不多是 1（因為圖形膨脹了）。又因為 $\|\hat{\phi}(\omega)\| = \sqrt{2\pi} \approx 2.5$ ，故當 $|\omega|$ 遠離 0 的時候 $|\hat{\phi}(\omega)|$ 應該很小。所以我們得到一個大致的印象：

$$\hat{f}(\omega) \mathcal{F}[\phi(2^j x - k)](\omega) \approx \begin{cases} \hat{f}(\omega) & \text{當 } |\omega| \text{ 靠近 } 0 \\ 0 & \text{當 } |\omega| \text{ 遠離 } 0 \end{cases}$$

意思是說， $\hat{f}(\omega) \mathcal{F}[\phi(2^j x - k)](\omega)$ 相乘的結果，使得 $\hat{f}(\omega)$ 的低頻部分 ($|\omega|$ 靠近 0) 被留下來，而高頻部分 ($|\omega|$ 遠離 0) 被刪成 0。就好像 $f(x)$ 的低頻部分通過了一個濾器，而高頻部分就被濾掉了。在這個意義之下，我們稱 $\phi(x)$ 是一個低頻濾波 (*low-pass filter*) 函數。而 $\langle f(x), \phi(2^j x - k) \rangle$ 這個探測結果，也就差不多是在 $\text{supp } \phi(2^j x - k)$ 這個小尺度之內對 $\hat{f}(\omega)$ 的低頻做探測。

idx:lowpassfu

當 j 越大時，前面所謂之低頻的頻率就越高而且範圍越寬。總之，相對而言， $\phi(2^j x - k)$ 的頻率反應範圍比 $\phi(2^{j+1} x - k)$ 來得低。基於這個道理，我們稱 $\mathcal{P}_j f \in V_j$ 是 $\mathcal{P}_{j+1} f \in V_{j+1}$ 的低頻部分 (*low frequency*)

part)。從 V_{j+1} 分解到 V_j 所使用的那組係數 $\frac{1}{2}c_k$ (參閱 (3.43))，稱為 idx:lowfreq
低頻濾波係數 (*low-pass filter coefficients*)。

現在我們探討波函數 $\psi(x)$ 的尺度與頻率之關係。一方面， ψ 與 ϕ 的性質相同。在頻率域上，

$$\mathcal{F}[\psi(2^j x - k)](\omega) = \frac{1}{2^j} e^{-i\frac{k}{2^j}\omega} \hat{\psi}\left(\frac{\omega}{2^j}\right)$$

所以，當 j 較大時， $\psi(2^j x - k)$ 的探測密度較高，它在較小的尺度下探測較高亦較寬的頻率範圍。另一方面， ψ 和 ϕ 的最重要不同是， $\int \psi dx = 0$ ，故 $\hat{\psi}(0) = 0$ 。因為 $\hat{\psi}(\omega)$ 也是連續函數，故 $\hat{\psi}(\omega)$ 在 $\omega = 0$ 的附近差不多是 0。當 j 越大時， $\mathcal{F}[\psi(2^j x - k)](\omega)$ 在 $\omega = 0$ 的附近就有越大的範圍差不多是 0。又因為 $V_j \oplus W_j = V_{j+1}$ ，所以 $\mathcal{F}[\psi(2^j x - k)](\omega)$ 和 $\mathcal{F}[\phi(2^j x - k)](\omega)$ 的頻率反應範圍應該可以合成 $\mathcal{F}[\phi(2^{j+1} x - k)](\omega)$ 的頻率反應範圍。換句話說， $\mathcal{F}[\psi(2^j x - k)](\omega)$ 的頻率反應範圍應該是 $\mathcal{F}[\phi(2^{j+1} x - k)](\omega)$ 與 $\mathcal{F}[\phi(2^j x - k)](\omega)$ 的頻率反應範圍之差集。所以我們得到一個大致的印象：

$$\hat{f}(\omega) \mathcal{F}[\psi(2^j x - k)](\omega) \approx \begin{cases} \hat{0} & \text{當 } |\omega| \text{ 靠近 } 0 \\ \hat{f}(\omega) & \text{當 } |\omega| \text{ 在遠離 } 0 \text{ 的某個範圍內} \end{cases}$$

意思是說， $\hat{f}(\omega) \mathcal{F}[\phi(2^j x - k)](\omega)$ 相乘的結果，使得 $\hat{f}(\omega)$ 的低頻部分被濾掉了，反而是某一段的高頻部分通過了濾波器。在這個意義之下，我們稱 $\psi(x)$ 是一個高頻濾波 (*high-pass filter*) 函數。而在時間域的探測結果 $\langle f(x), \phi(2^j x - k) \rangle$ ，也就差不多是在 $\text{supp } \psi(2^j x - k)$ 這個小尺度之內對 $\hat{f}(\omega)$ 的某一段高頻做探測。 idx:highpas

當 j 越大時，前面所謂之高頻的頻率就越高而且範圍越寬。相對而言， $\psi(2^j x - k)$ 的頻率反應範圍比 $\phi(2^j x - k)$ 來得高，而且它們的頻率反應範圍都在 $\phi(2^{j+1} x - k)$ 的頻率反應範圍之內。基於這個道理，我們

稱 W_j 是 $f \in V_{j+1}$ 的高頻部分 (*low frequency part*)。從 V_{j+1} 分解到 W_j 所使用的那組係數 $\frac{1}{2}d_k$ (參閱 (3.44))，稱為高頻濾波係數 (*low-pass filter coefficients*)。

idx:highfreq

利用這一套新的觀念，我們可以重新闡述凌波分解的意義，就是將 V_{j+1} 中的函數以低頻和高頻濾波係數，分解成兩相正交的低頻部分和高頻部分，它們分別在 V_j 和 W_j 裡面。高頻濾波係數必須針對個別的低頻濾波係數來設計，使得被分解的兩部分，可以被完美地合成 (PR: perfectly reconstructable)。這樣的一對高低頻濾波係數，可以利用凌波理論來設計。亦即，低頻濾波係數來自於自格係數 c_k ，高頻濾波係數來自於凌波函數在 V_1 中的展開係數 d_k 。

最後，我們以一個理想化的例子，呈現前面所述的低頻與高頻關係。假若 $\varphi(x)$ 是一個“理想”的自格函數，亦即，它是一個“理想”的低頻濾波函數：

$$\hat{\varphi}(\omega) = \chi_{[-\frac{\pi}{2}, \frac{\pi}{2}]} = \begin{cases} 1 & \text{if } \omega \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\ 0 & \text{otherwise} \end{cases}$$

亦即 $\varphi(x)$ 可以把 $[-\frac{\pi}{2}, \frac{\pi}{2}]$ 之內的低頻部分一成不變地保留下來，而將此頻率範圍之外的高頻完全濾掉 (這就是它為什麼稱作“理想”的)。因為

$$\text{supp } \mathcal{F}[\varphi(2^j x - k)] = [-2^{j-1}\pi, 2^{j-1}\pi]$$

可見 $\varphi(2^j x - k)$ 的頻率反應範圍越來越寬，而且包含了越來越多的高頻。當 $j \rightarrow \infty$ 時， $\varphi(2^j x - k)$ 的頻率反應範圍趨近於整個 \mathbb{R} 。這也就反應了 $\cup V_j$ 將要稠密於 L^2 的事實。

考慮 $f(x) \in V_{j+1}$ ，因為 $f = \sum v_k \varphi(2^{j+1}x - k)$ ，所以

$$\text{supp } \hat{f}(\omega) \subset [-2^j\pi, 2^j\pi]$$

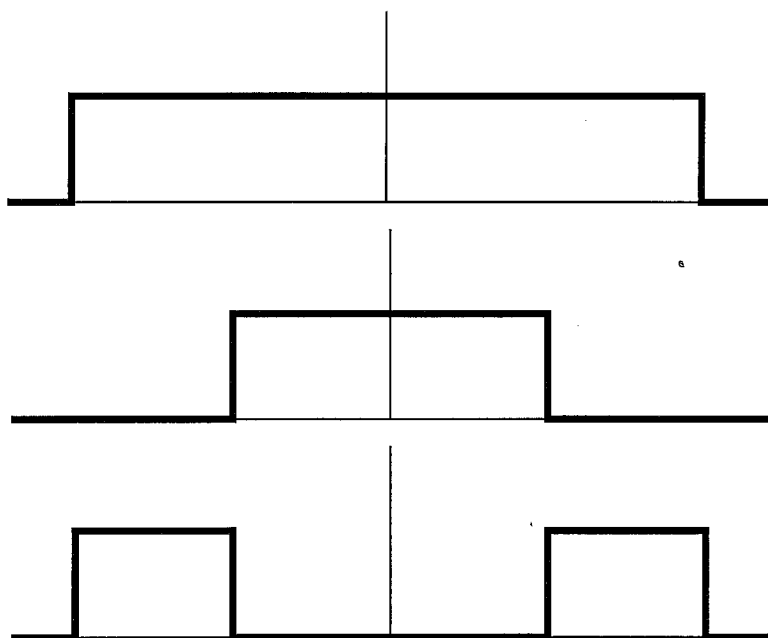


fig:hilo 圖 24 由上而下，依序是理想中 V_{j+1} 、 V_j 、 W_j 之頻率反應範圍

像這種函數，我們稱為有限頻寬 (band-limited)。若我們將 f 分解到 V_j 中，亦即取 $\mathcal{P}_j f(x)$ 。因為

$$\text{supp } \mathcal{F}[\mathcal{P}_j f(x)](\omega) \subset [-2^{j-1}\pi, 2^{j-1}\pi]$$

可見我們擷取了 $f(x)$ 的低頻部分。高低頻之間的差集，就是

$$[-2^j\pi, -2^{j-1}\pi] \cup [2^{j-1}\pi, 2^j\pi] \tag{3.50} \quad \text{eq:fhi}$$

如果我們定義 $\eta(x)$ 是“理想”高頻濾波函數，亦即

$$\hat{\eta}(\omega) = \begin{cases} 1 & \text{if } \omega \in [-\pi, -\frac{\pi}{2}] \cup (\frac{\pi}{2}, \pi] \\ 0 & \text{otherwise} \end{cases}$$

所以 $\text{supp } \mathcal{F}[\eta(2^j x - k)]$ 就是 (3.50) 中所需要的範圍。因此，我們知道， $\eta(x)$ 可以是對應 $\varphi(x)$ 的一個波函數。參閱圖 24 的圖示。

像 φ 和 η 這樣理想的低頻與高頻濾波函數，實際上有其缺點。因為 $\hat{\varphi}$ 和 $\hat{\eta}$ 在頻率域是有限函蓋函數，所以 φ 和 η 在時間域必定不是有限函蓋。而且 $\hat{\varphi}$ 和 $\hat{\eta}$ 都類似於方盒函數，所以 φ 和 η 在時間域類似於 sinc 函數；它們消散得很慢 ($O(\frac{1}{|\omega|})$ 的消散速度)。因此，它們固然能在頻率域做精確的探測，卻不能在時間域做精確的探測。在第二章十三節中，我們曾經作此闡述。在下一節中，我們將作明確的論述。

習題 3.18 證明 $\hat{\phi}(0) = 1$, $\hat{\psi}(0) = 0$ 。

3.10 相位平面

在平面上定義一組直角坐標系統。令橫軸是代表時間域的 x 變數，令縱軸是代表頻率域的 ω 變數，這個平面稱為相位平面。相位平面上的

idx:phase

一個點 (x, ω) 代表時間是 x 而頻率是 ω 。

定義在相位平面上的函數 $P(x, \omega)$ ，可以被理解作：在 x 時間發生 ω 頻率的強度。人類自古以來就使用相位平面上的函數來記錄或描述音樂。比如說，現在習用的五線譜，就是這種函數的最佳範例。如圖 25*。五線譜的目的就是要明確的標定，在什麼時間，出現什麼頻率的樂音，有時候還註明其強度、感情與音色。演奏者根據樂譜的描述，將這個相位平面上的函數轉變成振動空氣的聲波。

我們曾說 $\langle f, g \rangle$ 可以視為以 $g(x)$ 探測 $f(x)$ 。又因為 Parseval 等式，這個探測的結果幾乎等於 $\langle \hat{f}, \hat{g} \rangle$ ，也就是用 $\hat{g}(\omega)$ 探測 $\hat{f}(\omega)$ 。如果 $g(x)$ 的函蓋範圍是 I_x ，而 $\hat{g}(\omega)$ 的函蓋範圍是 I_ω ，則 g 在相位平面上函蓋了 $I_x \times I_\omega$ 的矩形區域。所以 $\langle f, g \rangle$ 的探測結果，就是訊號 f 在 I_x 時間範圍內發生 I_ω 範圍內之頻率的某種平均值。換句話說，訊號 f 在

* 這個樂譜並不是瞎掰的，我相信它是一首幾乎人人聽過的著名樂曲。您知道它是什麼曲子嗎？

Allegro con brio
Strings doubled in tow octaves below.

fff

STRINGS.

ff HORNS.

dolce.

fig:score 圖 25 五線譜就是相位平面上的一個函數

I_x 時間發生 I_ω 頻率的某種平均強度是 $\langle f, g \rangle$ 。顯然，我們會“希望” I_x 和 I_ω 都越窄越好。但是，從幾個已知例子裡面，我們發現它們無法同時很窄。例如方盒函數 $B_1(x)$ ，它在時間域的函蓋是 $[0, 1]$ ，但是在頻率域的函蓋是 \mathbb{R} 。而前一節所說的理想低頻濾波函數 $\varphi(x)$ ，雖然它在頻率域的函蓋是 $[0, 1]$ ，但是在時間域的函蓋是 \mathbb{R} 。使用 $B_1(x)$ 做探測函數時，比較確定訊號發生的時間，卻不易確定訊號發生的頻率；反之，使用 $\varphi(x)$ 做探測函數時，比較確定訊號發生的頻率，卻不易確定訊號發生的時間。這就是為什麼，所謂“理想的”低頻濾波函數，在實用上並不甚理想。

任取一個非零的 $g(x) \in L^2$ ，我們可以將其正規化使得 $\|g\| = 1$ (如果 $\|g\| \neq 1$ ，取 $\frac{1}{\|g\|} g(x)$)。此外，假設 $g(x)$ 及 $\hat{g}(\omega)$ 均消散得夠快，使

得以下所需要的積分都存在。定義

$$\bar{x} = \int x |g(x)|^2 dx, \quad \bar{\omega} = \frac{1}{2\pi} \int \omega |\hat{g}(\omega)|^2 d\omega \quad (3.51) \quad \text{eq: xbar}$$

因為 $\int |g|^2 dx = \frac{1}{2\pi} \int |\hat{g}|^2 d\omega = 1$ 所以 $|g|^2$ 和 $\frac{1}{2\pi} |\hat{g}|^2$ 都可以被視為機率分佈函數。在此意義下， \bar{x} 和 $\bar{\omega}$ 分別是 $|g|^2$ 和 $\frac{1}{2\pi} |\hat{g}|^2$ 的平均值 (mean value)。在物理意義下，則它們是質量中心。再定義

$$\sigma^2 = \int (x - \bar{x})^2 |g(x)|^2 dx, \quad \hat{\sigma}^2 = \frac{1}{2\pi} \int (\omega - \bar{\omega})^2 |\hat{g}(\omega)|^2 d\omega \quad (3.52) \quad \text{eq: xvar}$$

若仍以機率分佈解釋，則 σ^2 和 $\hat{\sigma}^2$ 分別是 $|g|^2$ 和 $\frac{1}{2\pi} |\hat{g}|^2$ 的變異數 (variance)，而 σ 和 $\hat{\sigma}$ (皆取正值) 則是標準差 (standard deviation)。在訊號處理的術語中，若 $g(x)$ 被視為一筆訊號，則 σ 稱為 g 的 RMS (Root Mean Square) 時域 (duration)， $\hat{\sigma}$ 稱為 g 的 RMS 頻寬 (bandwidth)。

在相位平面上定義一個矩形區域

$$[\bar{x} - \sigma, \bar{x} + \sigma] \times [\bar{\omega} - \hat{\sigma}, \bar{\omega} + \hat{\sigma}]$$

我們稱此區域為 $g(x)$ 的有效函蓋。就好像 g 在相位平面上開了一個窗口，而 $\langle f, g \rangle$ 就是透過這個窗口去探測 $f(x)$ 。為了探測的精確度，我們當然希望這個窗口越小越好。但是下面這個定理，顯示此願望是無法達成的。它的意思是說，有效函蓋之面積 $4\sigma\hat{\sigma}$ 不小於一個常數。所以，如果這個窗口在時間域很窄 (它可以精確探測訊號發生的時間)，則它在頻率域就必定很寬 (它不能精確探測訊號發生的頻率)；反之亦然。這個結果應該符合我們的生活經驗：當我們只聽到很短的一段音樂，比如說播放了一秒鐘的唱片，幾乎無法分辨音樂的內容。這種不可能性，讓我們聯想到量子力學中的海森堡測不準原理：無法同時測知一個粒子的位置和動量。以下這個定理，可以說是數學版的海森堡測不準原理。

idx: phasesupp

海森堡測不準定理 若 $g \in L^2(\mathbb{R})$ 且 $\|g\| = 1$ 。此外，若 $xg(x)$ 與 $\omega\hat{g}(\omega)$ 均為 $L^2(\mathbb{R})$ 之函數，而且 $g(x)$ 消散得夠快足以使得

$$\lim_{x \rightarrow \pm\infty} xg^2(x) = 0 \tag{3.53} \quad \text{eq: xgzero}$$

則 (3.52) 所定義之 σ 與 $\hat{\sigma}$ 符合 idx:Hess

$$\sigma\hat{\sigma} \geq \frac{1}{2} \tag{3.54} \quad \text{eq:Hess}$$

證明：因為 $xg(x)$ 與 $\omega\hat{g}(\omega)$ 均為 $L^2(\mathbb{R})$ 函數，使得 (3.51) 與 (3.52) 中的積分皆存在。再加上 (3.53)，使得 $g'(x)$ 存在而且是 L^2 函數，以及 $xg(x)g'(x)$ 可積。我們需要這些性質來證明 (3.54)。

爲了行文的方便，我們假設 $\bar{x} = 0$ 與 $\bar{\omega} = 0$ 。這樣做並不會失去一般性，只是坐標平移而已。如此一來，則

$$\sigma^2 = \int x^2 g^2 dx = \|xg\|^2, \quad \hat{\sigma}^2 = \frac{1}{2\pi} \int |\omega|^2 |\hat{g}|^2 d\omega = \frac{1}{2\pi} \|i\omega\hat{g}\|^2$$

但是，由分部積分 (integration by parts) 得知

$$\begin{aligned} i\omega\hat{g}(\omega) &= \int (i\omega)g(x)e^{-i\omega x} dx = - \int g(x) de^{-i\omega x} \\ &= -g(x)e^{-i\omega x} \Big|_{x=-\infty}^{\infty} + \int e^{-i\omega x} dg(x) \end{aligned}$$

由於 (3.53)，勢必 $\lim_{x \rightarrow \pm\infty} g(x) = 0$ ，所以前式之第一項爲 0。取 $i\omega\hat{g}(\omega)$ 之傅立葉逆轉換，即得 $g'(x)$ 而且 $g' \in L^2$ 。可見

$$\hat{\sigma}^2 = \frac{1}{2\pi} \|\mathcal{F}[g'](\omega)\|^2 = \|g'\|^2$$

一方面，根據席瓦茲不等式，

$$\left| \int x g g' dx \right|^2 \leq \|xg\|^2 \|g'\|^2 = \sigma^2 \hat{\sigma}^2 \quad (3.55) \quad \text{eq: xgSchwarz}$$

另一方面，

$$\int x g g' dx = \frac{1}{2} \int x dg^2 = \frac{1}{2} x g^2 \Big|_{x=-\infty}^{\infty} - \frac{1}{2} \int g^2(x) dx$$

由於 (3.53)，前式之第一項為 0，故

$$\left| \int x g g' dx \right|^2 = \frac{1}{4} \|g\|^4 = \frac{1}{4}$$

所以 $\frac{1}{4} \leq \sigma^2 \hat{\sigma}^2$ ，也就得到了 (3.54)。

最後，我們檢查一些細節。若 $g \in L^2$ 且 $xg \in L^2$ ，則

$$\begin{aligned} \int x |g(x)|^2 dx &\leq \int |x| |g(x)|^2 dx \leq \int (1 + |x|^2) |g(x)|^2 dx \\ &= \|g\|^2 + \|xg\|^2 < \infty \end{aligned}$$

讀者應可從此推導 (3.51) 及 (3.52) 中的積分都存在。 \square

當 (3.55) 裡面的 \leq 達到 $=$ 的時候，(3.54) 達到其最小之可能： $\sigma \hat{\sigma} = \frac{1}{2}$ 。根據席瓦茲不等式之性質，此時存在常數 c 使得 $g'(x) = c_1 x g(x)$ 。所以 $g(x)$ 必具備以下形式

$$g(x) = c_1 e^{c_2 x^2}$$

為使 $g \in L^2$ ，必須 $c_2 < 0$ 。為使 $\|g\| = 1$ ， c_1 與 c_2 必須具備某種關係。我們發現，符合這些條件的 $g(x)$ 必為所謂的高斯分佈函數 (*Gaussian function*)：

$$g_\sigma(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2\sigma^2}x^2} \quad (3.56) \quad \begin{array}{l} \text{id:Gaussian} \\ \text{eq:Gaussian} \end{array}$$

$g_{\sigma}(x - \bar{x})$ 的平均值與標準差分別就是 \bar{x} 和 σ 。而其傅立葉轉換為

$$\mathcal{F}[g_{\sigma}(x - \bar{x})](\omega) = e^{-i\bar{x}\omega} e^{-\frac{1}{2}\sigma^2\omega^2} \quad (3.57) \quad \text{eq:Gaussian}$$

雖然 $g_{\bar{x},\sigma}$ 與其傅立葉轉換均為無窮函蓋之函數，但在實際應用上，這並不造成多大的困難。因為它們都以 e^{-x^2} 之速度消散，這是非常快的消散速度。如果容許相當的誤差，它們差不多可以被視為有限函蓋函數。由於 $\hat{g}_{\sigma}(0) = 1$ ，可見高斯分佈函數是一個低頻濾波函數。 \bar{x} 決定它的位置 (平移)， σ 決定它的尺度 (脹縮) 和頻率反應範圍。

既然高斯分佈函數達到最小之有效函蓋，自格函數 $\phi(x)$ 的有效函蓋面積當然就不是最好的。但是自格函數仍有其特色：它可以產生對應的高頻濾波函數—凌波函數，而且有一套非常簡單的分解與合成之演算法。此外，自格函數與凌波函數在其尺度與頻率反應範圍之間，有一種自動的調適性。這是我們以下要解釋的現象。

所謂調適性之基本關係，來自於脹縮平移之傅立葉轉換公式 (3.48) 和 (3.49)。每當尺度提高一階 (從第 j 層到 $j+1$ 層)， $\psi_{jk}(x)$ 在相位平面上的有效函蓋矩形，就會變窄，拉高，並且整個提高其頻率反應範圍。不同尺度的有效函蓋矩形之面積，基本上保持不變。只是在大尺度、低頻時，探測一個寬闊的時間範圍、和一個低而窄的頻率範圍。反之，在小尺度、高頻時，探測一個狹窄的時間範圍、和一個高而寬的頻率範圍。請看圖 26。這是一份簡化的示意圖，旨在表明有效函蓋矩形的調適性。真正的情況下，這些矩形應彼此部分重疊。圖 26 之左上方表現的是方盒函數 $B_1(x)$ ，右上方表現的是理想低頻與高頻濾波函數 $\varphi(x)$ 及 $\eta(x)$ ，左下方表現的是一般 windowed 傅立葉或餘弦轉換 (DCT: Discrete Cosine Transform)，右下方就是表現凌波函數的有效函蓋了。

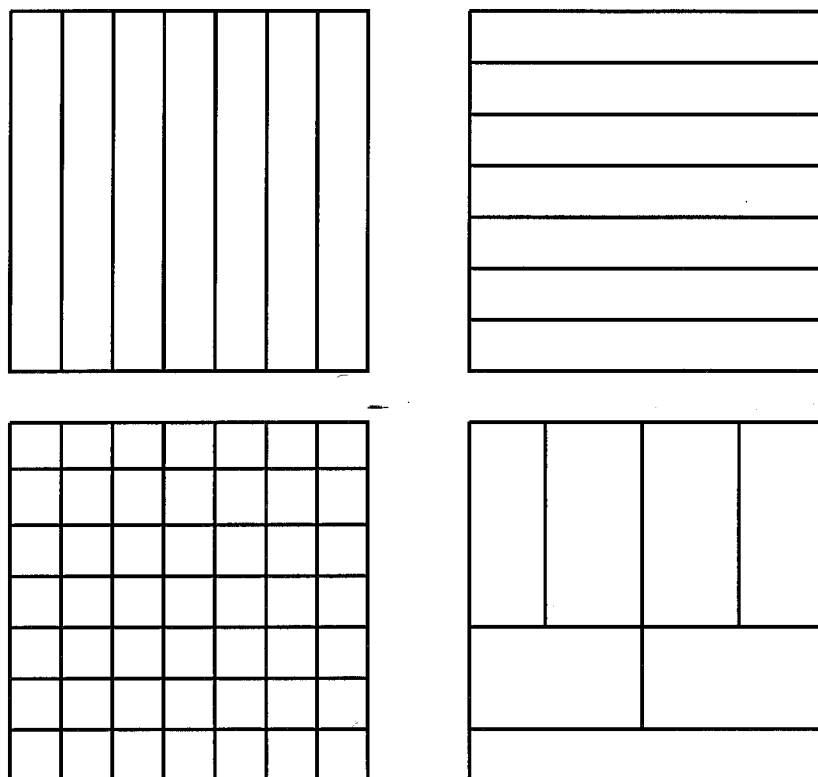


fig:phase 圖 26 相位平面上之各種有效函蓋示意圖

這樣的調適性，很符合我們在概念上對訊號的認識。低頻訊號的週期比較長，如果探測的時間很短，可能連一個週期都看不全，怎麼能斷定其頻率？而且，一般來說，低頻的資訊量豐富而重要，通常有必要將它們仔細地分辨。所以，一個有效率的低頻濾波函數，應該在時間域上寬，而在頻率域上窄。反之，高頻訊號的週期比較短，因此總是局部現象，如果使用了太長的探測時間，可能無法捕捉那些細微的變化；再者，通常高頻的資訊量稀疏而較不重要（它們可能根本就是雜訊），若在頻率域上細細探索，可能造成資源的浪費。所以，一個

有效率的高頻濾波函數，應該在時間域上窄，而在頻率域上寬。當然這並非定律，對於某些應用的學門（例如海洋探測），可能真的需要在高頻上仔細地探索。這時候，或許會發覺普通的波函數在處理高頻時太過粗糙，而波包 (wavelets packets) 或許有所幫助。波包的想法，就是要將小尺度、高頻譜的波函數再分解，以達到將高頻分割的目的。它的理論基礎不外乎本章前四節的內容，此處不再深究。

關於高頻的資訊量較為稀疏這個看法，至少對聲波而言是符合自然現象的。以頻率而言，我們的耳朵大約可以聽到 2000 Hz 範圍內的聲波。但是，以樂音而言，耳朵聽起來似乎呈等差變化的音階，其實是個指數函數。以鋼琴為例，參照圖 27。以中央 C 的音為準，它的頻率是 264，比它高八度的 C 頻率是 2×264 ，再高八度的 C 頻率是 4×264 ，高三個八度的 C 頻率是 $2^3 \times 264$ 。我們觀察到，在頻率 264 與 528 之間，鋼琴鍵盤上有十二個音（全音與半音），但是在頻率 1056 與 2112 之間，也是只有十二個音。可見在比較低頻的地方，耳朵所能接收的聲波訊號是比較密的，越高頻處訊號就越稀疏。

20000

習題 3.19 這個習題的目的，在於提示為什麼需要 (3.53) 這個條件。我們舉一個簡單的例子，以表現 $\int u(x) dx$ 可積並不能保證 $u(x) \rightarrow 0$ 。令

$$u(x) = \begin{cases} 1 & \text{if } x \in [n, n + \frac{1}{n^2}], \text{ for } n \in \mathbb{N} \\ 0 & \text{otherwise} \end{cases}$$

證明

$$\int_{-\infty}^{\infty} u(x) dx < \infty \quad \text{但是} \quad \lim_{x \rightarrow \infty} u(x) \neq 0$$

習題 3.20 若 $g(x)$ 恆正。證明若 $g'(x) = cxg(x)$ 則 $g(x) = \alpha e^{\frac{1}{2}cx^2}$ ，其中 $\alpha > 0$ 。

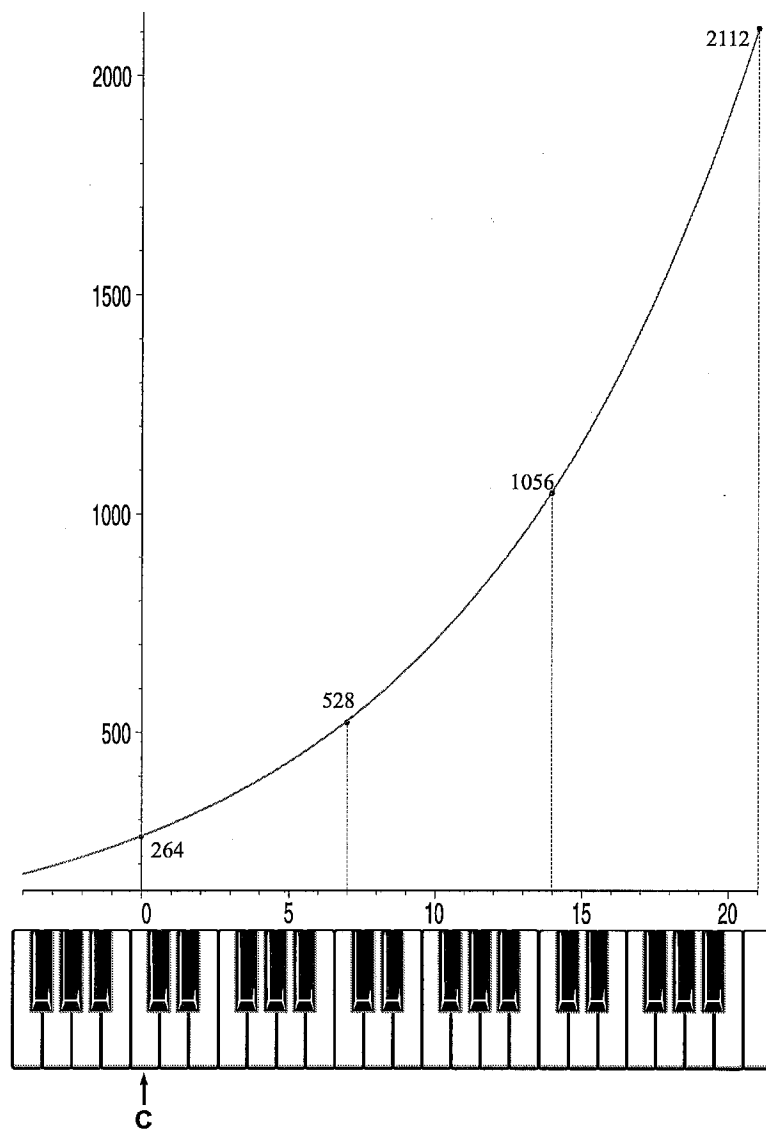


fig:piano 圖 27 鋼琴的音階與頻率

習題 3.21 若已知 $\int e^{-x^2} dx = \sqrt{\pi}$ ，證明*

$$\int g_{\sigma}(x - \bar{x}) dx = 1$$

而且 $g_{\sigma}(x - \bar{x})$ 之平均值是 \bar{x} ，標準差是 σ 。

習題 3.22 由 $\int e^{-x^2} dx = \sqrt{\pi}$ ，證明

$$\int e^{-ax^2+bx} dx = \sqrt{\frac{\pi}{a}} e^{\frac{b^2}{4a}}$$

其中 a, b 是常數。是否可以由此等式推導 (3.57)？(提示：做 $-ax^2 + bx$ 的完全平方。)

3.11 波轉換與泰勒展開

前面我們探討了波函數的尺度與頻譜之間的關係。波分解，相當於把 V_{j+1} 中的訊號分成低頻與高頻兩部分，低頻的放在 V_j 中，高頻的放在 W_j 中。在這一節裡面，我們將探討波分解與泰勒展開的關係。籠統地說，泰勒多項式中的低階項對應了低頻的訊號，而高階項對應了高頻的訊號。

根據 (3.19)，我們可以取高頻濾波係數為 $d_k = (-1)^k c_{2p-1-k}$ ，其中 $k = 0, 1, \dots, 2p-1$ 。由 (3.36) 知道 $\sum (-1)^k k^m c_k = 0$ ，其中 $m = 0, 1, \dots, p-1$ 。其實，可以推論

$$\sum_k k^m d_k = 0, \quad \text{for } 0 \leq m \leq p-1 \quad (3.58)$$

eq:dkzero

* $\int e^{-x^2} dx = \sqrt{\pi}$ 之證明需要二重積分與極坐標，不能在此詳述。讀者應該可以在大部分的微積分教科書中找到證明。

任選整數 N 及 $0 \leq m \leq p-1$ ，根據 (3.36) 可得

$$\begin{aligned} \sum_k (-1)^k (N-k)^m c_k &= \sum_k (-1)^k \left[\sum_{n=0}^m \binom{m}{n} N^{m-n} (-k)^n \right] c_k \\ &= \sum_{n=0}^m \binom{m}{n} N^{m-n} (-1)^n \sum_k (-1)^k k^n c_k = 0 \end{aligned}$$

代入 $N = 2p-1$ 再以足標變換 $k \mapsto N-k$ ，則

$$0 = \sum_k (-1)^k (N-k)^m c_k = - \sum_k (-1)^k k^m c_{2p-1-k} = - \sum_k k^m d_k$$

故得 (3.58)。以 $p=2$ 、 $m=1$ 為例 (參見 (3.37))，即

$$\begin{aligned} d_1 + 2d_2 + 3d_3 &= -c_2 + 2c_1 - 3c_0 \\ &= -3(c_0 - c_1 + c_2 - c_3) + (-c_1 + 2c_2 - 3c_3) = 0 \end{aligned}$$

現在，假設一筆離散訊號 v_k 是一個連續訊號的均勻樣本值。亦即存在一個函數 $f(x)$ 使得 $v_k = f(x_k)$ ，其中 x_k 是等距節點。假設 $x_k - x_{k-1} = h$ ，而且 $f(x)$ 在 (x_n, x_{n+2p-1}) 之間是個至少 $p+1$ 次連續可微的函數。所以 $f(x)$ 可以用 x_n 為參考點在 (x_n, x_{n+2p-1}) 做 p 階的泰勒展開。我們特別關注當 $k=0, 1, \dots, 2p-1$ 的情形：

$$\begin{aligned} f(x_{n+k}) &= f(x_n + kh) \\ &= \sum_{s=0}^p \frac{1}{s!} f^{(s)}(x_n) (kh)^s + \frac{1}{(p+1)!} f^{(p+1)}(\xi_k) (kh)^{p+1} \end{aligned}$$

其中 ξ_k 是介於 x_n 和 x_{n+k} 之間的某數。現在，若我們將 $v_n, v_{n+1}, \dots, v_{n+2p-1}$ 做一次高頻濾波，則結果是

$$\begin{aligned} \sum_{k=0}^{2p-1} d_k v_{n+k} &= \sum_{s=0}^{p-1} \left[\sum_{k=0}^{2p-1} k^s d_k \right] \frac{1}{s!} f^{(s)}(x_n) h^s \\ &\quad + \frac{1}{p!} f^{(p)}(x_n) h^p \sum_{k=0}^{2p-1} k^p d_k + O(h^{p+1}) \end{aligned}$$

令 $K = \sum k^p d_k$ ，根據 (3.58)，我們得到

$$\sum_{k=0}^{2p-1} d_k v_{n+k} = \frac{K h^p}{p!} f^{(p)}(x_n) + O(h^{p+1})$$

我們獲得的結論是，高頻濾波的過程，相當於消去泰勒展開的低階項。因為泰勒展開的係數通常會隨著項次的增高而變小，可見高頻過濾之後的訊號值應該變小。另一個有趣的觀察是，高頻濾波係數 d_k 也可以拿來當做差分係數。意思是說，只要 $f(x)$ 足夠可微，則

$$\frac{p!}{K h^p} \sum_{k=0}^{2p-1} d_k f(x_{n+k}) = f^{(p)}(x_n) + O(h^{p+1}) \quad (3.59)$$

eq:dkdiff

所以我們可以綜合一個結論：

低頻濾波過程相當於取平均，高頻濾波過程相當於做差分

這一種類比，對於比較熟悉微分方程的讀者而言，或許比較有用。

3.12 特徵函數

在這一章的最後兩節中，我們以傅立葉轉換為工具，為自格函數與凌波函數推導出更多的性質；或是為前面敘述的某些性質提出證明。讀者必須瞭解，在凌波函數發展之初，傅立葉轉換之相關技巧是必備的工具與基礎。所以，若要閱讀相關的書籍或文獻，必須具備相當程度的傅立葉技巧。這本書的推演方式，盡量減少對傅立葉技巧的依賴，並延後傅立葉技巧的介紹。其目的純粹是為了減輕初學讀者的負擔，絕非因為這些理論都不重要。雖然我們可以不使用傅立葉技巧而導出許多性質，但是，對於這些技巧的掌握，將有助於更深刻地瞭解凌波函數。

令 $\hat{\phi}(\omega)$ 是 $\phi(x)$ 的傅立葉轉換。則根據自格等式 (3.4) 以及尺度與頻譜之關係 (3.48) 和 (3.49)，得到

$$\begin{aligned}\hat{\phi}(\omega) &= \sum_k c_k \mathcal{F}[\phi(2x - k)](\omega) \\ &= \frac{1}{2} \sum_k c_k e^{-i\frac{k}{2}\omega} \hat{\phi}\left(\frac{\omega}{2}\right)\end{aligned}$$

令

$$m_0(\omega) = \frac{1}{2} \sum_k c_k e^{-ik\omega} \quad (3.60)$$

eq:mzerodef

顯然 $m_0(\omega)$ 是一個 2π 週期函數，而 $\frac{1}{2}c_k$ 就是 m_0 的傅立葉係數。稱 $m_0(\omega)$ 為 $\phi(x)$ 的特徵函數。則

idx:charfcn

$$\hat{\phi}(\omega) = m_0\left(\frac{\omega}{2}\right)\hat{\phi}\left(\frac{\omega}{2}\right) \quad (3.61)$$

eq:phihat

由於 $\sum c_k = 2$ ，故 $m_0(0) = 1$ 。所以，在 (3.61) 中代入 $\omega = 0$ 時，得到恆等式 $\hat{\phi}(0) = \hat{\phi}(0)$ 。因為 $\hat{\phi}(0) = \int \phi(x) dx$ ，爲了唯一性我們規定 $\int \phi dx = 1$ ，也就是 $\hat{\phi}(0) = 1$ 。

反過來，若已知一個函數 $\hat{\phi}(\omega)$ ，並存在一個 2π 週期函數 $m_0(\omega)$ 使得 (3.61) 成立，而且 $m_0(0) = 1$ ，則 m_0 有傅立葉級數 $\sum \gamma_k e^{-ik\omega}$ 。取 $\hat{\phi}(\omega)$ 的傅立葉逆轉換得到 $\phi(x)$ ，則這個 $\phi(x)$ 是一個自格函數，它的自格係數是 $c_k = 2\gamma_k$ 。

例如方盒函數 $B_1(x)$ 的非零自格係數是 $c_0 = 1$ 、 $c_1 = 1$ ，其特徵函數就是

$$\frac{1 + e^{-i\omega}}{2}$$

根據 (2.51) 和 (2.44) ,

$$\begin{aligned}\hat{B}_p(\omega) &= \hat{B}_1(\omega)^p \\ &= \left(\frac{1 + e^{-i\frac{\omega}{2}}}{2}\right)^p \hat{B}_1\left(\frac{\omega}{2}\right)^p \\ &= \frac{1}{2^p} \left(\sum_{k=0}^p \binom{p}{k} e^{-ik\frac{\omega}{2}}\right) \hat{B}_p\left(\frac{\omega}{2}\right)\end{aligned}$$

所以 $B_p(x)$ 的特徵函數就是

$$\frac{1}{2} \sum_{k=0}^p \frac{1}{2^{p-1}} \binom{p}{k} e^{-ik\omega}$$

可見 $B_p(x)$ 的非零自格係數就是 $\frac{1}{2^{p-1}} \binom{p}{k}$ 。因此

$$B_p(x) = \sum_{k=0}^p \frac{1}{2^{p-1}} \binom{p}{k} B_p(2x - k)$$

也就證明了性質 3.1 。

在第二節中，我們由 $\{\phi(x - k)\}$ 的正則關係，導出自格係數 c_k 所應符合的關係 (3.6) 。在這一節裡面，我們則是由 $\{\phi(x - k)\}$ 的正則關係，導出一些 m_0 與 $\hat{\phi}$ 的基本性質。

由於 $\phi(x)$ 是正則自格函數，根據 Parseval 等式 (2.43) 得到

$$\begin{aligned}\delta_{0k} &= \langle \phi(x), \phi(x - k) \rangle \\ &= \frac{1}{2\pi} \int \hat{\phi}(\omega) \overline{e^{-ik\omega} \hat{\phi}(\omega)} d\omega \\ &= \frac{1}{2\pi} \int |\hat{\phi}(\omega)|^2 e^{ik\omega} d\omega\end{aligned}\tag{3.62}$$

eq:phihatsu

我們將 $|\hat{\phi}(\omega)|^2$ 週期化 (參見 (2.33)) 成爲

$$F_{\phi}(\omega) = \sum_n |\hat{\phi}(\omega - 2n\pi)|^2 \quad (3.63)$$

idx:periodize

eq:Fphi

請注意 $F_{\phi}(\omega)$ 是一個 2π 週期函數。而且，因爲 $\hat{\phi} \in L^2(\mathbb{R})$ ，所以 $F_{\phi} \in L^2_p([-\pi, \pi])$ 。由於 $e^{ik\omega}$ 是 2π 週期函數，所以 (3.62) 可以寫成

$$\delta_{0k} = \frac{1}{2\pi} \int_{-\pi}^{\pi} F_{\phi}(\omega) e^{ik\omega} d\omega$$

現在我們假設 $F_{\phi}(\omega)$ 是連續函數 (通常的確如此)，則根據傅立葉級數之合成公式 (2.28)，得到

$$F_{\phi}(\omega) = \frac{1}{2\pi} \sum_k e^{ik\omega} \int_{-\pi}^{\pi} F_{\phi}(\omega) e^{-ik\omega} d\omega = \sum_k \delta_{0k} e^{ik\omega} = 1 \quad (3.64)$$

eq:Fphone

反過來，若 $F_{\phi}(\omega) = 1$ ，則根據 (3.62)，

$$\langle \phi(x), \phi(x - k) \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} F_{\phi}(\omega) e^{ik\omega} d\omega = \delta_{0k} \quad (3.65)$$

eq:Fphik

所以 $\phi(x)$ 是正則自格函數。我們的結論如下。

性質 3.4 若 $\phi(x) \in L^2(\mathbb{R})$ 是一個自格函數，且 (3.63) 中定義的 $F_{\phi}(\omega)$ 爲連續之 2π 週期函數。則 $\phi(x)$ 成爲正則自格函數的充分且必要條件是 $F_{\phi}(\omega) = 1$ 。

更進一步，我們可以從 $\{\phi(x - k)\}$ 的正則性質推衍出特徵函數 $m_0(\omega)$ 的性質。由於 F_{ϕ} 是個常數函數，我們可以將它縮窄：

$$1 = F_{\phi}(\omega) = F_{\phi}(2\omega)$$

代入 (3.61)，得到

$$F_\phi(2\omega) = \sum_n |\hat{\phi}(2\omega - 2n\pi)|^2 = \sum_n |m_0(\omega - n\pi)|^2 |\hat{\phi}(\omega - n\pi)|^2$$

現在，我們將整數 n 分成奇數 ($n = 2\ell - 1$) 與偶數 ($n = 2\ell$) 兩部分，將上式寫成

$$\begin{aligned} F_\phi(2\omega) &= \sum_\ell |m_0(\omega - 2\ell\pi + \pi)|^2 |\hat{\phi}(\omega - 2\ell\pi + \pi)|^2 \\ &\quad + \sum_\ell |m_0(\omega - 2\ell\pi)|^2 |\hat{\phi}(\omega - 2\ell\pi)|^2 \end{aligned}$$

由於 m_0 是 2π 週期函數，所以 $m_0(\omega - 2\ell\pi) = m_0(\omega)$ 而 $m_0(\omega - 2\ell\pi + \pi) = m_0(\omega + \pi)$ 。故上式又可以簡化成

$$F_\phi(2\omega) = |m_0(\omega + \pi)|^2 F_\phi(\omega + \pi) + |m_0(\omega)|^2 F_\phi(\omega) \quad (3.66) \quad \text{eq:Fphifix}$$

但是 $F_\phi(2\omega) = F_\phi(\omega + \pi) = F_\phi(\omega) = 1$ ，所以 $|m_0(\omega)|^2 + |m_0(\omega + \pi)|^2 = 1$ 。我們獲得結論如下。

性質 3.5 若 $\phi(x) \in L^2(\mathbb{R})$ 是一個正則自格函數， $m_0(\omega)$ 如 (3.60) 所定義為其特徵函數，則

$$|m_0(\omega)|^2 + |m_0(\omega + \pi)|^2 = 1 \quad (3.67) \quad \text{eq:mzerosur}$$

在第二節中，我們看到 $\phi(x)$ 的正則性和 $\int \phi dx = 1$ 這兩個性質，分別導致自格係數 c_k 的 (3.6) 和 (3.10) 兩個條件。現在我們知道，這兩個性質分別導致 (3.67) 和 $m_0(0) = 1$ 這兩個條件。亦如第三節中之討論，這兩個條件只是 $\phi(x)$ 成為正則自格函數的必要條件。在這一節剩下的篇幅中，我們介紹一些提供充分條件的理論。

其實，保證自格函數 $\phi(x)$ 存在的充分條件可以非常地簡單。只要 $\sum c_k = 2$ 再加上 $\int \phi dx = 1$ ，換句話說就是 $m_0(0) = 1$ 和 $\hat{\phi}(0) = 1$ ，就存在唯一一個以 c_k 為係數的自格函數 $\phi(x)$ 。更甚者，如果這些 c_k 的非零項只有 c_0, \dots, c_N ，則 $\text{supp } \phi(x) \subset [0, N]$ 。但是，這個 ϕ 不一定連續、也未必在 L^2 ；它是一個所謂的廣義函數 (*generalized function or distribution*)。我們並不打算討論這種非常一般性的狀況。底下我們勾勒 Villemoes 的定理 [18]，以便明瞭什麼時候可以保證 $\phi(x)$ 在 L^2 內。讀者亦可參考曾正男先生的碩士論文 [17]。

若給定一組係數 c_k ，依 (3.60) 製造 2π 週期函數 $m_0(\omega)$ 。針對這個 m_0 定義一個算子 \mathcal{H} 。 \mathcal{H} 將一個 2π 週期函數 $\Gamma(\omega)$ 映射到另一個 2π 週期函數 $\mathcal{H}\Gamma(\omega)$ ：

$$\mathcal{H}\Gamma(\omega) = |m_0(\frac{\omega}{2})|^2 \Gamma(\frac{\omega}{2}) + |m_0(\frac{\omega}{2} + \pi)|^2 \Gamma(\frac{\omega}{2} + \pi)$$

所謂 Villemoes 條件就是：找得到一個實值*非負之 2π 週期函數 $f(\omega)$ 滿足 $f(0) > 0$ 而且

$$\mathcal{H}f(\omega) = f(\omega) \tag{3.68} \quad \text{eq:Hfix}$$

若此條件成立，則存在唯一一個以 c_k 為係數的自格函數 $\phi(x)$ ，它符合 $\phi(x) \in L^2$ 和 $\hat{\phi}(0) = 1$ 。

等式 (3.68) 的意義，就是說 $f(\omega)$ 是 \mathcal{H} 算子的一個不動點 (*fixed-point*)。我們在前面已經看到，當 $\phi(x)$ 是正則自格函數的時候，常數函數 $f(\omega) = 1$ 就是符合 (3.68) 的一個不動點。一般而言，只要 $\phi \in L^2$ 是個自格函數，則 (3.66) 成立。可見 $F_\phi(\omega)$ 必然符合 Villemoes 條件。因此，Villemoes 條件是自格函數 ϕ 落在 L^2 中的充分且必要條件。

* 亦即函數值是實數 (real-valued)。

如果確定了 $\phi(x)$ 在 L^2 之內，則可以利用 Lawton 條件來檢查 $\phi(x)$ 是否為正則自格函數。其實，Lawton 條件和以下敘述是等價*的：

idx:equiv

只有當 $f(\omega) = c$ 是常數函數的時候，才符合 (3.68)。

我們另外可以提供幾個檢查 $\phi(x)$ 是否為正則自格函數的充分條件：若 $m_0(0) = 1$ 且 (3.67) 成立，而且

$$m_0(\omega) \neq 0, \quad \forall \omega \in \left[-\frac{\pi}{3}, \frac{\pi}{3}\right]$$

或是

$$m_0(\omega) \neq 0, \quad \forall \omega \in \left[-\frac{\pi}{5}, \frac{\pi}{5}\right] \quad \text{and} \quad m_0\left(\frac{\pi}{3}\right) \neq 0$$

或是

$$m_0(\omega) \neq 0, \quad \forall \omega \in \left[-\frac{\pi}{17}, \frac{\pi}{17}\right] \quad \text{and} \quad m_0\left(\frac{\pi}{3}\right) \neq 0, \quad m_0\left(\frac{\pi}{5}\right) \neq 0$$

則 $\phi(x)$ 是正則自格函數。讀者亦可參考嚴健彰先生的碩士論文 [19]。

Daubechies [3] 從等式 (3.67) 中經過類似開根號的步驟得到三角多項式 $m_0(\omega)$ ，然後從 $m_0(\omega)$ 中取得 c_k 。當 $p > 5$ 的時候，仍無法求得這些 c_k 的真確解，故而必須以數值解估計。我們則是利用 c_k 所符合的 $2p$ 個條件 ((3.6)、(3.10) 和 (3.36))，去求聯立方程式的數值解而得到 c_k 。我們不在此介紹 Daubechies 求解的方法，有需要知道細節的讀者，亦可參考 [12]。

習題 3.23 若 $\phi(x) \in L^2(\mathbb{R})$ ， $F_\phi(\omega)$ 如 (3.63) 所定義，證明

$$F_\phi \in L_p^2([-\pi, \pi])$$

* 亦即兩個敘述同時成立或同時不成立。

習題 3.24 若 $\phi(x)$ 是 $L^2(\mathbb{R})$ 中的一個自格函數，且

$$\sum_n |\hat{\phi}(\omega - 2n\pi)|^2 > 0, \quad \text{for } \omega \in [-\pi, \pi]$$

令 $\varphi(x)$ 是以下函數的傅立葉逆轉換函數：

$$\frac{1}{\sum_n |\hat{\phi}(\omega - 2n\pi)|^2} \hat{\phi}(\omega)$$

試證明 $\varphi(x)$ 將是一個正則自格函數。

習題 3.25 若將 (3.63) 中 $F_\phi(\omega)$ 之定義改成 $\sum_n |\hat{\phi}(\omega + 2n\pi)|^2$ ，會產生什麼不同的結果嗎？

習題 3.26 比較 Villemoes 的理論和第三節中 Lawton 的理論，Villemoes 多說了些什麼？

習題 3.27 如果 c_k (也就是 $m_0(\omega)$) 符合 Villemoes 條件且 $\hat{\phi}(0) = 1$ ，證明

$$\hat{\phi}(\omega) = \prod_{j=1}^{\infty} m_0\left(\frac{\omega}{2^j}\right) \quad (3.69)$$

eq:infprod

其中 \prod 是連乘的意思。

習題 3.28 若 $\phi(x)$ 是正則自格函數，試利用 Poisson 求和公式的形態之一 (2.49) 來證明 $F_\phi(\omega) = 1$ 。

習題 3.29 若 $m_0(\omega)$ 符合 (3.67)，證明 $m_0(\pi) = 0$ 。據此證明 (3.11)，亦即 $\sum (-1)^k c_k = 0$ 。

習題 3.30 根據 $m_0(\omega)$ 的定義，證明

$$|m_0(\omega)|^2 + |m_0(\omega + \pi)|^2 = \frac{1}{2} \sum_m \sum_k c_k c_{k+2m} e^{im\omega}$$

然後證明 (3.6) 和 (3.67) 是等價的。

3.13 波函數與多項式

在這一節中，我們應用 $\phi(x)$ 的特徵函數 $m_0(\omega)$ 來討論自格函數與多項式之間的關係。然後我們討論 $m_0(\omega)$ 與波函數、以及與逼近能力之間的關係。

在前一節中我們看到，當 $\phi(x)$ 是正則自格函數時， $F_\phi(\omega) = 1$ 。代入 $\omega = 0$ 則是

$$F_\phi(0) = \sum_n |\hat{\phi}(2n\pi)|^2 = 1$$

但是 $\hat{\phi}(0) = 1$ ，所以

$$\hat{\phi}(2n\pi) = \delta_{0n} \quad (3.70) \quad \text{eq:phatint}$$

考慮

$$g(x) = \sum_k \phi(x - k), \quad x \in [0, 1]$$

因為 $\phi(x)$ 是有限函蓋函數，所以上式中的和其實只有有限多項。明顯地 $g(x) \in L^2[0, 1]$ 。原本 $g(x)$ 定義在 $[0, 1]$ 內，我們可以將 $g(x)$ 週期化成為 1 週期函數。經過變數變換 (參考 (2.34))，可以得到 $g(x)$ 的傅立葉級數展開

$$g(x) = \sum_n a_n e^{i2n\pi x}$$

其中的傅立葉係數 a_n 是

$$a_n = \int_0^1 g(x) \overline{e^{i2n\pi x}} dx = \sum_k \int_0^1 \phi(x - k) e^{-i2n\pi x} dx$$

因為 $e^{-i2n\pi x}$ 是 1 週期函數，也就是說 $e^{-i2n\pi x} = e^{-i2n\pi(x-k)}$ ，所以我們可以合併這些積分，得到

$$a_n = \int \phi(x) e^{-i2n\pi x} dx = \hat{\phi}(2n\pi) = \delta_{0n}$$

可見 $g(x) = 1$ ，亦即

$$\sum_k \phi(x-k) = 1 \quad (3.71)$$

eq:sumone

(3.71) 的意義是 $\{\phi(x-k)\}$ 可以組成常數函數。但是我們並不能說 $1 \in V_0$ ，因為 $1 \notin L^2(\mathbb{R})$ 。比較嚴格的說法是，任意給定一個有限區間 $[a, b]$ ， $\{\phi(x-k)\}$ 可以組成 $[a, b]$ 內的常數函數。但是這樣說很麻煩，所以通常我們還是簡單地講： V_0 包含常數函數。其實，經過變數變換（尺度的變換），我們知道 V_j 包含常數函數，對任何的 j 。

在第五節中，我們曾探討，若 $\psi(x)$ 正交於越多的多項式，則它的逼近能力越好。因此我們設計

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

雖然 ψ 正交於多項式 x^m ，亦正交於 V_0 ，但這並不能保證 V_0 包含了 x^m 。前面我們證明了 V_0 包含 0 階多項式。以下，我們要勾勒一套理論，以表明 V_0 的確包含了階數 $< p$ 的多項式。此處所謂之“包含”的意義，就是 $\{\phi(x-k)\}$ 可以在 $[a, b]$ 區間內組成階數 $< p$ 的多項式。

首先，我們觀察，給定一非負整數 s ，

$$m_0^{(s)}(\omega) = \frac{(-i)^s}{2} \sum_k k^s c_k e^{-ik\omega}$$

由於我們將 $p-1$ 個逼近能力的條件 (3.36) 加諸於 c_k 上，可見

$$m_0^{(s)}(\pi) = \frac{(-i)^s}{2} \sum_k (-1)^k k^s c_k = 0, \quad \text{for } 0 \leq s \leq p-1 \quad (3.72)$$

eq:mzerodiff

根據 (3.61)

$$\hat{\phi}'(\omega) = \frac{1}{2} m_0'(\frac{\omega}{2}) \hat{\phi}(\frac{\omega}{2}) + \frac{1}{2} m_0(\frac{\omega}{2}) \hat{\phi}'(\frac{\omega}{2})$$

給定任一整數 $n \neq 0$ ，代入 $\omega = 2n\pi$ 得

$$\hat{\phi}'(2n\pi) = \frac{1}{2}m'_0(n\pi)\hat{\phi}(n\pi) + \frac{1}{2}m_0(n\pi)\hat{\phi}'(n\pi)$$

先看前一項，若 n 為偶數，亦即 $n = 2n_1$ ，根據 (3.70)

$$\frac{1}{2}m'_0(n\pi)\hat{\phi}(n\pi) = \frac{1}{2}m'_0(0)\hat{\phi}(2n_1\pi) = 0$$

若 n 為奇數，根據 (3.72)，只要 $p \geq 2$ 就有

$$\frac{1}{2}m'_0(n\pi)\hat{\phi}(n\pi) = \frac{1}{2}m'_0(\pi)\hat{\phi}(n\pi) = 0$$

所以

$$\begin{aligned} \hat{\phi}'(2n\pi) &= \frac{1}{2}m_0(n\pi)\hat{\phi}'(n\pi) \\ &= \frac{1}{2} \begin{cases} m_0(0)\hat{\phi}'(n\pi) = \hat{\phi}'(n\pi) & \text{若 } n \text{ 為偶數} \\ m_0(\pi)\hat{\phi}'(n\pi) = 0 & \text{若 } n \text{ 為奇數} \end{cases} \end{aligned}$$

但是，若 n 為偶數，亦即 $n = 2n_1$ ，則

$$\hat{\phi}'(2n\pi) = \frac{1}{2}\hat{\phi}'(2n_1\pi) = \frac{1}{2} \begin{cases} \hat{\phi}'(n_1\pi) & \text{若 } n_1 \text{ 為偶數} \\ 0 & \text{若 } n_1 \text{ 為奇數} \end{cases}$$

若 n_1 仍為偶數，則再除以 2。依此類推，總在有限步驟後會出現奇數（最糟的情況也不過是 $n = 2^\ell$ ，則在 ℓ 次步驟後出現 1）。因此，我們得到結論：若 $p \geq 2$ 則

$$\hat{\phi}'(2n\pi) = 0, \quad \forall 0 \neq n \in \mathbb{Z}$$

我們留給讀者自己去推導

$$\hat{\phi}^{(m)}(2n\pi) = 0, \quad \forall 0 \neq n \in \mathbb{Z}, \quad 0 \leq m \leq p-1 \quad (3.73) \quad \text{eq:phidiff}$$

現在，令

$$f(x) = x\phi(x), \quad g(x) = \sum_k f(x-k)$$

則 $g(x)$ 是一個 1 週期函數。它的傅立葉級數展開是

$$g(x) = \sum_n a_n e^{i2n\pi x}$$

其中的傅立葉係數是

$$a_n = \int f(x) e^{-i2n\pi x} dx$$

若 $n = 0$ ，則 (參見 (3.35))

$$a_0 = \int f(x) dx = \int x\phi(x) dx = M_1$$

一般而言，

$$\begin{aligned} \int f(x) e^{-i\omega x} dx &= \int x\phi(x) e^{-i\omega x} dx \\ &= i \frac{\partial}{\partial \omega} \int \phi(x) e^{-i\omega x} dx = i \frac{d}{d\omega} \hat{\phi}(\omega) \end{aligned}$$

我們將 $\frac{d}{d\omega} \hat{\phi}(\omega)$ 簡記做 $\hat{\phi}'(\omega)$ 。所以，若 n 為一非零整數，則

$$a_n = i \hat{\phi}'(2n\pi) = 0$$

可見只要 $p \geq 2$ 則

$$\sum_k (x-k)\phi(x-k) = M_1 \tag{3.74}$$

eq:polyone

由於 $\sum \phi(x-k) = 1$ ，故得

$$x - \sum_k k\phi(x-k) = M_1, \quad \text{or} \quad \sum_k k\phi(x-k) = x - M_1$$

我們留給讀者自己去推導

$$\sum_k (x-k)^m \phi(x-k) = M_m, \quad \text{for } 0 \leq m \leq p-1 \quad (3.75)$$

eq:polym

並且得到以下結論。

性質 3.6 給定參數 $p \geq 1$ 。若 c_k 滿足條件 (3.36)，而 $\phi(x)$ 是以 c_k 為係數的自格函數，則 $\sum k^m \phi(x-k)$ 是一個首項係數為 1 的 m 階多項式，其中 $0 \leq m \leq p-1$ 。

prop:polysu

最後，我們探討一些凌波函數 $\psi(x)$ 與特徵函數 $m_0(\omega)$ 之間的關係。首先，我們套用 m_0 的定義方式，為 ψ 找一個特徵函數。由於 $\psi(x) = \sum d_k \phi(2x-k)$ ，所以

$$\hat{\psi}(\omega) = \frac{1}{2} \sum_k d_k e^{-ik\frac{\omega}{2}} \hat{\phi}\left(\frac{\omega}{2}\right)$$

若定義 $m_1(\omega) = \frac{1}{2} \sum d_k e^{-ik\omega}$ ，則 m_1 可謂是 ψ 的特徵函數。也就是：

$$\hat{\psi}(\omega) = m_1\left(\frac{\omega}{2}\right) \hat{\phi}\left(\frac{\omega}{2}\right) \quad (3.76)$$

eq:psihat

但是，由於 d_k 和 c_k 之間的關係*，其實 m_1 可以由 m_0 決定。我們採用 (3.20) 的定義方式，則

$$\begin{aligned} m_1(\omega) &= \frac{1}{2} \sum_k d_k e^{-ik\omega} = \frac{1}{2} \sum_k (-1)^k c_{1-k} e^{-ik\omega} \\ &= \frac{1}{2} \sum_k c_{1-k} e^{-ik(\omega+\pi)} = \frac{1}{2} \sum_n c_n e^{-i(1-n)(\omega+\pi)} \\ &= \frac{1}{2} \sum_n c_n e^{-i\omega} e^{-i\pi} e^{in(\omega+\pi)} = -e^{-i\omega} \frac{1}{2} \sum_k c_k e^{ik(\omega+\pi)} \end{aligned}$$

* 而這份關係也就是使得 $\psi(x)$ 成為正交凌波函數的原因。

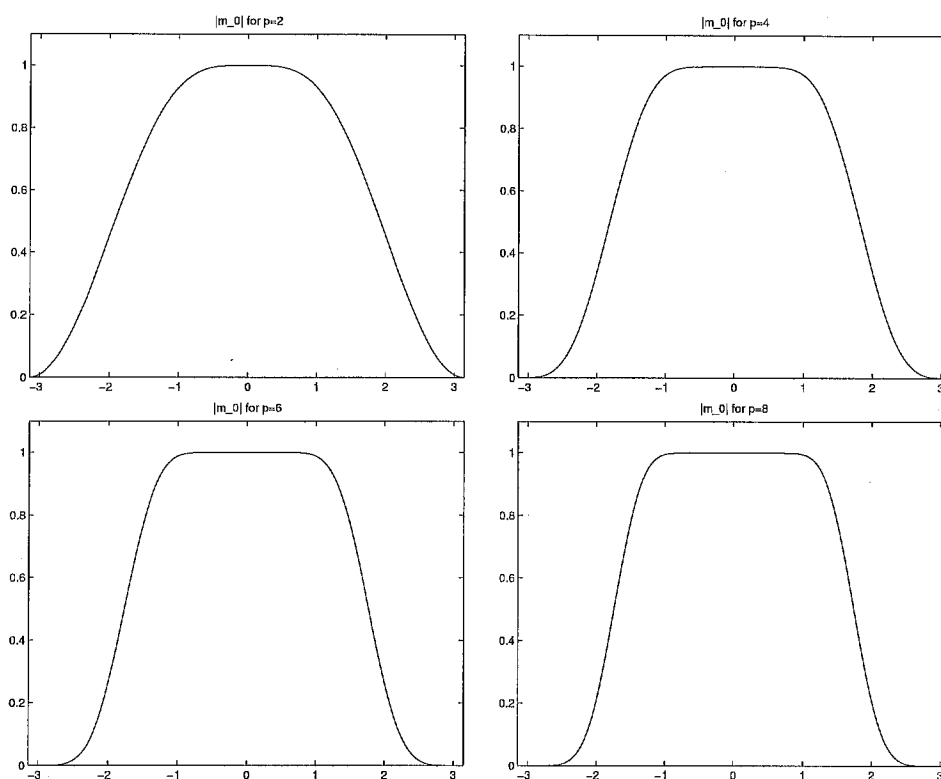


fig:mzeroabs 圖 28 $p = 2, 4, 6, 8$ 的 $|m_0(\omega)|$ 在 $[-\pi, \pi]$ 之內

將 m_0 代入，得知

$$m_1(\omega) = -e^{-i\omega} \overline{m_0(\omega + \pi)} \quad (3.77) \quad \text{eq:monedef}$$

讓我們看幾個簡單的推論。首先，根據 (3.67) 得到

$$|m_0(\omega)|^2 + |m_1(\omega)|^2 = 1 \quad (3.78) \quad \text{eq:monesum}$$

此外，由於 $m_1(0) = \overline{m_0(\pi)}$ ，根據 (3.72) 得知

$$m_1^{(s)}(0) = 0, \quad \text{for } 0 \leq s \leq p - 1$$

代回 (3.76) 得到

$$\hat{\psi}^{(m)}(0) = 0, \quad \text{for } 0 \leq m \leq p - 1 \quad (3.79) \quad \text{eq:psihatze}$$

也就是

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

從 (3.72) 中，我們看到

$$m_0(\pi) = m_0'(\pi) = \dots = m_0^{(p-1)}(\pi) = 0$$

故 $m_0(\omega)$ 在 $\omega = \pi$ 處有 p 重根。 $m_0(\omega)$ 是一個三角多項式，我們可以仿照多項式的特性，提出這 p 重根的因式。我們選擇以下的形式：

$$m_0(\omega) = \left(\frac{1 + e^{-i\omega}}{2} \right)^p \mathcal{L}(\omega) \quad (3.80) \quad \text{eq:mzerofac}$$

其中的商式 $\mathcal{L}(0) = 1$ 且 $\mathcal{L}(\pi) \neq 0$ 。根據我們對多項式的瞭解，一個根的重複次數越高，其函數圖形在根的附近就越平坦。這其實可以從函數的泰勒展開看出來。所以，當 p 越大， $|m_0(\omega)|$ 在 $\omega = \pi$ 附近應該越平坦。又由於 (3.78)，可見當 p 越大， $|m_0(\omega)|$ 在 $\omega = 0$ 附近也應該越平坦。而參數 p 代表著 V_j 所包含多項式的階數，或 $\{V_j\}$ 的逼近能力，或 $\psi(x)$ 的凌波係數消散速度。所以，這相當於，觀察 $|m_0(\omega)|$ 的函數圖形，也可以大致地得知上述資訊。當 $|m_0(\omega)|$ 在 $\omega = 0$ 附近越平坦， $\phi(x)$ 的逼近效果越好，而 $\psi(x)$ 的凌波係數消散越快。參考圖 28。

我們在 $[-\pi, \pi]$ 之間，以 $h = \frac{1}{1024}$ 做均勻分割得到 1025 個節點，然後在這些點上取 $m_0(\omega)$ 的值。我們提供一個 Matlab 格式的資料檔案

m0.mat

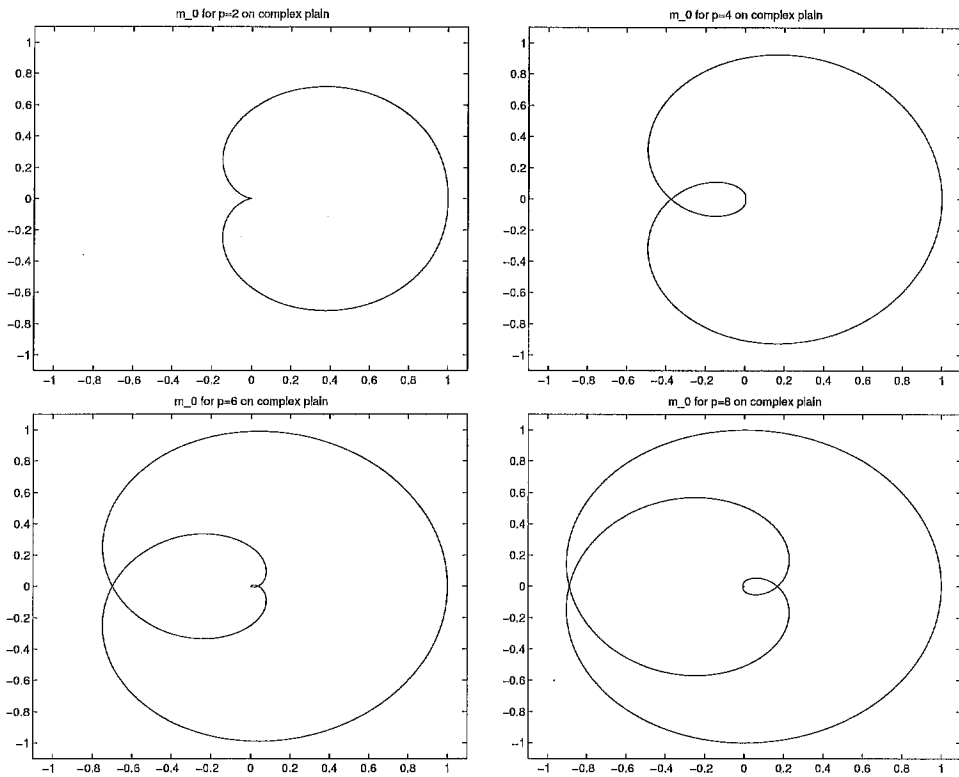


fig:mzerocmpx 圖 29 $p = 2, 4, 6, 8$ 的 $m_0(\omega)$ 在複數平面上的曲線

檔案內有十個向量 `m0_1`、`m0_9` 和 `m0_A`。它們分別是 $p = 1, \dots, 10$ 的特徵函數 $m_0(\omega)$ 在上述節點的函數值。這些函數值都是複數值。將這些函數在複數平面上畫出圖來 (例如說 `plot(m0_2);`)，會出現頗為有趣的花樣。如圖 29。

習題 3.31 試利用一種 Poisson 求和公式 (2.47) 來證明 (3.71)。

習題 3.32 試利用一種 Poisson 求和公式 (2.47) 來證明 (3.75)。並證明性質 3.6。

習題 3.33 證明 (3.73)。

習題 3.34 證明 (3.75) 和性質 3.6。

習題 3.35 若 $r(x)$ 是一個階數 $< p$ 的多項式，證明

$$\sum_k r(k)\phi(x-k)$$

也是一個階數 $< p$ 的多項式。

習題 3.36 證明 (3.79)。

習題 3.37 試由 $\hat{\psi}^{(m)}(0) = 0$ 證明 $\int x^m \psi(x) dx = 0$ 。

習題 3.38 試問 p 階樣條函數 $B_p(x)$ 的特徵函數是什麼？它所對應 (3.80) 中的 $\mathcal{L}(\omega)$ 是什麼？它在 $\omega = \pi$ 處有幾重根？

第四章

正交凌波函數 的訊號處理範例

在前三章中，我們探討了凌波函數的研究動機、準備了一些的數學基礎、並且推演了一些正交凌波函數的基本理論。在此後的兩章中，我們要展示有關自格函數或凌波函數的演算法。任何數學工具的應用，都是不可預設立場的：我們不能預先設定什麼數學就是做什麼用的。科學家、工程師與數學家的創意，使得數學工具往往出現意想不到的用途。何況，即使是現在，凌波函數的應用領域已經多得超過這本書的篇幅範圍。所以，我們在此並非列舉所有的應用，而是舉出幾個應用的範例、和它們的演算法。

我們將應用的範例分成兩大類。一類是訊號處理、另一類是數值計算。凌波函數在訊號處理中的角色，大致來說就是提供一個另類基底，使得訊號在此基底之下更容易突顯某些特色。因此，訊號處理之演算法，例如壓縮、除雜訊與辨識，在凌波基底之下能夠更有效率。所以，在訊號處理這個應用範疇中，凌波轉換是共同的核心技術。

在這一章中，我們先描述凌波轉換的演算法，並且以 Matlab 巨集語言為例，來實現這些演算法。我們會解釋個別遇到的 Matlab 指令，

讀者亦可參考附錄，以便對 Matlab 產生較為完整的認識。誠然，市面上或者網路上，已經有超過十種的凌波轉換軟體；有些是套裝軟體，有些是程式庫，也有些是以工具箱的形式呈現。而我們在本章中所開發的程式，並沒有比前述的軟體更為完備，它們的主要功能是教育性的：將凌波轉換的演算過程展現在讀者面前。期望的效果是，讀者在使用那些軟體時，不至於過分將之視為神秘的黑盒子；而且在將來有必要的時候，能夠開發自己的軟體。

4.1 離散凌波轉換 (DWT)

這一節和下一節的內容，分別對應第一章十六、十七節的內容。但是我們不再贅述發展的動機，也不重複推導的過程。有必要的時候，請讀者回顧第一章或第三章的相關部分。

已知

$$V_j = W_{j-1} \oplus W_{j-2} \oplus \cdots \oplus W_0 \oplus V_0$$

但是只要我們知道如何在兩層之間做凌波轉換 (分解與合成)，亦即

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

其他的步驟就可以遞迴執行。所謂分解就是將 V_j 中的函數，從 V_j 的基底變換到 V_{j-1} 和 W_{j-1} 的基底。反過來，所謂合成就是將 V_{j-1} 和 W_{j-1} 中的函數組合回到 V_j 。請參照圖 6 的圖示。

將一個一維訊號 (例如聲波) 以函數形式表達，寫成 $f(x)$ 。實際上我們只能處理有限範圍 (例如有限的時間) 內的有限能量訊號，所以在不失現實性的情況下，我們可以假設 $f \in L^2(\mathbb{R})$ 而且 $\text{supp } f = [a, b]$ 。當 $x \in [a, b]$ ， $f(x)$ 是一個無窮解析度的訊號。而我們想要的是一個有限解

析度的數位訊號。令 $\mathcal{P}_j : L^2(\mathbb{R}) \rightarrow V_j$ 是個正交投影。則

$$\mathcal{P}_j f(x) = \sum_k \langle f, \phi_{jk} \rangle \phi_{jk}(x) = \sum_k 2^j \langle f, \phi(2^j x - k) \rangle \phi(2^j x - k)$$

若

$$\mathcal{P}_j f(x) = \sum_k v_k \phi(2^j x - k) \quad (4.1)$$

eq:vkmodel

我們得到 $f(x)$ 在 V_j 的投影係數 v_k 。在離散凌波轉換的數學模型中，數位訊號乃是對某一個 $f(x)$ 和某一個 $j > 0$ ，如 (4.1) 所定義的一組數列 (v_k) 。

假設數位訊號是

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

爲了行文的方便，我們令 $N = 2^j$ ，則有

$$\begin{aligned} V_j &\ni \sum_{n=0}^{N-1} v_n \phi(2^j x - n) \\ &= \sum_{n=0}^{\frac{N}{2}-1} u_n \phi(2^{j-1} x - n) + \sum_{n=0}^{\frac{N}{2}-1} w_n \psi(2^{j-1} x - n) \in V_{j-1} \oplus W_{j-1} \end{aligned}$$

離散凌波轉換就是要將 $\{v_n\} \mapsto \{u_n, w_n\}$ 和 $\{u_n, w_n\} \mapsto \{v_n\}$ ；前者稱爲分解，後者稱爲合成。三組係數 v_n 、 u_n 與 w_n 之間的關係，已經在第三章七節中推導過。在此只覆述其結果。若有 $\phi(x) = \sum_k c_k \phi(2x - k)$ 和 $\psi(x) = \sum_k d_k \phi(2x - k)$ ，而且 $\phi(2x)$ 與 $\phi(2x - 1)$ 可以寫成 (3.24) 的形式，則由 (3.43) 及 (3.44)：

idx:dwt-de-

$$u_n = \frac{1}{2} \sum_k c_k v_{k+2n}, \quad w_n = \frac{1}{2} \sum_k d_k v_{k+2n} \quad (4.2)$$

eq:unwn

由 (3.47) :

$$\begin{aligned} v_{2n} &= \sum_k p_{2k} u_{n-k} + q_{2k} w_{n-k} \\ v_{2n+1} &= \sum_k p_{2k+1} u_{n-k} + q_{2k+1} w_{n-k} \end{aligned} \quad (4.3)$$

eq:vnagain

若 $\phi(x)$ 是正交自格函數， $\psi(x)$ 是所對應的正交凌波函數。由 (3.25) 知 $p_k = c_k$ ， $q_k = d_k$ ；由 (3.17) 知 $d_k = (-1)^k c_{2p-1-k}$ 。所以，分解與合成的計算全部可以由自格係數 c_k 決定。

在 (4.2) 中，可以假設足標 k 總是在 $0, \dots, 2p-1$ 的範圍內。但是，對於某些 n ，將使得 v_{k+2n} 的足標超出 $0, \dots, N-1$ 的範圍。同樣地，也有些 n 會使得 (4.3) 中 u_{n-k} 和 w_{n-k} 的足標超出 $0, \dots, \frac{N}{2}-1$ 的範圍。這些狀況在算法上會造成小小的麻煩，我們將在下一節討論。

習題 4.1 若 $p > 1$ ，試討論 (4.2) 中哪些 n 使得 $k+2n$ 超出 $0, \dots, N-1$ 的範圍？(4.3) 中哪些 n 使得 $n-k$ 超出 $0, \dots, \frac{N}{2}-1$ 的範圍？當 $p = 1$ 的時候，會不會發生這類的狀況？

4.2 離散凌波轉換 (DWT) 之矩陣表達

令數位訊號 (v_k) 是 \mathbb{R}^N 中的向量，而 $N = 2^j$ ，其中 $j \in \mathbb{N}$ 。亦即，令輸入向量為

$$\mathbf{v} = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{N-1} \end{pmatrix}_{N \times 1}$$

若分解 v_n 成低解析度係數 u_n 及細節部分之係數 w_n ，可以將

(u_n) 與 (w_n) 分別寫成向量形式：

$$\mathbf{u} = \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{\frac{N}{2}-1} \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{\frac{N}{2}-1} \end{pmatrix}$$

我們已經知道凌波分解的步驟相當於兩個矩陣的作用。一個稱作低頻濾波器 (low pass filter)，寫作 \mathcal{L} ；另一個稱作高頻濾波器 (high pass filter)，寫作 \mathcal{H} 。而

$$\mathbf{u} = \mathcal{L}\mathbf{v}, \quad \mathbf{w} = \mathcal{H}\mathbf{v}$$

\mathcal{L} 與 \mathcal{H} 的矩陣維度由輸入訊號的長度而定，但通常要求列數 (row) 是行數 (column) 的一半。也就是說， \mathbf{u} 和 \mathbf{w} 的長度都是 \mathbf{v} 長度的一半。

由 (4.2)， \mathcal{L} 和 \mathcal{H} 的前兩列如下：

$$\mathcal{L} = \frac{1}{2} \begin{pmatrix} c_0 & c_1 & c_2 & c_3 & \cdots \\ & & c_0 & c_1 & \cdots \end{pmatrix}$$

$$\mathcal{H} = \frac{1}{2} \begin{pmatrix} c_{2p-1} & -c_{2p-2} & c_{2p-3} & -c_{2p-4} & \cdots \\ & & c_{2p-1} & -c_{2p-2} & \cdots \end{pmatrix} \quad (4.4)$$

eq:matLH

將 \mathcal{L} 和 \mathcal{H} 上下堆在一起，就成爲一個 $N \times N$ 的方陣：

$$\mathcal{W} = \begin{pmatrix} \mathcal{L} \\ \mathcal{H} \end{pmatrix}_{N \times N} \quad (4.5)$$

eq:matW

則 \mathcal{W} 可以被視爲凌波轉換矩陣。因此得到分解公式

idx:wav-tra

$$\mathcal{W}\mathbf{v} = \begin{pmatrix} \mathbf{u} \\ \mathbf{w} \end{pmatrix} \quad (4.6)$$

eq:Wdecomp

當 $p > 1$ ， \mathcal{L} 和 \mathcal{H} 的最後幾列並不能按照這種每列向右邊移兩項的方式造成。原因是某些非零係數 c_k 會落到矩陣的邊外面。以 $p = 3$ 為例， \mathcal{L} 的最後三列之形式如下：

$$\begin{pmatrix} & & c_0 & c_1 & c_2 & c_3 & c_4 & c_5 \\ & & & & c_0 & c_1 & c_2 & c_3 \\ & & & & & & c_0 & c_1 \end{pmatrix} \begin{matrix} c_4 & c_5 \\ c_2 & c_3 & c_4 & c_5 \end{matrix}$$

這些落在矩陣邊外面的係數，對應於 (4.2) 中 $k+2n$ 超出 $0, \dots, N-1$ 範圍的 v_{k+2n} 。

我們介紹兩種處理的方法。一種處理方式，就是不理會那些落到矩陣外面的係數。這樣做的意義，就是假設 (4.2) 中的

$$v_{k+2n} = 0, \quad \text{for } k+2n \geq N$$

也就是說，假設數位訊號是

$$\mathbf{v} = (v_0, v_1, \dots, v_{N-1}, 0, 0, \dots, 0)$$

(尾端有 $2p - 2$ 個 0)。這樣做所引起的麻煩是

$$2\mathcal{W}^T \mathcal{W} = 2(\mathcal{L}^T \mathcal{L} + \mathcal{H}^T \mathcal{H}) = \begin{pmatrix} \mathbf{B} & \\ & \mathbf{I} \end{pmatrix} \neq \mathbf{I} \quad (4.7) \quad \text{eq:WB}$$

所以 $2\mathcal{W}^T \neq \mathcal{W}^{-1}$ 。也就是說，合成的步驟就不再是單純的轉置矩陣。其中 \mathbf{B} 是一個維度 $2p - 2$ 的方陣。矯正的方法就是在合成的步驟中，先乘上 $2\mathcal{W}^T$ ，再多乘上一個 \mathbf{B}^{-1} (如果它存在)。

另一個處理方式可以保持 $\mathcal{W}^{-1} = 2\mathcal{W}^T$ 。那就是週期化。當某一列的非零係數落到矩陣的邊外面時，把它們折回來放到此列向量的前面。以 $p = 3$ 為例， \mathcal{L} 的最後三列之形式如下：

$$\begin{pmatrix} & & & & c_0 & c_1 & c_2 & c_3 & c_4 & c_5 \\ c_4 & c_5 & & & & & c_0 & c_1 & c_2 & c_3 \\ c_2 & c_3 & c_4 & c_5 & & & & & c_0 & c_1 \end{pmatrix}$$

一般而言， \mathcal{L} 的最後 $p-1$ 列會出現折回來的係數。 \mathcal{H} 亦同。

這樣做的意義，就是假設 (4.2) 中的

$$v_{k+2n} = v_{k+2n-N}, \quad \text{for } k+2n \geq N$$

也就是說，假設數位訊號是

$$\mathbf{v} = (v_0, v_1, \dots, v_{N-1}, v_0, v_1, \dots, v_{2p-3})$$

(尾端有 $2p-2$ 個與前端相同的元素)。這麼一來，

$$\mathcal{W}(2\mathcal{W}^T) = \begin{pmatrix} 2\mathcal{L}\mathcal{L}^T & 2\mathcal{L}\mathcal{H}^T \\ 2\mathcal{H}\mathcal{L}^T & 2\mathcal{H}\mathcal{H}^T \end{pmatrix} = \mathbf{I} \quad (4.8) \quad \text{eq:WWI}$$

也就是說， $\mathcal{W}^{-1} = 2\mathcal{W}^T$ 。所以

$$\mathbf{v} = \mathcal{W}^{-1} \begin{pmatrix} \mathbf{u} \\ \mathbf{w} \end{pmatrix} = (2\mathcal{L}^T \quad 2\mathcal{H}^T) \begin{pmatrix} \mathbf{u} \\ \mathbf{w} \end{pmatrix}$$

因此得到合成公式

$$\mathcal{W}^{-1} \begin{pmatrix} \mathbf{u} \\ \mathbf{w} \end{pmatrix} = 2\mathcal{L}^T \mathbf{u} + 2\mathcal{H}^T \mathbf{w} = \mathbf{v} \quad (4.9) \quad \text{eq:Wreconst}$$

最後，我們提醒讀者，有些書上會令 (4.4) 中 \mathcal{L} 和 \mathcal{H} 的係數為 $\frac{1}{\sqrt{2}}$ 而不是 $\frac{1}{2}$ 。如此一來，就相當於令

$$\sqrt{2}\mathcal{W} = \begin{pmatrix} \sqrt{2}\mathcal{L} \\ \sqrt{2}\mathcal{H} \end{pmatrix}$$

為凌波轉換矩陣。這樣寫的好處是

$$(\sqrt{2}\mathcal{W})^T (\sqrt{2}\mathcal{W}) = 2\mathcal{W}^T \mathcal{W} = \mathbf{I}$$

因此 $(\sqrt{2}\mathcal{W})^{-1} = (\sqrt{2}\mathcal{W})^T$ 。所以整個理論看起來比較對稱。而我們則是為了突顯低頻濾波就是做加權平均的特性，故取 \mathcal{L} 和 \mathcal{H} 的係數為 $\frac{1}{2}$ 。所以 \mathcal{L} 每一列的係數和都是 1，相當於一組加權平均係數。

習題 4.2 求 (4.7) 中的 \mathbf{B} 方陣。

習題 4.3 驗證 (4.8)。

4.3 一維離散凌波轉換—演算法

在前一節的 (4.6) 和 (4.9) 中，我們用矩陣與向量形式表明了：凌波轉換在兩層解析度之間的分解與合成步驟。如果要在電腦中實作 (implement) 這套算法，並不需要真的組成轉換矩陣 \mathcal{W} 。回顧第一章十八節中，將 \mathcal{L} 與 \mathcal{H} 的列向量交叉排列的想法，這種做法將可以節省記憶體空間。而且，觀察 \mathcal{L} 與 \mathcal{H} 的元素，其實每列都一樣，只是位置向右移了兩個單位。所以， \mathcal{W} 和 $2\mathcal{W}^T$ 的矩陣向量乘法，其實可以用程式語言中的迭代迴圈來實現。根據這兩個基本想法，我們可以設計快速的凌波轉換演算法。除此之外，我們還需要一些附屬程式，幫我們做資料結構的工作。我們使用 Matlab 為實作的範例平台。一方面因為它是物件 (矩陣) 導向語言，可以省去許多資料結構上的細節；另一方面因為它是一個計算與繪圖的整合環境，可以省去在螢幕上繪圖的麻煩。

回顧第一章十八節中，將 \mathcal{L} 與 \mathcal{H} 的列向量交叉排列的想法。也就是說，定義

$$\widetilde{\mathcal{W}} = \frac{1}{2} \begin{pmatrix} c_0 & c_1 & c_2 & c_3 & \cdots \\ c_{2p-1} & -c_{2p-2} & c_{2p-3} & -c_{2p-4} & \cdots \\ & c_0 & c_1 & \cdots \\ & c_{2p-1} & -c_{2p-2} & \cdots \\ & & & \cdots \end{pmatrix}_{N \times N}$$

其中 $\widetilde{\mathcal{W}}$ 的最後 $p-1$ 列，用週期化的方式將超過邊界的係數折回到前面。

令 v 的長度是 $N = 2^j$ ，其中 $j \geq 1$ ，則 u 的元素恰是 $\widetilde{\mathcal{W}}v$ 的奇數項元素，而 w 的元素恰是 $\widetilde{\mathcal{W}}v$ 的偶數項元素。而且， u 和 w 的長度各是 $\frac{N}{2} = 2^{j-1}$ ，所以可以儲存在 v 原來的位置上。如此，我們可以結省計憶空間。這種新的資料結構方式，使得我們只需要一個額外的暫存空間。

但是，真正寫程式的時候，當然不需要真正組成矩陣 \widetilde{W} 。我們只需要知道 \mathcal{L} 和 \mathcal{H} 的列向量就夠了。現在，我們使用 Matlab 作為工具，發展一個一維的離散凌波轉換 (discrete wavelet transform) 程式，稱為 `dwt`。

這是一個 Matlab 的 function。假設輸入向量屬於某一個 V_J 。為了對應數學符號的方便，我們假設輸入向量 v 的長度是 $N = 2^J$ 。我們要把這個訊號分解到

$$W_{J-1} \oplus W_{J-2} \oplus \cdots \oplus W_{j_0} \oplus V_{j_0}$$

其中 j_0 是最低解析度空間。理論上 $j_0 \geq 0$ 即可，但是實際上， j_0 選得太小反而顯得沒有意義。 v 在每個子空間 W_j 中的元素個數是 2^j ，則分解後的資料量仍是

$$2^{J-1} + 2^{J-2} + \cdots + 2^{j_0} + 2^{j_0} = 2^J$$

在程式中我們用 `Jend-1` 來代表 j_0 。也就是說，我們對 $J \geq j \geq Jend$ 執行

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

的分解步驟。

先看 function `dwt` 的宣告。

```
function x = dwt(p,x,steps)
```

其中 p 是指 Daubechies 正交凌波函數的階數，有意義的範圍是 $1 \leq p \leq 14$ 。 x 是輸入向量，亦即數學符號的 v 。在 `dwt` 的運算過程中，我們要求 x 是個行向量 (column vector)。這樣，以後的運算符號才會一致。但是，如果使用者輸入的是序列 (列向量)，則輸出的答案也將

是序列。而參數 `steps` 用來決定 j_0 和凌波轉換的方向。當 $steps \geq 0$ 則 `dwt` 做分解，否則它做合成。而且一共做 $|steps|$ 次的分解或合成。亦即

$$j_0 = J - |steps|$$

如果這個 `steps` 太大以至於沒有分解的意義，則程式會自動取一個有意義的最大值。

目前，我們只討論一維凌波轉換程式。其實，只要經過一點點修改，`dwt` 就可以做二維轉換。這個留到第六節再說。

首先，是一些輸入資料的正確性檢驗。令 `ndim` 是 `x` 的列數，`mdim` 是 `x` 的行數。先確定 `x` 是個行向量。如果不是，將其轉置，並留下一個旗標 `transp`，以備程式結束之前使用。爲了容許輸入的 `p` 不是整數，我們將取它的四捨五入整數值。

```
[ndim,mdim] = size(x);
transp = 0;
if (ndim < mdim)
    x = x';
    [ndim,mdim] = size(x);
    transp = 1;
end
p = round(p);
```

然後 `dwt` 要檢查 `x` 是否真的是一個長度 $N = 2^J$ 的向量。由於 `dwt` 是一個教育與實驗性質的程式，我們不打算處理一般情形的 N 。所以如果 `ndim` 不是 2 的整數次方，我們就不繼續計算了。取 $N \geq ndim$ 是最靠近 `ndim` 的 2 的次方數。一般而言，取

$$J = \lceil \log_2 ndim \rceil$$

如果 $2^J > \text{ndim}$ ，就代表 x 的長度不對。此段程式中的某些部分有關二維轉換，留待第六節再說。若 steps 是個整數， $1 \leq p \leq 14$ 而且 x 符合規格，則送入副程式 `dwt1d` 處理。否則就不做任何計算，並寫出錯誤訊息。以下之程式看似冗長，其實只不過是——確認各參數的意義。

code:checkr

Matlab Code-1

```
J = ceil(log2(ndim));
N = 2^J;
J = ceil ( log2(mdim) );
M = 2^J;
if (floor(steps) ~= steps)
    disp('dwt: steps must be an integer')
elseif (ndim < N)
    disp ('dwt: size not power of 2. Not implemented.')
else
    if (p<1)
        disp ('dwt: p<1 not implemented.')
    elseif (p<15)
        if (mdim == 1)
            disp ('dwt: Doing 1-d Daubechies DWT...')
            x = dwt1d(p,x,steps,N);
        elseif (mdim == M)
            disp ('dwt: Doing 2-d Daubechies DWT...')
            x = dwt2d(p,x,steps,N,M);
        else
            disp ('dwt: not a vector or not power 2...')
```

```

        end;
    else
        disp ('dwt: p>15 not implemented.')
    end;
end;

```

其中 dwt: Doing ... 的訊息是爲了讓使用者知道程式進行的情形。如果您覺得這些訊息很討厭，可以刪除那兩行。

如果我們曾經在前面轉置了 x ，最後要將它轉回來。

```

if (transp==1)
    x = x';
end;

```

這就結束了 dwt 這個程式的介紹。

如果讀者有一組訊號想要利用 dwt 做凌波轉換，但是它的長度 N 不是 2 的次方。那麼您可以只截取其中的一段，使其長度較短但符合規格。您也可以利用一個小程式，將此訊號鏡射爲長度 $2^{\lceil \log_2 N \rceil}$ 的向量。這個程式是 reflect(x)。例如

$$\text{reflect}([1 \ 2 \ 3 \ 4 \ 5]) = ([1 \ 2 \ 3 \ 4 \ 5 \ 5 \ 4 \ 3])$$

以下是 reflect 的內容。

code:reflect

Matlab Code-2

```

function x = reflect(x)
[n,m] = size(x);
N = 2^(ceil(log2(n)));
M = 2^(ceil(log2(m)));

```

```

k = N-n;
if (k>0) x=[x; x(n:-1:n-k+1,:)]; end;
k = M-m;
if (k>0) x=[x x(:,m:-1:m-k+1)]; end;

```

接下來，我們描述 `dwt1d(p,x,steps,N)`。因為這個程式的輸入值已經被 `dwt` 檢查過，所以我們假設輸入的 `x` 符合規格：它是一個長度 $N = 2^J$ 的向量。`dwt1d` 首先根據 `p` 從子程式 `getdaubc` 取得 c_k ，存在序列 `coef` 中。其中 $\text{coef}(k+1) = c_k$ ， $0 \leq k \leq 2p - 1$ 。我們已經將這些 c_k 存進一個 Matlab 資料檔案

daub_coef.mat

它們是以雙精度浮點數的格式儲存。這些值是用四倍精度在 VAX 6510 上計算的。然後它準備一些參數：令 $\phi(x)$ 和 $\psi(x)$ 的函蓋是 $[0, R] := [0, 2p - 1]$ ，而非零自格係數的個數是 $2p = R + 1 = NC$ 。

```

coef = getdaubc(p)';
R = 2*p - 1;
NC = R+1;
J = log2(N);

```

副程式 `getdaubc` 只是提供 `p` 階正交自格函數的自格係數。若 $p \leq 0$ 或 $p \geq 15$ ，則回應錯誤訊息，並將係數全設成 0。

code:getdai

Matlab Code-3

```

function c = getdaubc(p)
load daub_coef
if ((0<p) & (p<15))

```

```

    c = C(1:2*p,p);
else
    disp('getdaubc: Not implemented.');
```

```

    c = zeros(2*p,1);
end
```

在副程式 `getdaubc` 中開啓的資料檔 `daub_coef.mat`，當副程式執行結束之後，就會被清除，所以不佔記憶體。在 `dwt` 中我們已經把 `p` 變成整數。一般而言，若 `p` 不是整數，Matlab 在將 `p` 用做矩陣足標的時候，會自動做四捨五入。

接著檢查 `steps`。若它是正數，代表要做分解，若是負數，代表要做合成。我們取 `steps` 的絕對值作為分解或合成的次數，用 `case = ±1` 作為分解或合成的分辨。

```

if (steps > 0)
    case = 1;
else
    case = -1;
    steps = -steps;
end;
```

`dwt1d` 的分解步驟將由 $j = J$ 依次遞減做到 $j = J_{end}$ ， J_{end} 層的分解結果就是 $J_{end}-1 = j_0$ 層。回顧週期化的矩陣設計，使得 \mathcal{L} 和 \mathcal{H} 的最後一列有 $2p - 2$ 個係數要折到前面。當 x 已經被分解到 V_j 而 $2^j < 2p - 2$ 之後，就不容易做這個週期化的動作。其實 $2p - 2$ 已經夠小了，做到這裡停下來並沒有損失什麼。為了方便起見，我們就取最小可能的 j_0 是

$$\min j_0 = \lceil \log_2 2p \rceil \quad (4.10) \quad \begin{array}{l} \text{idx:minj0} \\ \text{eq:minj} \end{array}$$

所以 $J_{\text{end}} = J - \text{steps} + 1$ 而且 $0 \leq \text{steps} \leq J - \min j_0$ 。如果使用者要求的分解或合成次數 steps 太多或太少，我們就取其最小值或最大值。

```
steps = min(steps, J-ceil(log2(2*p)));
Jend = J - steps + 1;
```

如果 $\text{case} = 1$ ，就是要做分解。我們將低頻與高頻分解係數，也就是 c_k 與 d_k ，分別儲存在長度 $2p$ 的序列 DL 與 DH 中。然後寫出一句訊息，告訴使用者，實際上做了幾次的分解步驟。

```
if (case == 1)
    DL = coef / 2;
    DH = (coef(NC:-1:1) .* (-1).^ (0:R)) / 2;
    disp(['dwt1d: decomposing ' int2str(steps) ' steps']);
```

只有 DH 那一行需要解釋。其中 $\text{coef}(\text{NC}:-1:1)$ 就是將自格係數 c_k 的排列順序反過來，產生 $(c_{2p-1}, c_{2p-2}, \dots, c_0)$ 。 $(-1).^ (0:R)$ 就是製造 $(1, -1, 1, -1, \dots, -1)$ 這樣的序列。而 $*$ 的結果就是將上述兩個序列對應相乘，得到 $(c_{2p-1}, -c_{2p-2}, \dots, -c_0)$ ，也就是 d_k (見 (3.17))。

接著我們就一層一層地做

$$V_j = V_{j-1} \oplus W_{j-1}, \quad \text{for } j = J, J-1, \dots, J_{\text{end}}$$

我們令 x 向量中的 x_1, \dots, x_{2^j} 儲存的是在 V_j 中的投影係數， $x_{2^j+1}, \dots, x_{2^{j+1}}$ 儲存的是在 W_j 中的投影係數， $x_{2^{j+1}+1}, \dots, x_{2^{j+2}}$ 存的是在 W_{j+1} 中的投影係數，依此類推到 W_{J-1} 。

對每一個 j ，我們要取出 x 的前面 2^j 個元素做分解。 x 在 $2^j + 1$ 項以後的元素都是已經分解完成的元素，不必再去動它們。子程式 `pwd`

將訊號在邊界附近做週期化，然後執行核心的計算步驟。pwd 的輸出仍是長度 2^j 的向量，其奇數項是 V_{j-1} 中的低解析投影，而偶數項是 W_{j-1} 的細節部份。我們利用一個臨時變數 w ，將計算結果的奇數項放入 x 的前 2^{j-1} 個位置，將偶數項放入 x 的次 2^{j-1} 個位置。這樣，就完成了一個從 V_j 到 $V_{j-1} \oplus W_{j-1}$ 的分解步驟。

code:pwdone

 Matlab Code-4

```
for j = J:-1:Jend,
    m = 2^j;
    x(1:m,:) = pwd(p,x(1:m,:),DL,DH);
    x(1:m,:) = [x(1:2:(m-1),:); x(2:2:m,:)];
end;
```

在上面那個程式片段中，以及稍後會看到的許多程式中，我們用了類似 $x(1:m,:)$ 這樣的語法。這個語法的意思是，取出 x 中從第 1 列到第 m 列的所有元素，不管每一列有多少行。如果 x 是一個向量，當然每列只有一行。但是，我們在此將程式寫得較一般化，使得 x 也可以是個矩陣。這樣做的目的，等到我們在第六節中描述二維離散凌波轉換的時候，就會明白。

我們曾經闡述，根據週期化的想法，把矩陣 \mathcal{L} 和 \mathcal{H} 的最後幾列週期化。其實，這等於將輸入向量 x 的最前面 $2p-2$ 項抄到 x 的後面，然後和 \mathcal{L} 與 \mathcal{H} 的列向量做內積。這樣的處理，使得濾波係數 DL 和 DH 完全不需改變。pwd 就是先做分解前的週期化，然後呼叫 wf 做大量的實際計算工作，做完之後再截去尾巴上爲了週期化而加長的幾項。

idx:periodize

code:pwd

 Matlab Code-5

```
function x = pwd(p,x,L,H)
[m,n] = size(x);
```

```
x(m+1:m+2*p-2, :) = x(1:2*p-2, :);
x = wf(x,m,L,H);
x = x(1:m, :);
```

真正的大量計算工作由 wf 完成。它接受 x 為輸入向量，以 L 和 H 為低頻與高頻濾波係數 (列向量格式)，作 m 次濾波。

code:wf

Matlab Code-6

```
function x = wf(x,m,L,H)
R = length(L)-1;
for i = 1:2:m,
    t1 = L*x(i:i+R, :);
    t2 = H*x(i:i+R, :);
    x(i,:) = t1;
    x(i+1,:) = t2;
end;
```

子程式 wf 是真正的計算所在。它也是耗費最多計算時間的部分。所以，如果要追求更快的速度，在此外接 C 或其他語言所編譯的程式，將會得到速度上最大的改進*。或者，使用 Matlab Compiler 也應該很有幫助，但是我沒有使用經驗。請注意，如果有同檔名的 M 檔案和 MEX 檔案存在於同一個資料夾內，則 MEX 檔案會優先執行。在磁碟片中，附了三種 wf 程式

wf.m wf.f wf.c

* Matlab 可以呼叫一個外部程式，該程式可以是由 ANSI C 或 FORTRAN 寫成的。詳情請看 Matlab 使用手冊，關於 CMEX 之功能。

依序

分別是 Matlab 程式、FORTRAN 程式和 C 程式，請參考。

~~這些程式~~

pwd 的輸出結果仍是長度 2^j 的向量。但是其低頻係數與高頻係數乃是交叉出現。為了方便下一個迴圈的計算，我們將它重新排好。亦即

$$\begin{pmatrix} u_1 \\ w_1 \\ u_2 \\ w_2 \\ \vdots \\ \vdots \\ \vdots \end{pmatrix} \mapsto \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ w_1 \\ w_2 \\ \vdots \\ \vdots \end{pmatrix}$$

至此，我們完成了 dwt1d 分解步驟的描述。除了子程式 wf 之外，其他的步驟全都是在做資料確認和資料結構的工作。在 Matlab 中，有關矩陣與向量的資料結構管理，都非常有效率。為了測試資料結構所佔去的時間，我取 x 的長度為 2048， $p = 4$ ，執行 `dwt(4,x,11)`。在我的工作站上，如果執行 wf 子程式，費時 1.65 秒，如果略過 wf 子程式，費時 0.10 秒。可見處理資料結構的時間大約佔去整個計算時間的 6%。有必要的時候，讀者應可改寫 Code-4 和 Code-5 的流程，使得資料結構所佔的時間更少。但是，提高 wf 的速度，對整體效率的提升才是最有效的。

以下我們介紹 dwt1d 的合成部分。

如果 $case = -1$ ，就是要做合成。其實 $case$ 只有兩種可能的值： ± 1 ，所以在程式中只要寫 `else` 就可以了。我們將低頻與高頻的合成係數 p_k 與 q_k 交叉儲存在長度 $2p$ 的序列 RE 與 RO 中。然後寫出一句訊息，告訴使用者，實際上做了幾次的合成步驟。參照 (4.3)，RE 將由 u_k 和 w_k 去合成 v_{2n} (偶數項)，而 RO 將由 u_k 和 w_k 去合成 v_{2n+1} (奇數項)。RE 與 RO 其實是取自 $2\widetilde{W}^T$ 的列向量。讀者只要比較細心地觀

察，就可以得知 RE 與 RO 中的元素和 p_k 與 q_k 的關係，然後再推得它們和 c_k 的關係。

code:RE0

Matlab Code-7

```

else
    a=coef(1:2:NC); b=coef(2:2:NC);
    RE(1:2:NC)=a(p:-1:1); RE(2:2:NC)=b;
    RO(1:2:NC)=b(p:-1:1); RO(2:2:NC)=-a;
    clear a b
    disp(['dwt1d: reconstructing ' int2str(steps) ...]);

```

接著我們就一層一層地做

$$V_{j-1} \oplus W_{j-1} = V_j, \quad \text{for } j = J_{\text{end}}, \dots, J-1, J$$

我們假設 x 向量中 x_1, \dots, x_{2^j} 儲存的是在 V_j 中的投影係數， $x_{2^j+1}, \dots, x_{2^{j+1}}$ 儲存的是在 W_j 中的投影係數， $x_{2^{j+1}+1}, \dots, x_{2^{j+2}}$ 存的是在 W_{j+1} 中的投影係數，依此類推到 W_{J-1} 。

做合成時，就要將分解的步驟逆其道而行。對每一個 j ，我們要取出 x 的前面 2^j 個元素做合成。其餘的元素還沒有要做合成，所以留著不動。這 2^j 個元素的前面一半是 V_{j-1} 中的係數，後面一半是 W_{j-1} 中的係數。首先要把 V_{j-1} 和 W_{j-1} 中的係數交叉重排，亦即

$$\begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ w_1 \\ w_2 \\ \vdots \end{pmatrix} \mapsto \begin{pmatrix} u_1 \\ w_1 \\ u_2 \\ w_2 \\ \vdots \\ \vdots \end{pmatrix}$$

這個步驟的效率極為重要。在 Matlab 中比較好寫，這由子程式 `interlace` 來完成。若使用一般的程式語言，則要考慮其他的實作方法。重排後的 x 交由 `pwc` 將其週期化，再呼叫 `wf` 做實際的計算。這樣，就完成了一個從 $V_{j-1} \oplus W_{j-1}$ 到 V_j 的合成步驟。

code:pwcone

Matlab Code-8

```

for j = Jend:J,
    m=2^j;
    x(1:m,:) = interlace(x(1:m,:));
    x(1:m,:) = pwc(p,x(1:m,:),RL,RH);
end;
end;

```

這就結束了 `dwt1d` 這個程式的介紹。

以下我們再交代兩個子程式的內容。先說 `interlace`，它不管輸入的是矩陣還是向量，只檢查輸入物件是否有偶數列。如果是，則把此向量或矩陣的上面一半依序放到奇數列，下面一半依序放到偶數列。如果不是偶數列，則給一個錯誤訊息。

code:interlac

Matlab Code-9

```

function w = interlace(v)
[n,m] = size(v);
ifodd = 'n - 2*floor(n/2)';
if eval(ifodd) == 0
    w(1:2:n-1,:) = v(1:n/2,:);
    w(2:2:n,:) = v(n/2+1:n,:);
else
    disp('interlace: Cannot interlace ...')

```

```
w = v;
end
```

而 `pwc` 和 `pwd` 所作的工作很像，就是把週期化的結構準備好，再把實際的計算工作交給 `wf`。注意，只要資料給得對，分解和合成的計算步驟都是由 `wf` 完成的。在 `wf` 之後，要把最後的幾個因為週期化加長的項移到前面；它們才是正確的值。

code:pwc

Matlab Code-10

```
function x = pwc(p,x,L,H)
[n,m] = size(x);
x(n+1:n+2*p-2,:) = x(1:2*p-2,:);
x = wf(x,n,L,H);
z=x(n-2*p+3:n,:);
x = [z; x(1:n-2*p+2,:)];
```

我們可以用 Matlab 中的 `flops` 指令來測量計算複雜度。Matlab 把一個實浮點數的相加或相乘各算一個 flop。這個定義和某些數值分析的書上寫的不一樣。下指令 `flops(0)` 將計數器歸零，執行完計算之後，再輸入 `flops` 就獲知整個計算的 `flops` 總數。以分解過程為測量標準，令輸入向量的長度分別是

2048, 1024, 512, 256, 128

當 $p = 4$ 時，做最多可能的分解步驟 (就是取 $j_0 = \min j_0$)，Matlab 計算的 `flops` 總數分別是

71538, 35686, 17754, 8782, 4290

而 $p = 8$ 時，flops 總數分別是

$$136302, \quad 67682, \quad 33366, \quad 16202, \quad 7614$$

可見計算複雜度與 pN 呈線性關係。平均而言，這個程式的計算複雜度大約是

$$8.35pN$$

因為濾波係數的長度是 $2p$ ，依照 Matlab 對 flop 的定義，理論上的複雜度是應該是 $8pN$ 。實際測量的結果很接近理論估計值。

不論分解還是合成，主要的計算工作都是由 wf 完成。所以兩者的計算量差不多是一樣的。例如我取 x 的長度是 $N = 2048$ ，其元素就是從 1 到 2048 的整數。執行 $y = \text{dwt}(4, x, 11)$ ；所耗費的計算量是 71538 flops，執行 $z = \text{dwt}(4, y, -11)$ ；所耗費的計算量是 71594 flops。 x 和 z 理論上應該相等，實際上它們之間有 10^{-14} 的相對誤差，這是有限精度計算所造成的 round-off 誤差。

一般而言，我們必然會面臨 $\log_2 N \notin \mathbb{N}$ 的情形。關於這方面的考慮，我們已經在第一章十八節中討論過（在 (1.64) 的附近）。因此不再贅述。總之，此後我們只考慮 N 是 2^J 的情形，其中 $J \in \mathbb{N}$ 。

習題 4.4 根據 (4.3) 證明 Code-7 中的 RE 和 RO 是正確的。

習題 4.5 令 $v \in V_J$ 而 v 的向量長度是 $N = 2^J$ 。驗證將 v 分解到 $V_0 \oplus W_0 \oplus W_1 \oplus \dots \oplus W_{J-1}$ 的計算複雜度應是 $8pN$ 。

4.4 一維離散凌波轉換—範例

概括來說，舉凡利用凌波轉換所做的應用，都是根據兩個事實：(一) 凌波轉換的快速演算法，(二) 凌波係數消散定理。由前一節已知，如

果輸入訊號的長度是 N ，凌波轉換的計算複雜度*是 $O(N)$ 。也就是說，當 N 夠大的時候，計算複雜度 $\approx CN$ 。我們知道其中 $C \approx 8p$ 。而凌波係數消散定理，基本上就是說 $f(x)$ 做了凌波轉換之後，對應於 $\psi_{jk}(x)$ 的係數只和 $f(x)$ 在 $\text{supp} \psi_{jk}$ 區間內的平滑度有關；函數越平滑，係數就越小。所以，一方面我們可以在平滑的部分得到高比例的資料壓縮，另一方面我們可以利用係數的大小來做影像辨識類的應用。

在這一節裡面，我們先闡述因週期化而造成的邊界問題。然後非常簡要地介紹 (一) 高頻或細節係數的特性，以及它們的應用；(二) 一種消除雜訊的做法。

我們先看一個最簡單的轉換。由於 $\psi(x) \perp 1$ ，所以高頻濾波係數也正交於常數訊號。也就是說，如果輸入的向量全是 1，則轉換後 W_j 中的分量全是 0。因為低頻濾波相當於取平均，1 的平均仍是 1，所以 V_{j_0} 中的分量仍然是 1。讀者可以實驗

```
dwt(2,ones(1,16),3)
```

亦即輸入一個 1×16 的序列，其元素全是 1，用 $p = 2$ 的凌波函數做至多 3 次轉換。因此 $N = 16$ 而 $J = 4$ 。但因為 $\min j_0 = 2$ ，所以 dwt 只做兩次轉換，得到答案

1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0

其中前四個 1 是在 V_2 中的係數，次四個 0 是在 W_2 中的係數，後面的八個 0 是在 W_3 中的係數。

當 $p \geq 2$ 時， $\psi(x) \perp x$ 。所以高頻濾波係數也應該正交於由一階多項式所產生的訊號。例如令 $f(x) = x$ ，在 $[0, 1]$ 之間每隔 $\frac{1}{1023}$ 取一個樣

* 所謂計算複雜度就是所需的計算量的意思。在此我們一律以 Matlab 所定義的 flop 為測量單位。

2627

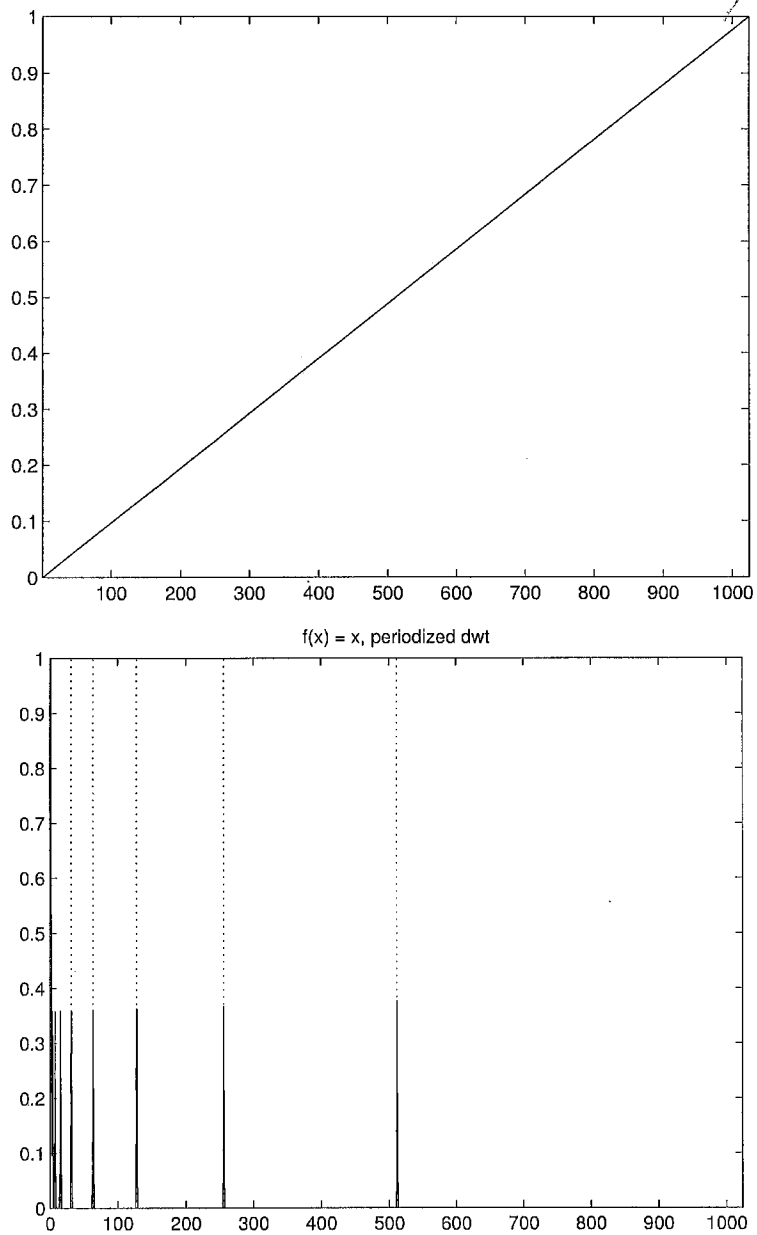


fig:testx. 圖 30 $f(x) = x$ 的原始訊號與轉換後訊號

本點，組成向量 x 。則 x 的長度是 $N = 2^{10}$ ，我們假設 $x \in V_{10}$ 。我們直接用 `plot(x)` 將此向量畫出來，得到一張很單調的圖片，在圖 30 的上方。將 x 做 $p = 2$ 的凌波轉換得到轉換後訊號 y ：

```
y = dwt(2,x,10);
```

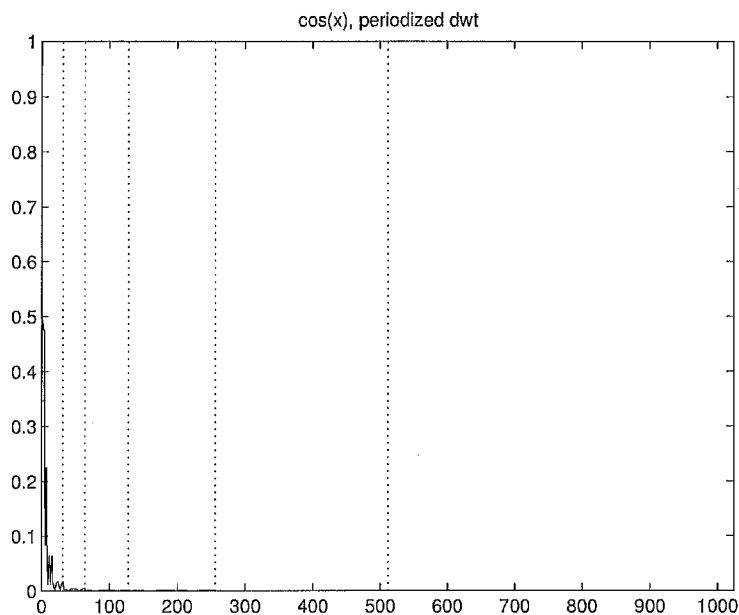
我們將 y 的絕對值畫在圖 30 的下方。直接用 y 的向量足標做橫軸坐標。圖中的虛線提示多層解析空間的區隔。亦即，從右邊數起，第一條虛線的右邊有 512 筆資料，是在 W_9 中的係數，第一條虛線與第二條虛線之間的 256 筆資料，是在 W_8 中的係數，依此類推，最左邊比較大的資料，是在 V_2 中的係數。

從這個例子中，我們看到 W_j 中的大部分係數的確是 0，如理論所預期。但是，總有一小部分的係數 $\neq 0$ 。這就是因為我們在寫程式的時候，在邊界外做週期化所造成的。當我們把 $f(x) = x$ 、 $x \in [0, 1]$ 週期化成爲 1-週期函數之後，它就不再是個連續函數：它在 $x \in \mathbb{Z}$ 的時候都是不連續的。而凌波係數在函數不連續的地方消散得很慢，所以在圖 30 的下圖中，我們看到每一段 W_j 訊號，都在靠邊界的地方有大的係數。這些大係數的個數與 j 無關，在每一層 W_j 中，都大約有 $2p$ 個靠近邊界的元素比較大。因為 $2p$ 相對於 N 而言是很小的量。所以在實際上它們並不會影響壓縮率。有一種有限區間內的凌波函數，可以排除這個現象。我們暫時不去探討那種凌波函數。

idx:periodi

如果原始訊號是一個平滑的週期性訊號，例如從 $f(x) = \cos x$ 、 $x \in [0, 2\pi]$ 中取樣出來的 1024 筆資料，在週期化之後還是平滑的函數。那麼前述的邊界現象就不會發生。請看圖 31。所有的高頻係數幾乎都是 0。

從圖 30 和圖 31 中，我們可以看出來，轉換後的訊號有相當多幾乎是 0 的資料。這就增加了訊號的同質性。我們可以不必真的儲存這

fig:testcos 圖 31 $\cos x$ 的轉換後訊號

些幾乎是 0 的資料，而利用資料結構的技巧，來記錄它們的存在。這就是資料壓縮 (*data compression*) 的基本想法。但是我們不打算在這本書裡詳述資料壓縮這門學問。

idx:compress

接下來，我們設計一個測試訊號。這份數位訊號共有 1024 筆資料，它是從一個單變數函數取樣得來。我們用這個測試訊號，或者說這個測試函數，來呈現凌波係數與函數平滑度的關係。此測試函數乃是由四片函數銜接起來，定義在 $[0, 4]$ 之間。在 $[0, 1]$ 之間，它是個四階多項式。在 $[1, 2]$ 之間，它是個 $\sin \frac{1}{x}$ 形式的高振盪函數 ($x \neq 0$)。其實是平移放大以下函數到 $[1, 2]$ 區間：

$$\frac{7}{8} - \frac{1}{8} \sin \frac{1}{x}, \quad x \in \left[-\frac{2}{5\pi}, \frac{1}{50}\right]$$

我們設計它在 $x = 1$ 處是一次可微連續，所以在 $x = 1$ 是平滑地銜接。

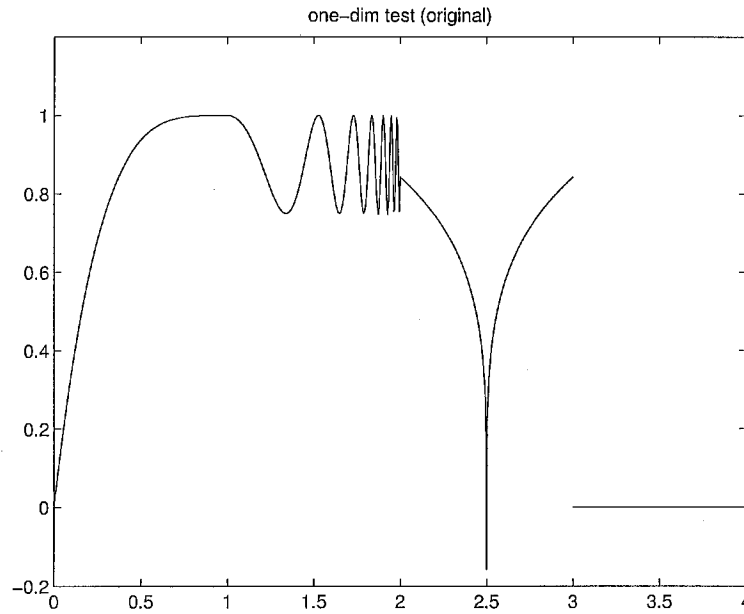


fig:testoned 圖 32 一維測試函數

在 $[2, 3]$ 之間，它是個 $|\sqrt[5]{x}|$ 形式的函數，我們設計它在 $x = 2.5$ 處出現尖銳的圖形，稱為 cusp；而在 $x = 2$ 處是連續的。在 $[3, 4]$ 之間，它是個常數函數 0。所以 $x = 3$ 處是不連續的。但週期化之後，它在 $x = 0$ 或 $x = 4$ 處是連續函數。請參照圖 32。製造這個測試函數的 Matlab 程式，和它的數據資料，分別放在以下檔案內。

```
test1d.m test1d.mat
```

這個測試函數在區間 $(0, 1)$ 、 $(1, 2)$ 、 $(2, \frac{5}{2})$ 、 $(\frac{5}{2}, 3)$ 、 $(3, 4)$ 之間都是平滑的。它在 $x = 1$ 、 $x = 2$ 、 $x = \frac{5}{2}$ 處連續但是分別有不同程度的不平滑 (分別是 C^1 、 $C^{0,1}$ 和 $C^{0, \frac{1}{5}}$)。而且在 $x = 2$ 的附近有高頻率的振盪。它在 $x = 3$ 處有一個不連續點。

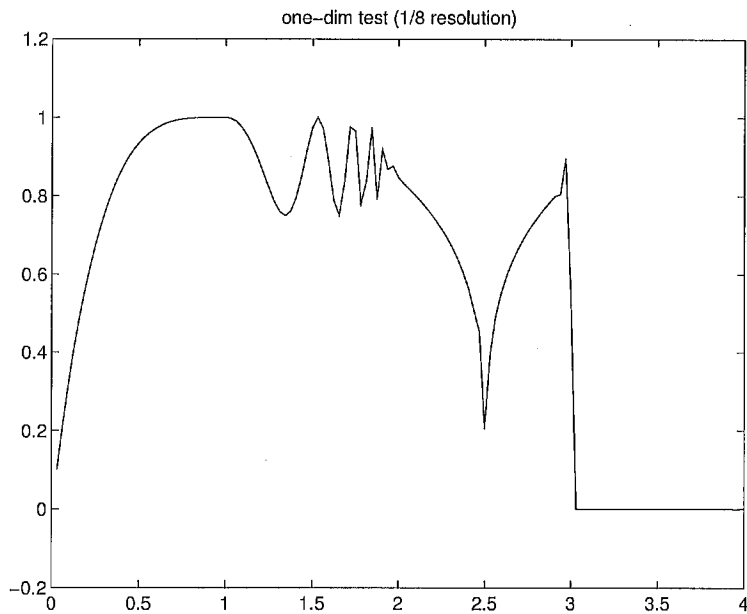


fig:testonedg 圖 33 一維測試訊號的 $\frac{1}{8}$ 解析度

令 y 代表圖 32 中的測試訊號。因為 y 共有 1024 筆資料，我們假設 $y \in V_{10}$ 。以 $p = 3$ 的凌波函數做三次分解：

```
z=dwt(3,y,3);
```

則 z 的前 128 個元素是 V_7 中的係數。將這些資料畫在圖 33。 V_7 的解析度是 V_{10} 的 $\frac{1}{8}$ 。讀者若將圖 33 與原訊號比較，會發現低解析度的訊號大致保持了原訊號的外觀。在越不平滑的地方，會有越大的失真。可見凌波轉換也提供了一個降低解析度的有力工具。

在圖 34 的上圖中，我們畫出凌波轉換後的一維測試訊號之絕對值。觀察在每一個解析度的 W_j 中，都明顯地有三個高突。這三個高突分別對應測試函數在 $x = 2$ 、 $x = \frac{5}{2}$ 和 $x = 3$ 處的不平滑。但是，若仔細觀察這些高突在不同解析度下的改變，則可以更清楚地分辨這些點

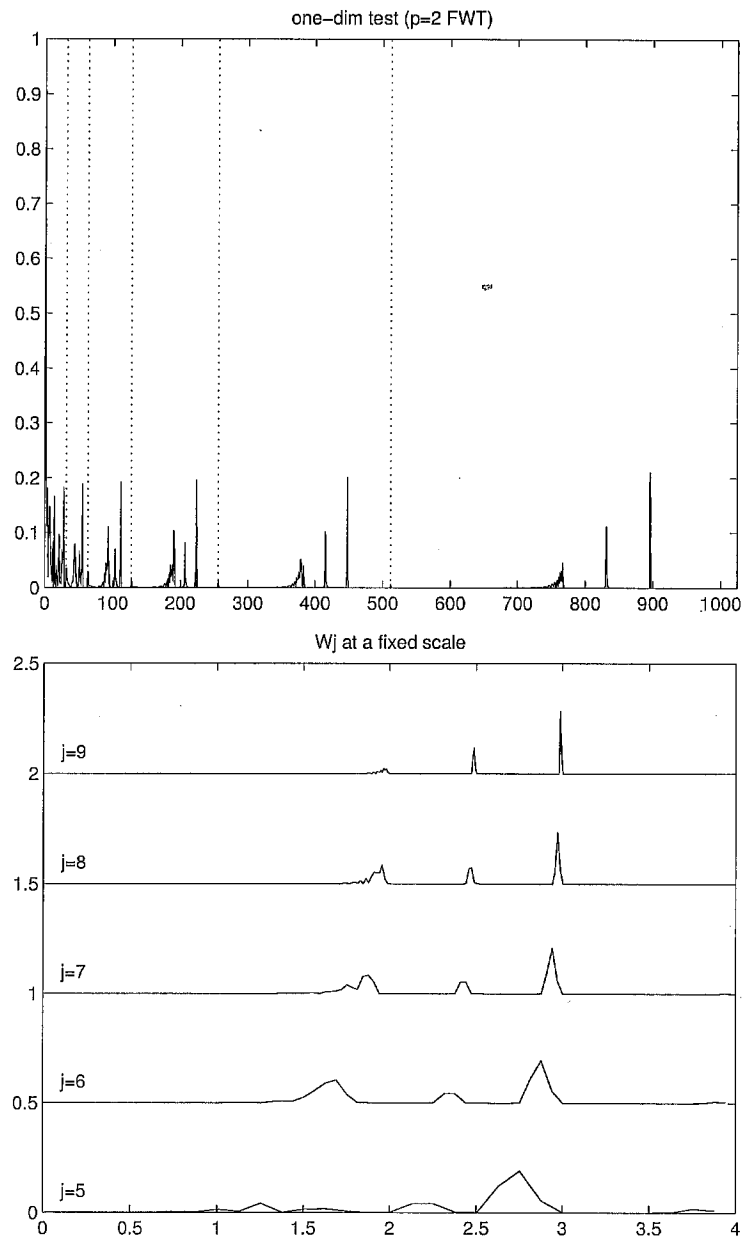


fig:testonedw 圖 34 凌波轉換後的一維測試訊號

的不平滑程度。

在圖 34 的下圖中，我們換一種方式來表達凌波係數。我們將橫軸的尺度一律固定在原函數的尺度 $[0, 4]$ 。然後將 W_j 中的訊號 (取絕對值後) 還原到 $[0, 4]$ 坐標上畫圖。這樣就可以觀察，在每一個不平滑處，各解析層 W_j 中高頻係數的變化情形。我們看到，當解析度 $j > 5$ 時，凌波係數都不會在 $x = 1$ 處有大的值。而在 $x = 2$ 附近，低解析度的凌波係數較大，但是隨著解析度的提高，就好像它比較認清了真相，發現 $x = 2$ 附近其實並不太壞，所以凌波係數就變小了。只有 $x = \frac{5}{2}$ 的 cusp 和 $x = 3$ 的不連續點，才真正在每一層解析度造成大的凌波係數。但是 cusp 和不連續點的不平滑程度不同，因此也造成凌波係數大小的不同。利用不同解析度中凌波係數的不同變化情形，我們可以利用凌波轉換後的資料，來判斷原訊號在哪些地方不平滑，而且還可以估計它們不平滑的程度。這一個性質，是 pattern recognition 和 edge detection 這一類應用的基礎。

最後，我們來看一個利用凌波轉換來除雜訊 (denoising) 的方法。這是 Donoho 發明的凌波縮減法 (wavelets shrinkage)。請參照圖 35。第一列的左圖，是一個單純的 sin 波形訊號。它的右邊是凌波轉換後的訊號，我們看到高頻部分幾乎都是 0。第二列的左圖，是加上了雜訊的 sin 波形訊號，我們用平均為 0 的隨機亂數來模擬雜訊。它的右邊是凌波轉換後的訊號，我們看到雜訊使得高頻部分有小幅振動。所謂凌波縮減，就是決定一個 $\epsilon > 0$ ，然後對凌波係數做以下計算：

$$(w_{jk})_{\pm\epsilon} = \begin{cases} w_{jk} - \epsilon & \text{if } w_{jk} > \epsilon \\ w_{jk} + \epsilon & \text{if } w_{jk} < -\epsilon \\ 0 & \text{if } |w_{jk}| \leq \epsilon \end{cases} \quad (4.11)$$

就好像我們畫兩條水平線 $y = \epsilon$ 和 $y = -\epsilon$ ，然後將兩條線夾到橫軸上，把兩條線之間的量都清除掉。這就是圖 35 第三列的左圖所要表現

idx:denoise

idx:shrink

eq:shrink

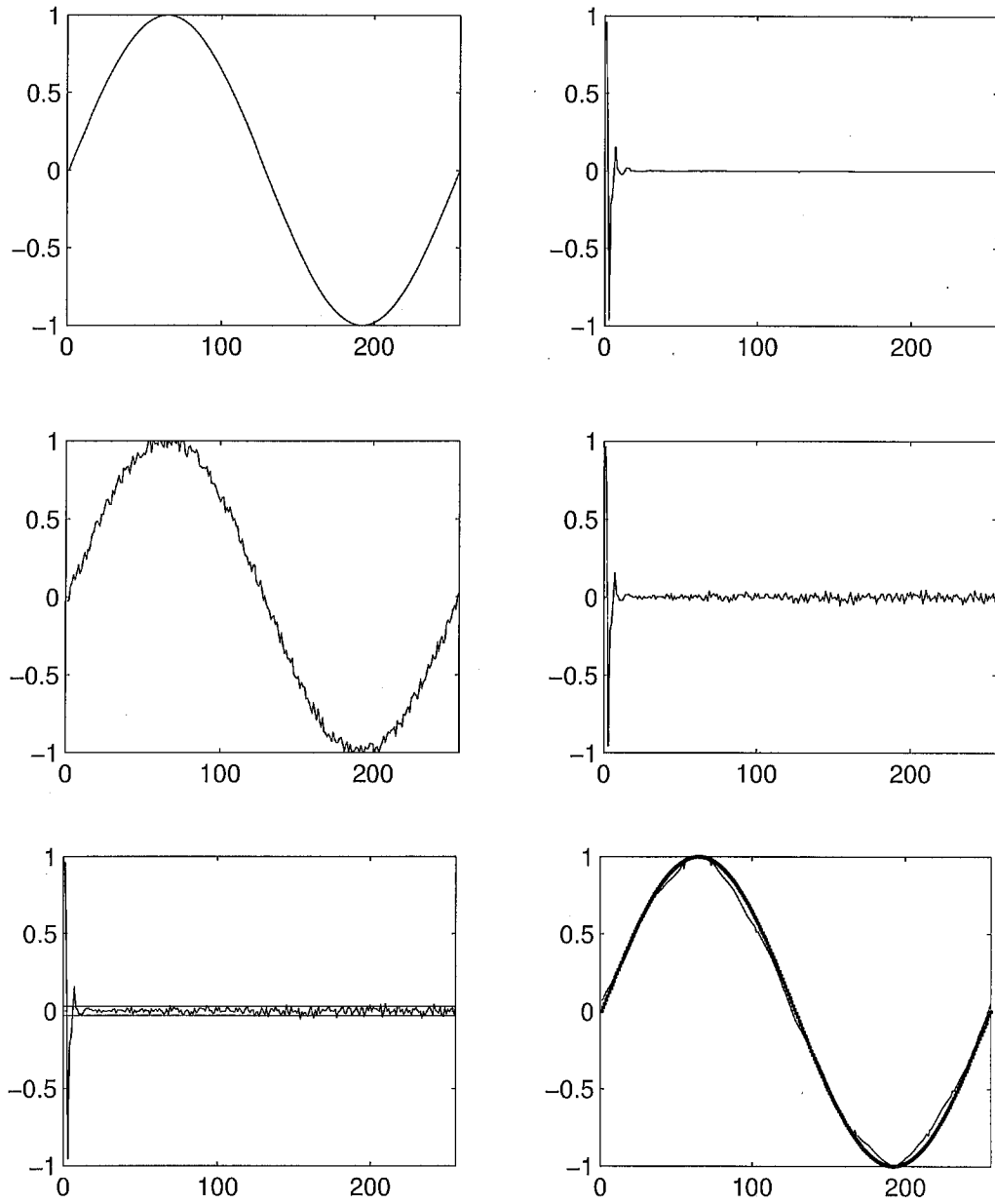


fig:shrink 圖 35 凌波縮減除雜訊法之示意圖

的意思。做完了凌波縮減之後，再做合成。合成後的訊號未必是原來沒有加雜訊的乾淨訊號，但是已經好多了。圖 35 第三列的右圖中，比較粗的線是單純的 \sin 曲線，比較細的線是經過凌波縮減之後的還原訊號。雖然兩者不同，但已經相去不遠。高頻的雜訊幾乎沒有了，但是低頻的訊號也多少受到影響。

凌波縮減的技術固然簡單，但是要有效地消除雜訊，最要緊的還是如何決定那個門檻數 ϵ 。這個 ϵ 未必是個常數，它有可能與解析層次 j 有關。決定 ϵ 的理論與技術，則牽涉統計和訊號處理的專業知識。我們就此停住。

習題 4.6 請設計一個實驗，以凌波縮減的方法來消除雜訊。我們可以用 Matlab 的 `rand(1,n)` 指令來產生一個長度 n 的亂數序列，並利用它來模擬隨機雜訊。例如

```
noise = rand(1,1024); noise = (noise - mean(noise))/8;
```

即產生振幅介於 $\pm\frac{1}{8}$ 之間、平均值為 0 的雜訊。

4.5 與傅立葉轉換之比較

理論上，凡是凌波轉換辦得到的事，傅立葉轉換（或級數）也可以辦到。換句話說，在理論上傳立葉轉換辦不到的，凌波轉換也辦不到。所以它們兩者之間，只有效率上的差別：通常凌波轉換的效率較高。而凌波轉換之所以有較高的效率，說穿了只有一個原因，那就是凌波函數不論在時間域或頻率域都具有局部性，三角函數則不。（亦請參考圖 26 的圖示。）在這一節裡面，我們首先比較兩種轉換的計算複雜度，再以兩個例子來佐證前面所做的論述。第一個例子呈現 Gibbs 現象，在凌波轉換之下也會發生。第二個例子，以圖 32 中的測試訊號來做壓縮效率的比較。

若輸入訊號之長度為 N 。已知離散凌波轉換 (DWT) 的計算複雜度是 $O(N)$ 。離散傅立葉轉換原本的計算複雜度是 $O(N^2)$ ，但因為快速傅立葉轉換 (FFT) 的發明，使其計算複雜度降為 $O(N \log_2 N)$ 。若以理論上的計算複雜度相比，則 DWT 的 $O(N)$ 似乎比 FFT 的 $O(N \log_2 N)$ 略勝一籌。但是，以我們在 Matlab 中實作的 DWT 程式 `dwt` 與 Matlab 內建的 FFT 程式 `fft` 相比，會發現兩者在實際上沒有多大差別。`dwt` 的實際複雜度大約是 $8.35pN$ ，而經過實驗，`fft` 的實際複雜度大約是 $3.27N \log_2 N$ 。所以，實際上要相比的是 $8.35p$ 和 $3.27 \log_2 N$ 。理論上，如果選定 p ，則 N 越大就越顯得凌波轉換很有效率。可是如果 N 不太大的時候，FFT 還是可能比較快。例如當 $N = 1024$ ，若選 $p \geq 4$ ，則 `dwt` 的計算量就超過了 `fft`。

idx:FFT

idx:DWTvsFI

當然，也可能因為 `fft` 是 Matlab 的內建指令，它是一個寫得非常成熟與洗煉的程式。我們的 `dwt` 只是練習之作，還有改善的空間。但是有關 FFT 的演算法與程式寫作技巧，甚至包括平行化的程式寫作，都已經發展了四十餘年。它必定是比凌波轉換要來得成熟。我想要下的結論是，在計算複雜度方面，DWT 理論上比 FFT 有效率，但是實際上並沒有多少分別。

如果 $f(x)$ 是 2π 週期函數，則傅立葉轉換就是傅立葉級數。在圖 17 中，我們已經看到原始的 Gibbs 現象。其實這種現象是普遍性的。只要以傅立葉級數來逼近不連續函數，就會出現這樣的現象。我們現在以 $f(x) = H(x) - H(-x)$ 來做示範。其中 $H(x)$ 是 Heaviside 函數：

idx:Gibbx

idx:Heaviside

$$H(x) = \begin{cases} 1 & \text{for } x > 0 \\ 0 & \text{for } x \leq 0 \end{cases}$$

將 $f(x)$ 限制在 $[-\pi, \pi]$ 中，再將其週期化。由於 $f(x)$ 是個奇函數，它的傅立葉級數只有 $\sin kx$ 項。圖 36 分別是取前 16 項與前 50 項傅立

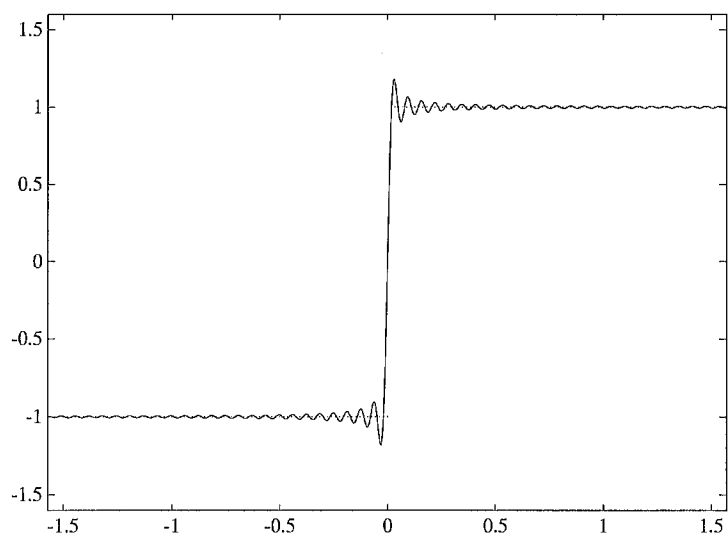
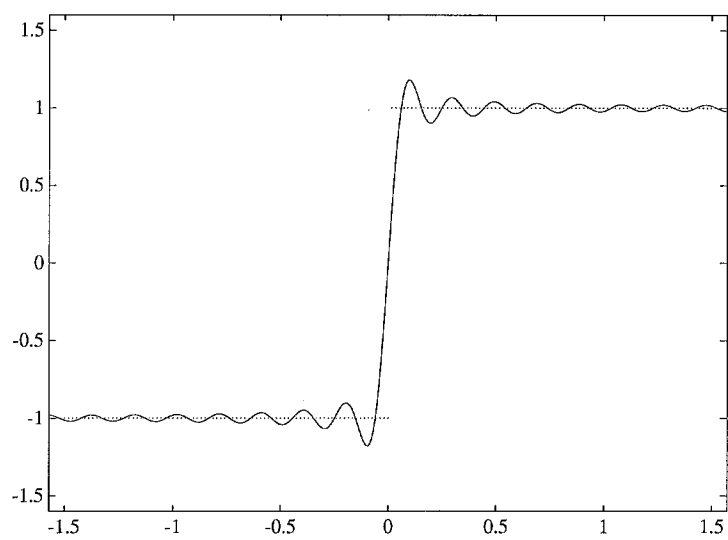


fig:GibbsH 圖 36 $H(x) - H(-x)$ 的 16 項與 50 項傅立葉級數

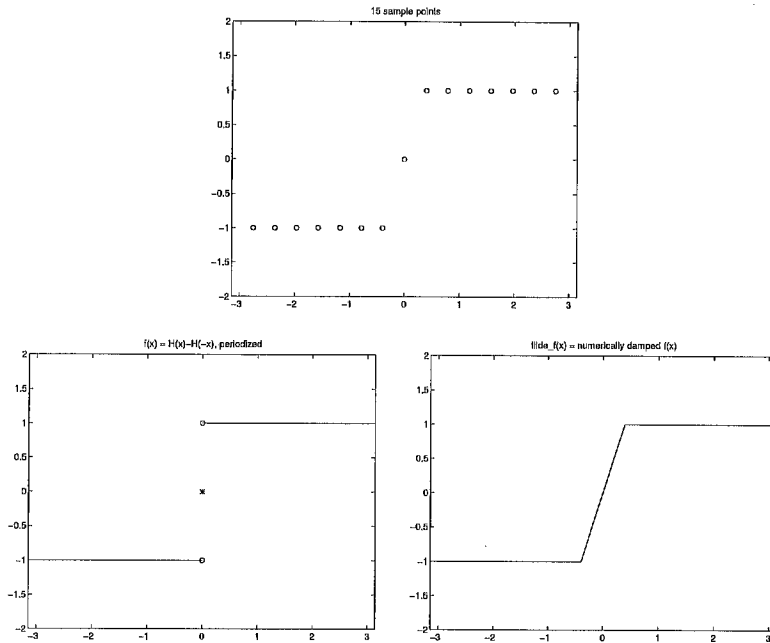


fig:discont 圖 37 數值黏滯性之圖例

葉級數的合成圖形。我們仍然可以在圖中看到 Gibbs 現象：在不連續點的兩側，傅立葉級數有超射的現象，而且超射的程度趨近於一個常數。

在此要闡述一個細節。當我們做此類實驗的時候，一定要認真地用以下方式計算傅立葉級數的係數：

$$\int_{-\pi}^{\pi} f(x) \sin kx \, dx = \int_0^{\pi} \sin kx \, dx - \int_{-\pi}^0 \sin kx \, dx$$

不可以直接拿 $f(x)$ 在 $[-\pi, \pi]$ 中做數值積分（我們將在第五章五節談到數值積分）。因為數值積分都是取 $f(x)$ 的樣本點去做計算。但是在取了樣本點之後，電腦的計算法則無法判別不連續的函數。圖 37 上圖中的圓點就是 $f(x)$ 在 $(-\pi, \pi)$ 之間取的 15 個均勻分佈樣本點。試想，只看

到這 15 個點，如何分辨它們是圖 37 左下圖函數 ($f(x)$) 的樣本點，還是右下圖函數 ($\tilde{f}(x)$) 的樣本點？

圖 37 中的 $f(x)$ 是不連續函數，但 $\tilde{f}(x)$ 是連續函數。當我們以圖 37 上圖中的那 15 個樣本點去做 $\int f(x) \sin kx dx$ 的數值積分，其效果就等於做 $\int \tilde{f}(x) \sin kx dx$ 的數值積分。但因為 $\tilde{f}(x)$ 是連續函數，所以由這些數值積分產生的傅立葉係數，就不會造成 Gibbs 現象。這是由於數值計算的有限性，導至不連續的曲線被誤認為連續但很陡的曲線，就好像曲線下方有非常黏稠的物質，使得它不易被拉斷。這種誤導的現象，稱為數值黏滯性。

如果我們同樣將 $f(x) = H(x) - H(-x)$ 做凌波投影。亦即將 $f(x)$ 正交投影到 $V_j = V_0 \oplus W_0 \oplus \cdots \oplus W_{j-1}$ ，則 Gibbs 現象仍然會發生。所以在理論上，投影後的曲線仍然在不連續點附近會發出超射的現象。但是，實際上，凌波轉換的效果卻比傅立葉級數好得多。看圖 38：上圖是取到 $j = 2$ 的投影函數，下圖取到 $j = 4$ 。

比較圖 36 和圖 38，我們可以認識到，因為 \sin 函數的全域性 (global)，使得就因為 $f(x)$ 在 $x = 0$ 處的一個不連續點，而干擾了傅立葉級數的整體表現。而因為凌波函數的局部性 (local)，使得凌波空間中的投影函數只有在 $x = 0$ 附近受到影響。所以，局部性的特色，使得凌波空間中的投影，不會因為一顆老鼠屎而壞了整鍋粥。固然在理論上，若 $k \rightarrow \infty$ 則傅立葉級數會收斂到 $f(x)$ ，同樣地若 $j \rightarrow \infty$ 則凌波空間的投影函數也會收斂到 $f(x)$ ，但是凌波投影函數顯然收斂地更快，而且有更好的逼近效果。

全域性相對於局部性，也同樣地影響其他訊號處理工作的效率。以下舉一個訊號壓縮的例子。我們將圖 32 的一維測試訊號，模擬 1:4 和 1:8 的壓縮，然後比較壓縮後的合成訊號。所謂 1:n 的壓縮，就是只儲存 $\frac{N}{n}$ 的資料量 (其中 N 是原資料量)。我們並沒有真的做壓縮，

idx:local

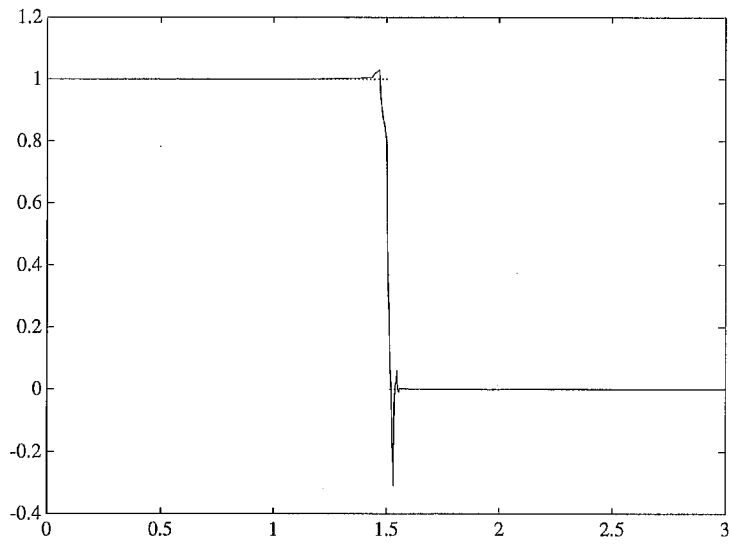
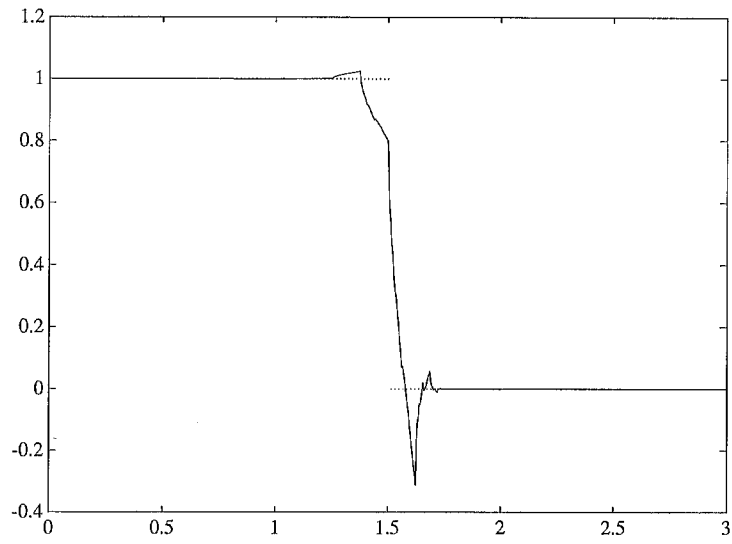


fig:GibbsW 圖 38 $H(x) - H(-x)$ 在 V_2 和 V_4 的正交投影

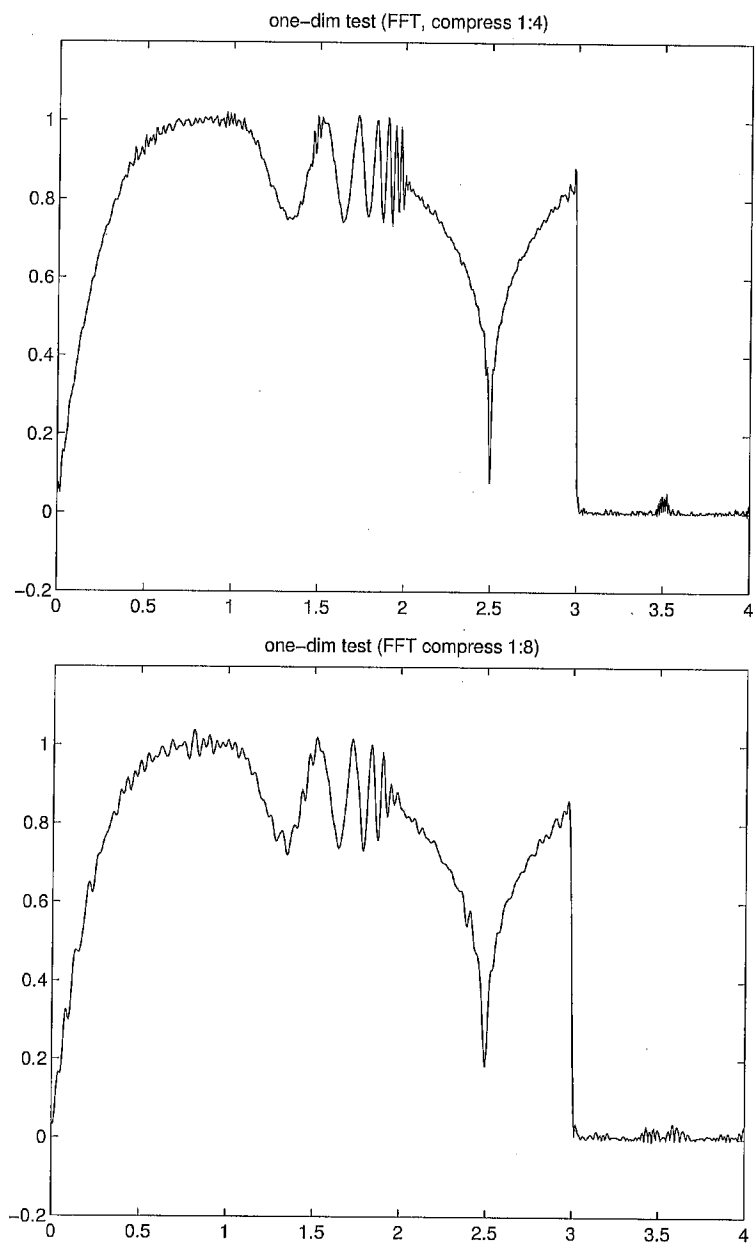


fig:testonedcf 圖 39 一維測試訊號以 FFT 做 1:4 與 1:8 之壓縮與還原

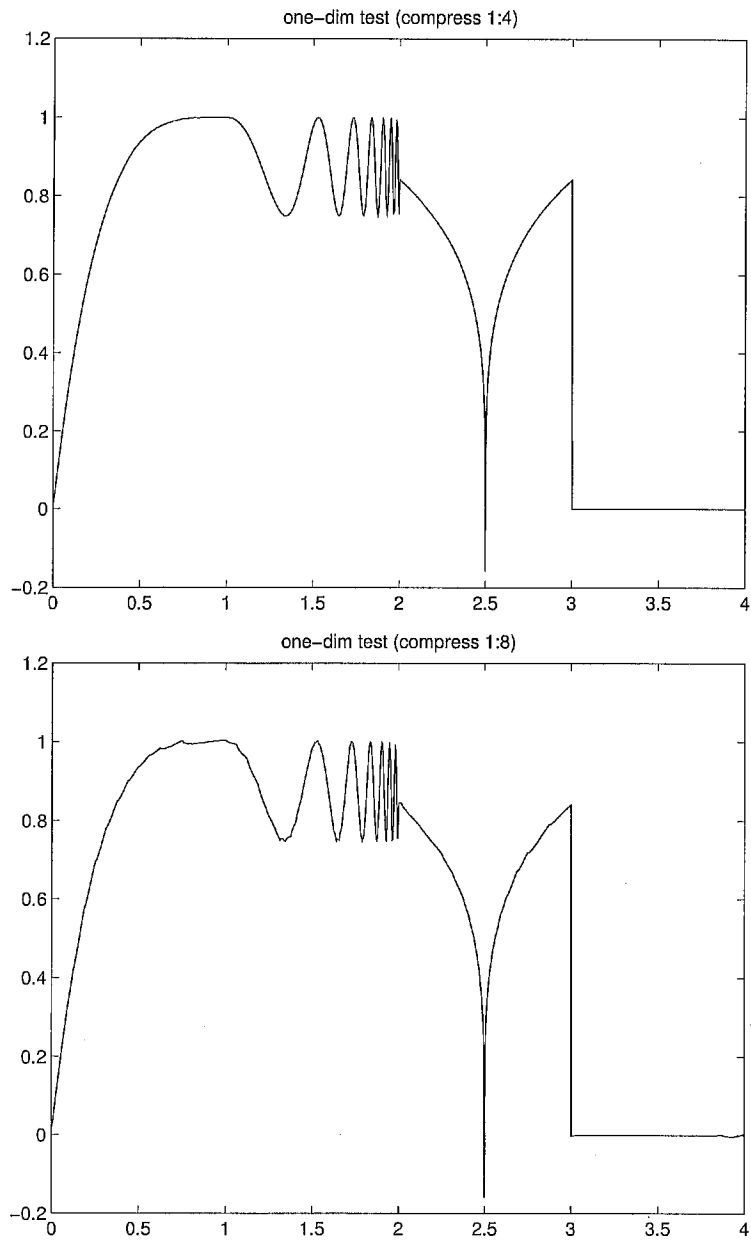


fig:testonedcw 圖 40 一維測試訊號以 DWT 做 1:4 與 1:8 之壓縮與還原

只是模擬這個過程。模擬的方法是，先做基底轉換 (傅立葉或凌波)。在轉換後之係數中，保留絕對值最大的 $\frac{N}{n}$ 個元素，將其他的全部設為 0。然後，我們合成這些模擬的壓縮訊號，將之與原始訊號做比較。圖 39 是以 FFT 作為轉換工具，圖 40 是以 DWT 作為轉換工具。兩張圖的上方皆是 1:4 的壓縮比，下方皆是 1:8 的壓縮比。

FFT 的還原結果，受到三角函數之全域性影響，使得原訊號的少數幾處不平滑點，污染了整個合成訊號。相對地，DWT 的還原結果，則只有在不平滑處的附近才受影響。注意在這兩張圖中不見 Gibbs 現象。因為我們在此並沒有真正以積分求投影係數，而是直接拿測試訊號的 1024 筆數位資料來做 FFT 或 DWT 的轉換。所以數值黏滯性就起了作用。

習題 4.7 以 Matlab 或其他程式語言，實作一套模擬訊號壓縮的程式。如果您想要模擬得更逼真一點，可以將模擬壓縮的資料寫入一個 mat 檔案，然後以一般性的壓縮程式 (例如 ZIP) 來壓縮那個 mat 檔。

4.6 二維離散凌波轉換—演算法

關於如何將單變數的凌波函數觀念推廣到多變數，以至於將一維的凌波轉換演算法推廣到高維度，數學家發展了幾種不同的方法或模式。在此我們只介紹最常見的一種模式：直積 (*Cartesian product*)。其實我們只考慮二維 (雙變數) 的情形，並不考慮更高的維度。換言之，我們只是將凌波函數從直線推廣到平面上，將凌波轉換演算法從一維的向量推廣到二維的矩陣。這一節的內容，曾經在第一章的十五節與十七節中預習過。

定義雙變數的自格函數 $\phi(x, y)$ 為

$$\phi(x, y) = \phi(x)\phi(y) \quad (4.12)$$

idx:Cartesian

idx:phitwod

eq:phixy

其中 $\phi(\cdot)$ 是 p 階正交自格函數。所以 $\phi(x)$ 和 $\phi(y)$ 分別是 x -軸和 y -軸上的自格函數。例如我們取 $p = 1$ ，則對應的 $\phi(x, y) = \chi_{[0,1]^2}$ 。現在，令其脹縮平移為

$$\phi_{jkl}(x, y) = \phi_{jk}(x)\phi_{jl}(y) = 2^j \phi(2^j x - k)\phi(2^j y - \ell) \quad (4.13) \quad \text{eq:phijkl}$$

則自格等式就可寫成

$$\phi(x, y) = \sum_k c_k \phi(2x - k) \sum_\ell c_\ell \phi(2y - \ell) = \frac{1}{2} \sum_{k, \ell} c_k c_\ell \phi_{1k\ell}(x, y) \quad (4.14) \quad \text{eq:sctwod}$$

如此，則 $\{\phi_{jkl}(x, y) \mid (j, k) \in \mathbb{Z}^2\}$ 形成一個正則集合。假設它們展成一個 $V_j(x, y)$ 空間，亦即

$$V_j(x, y) = \text{span} \{ \phi_{jkl}(x, y) \mid k, \ell \in \mathbb{Z} \} \quad (4.15) \quad \text{eq:Vjxy}$$

則根據 (4.14) 明顯可見 $V_j(x, y) \subset L^2(\mathbb{R}^2)$ 。事實上，可證明

$$\cdots \subset V_0(x, y) \subset V_1(x, y) \subset \cdots$$

形成 $L^2(\mathbb{R}^2)$ 上的一組多層解析空間。其中 $L^2(\mathbb{R}^2)$ 的內積定義成

$$\langle f(x, y), g(x, y) \rangle = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \overline{g(x, y)} dx dy \quad (4.16) \quad \text{eq:iptwod}$$

而長度定義成 $\|f(x, y)\| = \sqrt{\langle f, f \rangle}$ 。為了符號的簡潔，我們常省略前面重積分中的上下限，簡記做 \iint 。

根據 (4.15) 和 (4.13) 之定義，我們可以改寫

$$V_j(x, y) = V_j(x) \times V_j(y) \quad (4.17) \quad \text{eq:VxVy}$$

其中

$$V_j(x) = \text{span} \{ \phi_{jk}(x) \mid k \in \mathbb{Z} \}, \quad V_j(y) = \text{span} \{ \phi_{jk}(y) \mid k \in \mathbb{Z} \}$$

接著，我們要知道 $V_{j+1}(x, y)$ 和 $V_j(x, y)$ 之間的直差。因為 $V_j(x, y) \subset V_{j+1}(x, y)$ 且 $V_j(x, y) \neq V_{j+1}(x, y)$ ，所以必定存在一個 $W_j(x, y)$ 使得

$$V_j(x, y) \oplus W_j(x, y) = V_{j+1}(x, y)$$

問題是如何簡單地表達 $W_j(x, y)$ 這個空間？它和 $\phi(\cdot)$ 所對應的正交凌波函數 $\psi(\cdot)$ 有沒有關係？答案是肯定的：

$$W_j(x, y) = (V_j(x) \times W_j(y)) \oplus (W_j(x) \times V_j(y)) \oplus (W_j(x) \times W_j(y)) \quad (4.18) \quad \text{eq: WxWy}$$

其中

$$W_j(x) = \text{span} \{ \psi_{jk}(x) \mid k \in \mathbb{Z} \}, \quad W_j(y) = \text{span} \{ \psi_{jk}(y) \mid k \in \mathbb{Z} \}$$

原因很明顯：

$$\begin{aligned} V_{j+1}(x, y) &= V_{j+1}(x) \times V_{j+1}(y) \\ &= (V_j(x) \oplus W_j(x)) \times (V_j(y) \oplus W_j(y)) \\ &= V_j(x, y) \oplus (V_j(x) \times W_j(y)) \oplus (W_j(x) \times V_j(y)) \oplus (W_j(x) \times W_j(y)) \end{aligned}$$

不再像一維的凌波分解， W_j 是由一組函數 $\psi_{jk}(x)$ 所展成的；二維的 $W_j(x, y)$ 由三組函數展成：

$$W_j(x, y) = \text{span} \{ \psi_{jke}^1(x, y), \psi_{jke}^2(x, y), \psi_{jke}^3(x, y) \mid k, \ell \in \mathbb{Z} \} \quad (4.19) \quad \text{eq: Wjxydef}$$

其中

$$\begin{aligned}\psi_{jke}^1(x, y) &= \phi_{jk}(x)\psi_{je}(y) \\ \psi_{jke}^2(x, y) &= \psi_{jk}(x)\phi_{je}(y) \\ \psi_{jke}^3(x, y) &= \psi_{jk}(x)\psi_{je}(y)\end{aligned}\tag{4.20}$$

eq:psijkl

上述三族函數形成一個正則集合，亦即

$$\langle \psi_{jke}^i, \psi_{j'k'e'}^{i'} \rangle = \delta_{ii'}\delta_{jj'}\delta_{kk'}\delta_{ee'}$$

這種二維自格函數與凌波函數的定義方式，最大好處就是算法上的方便。二維的離散凌波轉換應用在二維或平面訊號上。二維訊號的意義，通常是一張影像。假設影像已經以離散數位的格式儲存，則它們逐點的顏色（或其他性質）已經被寫成數字。對二維離散凌波轉換的演算法而言，這些數字組成一個矩陣：

$$\mathbf{V} = (v_{kle})_{N \times M}$$

爲了敘述與程式中資料結構的方便，我們假設 $N = 2^J$ 、 $M = 2^L$ 。

在此我們先介紹二維凌波轉換的矩陣表達。如果 $N = M$ ，則 $\mathbf{V} \in \text{idx:dwtmat}$ $V_J(x, y)$ 。令 \mathcal{W} 是凌波轉換矩陣 (4.5)。則 $\mathcal{W}\mathbf{V}$ 相當於將 \mathbf{V} 中的每一個行向量 (column vector) 做凌波分解。也就是做了以下的分解步驟：

$$V_J(x, y) = V_J(x) \times (V_{J-1}(y) \oplus W_{J-1}(y))$$

而 $\mathbf{V}\mathcal{W}^T$ 相當於將 \mathbf{V} 中的每一個列向量 (row vector) 做凌波分解。也就是做了以下的分解步驟：

$$V_J(x, y) = (V_{J-1}(x) \oplus W_{J-1}(x)) \times V_J(y)$$

因此 $\mathcal{W}\mathbf{V}\mathcal{W}^T$ 就是將 \mathbf{V} 做了一層的凌波分解：

$$V_J(x, y) = V_{J-1}(x, y) \oplus W_{J-1}(x, y)$$

參照 (1.63)，分解後的矩陣可以寫成四塊：

$$\mathcal{W}\mathbf{V}\mathcal{W}^T = \begin{pmatrix} \mathbf{U} & \mathbf{W}_x \\ \mathbf{W}_y & \mathbf{W} \end{pmatrix} \quad (4.21) \quad \text{eq:UWWagain}$$

四塊子矩陣 \mathbf{U} 、 \mathbf{W}_y 、 \mathbf{W}_x 和 \mathbf{W} 的維度都是 $\frac{N}{2} \times \frac{N}{2}$ 。 \mathbf{U} 中的元素就是 $V_{J-1}(x, y)$ 中的係數，其他三塊矩陣中的元素，就是 $W_{J-1}(x, y)$ 中的係數。對照 $W_j(x, y)$ 的定義 (4.18)， \mathbf{W}_y 中的元素是 $V_{J-1}(x) \times W_{J-1}(y)$ 中的係數、此空間的正則基底是 $\psi_{J-1, k\ell}^1(x, y)$ ； \mathbf{W}_x 中的元素是 $W_{J-1}(x) \times V_{J-1}(y)$ 中的係數、此空間的正則基底是 $\psi_{J-1, k\ell}^2(x, y)$ ； \mathbf{W} 中的元素是 $W_{J-1}(x) \times W_{J-1}(y)$ 中的係數、此空間的正則基底是 $\psi_{J-1, k\ell}^3(x, y)$ 。

如果 $N \neq M$ ，可以假設 \mathbf{V} 中的元素是

$$V_L(x) \times V_J(y)$$

中的係數。因為 $J \geq L$ ，故 $V_L(x) \subset V_J(x)$ ，所以理論上可以認為 \mathbf{V} 中的元素是 $V_J(x, y)$ 中的係數。請讀者自行推廣前一段的描述。

底下我們解釋二維離散凌波轉換的演算法，以及一個在 Matlab 中實作的範例。此範例程式是 `dwt2d`。就像 `dwt1d` 一樣，我們用 `dwt` 作為它的控制程式。通常使用者呼叫 `dwt`，而不是 `dwt1d` 或 `dwt2d`。

回顧 `dwt(p, x, steps)` 的程式片段 Code-1，如果輸入 x 是一個矩陣，則我們假設 x 的列數 (row dimension) N 不小於行數 (column dimension) M 。換言之，假設 x 之行向量的維度是 N 、列向量的維度是

M，則 $N \geq M^*$ 。再換言之，如果 x 不是正方形，則它的外形是窄而高的，像一般的人像畫 (portrait)，而不是寬而矮的，像一般的風景畫 (landscape)。而且，我們要求 N 和 M 分別是 2 的正整數次方。如果通過了 `dwt` 的檢查，就會呼叫 `dwt2d(p,x,steps,N,M)`。 idx:dim

如果您有一個非 2 的整數次方大小的影像，可以用 `reflect` 程式 (Code-2) 將其擴大成最靠近的 2 的次方矩陣。雖然在第三節中只介紹了 `reflect` 施加於向量上的用法，其實它也可以作用在矩陣上。例如，若

$$x = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 3 & 2 & 1 \\ -2 & -1 & 0 & 1 & 2 \end{pmatrix}$$

則

$$\text{reflect}(x) = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 5 & 4 & 3 \\ 5 & 4 & 3 & 2 & 1 & 1 & 2 & 3 \\ -2 & -1 & 0 & 1 & 2 & 2 & 1 & 0 \\ -2 & -1 & 0 & 1 & 2 & 2 & 1 & 0 \end{pmatrix}$$

`dwt2d` 的前置作業和 `dwt1d` 幾乎完全一樣。所不同的只有最低分解層次的設定。一維訊號的凌波轉換只要設定 j_0 ，而二維訊號的凌波轉換需設定 j_0 和 ℓ_0 。定義

$$j_0 = \max\{J - |\text{steps}|, \min j_0\}, \quad \ell_0 = \max\{L - |\text{steps}|, \min \ell_0\}$$

其中 $\min j_0 = \min \ell_0 = \lceil \log_2 2p \rceil$ ，如 (4.10) 所列。由於 $J \geq L$ ， j_0 和 ℓ_0 可能不同。`steps` 的最大可能是 $J - \min j_0$ 。當 $J > L$ 而且 `steps` >

* 這裡或許出現了一點兒名詞上的混淆。當我們說訊號的維度時，指的是在連續狀態下，那個訊號是幾個變數的函數。當我們說向量的維度時，指的是向量中有幾個元素。當我們說矩陣的維度時，指的是矩陣共有幾列、幾行。

$L - \min \ell_0$ 的時候，水平 (x) 方向的分解會先停止，dwt2d 會呼叫 dwt1d 使得垂直 (y) 方向的分解繼續下去，直到 $\min j_0$ 層為止。

```
J = log2(N);
L = log2(M);
minj = ceil(log2(2*p));
steps = min(steps, J-minj);
Lsteps = min(steps, L-minj);
steps = steps - Lsteps;
```

程式中的 $\min j$ 就是 $\min j_0 = \min \ell_0$ 。Lsteps 是水平與垂直兩方向都可以做分解的次數，它最多是 $L - \min \ell_0$ 。如果 $\text{steps} > L - \min \ell_0$ ，則 Lsteps 次的分解，將 V 訊號

$$\text{從 } V_L(x) \times V_J(y) \text{ 分解到 } V_{\min \ell_0}(x) \times V_{j_0}(y)$$

其中 $j_0 = J - \text{Lsteps}$ 。然後重新定義 steps，準備以後使用。

如果 $\text{case} = 1$ ，我們做分解。仍然像 dwt1d 一樣地將低頻與高頻濾波係數分別組成序列 DL 和 DH。然後做 Lsteps 次的二維分解。回顧 pwd 和 wf 的設計 (Code-5 和 Code-6)，已經兼顧了矩陣運算。當 x 是矩陣的時候，pwd 和 wf 的運算自動會對每一行的元素執行。例如

```
x(m+1:m+2*p-2, :) = x(1:2*p-2, :);
```

會將 x 矩陣的前面 $2p - 2$ 列接到最後面。例如

```
L*x(i:i+R, :)
```

會將序列 L 與 x 每一行中從 i 到 $i+R$ 的向量做內積。

所以，在 `dwt2d` 中我們還可以呼叫 `pwd` 來做行向量（垂直方向）的分解。執行一次 `pwd`，相當於做 $\widetilde{W}V$ 。我們將第一次執行的結果轉置，再執行一次 `pwd`，然後再轉置回來，就相當於做了 $\widetilde{W}V\widetilde{W}^T$ 。然後我們按奇偶位置將矩陣重排，使其結構符合 (4.21) 的描述。

code:pwdtwc

Matlab Code-11

```
n = N; m = M;
for i = 1:Lsteps,
    x(1:n,1:m) = pwd(p, pwd(p,x(1:n,1:m),DL,DH)', DL,DH)';
    x(1:n,1:m) = [x(1:2:n,1:2:m) x(1:2:n,2:2:m); ...
                  x(2:2:n,1:2:m) x(2:2:n,2:2:m)];
    n=n/2; m=m/2;
end;
```

如果 `steps > 0`，則要將 $V_{j_0}(y)$ 中的元素再做 `steps` 次的分解。但是，此刻的分解只是對每個行向量做一維凌波分解，所以可以呼叫 `dwt1d` 來執行。

```
if (steps > 0)
    x(1:n,1:m) = dwt1d(p,x(1:n,1:m),steps,n);
end;
```

如果 `case ≠ 1`，則採取合成步驟。還是像 `dwt1d` 一樣地將組成序列 RL 和 RH 。有關 `Lsteps` 與 `steps` 的定義依舊。合成乃是反分解之道而行。所以，如果 `steps > 0`，則要先在垂直方向做 `steps` 次的合成，使得訊號回到 $V_{\min l_0}(x) \times V_{j_0}(y)$ 。這個步驟仍然可以呼叫 `dwt1d` 來執行。然後才能執行 `Lsteps` 次的二維凌波合成。

code:pwctwc

Matlab Code-12

```

if (steps > 0)
    n = 2^(steps+minj); m = 2^minj;
    x(1:n,1:m) = dwt1d(p,x(1:n,1:m),-steps,n);
else
    n = 2^(J-Lsteps); m = 2^(L-Lsteps);
end;
for i = 1:Lsteps,
    n=2*n; m=2*m;
    x(1:n,1:m) = pwc(p, interlace(x(1:n,1:m)), RL, RH);
    x(1:n,1:m) = pwc(p, interlace(x(1:n,1:m)'), RL, RH)';
end;

```

至此，我們完成了 $dwt2d$ 的描述。這個程式完全是由 $dwt1d$ 和它的子程式所構成。因此，其計算複雜度很明顯地應該是 $O(MN)$ 。當 $N = M$ ，計算複雜度大約是 $10pN^2$ 。一般來說，複雜度是 C_pNM 。當 N 越大於 M 時，真正執行二維轉換的運算次數越少，所以係數 C 就逐漸降低。例如以維度分別是

$$64 \times 64, \quad 128 \times 32, \quad 256 \times 16, \quad 512 \times 8$$

的矩陣輸入 $dwt(4,x,10)$ ；其計算次數分別大約是

$$10.5pMN, \quad 10.4pMN, \quad 9.9pMN, \quad 7.9pMN$$

習題 4.8 證明 $\{\phi_{jkl}(x,y) \mid k,l \in \mathbb{Z}\}$ 在 $L^2(\mathbb{R}^2)$ 的內積意義之下是一個正則集合。

習題 4.9 請問 $\text{supp } \phi_{jkl}(x,y)$ 是平面上的哪個區域？

習題 4.10 證明 (4.20) 中定義的三族函數彼此正交。

習題 4.11 若 $N \neq M$ ，請問 (4.21) 中四個子矩陣的維度分別是多少？令 $N = 2^J$ 、 $M = 2^L$ ，請問 V 、 U 、 W_y 、 W_x 和 W 各自是在多層解析空間的什麼空間裡面？

習題 4.12 設計一套實驗，檢查 dwt2d 這個程式的精確度。

4.7 二維離散凌波轉換—範例

在此後的兩節中，我們呈現一些二維離散凌波轉換的範例。首先介紹一張經常出現在文獻中的數位影像：Tiffany，然後介紹一個經常被拿來測試矩陣運算效率的 CZO 型式矩陣。我們利用 Tiffany 數位影像，介紹 dwt2d 的基本操作程序，並模擬資料壓縮的效果。

Tiffany 數位影像儲存在 Matlab 的資料檔內

Tiffany.mat

這是一張 256 階的灰階影像。在 load Tiffany 之後，將載入兩個矩陣 Origin 和 g256。其中 Origin 是 256×256 的數位影像訊號，它的每個元素是一個介於 1 和 256 之間的整數，代表該位置的灰階。而 g256 是色盤。它是個 256×3 的矩陣，每一列代表一個顏色。有必要的話，請參考附錄 A.3。這個影像矩陣，以 Matlab 的 image 指令顯像出來，就是圖 41。

我們假設 Origin 是一張真實影像在 $V_8(x, y)$ 內的投影係數。執行 $\text{pwd}(p, x, DL, DH)$ 一次，相當於在影像矩陣的左邊乘一次 \widetilde{W} 。其效果就是將矩陣的每個行向量逐一作分解，分解後的低頻（低解析度）部分放在上方，高頻（細節）部分放在下方。圖 42 的上圖就是 $V_8(x) \times (V_7(y) \oplus$

fig:Teight 圖 41 Tiffany at $V_8(x, y)$

$W_7(y)$)。如果 $\text{pwd}(p, x', DL, DH)$ 一次，相當於在影像矩陣的右邊乘一次 \widetilde{W} 。其效果就是將矩陣的每個列向量逐一作分解，分解後的低頻部分放在左方，高頻部分放在右方。圖 42 的下圖就是 $(V_7(x) \oplus W_7(x)) \times V_8(y)$ 。

dwt2d 在做二維凌波轉換的部分，會執行兩次 pwd ，先做行向量的分解，再將其結果做列向量的分解。執行這兩次 pwd 的結果是產生 $V_7(x, y) \oplus W_7(x, y)$ 中的係數，如圖 43。這張圖很明顯地有四個部份。對應 (4.21) 的矩陣結構，左上角是原影像在 $V_7(x, y)$ 的低解析度影像，右上角是 W_x ，左下角是 W_y ，右下角是 W 。

在此我們必須更加詳細地解釋圖 42 與圖 43 的意義。在這一節中，執行任何實驗的時候，我們都選 $p = 3$ 。這三張圖的低頻部分，其訊號

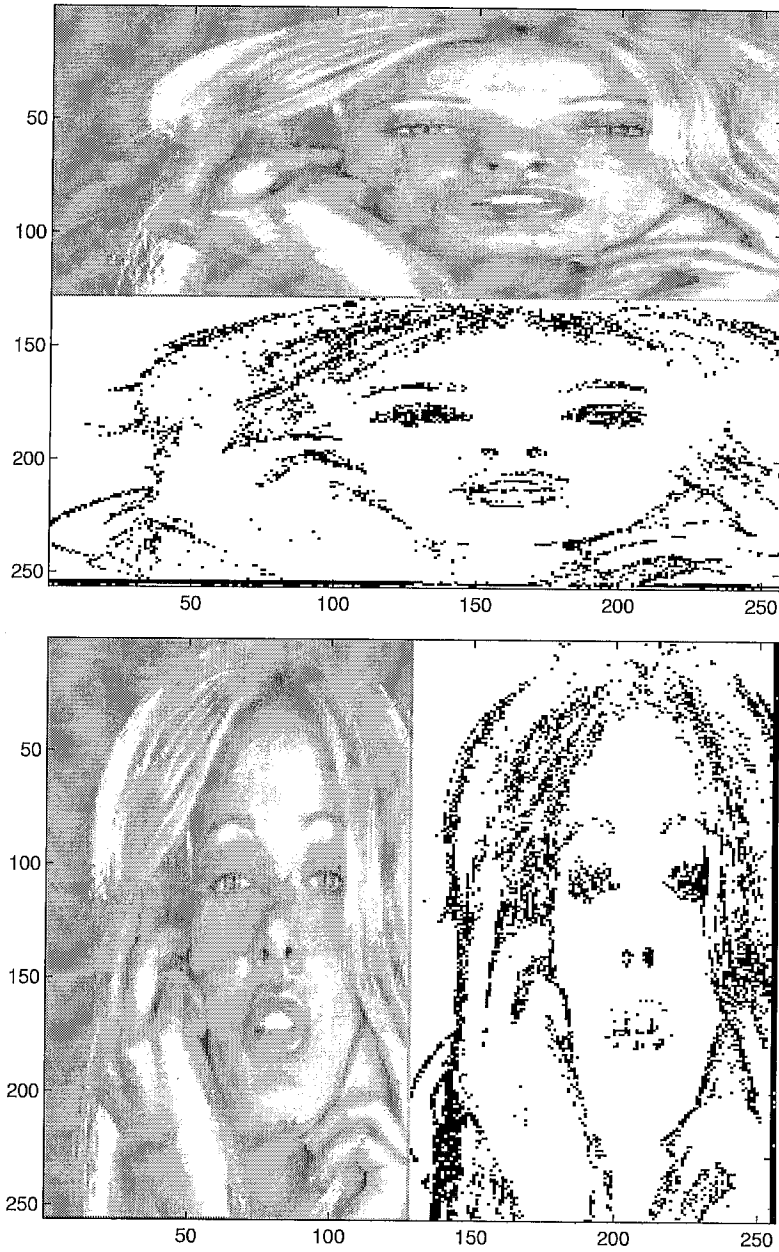


fig:Thalf 圖 42 $V_8(x) \times (V_7(y) \oplus W_7(y))$ 和 $(V_7(x) \oplus W_7(x)) \times V_8(y)$

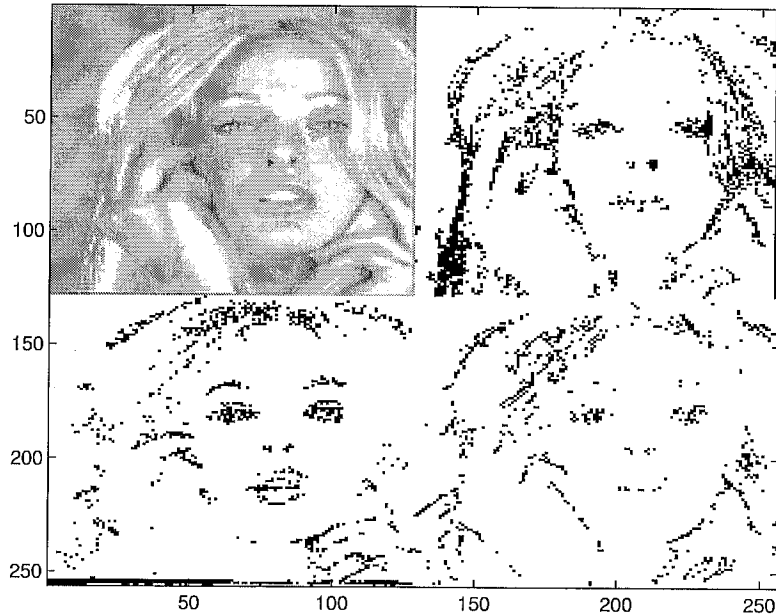


fig:Tdtwo 圖 43 Tiffany at $V_7(x, y) \oplus W_7(x, y)$

之意義仍然是影像，只是其數值（因為經過平均化）已經不是整數。正確的作法，應該是重新執行數位化（quantization）的處理，但是我們只簡單地將其數值做四捨五入（round）變成整數，再做飽和性切除使其介於 1 和 256 之間。但是這三張圖的高頻部分儲存的是 V_8 與 V_7 之間的差異性，它們的意義已經不是影像了（其數值甚至可能是負數）。爲了要製圖，我們選擇將比較大的高頻係數對應黑色，比較小的的高頻係數對應白色。所以讀者可以清楚地看出來，高頻係數的性質。在這三張圖中所見的黑點，代表高頻係數的絕對值 ≥ 4 。圖 43 中整個 $W_7(x, y)$ 的部份共有 49152 個點，其中有 4060 個黑點。底下就是製作圖 43 的過程。

idx:Matlab--r

```
Y=dwt(3,Origin,1);
```

```

[Ya Yb Yc] = split(Y,128,128);
Ya = saturate(round(Ya), 1, 256);
Yb = 1 + 255*(abs(Yb)<4);
Yc = 1 + 255*(abs(Yc)<4);
image([Ya Yb; Yc]);
colormap(g256);
print -deps Tdtwo
sum(sum(Yb==1)) + sum(sum(Yc==1))

```

上面的程式中，Origin 和 g256 從 Tiffany.mat 資料檔中取得。split 和 saturate 兩個子程式是常用的輔助工具。split 將一個矩陣切成三塊，saturate 做飽和性切除。稍後再解釋它們。總之 Ya 是分解後 $V_7(x, y)$ 的係數；Yb 是 W_x ，Yc 是 W_y 和 W 兩部分，合起來就是 $W_7(x, y)$ 的係數。我們將 $V_7(x, y)$ 係數做四捨五入及飽和性切除，使得它符合一個 256 色灰階影像的規格。然後處理 $W_7(x, y)$ 係數。若絕對值 < 4 ，令其對應白色 (256)；若絕對值 ≥ 4 ，令其對應黑色 (1)。然後就可以顯像了。顯像後定義色盤，然後存檔 Tdtwo.eps。最後，我們計算有多少黑點，答案是 4060。

split(X,m,n) 將一個矩陣 X 分成 XA、XB、XC 三個部份。其中 XA 是 X 左上角的 $m \times n$ 子矩陣，XB 是剩下的右上角部分，而 XC 是剩下的部分。參見圖 44。當 n 與 X 的行數相同時，XB 是一個空矩陣。當 m 與 X 的列數相同時，XC 是一個空矩陣。這個程式其實很簡單，大部分的步驟在檢查輸入值的正當性。

code:split

Matlab Code-13

```

function [XA, XB, XC] = split(X,m,n)
[M, N] = size(X);

```

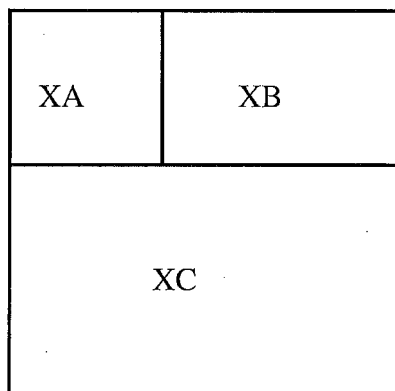


fig:split 圖 44 split 程式分解矩陣之圖示

```

if (m > M)
    XA = X;
    XC = [];
    disp('split: row cut point larger than matrix size.')
else
    XA = X(1:m,:);
    XC = X(m+1:M,:);
end
if (n > N)
    XB = [];
    disp('split: col cut point larger than matrix size.')
else
    XB = XA(:,n+1:N);
    XA = XA(:,1:n);
end

```

`saturate(X,m,M)` 將一個矩陣 X 中 $< m$ 的元素設定成 m ， $> M$ 的元素設定成 M 。這個程式只需要一行。但是請讀者注意，這個程式雖然寫起來簡單，但是執行起來效率並不高。

code:saturate

 Matlab Code-14

```
function A = saturate(A, m, M)
A = A.*((A<=M) & (A>=m)) + m*(A<m) + M*(A>M);
```

在得到 $V_7(x, y) \oplus W_7(x, y)$ 之後，如果還要再做一次凌波分解，`dwt2d` 就會取 $V_7(x, y)$ 的係數將它分解到 $V_6(x, y) \oplus W_6(x, y)$ 。由於分解前後元素的個數沒有改變，所以分解後的元素可以放回原來的位址。因此，顯像的情形就會像圖 45 的上圖。如果一直轉換到 $\min j_0$ ，就像圖 45 的下圖。

回顧圖 42 中，無論垂直或水平方向的高頻部分，在靠近邊界的地方都有大的係數，造成影像中的黑色邊框。我們在第四節中曾經闡述過，這是由於週期化所造成的結果。這些黑色的邊框，也在圖 43 中出現。我們稱之為邊框效應。此外，比較圖 43 的 W_x 與 W_y 部分。我們發現影像的眉毛部分在 W_x 中的高頻係數比較小，在 W_y 中的高頻係數比較大；而頭髮與右手指部分則在 W_x 中的高頻係數比較大，在 W_y 中的高頻係數比較小。這是因為，原影像的眉毛部分在水平方向很少變化，所以這個部位的 $W_7(x)$ 係數很小；眉毛部分在垂直方向有比較大的變化，所以這個部位的 $W_7(y)$ 係數較大。原影像的頭髮與右手指部分，則是相反的情形：主要的變化在水平方向而非垂直方向，所以在 W_x 中留下較大的高頻係數。

idx:periodic
idx:img-bdy

由於凌波係數只會在影像內容急遽變化的位置產生大的係數，而且這些係數的大小，與影像變化的急遽程度成正比。這就是凌波係數消散定理的一個應用，並成為影像辨識的基礎理論之一。

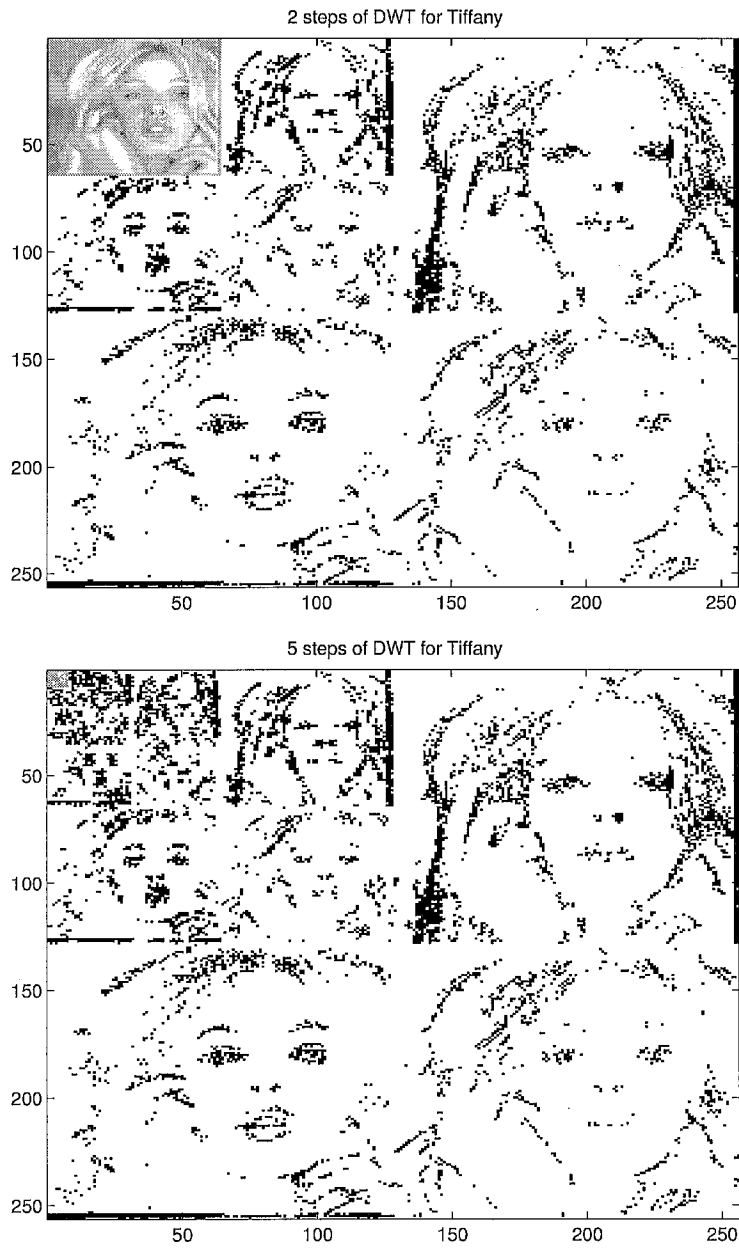


fig:Tdthree 圖 45 Tiffany 經過兩次和最多次的轉換結果

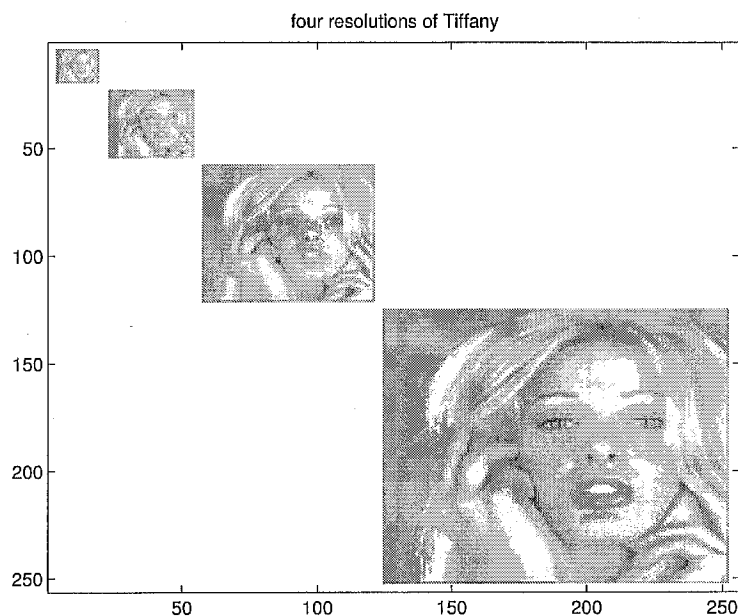


fig:Tfourpic 圖 46 Tiffany at $V_7(x, y)$ 、 $V_6(x, y)$ 、 $V_5(x, y)$ 、 $V_4(x, y)$

凌波轉換亦提供一個有效降低解析度的算法。回顧圖 45 中分解兩次的影像，其左上角的影像是原影像從 $V_8(x, y)$ 降低到 $V_6(x, y)$ 的結果。前者有 256×256 筆資料，後者只有 64×64 筆資料，但是與原圖圖 41 相比，兩者就好像以不同的距離觀賞同一幅圖畫的感覺。舉例來說，如果您以 50 公分的距離觀看圖 45 中 $V_6(x, y)$ 的影像，以四倍 (200 公分) 的距離觀看圖 41 中 $V_8(x, y)$ 的影像，兩者的視覺感受差不多是一樣的。在圖 46 中我們呈現四種不同解析度的相同影像。

如果我們把圖 43 高頻部分的白色點值全設成 0，相當於將高頻係數的絕對值 < 4 者全部設定成 0。保持其他的點值不變，只是四捨五入成整數值。這就相當於模擬影像壓縮。壓縮比是

idx:compress

$$(\text{非零係數個數}) : (\text{總數}) = (128^2 + 4060) : 256^2 \approx 1 : 3.2$$

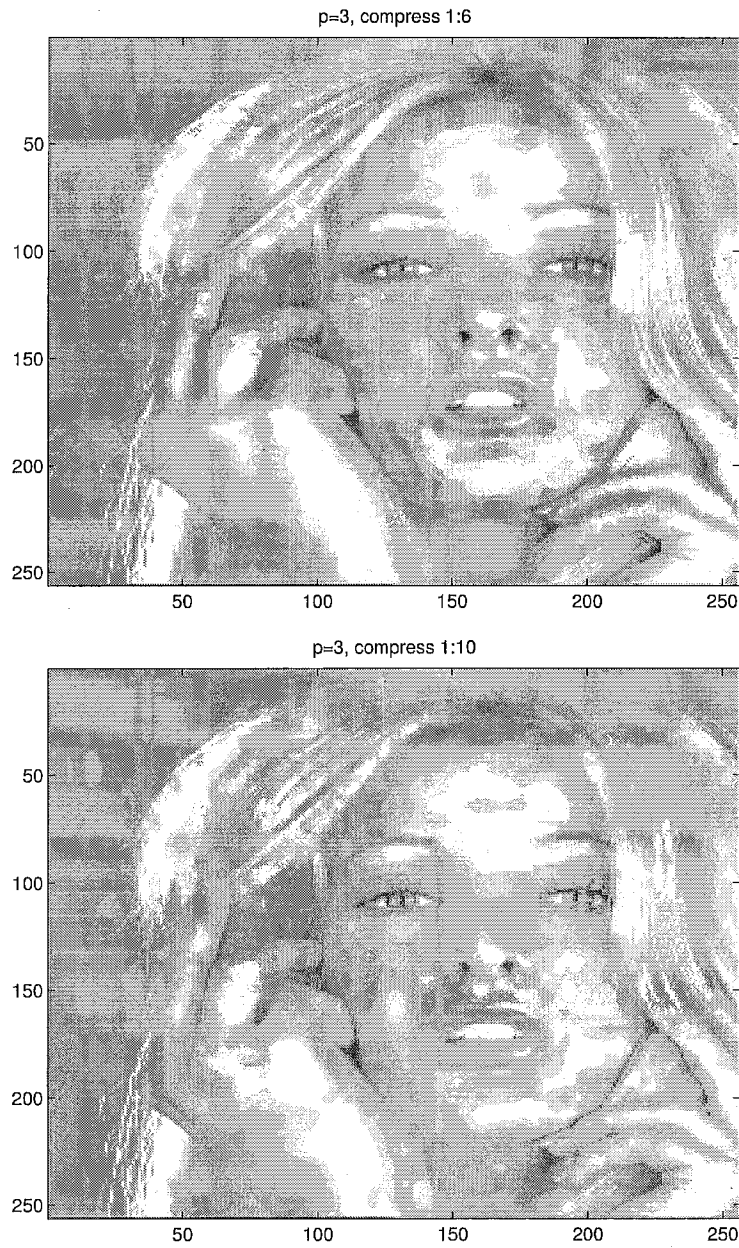


fig:Tcst 圖 47 Tiffany 經過 1:6 和 1:10 模擬壓縮後的合成影像

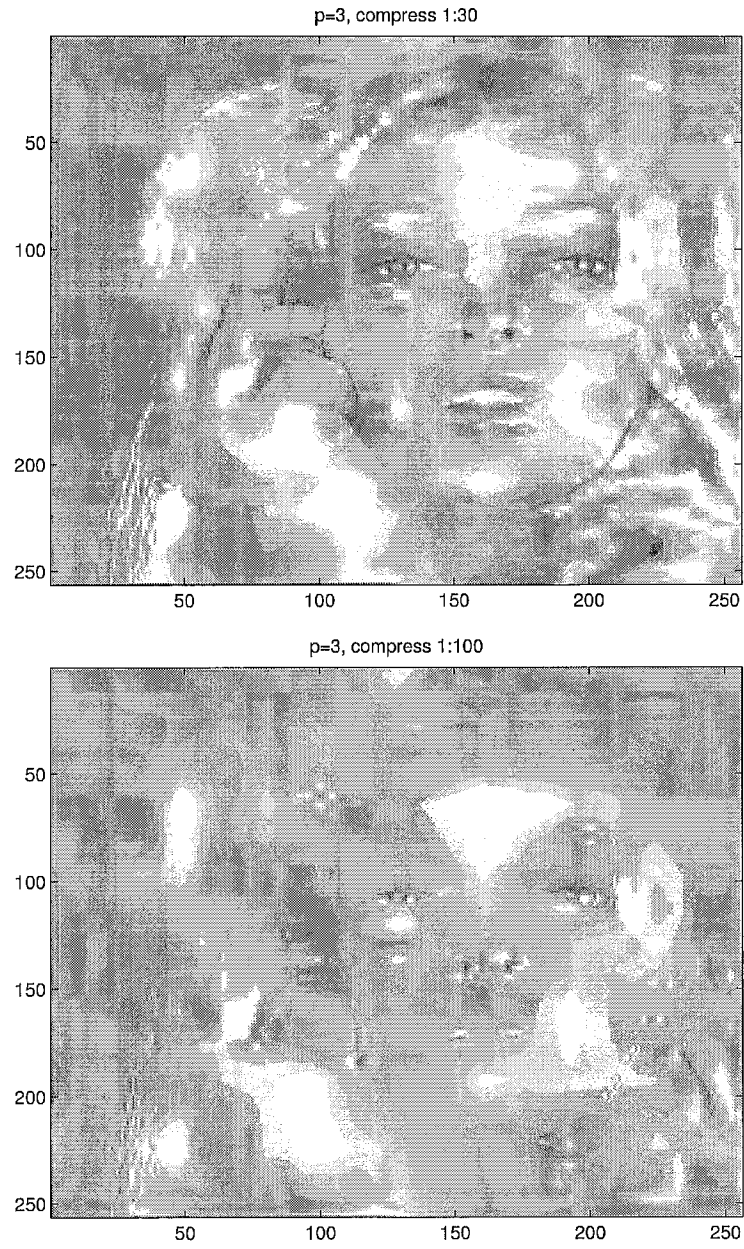


fig:Tchd 圖 48 Tiffany 經過 1:30 和 1:100 模擬壓縮後的合成影像

這當然不是一個令人激賞的壓縮比，我們只是做個實驗而已。將這個模擬壓縮的矩陣合成，所得的矩陣與原矩陣的相對誤差是 0.14%。因為這個誤差太小了，如果顯像出來，我們無法分辨它與原圖的差別，所以就不印了。以下是我們執行上述實驗的過程。注意，最後一個指令就是在計算合成訊號與原始訊號的相對誤差。

```

Y = round(dwt(3,Origin,1));
[Ya Yb Yc] = split(Y,128,128);
Yb = Yb .* (abs(Yb)≥4);
Yc = Yc .* (abs(Yc)≥4);
Y = [Ya Yb; Yc];
X = dwt(3, Y, -1);
X = saturate(round(X), 1, 256);
norm(Origin-X) / norm(Origin)

```

如果真的要模擬 1:k 的影像壓縮，我們應該將原始影像訊號做越多次的凌波轉換越好。轉換後的訊號只能保留大約

$$\Gamma = \frac{MN}{k}$$

個元素，其他的都要設定成零。我們當然應該要保留絕對值最大的 Γ 個元素。例如在圖 47 中所呈現的分別模擬是 1:6 (上圖) 和 1:10 (下圖) 的壓縮後合成影像。它們與原始影像的相對誤差分別是 0.49% 和 0.73%。如果再提高壓縮比會產生怎樣的 effects 呢？只有影像中最有“特色”的部分會突顯出來，例如在圖 48 中，我們欣賞 1:30 和 1:100 的壓縮後合成影像。它們與原始影像的相對誤差也不過分別是 1.71% 和 3.45%。其實，正交凌波函數這套基底，在視覺效果上表現得並不是很好。存在另一些凌波函數，在視覺效果上的表現比較好。這句話的意

思是說，以這些凌波做基底的時候，即使在計算上得到同樣大小的相對誤差，但是影像的視覺效果會比較好。

習題 4.13 設計一套程序，模擬 $1:k$ 的影像壓縮，計算合成後影像與原始影像的相對誤差，並顯像目視其差別。

4.8 CZO 型式的測試矩陣

在這一節中，我們提出一個 CZO 型式的測試矩陣，並將它做幾種二維凌波轉換。我們討論測試的結果、比較分層轉換和全轉換的不同、比較週期化轉換和區間內轉換在圖形上的差異。

idx:CZO

令 $\mathbf{A} = (a_{ij})$ 是一個方陣

$$a_{ij} = \begin{cases} -1 & \text{if } i = j, \\ \frac{1}{\sqrt{|i-j|}} & \text{otherwise} \end{cases} \quad (4.22)$$

eq:testCZO

則 \mathbf{A} 稱為 CZO 型式的矩陣。 \mathbf{A} 是一個對稱的，在對角線上不連續但其餘部份平滑遞減的矩陣 (或函數)。如圖 49。這個測試矩陣代表了某一類積分算子的特性。許多用凌波函數解積分方程的人，利用這個矩陣作為一個標準測試。此處我們暫且不談積分算子。若我們將 \mathbf{A} 當做一個影像來看，這個影像最刺眼的地方就在對角線上，其他的部位都很普通。

雖然 \mathbf{A} 是一個數值矩陣，我們仍然可以把它視為一個影像。將週期化的凌波轉換應用在 \mathbf{A} 這個影像上，相當於將矩陣 \mathbf{A} 變換基底到凌波函數基底。令轉換後的矩陣為 $\mathbf{B} = (b_{ij})$ 。亦即

$$\mathbf{B} = \mathcal{W} \mathbf{A} \mathcal{W}^T$$

在前一節中，我們只以黑白兩色來呈現 $W_j(x, y)$ 空間中的係數：係數絕對值夠大的是黑色，否則是白色。現在，我們想要更精確地觀

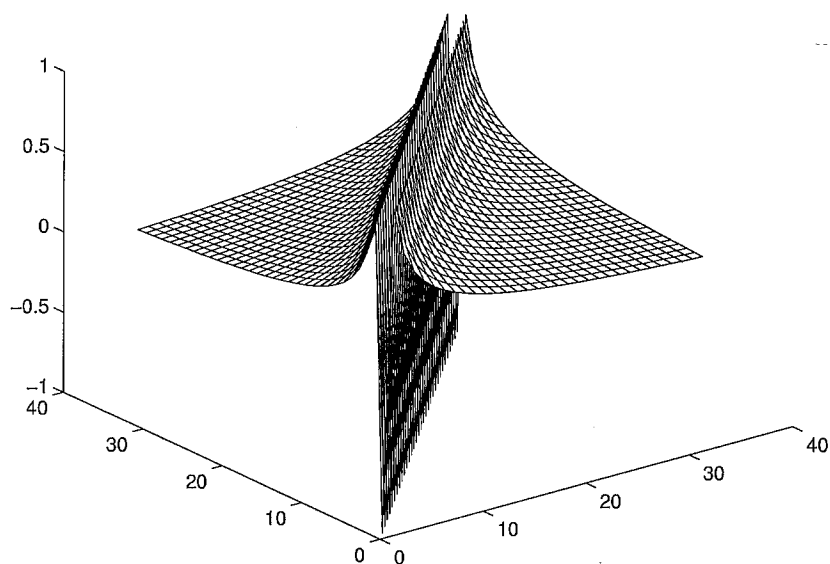


fig:testCZO 圖 49 CZO 型式的測試圖像

測凌波係數的大小，所以設計一種新的著色法。令 $M = \max_{i,j} |b_{ij}|$ ，將 B 中的元素用三種顏色呈現：

- if $|b_{ij}| < 10^{-6}M$ 用白色表示
- if $10^{-6}M \leq |b_{ij}| \leq 10^{-3}M$ 用灰色表示
- if $10^{-3}M < |b_{ij}| \leq M$ 用黑色表示

在圖 50 和 圖 51 中，我們列出使用 $p = 2, 4, 6, 8$ 的凌波函數作轉換之後的結果。

歸納圖 50 和 圖 51，其共同處如下：(1) 各做了四層的轉換。(2) 轉換後的大元素都集中在最低解析度的區塊、和每個區塊的對角線上。(3) 轉換後的大部分元素都很小 (白色)。(4) 上三角區塊的右邊和下三角區塊的下邊都有一些大的元素。基於 (2)-(3) 兩個理由，我們得以

壓縮 CZO 型式的影像或矩陣。等一下再談第四個現象。這四張圖的不同之處，在於當 p 比較小的時候，灰色地帶較大，可見壓縮比比較低；當 p 比較大的時候，灰色地帶較小，可見壓縮比比較高。這是因為 CZO 測試矩陣在對角線之外是平滑函數，所以高階的凌波函數可以發揮它的功能，迅速消除高解析度的係數。但是當 p 較大的時候，對角線上的黑色地帶也跟著放大。這是因為 CZO 測試矩陣在對角線上並不平滑，所以高階的凌波函數不能發揮它的功能，反而因為它的函蓋較寬（寬度是 $2p - 1$ ），所以有比較多的凌波跨過對角線，因而產生比較多的大元素。

二維凌波轉換的模式，除了前述的分層轉換之外，還有全轉換。當我們做影像處理的時候，所做的凌波轉換都是分層轉換。這是因為一旦影像被分解到高频空間 $W_j(x, y)$ 之後，就沒有必要再做分解（除非應用凌波包 (wavelet packets) 的想法）。

我們用數學空間記號來比較兩種轉換的不同。以兩層轉換為例，分層轉換的作法是

$$\begin{aligned} V_{j+1}(x, y) &= W_j(x, y) \oplus V_j(x, y) \\ &= W_j(x, y) \oplus (W_{j-1}(x, y) \oplus V_{j-1}(x, y)) \end{aligned}$$

全轉換的作法是 (以 7 層為例)

$$\begin{aligned} V_{j+1}(x, y) &= W_j(x, y) \oplus V_j(x, y) \\ &= W_j(x)W_j(y) \oplus W_j(x)V_j(y) \oplus V_j(x)W_j(y) \oplus V_j(x)V_j(y) \\ &= W_j(x)W_j(y) \oplus W_j(x)(W_{j-1}(y) \oplus V_{j-1}(y)) \\ &\quad \oplus (V_{j-1}(x) \oplus W_{j-1}(x))W_j(y) \\ &\quad \oplus (V_{j-1}(x) \oplus W_{j-1}(x))(W_{j-1}(y) \oplus V_{j-1}(y)) \end{aligned}$$

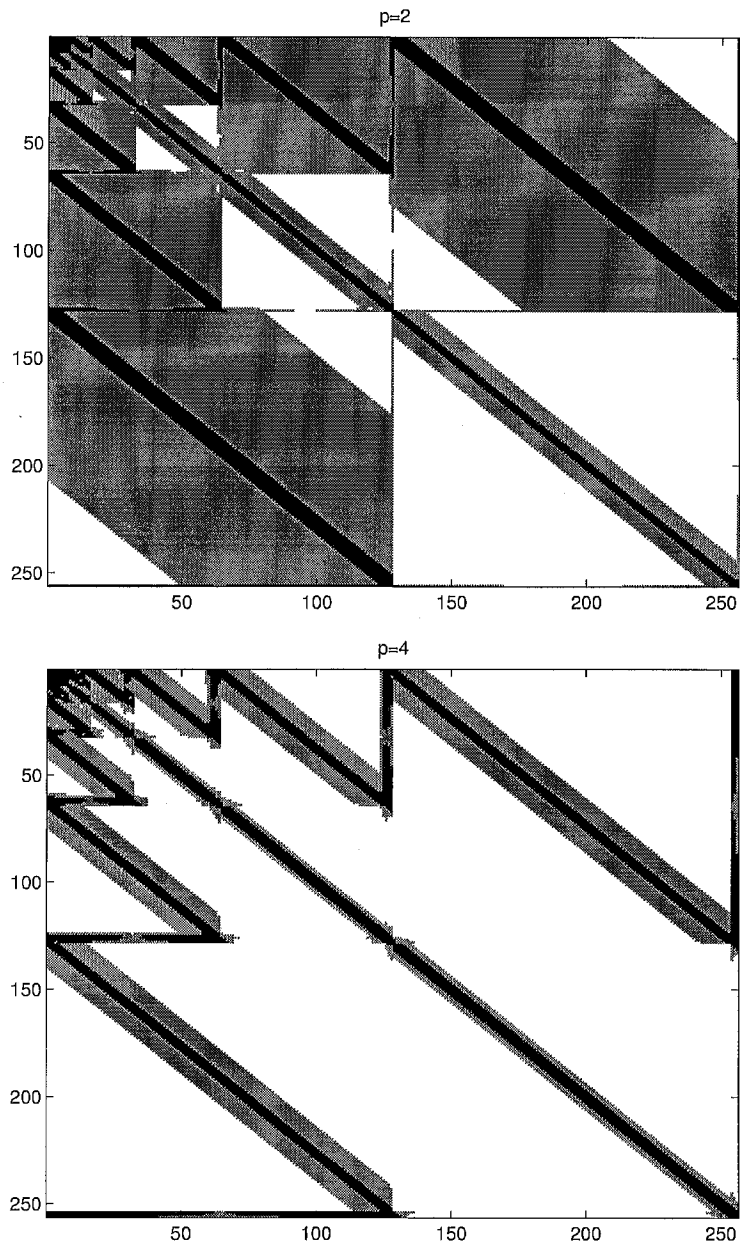


fig:twodfour 圖 50 CZO 測試矩陣經 $p = 2$ 和 $p = 4$ 凌波轉換

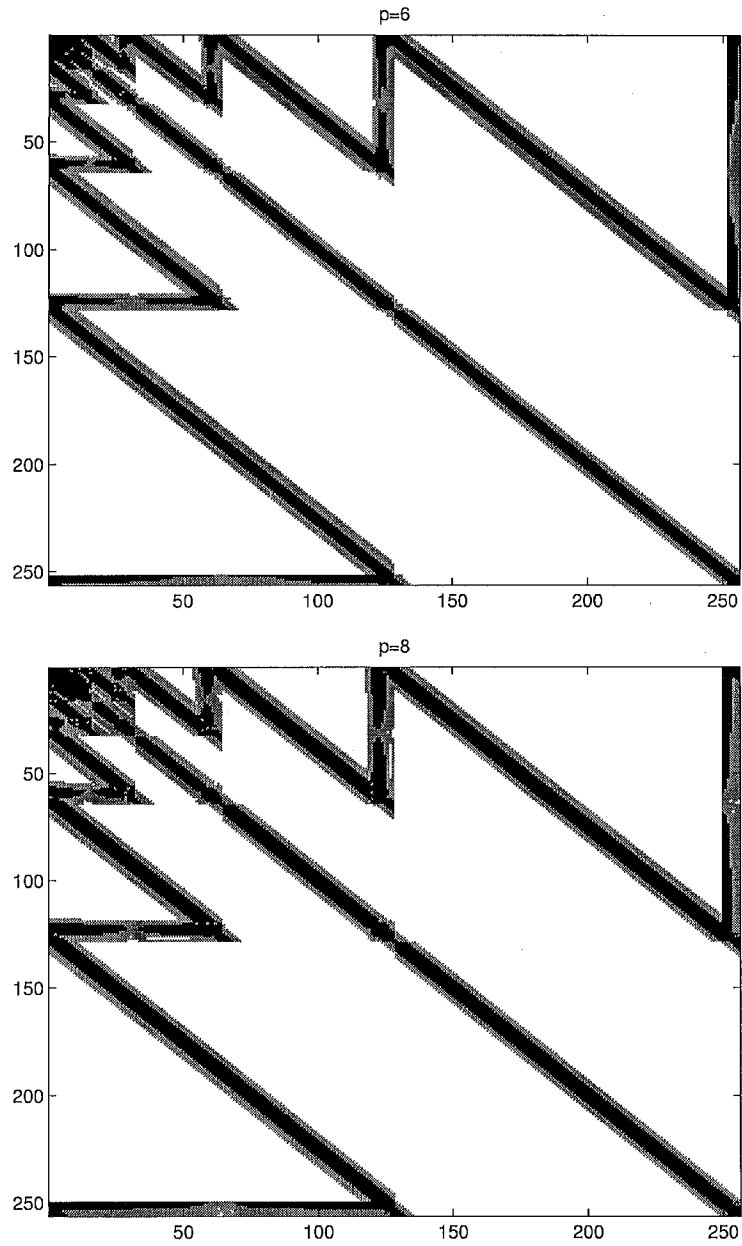


fig:twodeight 圖 51 CZO 測試矩陣經 $p = 6$ 和 $p = 8$ 凌波轉換

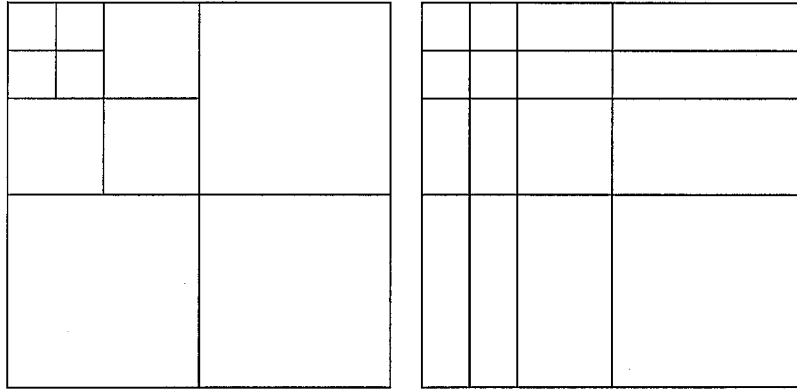


fig:decompA 圖 52 左邊是分層轉換、右邊是全轉換的圖示

請看圖 52 的圖示表達。

全轉換在實際運算上沒有任何困難，只是把輸入矩陣的每一行和每一列都個自作一維的凌波轉換。整體而言，全轉換的計算複雜度是分層轉換的兩倍，大致上它的轉換結果應該有比較多的小元素，所以應當有比較高的壓縮率。但是壓縮率未必能夠到達兩倍高。所以，拿計算複雜度比上其效能，這種轉換的效率比較低。但是如果做轉換的目的不是影像壓縮，而是解積分方程時候的矩陣壓縮，或是解微分方程時候的基底轉換，那麼都需要做全轉換。留待以後再談。

圖 53 就是 CZO 型式的測試矩陣經過全轉換的影像。對照圖 50 與圖 51 來看，就可以明顯看出來全轉換與分層轉換的不同。

最後，我們再觀察圖 51，發現只要使用週期化的凌波轉換，分解後的每個區塊的邊界上總會出現一些比較大的元素。這些元素在圖中表現得像畫框一樣，但是它們並非出現在每個區塊的四周。這又是在前一節提過的邊框效應。這些大元素來自於週期化所造成的不連續。將圖片週期化的結果，就是將圖片的左緣就接到了右邊的外側，上緣接到了下邊的外側。通常這樣接的結果就產生了不連續的情況，因為

idx:img-bdy

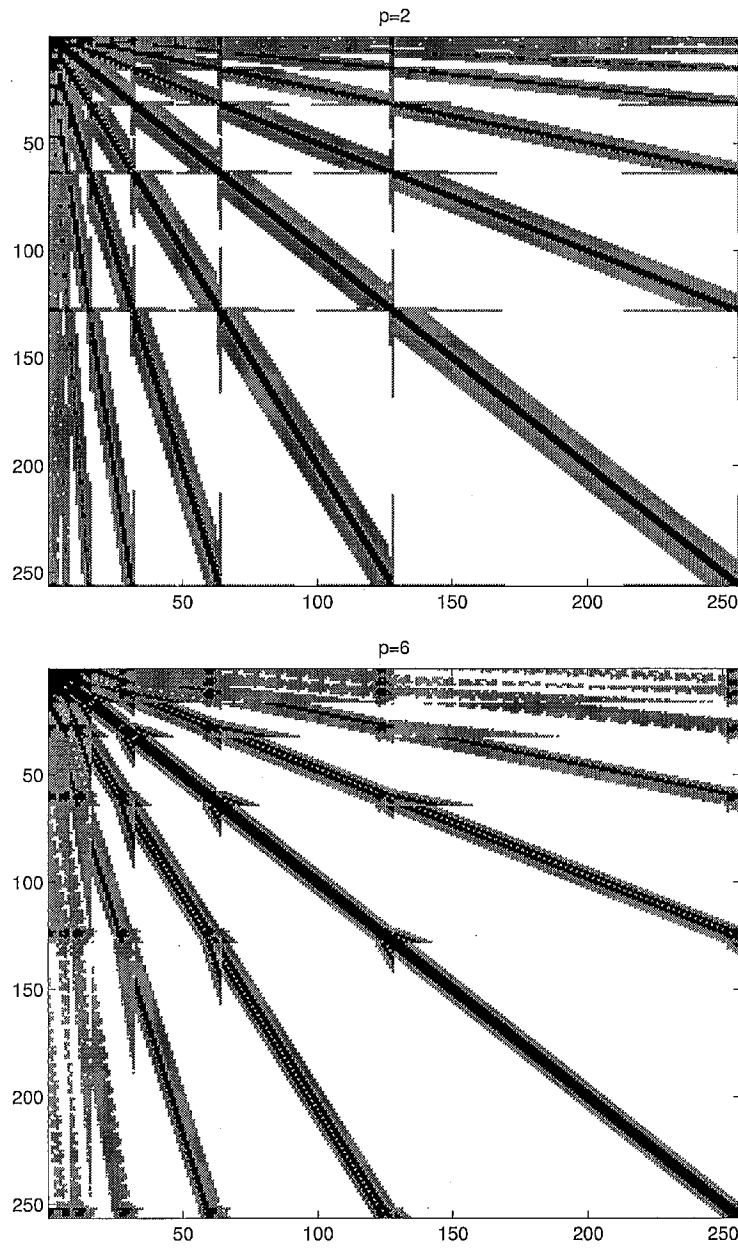


fig:twodfsix 圖 53 CZO 測試矩陣經 $p = 2$ 和 $p = 6$ 凌波做全轉換

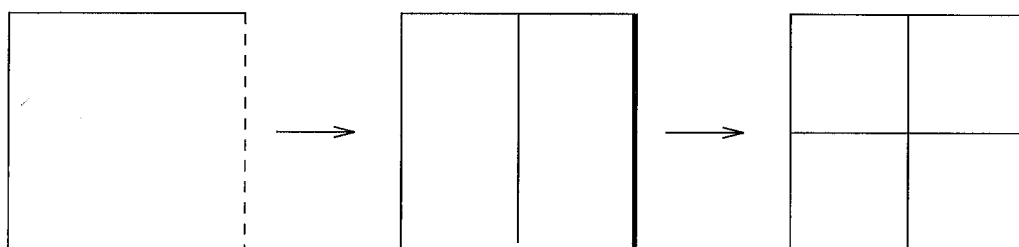


fig:imgbdy 圖 54 週期化凌波轉換的邊框效應

通常圖片的左緣並不會和右緣相連。由於在邊界產生了不連續或不平滑的一條介面，所以當凌波函數的函蓋跨過邊界的時候，就產生大的係數。這就解釋了，當 p 小的時候，由於函蓋較窄，故邊框效應的範圍較小。反而是當 p 大的時候，由於函蓋較寬，故邊框效應的範圍較大。

也值得一提的是，邊框效應並不發生在四個邊，觀察圖 51 最為明顯。這是因為由週期化造成的不連續介面只發生在右緣與下緣。以右緣為例，經 x 方向的凌波分解之後，造成高解析度部分靠右的大元素。再經過 y 方向的凌波分解，因為靠右的大元素並無 y 方向的劇烈變化，所以在分解後留在低解析度部分。如圖 54 所示，左圖的虛線部分代表不連續介面，中圖代表經過 x 方向的分解之後，右圖代表經過 y 方向分解的結果。

這些大的元素，會稍微降低壓縮率，或是會稍微提高壓縮後還原圖片在邊框附近的誤差。但是因為會產生較大誤差的部分在邊框，而且相對來說數目不多，所以通常人的眼睛不會察覺。

有一種定義在區間內的凌波函數。如果使用這種凌波函數來做轉換，就不需要將圖片做週期化的處理。因此就可以消除大部分在邊框附近因為週期化所造成的大元素。參見圖 55 (取自鄭儘女士的碩士論文 [20])，並與圖 43 和圖 50 對照。但是，區間內的凌波函數比較複雜，除

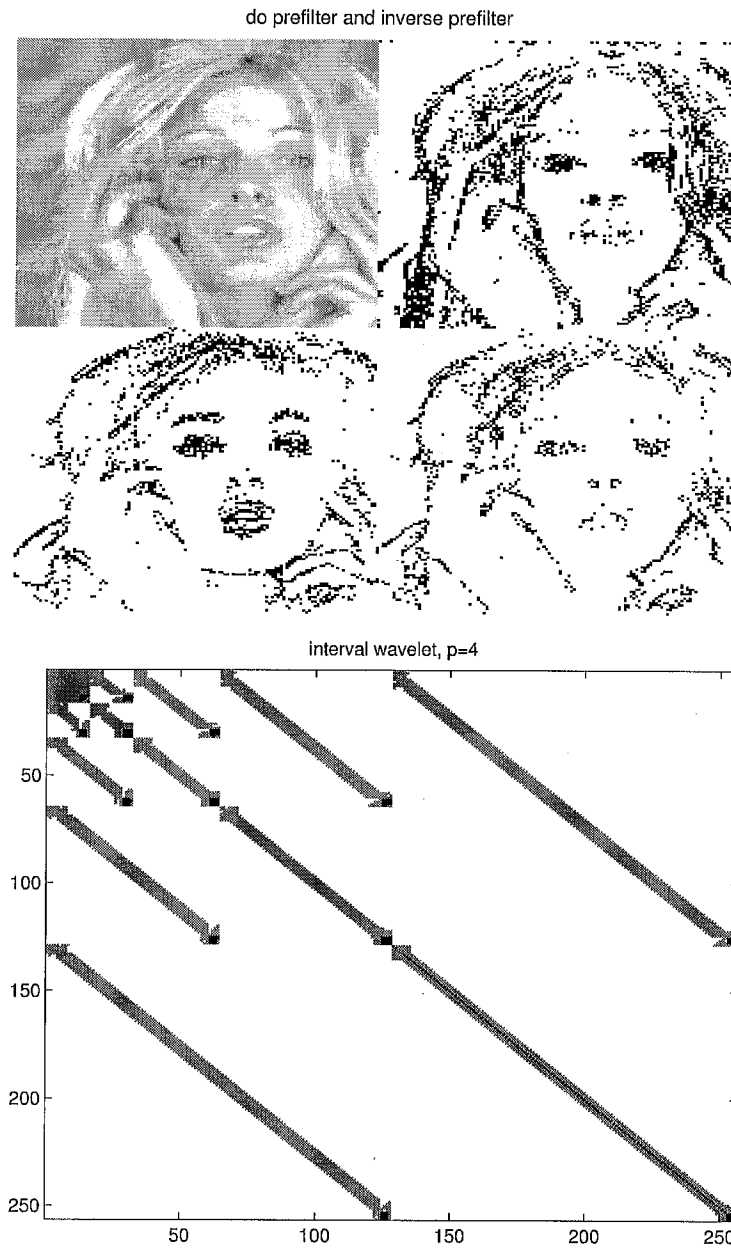


fig:twodjfour 圖 55 Tiffany 測試圖片和 CZO 測試矩陣經過區間內之凌波轉換

了需要兩組不同的濾波係數之外，在分解的前後還各需要執行預處理和後處理。因此，這些計算複雜度上的代價是否值得，還有待商榷。

4.9 一個數學之外的問題

當我們處理一組訊號或影像時，理論上應該是在處理一個連續函數。我們認定連續函數的解析度是無窮大，而人類的能量、資源和時間都是有限的，所以實際上人類只能處理有限解析度的數位訊號。數位訊號總是一組有限多的數字，以向量或矩陣的形態作為我們的輸入值。這一組數值是上述的連續函數，以某種方法將之離散化 (discretize) 與數位化 (quantize) 的結果。例如真正的影像是 Tiffany 的面容，而我們拿到的數位化影像是一個矩陣和色盤。在數位影像處理的領域中，所拿到的資料就是這個矩陣，所能處理的也就是這個矩陣。至於其數位化的方法與過程 (比如說，是用數位相機拍攝而得，還是拿沖印出來的照片掃描而得)，則不是我們所能控制的。

附帶一提，
...

為了敘述方便，我們假設一個單變數的函數 $f(x)$ 來代表真正的訊號，而它的數位化訊號是一個向量 $u = (u_k)$ 。根據我們的理論假設和數學模型，這一組數位訊號之數值 u_k 必須是 $f(x)$ 在某個解析空間 V_j 上的投影係數：

$$u_k = 2^j \langle f(x), \phi(2^j x - k) \rangle = \sqrt{2^j} \langle f(x), \phi_{jk}(x) \rangle$$

一般來說，我們無從驗證上述等式是否成立。有些讀者在仔細思考之後，會擔心這個問題 (確實值得擔心)。

其實我認為這已經不是個數學問題了，這是個哲學問題。您想想看，我們哪裡知道真實影像的函數式子呢？我們又哪裡可以肯定，經由電子設備取得的數位訊號，真的就是訊號函數映射到 V_j 空間的正交投影呢？這一切只是個數學模型。有了數學模型，我們才能做理論

推演，以便發現一些性質，或預測一些現象。數學模型總有一些假設或前提。而數學的推演，最多只能保證在這些假設或前提之下的正確性。

我們的確可以假想，存在另一個訊號函數 $\tilde{f}(x)$ ，使得

$$u_k = 2^j \langle \tilde{f}(x), \phi(2^j x - k) \rangle$$

那麼根據我們的數學模型，數位訊號 u 實際上是 $\tilde{f}(x)$ 而非 $f(x)$ 的數位化訊號。現在的問題是， $\tilde{f}(x)$ 和 $f(x)$ 到底相差多少呢？在哲學和科學的意義上，這個問題並不重要。因為即使它們不同，其不同處必須在高過 V_j 的解析層次上才看得出來。而我們根本無法得知那麼高的解析度；那一部分的資訊早已在數位化 $f(x)$ 而得到 u_k 的時候就喪失了。所以，在這個意義之下，我們既不必，也不能，分辨 $\tilde{f}(x)$ 和 $f(x)$ 的不同部分。

經過這番討論，從今後我們就認定數位訊號 u 就是訊號函數 $f(x)$ 映射到某個 V_j 空間的正交投影係數。這組數據 u_k 的意義是 V_j 中投影函數的基底分量。這個由 u_k 所代表的函數就是 $\mathcal{P}_j f(x) \in V_j$ ：

$$\mathcal{P}_j f(x) = \sum_k u_k \phi(2^j x - k) = \sum_k \frac{1}{\sqrt{2^j}} u_k \phi_{jk}(x) = \sum_k \langle f, \phi_{jk} \rangle \phi_{jk}(x)$$

$\mathcal{P}_j f(x)$ 就是我們所能處理的有限解析度訊號，而 u 就是我們所擁有的資訊。不要管它實際上是如何透過機械、電子或生物的工具，從 $f(x)$ 獲致 $\mathcal{P}_j f(x)$ 。因為那是數學模型之外的問題。

第五章

正交凌波函數 的數值運算範例

這一章的主題，是探討一些牽涉到自格函數或凌波函數的數值計算問題。這些問題包括投影、插值、製圖、數值積分。針對這些問題，我們提出一些方法，示範如何利用自格或凌波函數的特性來設計有效率的算法。這一章中大量地使用 Matlab 指令，但每道程序都有重點解釋。如有需要，亦請參考附錄。

5.1 繪製 V_j 中的函數圖形

這一節的主題是，給定一組係數 u_n ，令 $u(x) = \sum u_n \phi(2^j x - n)$ ，如何繪製 $u(x)$ 的圖形？首先我們用廣義平均值的觀念說明，當 j 夠大的時候只要繪製係數 u_n 的圖形就可以了。然後我們討論，當 j 不很大的時候要如何計算 $u(x)$ 的插值。我們介紹兩種計算方法。

假設一個函數 $u(x) \in V_j$ 。為了方便起見，我們考慮 $u(x)$ 在區間 $[0, N]$ 內的圖形，其中 N 是一個正整數。注意，因為

$$\text{supp } \phi(2^j x - n) = \left[\frac{n}{2^j}, \frac{n + 2p - 1}{2^j} \right]$$

所以函蓋住區間 $[0, N]$ 的 $\phi(2^j x - n)$ 包括

$$n = 2 - 2p, 3 - 2p, \dots, 0, 1, \dots, 2^j N - 1 \quad (5.1) \quad \text{eq:dimN}$$

共 $2^j N + 2p - 2$ 個。所以係數 u_n 也必須有 $2^j N + 2p - 2$ 個。所以

$$u(x) = \sum_{n=2-2p}^{2^j N-1} u_n \phi(2^j x - n) \quad (5.2) \quad \text{eq:xaa}$$

如果我們真的要製作 $y = u(x)$ 在區間 $[a, b]$ 內的圖形，嚴格來說應該根據 (5.2) 去組成 $u(x)$ ，並且在 x 坐標上標定均勻分佈的節點 x_k ，然後取 $u(x)$ 在節點 x_k 上的插值 $y_k = u(x_k)$ ，再利用標定點 (x_k, y_k) 製作圖形。以下我們要說明，當 j 夠大的時候，我們只需要簡約地製作向量 $u = (u_n)$ 的圖形。 idx:nodes

如果 j 夠大，我們令 $x_k = \frac{k}{2^j}$ ， $y_k = u(x_k)$ 。一般來說 $u_k \neq y_k$ ，所以製作 (x_k, u_k) 的圖形並不等於製作 (x_k, y_k) 的圖形。但是，其實只要 u_k 的數量夠多，也就是 j 夠大，那麼 $u_k \approx y_k$ ，所以直接製作 (x_k, u_k) 的圖形也就差不多是繪製了 $y = u(x)$ 的圖形。

現在我們在一個特殊情況下檢查 u_k 和 $u(x_k)$ 的差異。選取節點 $x_k = k/2^j$ ，則由 (5.2) 得

$$y_k = \sum_n u_n \phi(2^j x_k - n) = \sum_n u_n \phi(k - n)$$

經過足標變換 $k - n = m$ 我們得到

$$y_k = \sum_m u_{k-m} \phi(m)$$

但是 $\text{supp } \phi(x) = [0, 2p - 1]$ 而且它是連續函數 (當 $p > 1$)，所以 $\phi(m)$ 只有在 $m = 1, 2, \dots, 2p - 2$ 的時候才有非零的值。以 $p = 3$ 的 $\phi(x)$ 為

例，只有 $\phi(1)$ ， $\phi(2)$ ， $\phi(3)$ 和 $\phi(4)$ 的值是非零的，其他的整數點值都是 0。以 $k = 7$ 為例，則

$$y_7 = u(x_7) = u_6\phi(1) + u_5\phi(2) + u_4\phi(3) + u_3\phi(4)$$

但是，根據 (3.71) 我們知道 (代入 $x = 0$ 並做一次足標變換)

$$\sum_m \phi(m) = \sum_{m=1}^{2p-2} \phi(m) = 1$$

比如說當 $p = 3$ 的時候

$$\phi(1) \approx 1.2863 \quad \phi(2) \approx -0.3858 \quad \phi(3) \approx 0.0953 \quad \phi(4) \approx 0.0042$$

(我們將在後面介紹這些數值是怎麼算出來的。) 因此，我們發現 y_k 是

$$u_{k-1}, u_{k-2}, \dots, u_{k-2p+2}$$

的一個廣義平均值，或者說 y_k 是

$$u_{k-1}, u_{k-2}, \dots, u_{k-2p+2}$$

的一個外差值。

我們暫且對廣義平均值稍作解釋。任意給 n 個數 a_1, a_2, \dots, a_n ，則它們的平均值是

$$\frac{a_1 + a_2 + \dots + a_n}{n} \tag{5.3} \quad \text{eq:xab}$$

但是上式可以改寫成

$$\frac{1}{n}a_1 + \frac{1}{n}a_2 + \dots + \frac{1}{n}a_n := r_1 a_1 + r_2 a_2 + \dots + r_n a_n = \sum_{i=1}^n r_i a_i$$

此時的 $r_i = 1/n$ 所以顯然符合 $\sum r_i = 1$ 。因此，只要有任何一組參數 r_1, r_2, \dots, r_n 符合 $\sum r_i = 1$ ，則我們稱

$$\sum_{i=1}^n r_i a_i \quad (\text{where } \sum_{i=1}^n r_i = 1) \quad (5.4)$$

eq:xac

是 a_1, a_2, \dots, a_n 的廣義平均值 (*generalized average*)。

idx:gen-mean

如果 $u(x)$ 沒有急劇的變化，而且 x_k 相距夠近，也就是 j 夠大，那麼 $u_k \approx u_{k-1} \approx \dots \approx u_{k-2p+2}$ 。既然這些數值差不多相等，而 $y_k = u(x_k)$ 是 $u_{k-1}, u_{k-2}, \dots, u_{k-2p+2}$ 的廣義平均值，所以也就 $y_k \approx u_k$ 。

讓我們沿用前面的符號系統，做個簡短的結論：如果解析度的層次夠高，也就是 j 足夠大，那麼繪製 (x_k, u_k) 的圖形，差不多就等於繪製 $(x_k, u(x_k))$ 的圖形。因此沒有必要真的按照 (5.2) 組成 $u(x)$ 之後再製圖。

注意，這只是一個粗略的定性描述，至於比較詳細的分析，我們需要知道係數 u_k 的變化率。留到下一節再說。至於 j 要多大才夠大，也依賴於 u_k 的變化率。

如果 j 不夠大，那麼我們就需要按照 (5.2) 組成 $u(x)$ ，然後用比 $\frac{1}{2^j}$ 更密的節點來製圖。假設 $l > j$ 而我們令 $x_k = \frac{k}{2^l}$ 。則必存在唯一的整數 m 使得

$$\frac{m}{2^j} \leq \frac{k}{2^l} < \frac{m+1}{2^j}$$

其實這個 m 就是 $k \div 2^{\ell-j}$ 做整數除法的商。令

$$k = m \times 2^{\ell-j} + r \quad (\text{where } m, r \in \mathbb{Z}, 0 \leq r \leq 2^{\ell-j} - 1) \quad (5.5) \quad \text{eq:xad}$$

因為 (5.1)，可見函蓋住區間 $[\frac{m}{2^j}, \frac{m+1}{2^j})$ 的 $\phi(2^j x - n)$ 只有

$$n = m, m-1, \dots, m-(2p-2)$$

共 $2^p - 1$ 個。所以，由 (5.2) 和 (5.5) 得到

$$\begin{aligned}
 u(x_k) &= \sum_{i=0}^{2^p-2} u_{m-i} \phi\left(\frac{k}{2^{\ell-j}} - (m-i)\right) \\
 &= \sum_{i=0}^{2^p-2} u_{m-i} \phi\left(\frac{r + 2^{\ell-j} \cdot i}{2^{\ell-j}}\right)
 \end{aligned} \tag{5.6}$$

eq:compose

以 $p = 3, \ell = 4$ 和 $j = 2$ 為例，則

$$u(x_9) = u\left(\frac{9}{16}\right) = u_2\phi\left(\frac{1}{4}\right) + u_1\phi\left(\frac{5}{4}\right) + u_0\phi\left(\frac{9}{4}\right) + u_{-1}\phi\left(\frac{13}{4}\right) + u_{-2}\phi\left(\frac{17}{4}\right)$$

而且

$$u(x_{28}) = u\left(\frac{28}{16}\right) = u_7\phi(0) + u_6\phi(1) + u_5\phi(2) + u_4\phi(3) + u_3\phi(4)$$

以下，我們要根據 (5.6) 設計兩種 $u(x_k)$ 的方法。這兩種方法都將需要 $\phi\left(\frac{k}{2^{\ell-j}}\right)$ 的值。我們以後再回來解決這個問題。第一種方法可能比較適合當 $N \geq 2^{\ell-2j}$ 的情況，第二種方法可能比較適合當 $N < 2^{\ell-2j}$ 的情況。

第一種方法。回顧 (2.22)，我們發現 (5.6) 中的計算形式是 $u = (u_n)$ 與 $\phi\left(\frac{n}{2^{\ell-j}}\right)$ 的離散疊積。所以可以利用 Matlab 中的 `conv` 指令來實現上述的演算法。使用者輸入正整數 p, N, ℓ 和 j ，以及兩個序列

idx:dconvol

$$\begin{aligned}
 \mathbf{u} &= (u_{2-2^p}, \dots, u_{-1}, u_0, u_1, \dots, u_{2^j N - 1}) \\
 \mathbf{v} &= \left(\phi\left(\frac{0}{2^{\ell-j}}\right), \phi\left(\frac{1}{2^{\ell-j}}\right), \phi\left(\frac{2}{2^{\ell-j}}\right), \dots, \phi\left(\frac{2^{\ell-j}(2^p - 1) - 1}{2^{\ell-j}}\right)\right)
 \end{aligned} \tag{5.7}$$

eq:xae

注意 $\ell \geq j$ 而且 \mathbf{u} 和 \mathbf{v} 的長度分別是

$$\#\mathbf{u} = 2^j N + 2^p - 2, \quad \#\mathbf{v} = 2^{\ell-j}(2^p - 1)$$

而且要注意輸入 u 和 v 時，它們都是橫向量，稱為序列 (*array*)，不是直的向量 (*vector*)。令 $u(x)$ 如 (5.2) 所定義，我們用 Code-15 獲得以下向量：

$$y = (u(0), u(\frac{1}{2^\ell}), u(\frac{2}{2^\ell}), \dots, u(N - \frac{1}{2^\ell})) \quad (5.8)$$

idx:array
idx:vector

eq:xaf
code:composeC

Matlab Code-15

```
nv = length(v);
nu = length(u);
ny = 2^l*N;
nh = 2^(l-j);
ns = 2*p-1;
y = zeros(1,ny+1);
for r = 1:nh,
    t = conv(v(r:nh:nv),u);
    y(r:nh:ny) = t(ns:nu);
end
```

至於 $u(N)$ 的值，可以做一個內積獲得：

code:composeE

Matlab Code-16

```
y(ny+1) = v(1:nh:nv) * [0 u(nu:-1:(nu-2*p-3))]';
```

程序 Code-15 大概只有兩點需要解釋。conv 就是 Matlab 做離散疊積的指令。而每次做疊積的時候，我們從 v 中每隔 $2^{\ell-j}$ 取一個元素出來。依前面的例子 ($p = 3, \ell = 4$ 和 $j = 2$)，當 $r = 1$ 的時候取出來的序列是

$$\phi(\frac{0}{4}), \phi(\frac{4}{4}), \phi(\frac{8}{4}), \dots$$

而疊積的結果是

$$u\left(\frac{0}{16}\right), u\left(\frac{4}{16}\right), u\left(\frac{8}{16}\right), \dots$$

當 $r = 2$ 的時候取出來的序列是

$$\phi\left(\frac{1}{4}\right), \phi\left(\frac{5}{4}\right), \phi\left(\frac{9}{4}\right), \dots$$

而疊積的結果是

$$u\left(\frac{1}{16}\right), u\left(\frac{5}{16}\right), u\left(\frac{9}{16}\right), \dots$$

依此類推。這就是為什麼將疊積的結果間隔寫入 $y(r:nh:ny)$ 。但是並非所有的疊積結果都是有用的，我們必須去除前面的 $2^p - 2$ 個結果，並且只取

$$\frac{\#y}{2^{\ell-j}} = 2^j N$$

個元素出來。這就是為什麼只取 $t(ns:nu)$ 。

至於為什麼要去除前面幾個疊積的結果，就必須知道 `conv` 指令的性質。舉個簡單的例子來說明：比如 $u = (u_1, u_2, u_3)$ ， $v = (v_1, v_2)$ 則 `idx:conv` 疊積的結果 $x = \text{conv}(u, v)$ 長度是 $\#x = \#u + \#v - 1$ 。

$$\begin{aligned} x_1 &= u_1 v_1 \\ x_2 &= u_1 v_2 + u_2 v_1 \\ x_3 &= u_2 v_2 + u_3 v_1 \\ x_4 &= u_3 v_2 \end{aligned}$$

在前面的程序中，疊積後的前面 $2^p - 2$ 個元素依序是

$$\begin{aligned} &u_{2-2^p} \phi\left(\frac{r}{2^{\ell-j}}\right) \\ &u_{2-2^p} \phi\left(\frac{r+2^{\ell-j}}{2^{\ell-j}}\right) + u_{3-2^p} \phi\left(\frac{r}{2^{\ell-j}}\right) \\ &\vdots \\ &u_{2-2^p} \phi\left(\frac{r+2^{\ell-j}(2^p-3)}{2^{\ell-j}}\right) + \dots + u_{-1} \phi\left(\frac{r}{2^{\ell-j}}\right) \end{aligned}$$

所以這些結果是沒有用的。從第 $2p - 1$ 個結果開始，才是我們要求的

$$u(0) = u_{2-2p} \phi\left(\frac{r + 2^{\ell-j}(2p-2)}{2^{\ell-j}}\right) + \cdots + u_0 \phi\left(\frac{r}{2^{\ell-j}}\right)$$

第二種方法。我們將 $[0, N]$ 區間切成像以下

$$I_m = \left[\frac{m}{2^j}, \frac{m+1}{2^j}\right), \quad m = 0, 1, \dots, 2^j N - 1 \quad (5.9) \quad \text{eq: xaj}$$

這樣一段一段的區間來看。則 $u(x)$ 在 I_m 區間內的值是

$$u(x) \Big|_{I_m} = \sum_{i=0}^{2p-2} u_{m-i} \phi(2^j x - (m-i)) \Big|_{I_m}$$

其實，經過平移， $\phi(2^j x - (m-i))$ 在 I_m 內的值就是 $\phi(2^j x)$ 在 $[\frac{i}{2^j}, \frac{i+1}{2^j})$ 內的值。而經過脹縮， $\phi(2^j \cdot \frac{k}{2^{\ell}})$ 在 $[\frac{i}{2^j}, \frac{i+1}{2^j})$ 內的值就是 $\phi(x)$ 在 $[i, i+1)$ 之中

$$x = \frac{k}{2^{\ell-j}}$$

的點值。所以，如同第一個方法，輸入正整數 p, N, ℓ 和 j ，以及兩個序列 u 及 v 如 (5.7) 所述。則以下的 Matlab 程序將獲得 (5.8) 中的 y 。

code: composeF

Matlab Code-17

```

nh = 2^(l-j);
ns = 2*p-1;
nb = 1;
for i=1:ns,
    P(i,:) = v(nb:i*nh);
    nb = nb+nh;

```

```

end
y = zeros(1,2^l*N+1);
nb = 1;
ns = ns-1;
for m = 1:2^j*N,
    y(nb:m*nh) = u(m+ns:-1:m) * P;
    nb = nb+nh;
end

```

至於 $u(N)$ 的值，還是可以利用 Code-16 求得。

我們沿用前面的例子 ($p = 3$, $\ell = 4$ 和 $j = 2$) 來說明這個程序的工作。以程序中的 $m = 4$ 為例，其實它對應了 (5.9) 中的 $m = 3$ 。我們要在 $I_3 = [\frac{3}{4}, \frac{4}{4})$ 之間算出

$$u\left(\frac{12}{16}\right), u\left(\frac{13}{16}\right), u\left(\frac{14}{16}\right), u\left(\frac{15}{16}\right)$$

共四個數 (也就是 $2^{\ell-j}$)，這就是 y 中的一段 $y(nb:m*nh)$ 。而

$$\begin{aligned}
 u\left(\frac{12}{16}\right) &= \sum_{i=0}^4 u_{3-i} \phi\left(\frac{4i}{4}\right), & u\left(\frac{13}{16}\right) &= \sum_{i=0}^4 u_{3-i} \phi\left(\frac{1+4i}{4}\right) \\
 u\left(\frac{14}{16}\right) &= \sum_{i=0}^4 u_{3-i} \phi\left(\frac{2+4i}{4}\right), & u\left(\frac{15}{16}\right) &= \sum_{i=0}^4 u_{3-i} \phi\left(\frac{3+4i}{4}\right)
 \end{aligned}$$

當我們用 Matlab 編寫程序的時候，務必盡量將迭代的過程寫成向量或矩陣的運算，這樣才會利用到 Matlab 快速的矩陣計算功能，寫出來的程式才會比較有效率。根據前面的四條式子，觀察它們可以用向量與

idx:matlab

矩陣的乘法表達：

$$\begin{aligned} & \left(u\left(\frac{12}{16}\right), u\left(\frac{13}{16}\right), u\left(\frac{14}{16}\right), u\left(\frac{15}{16}\right) \right) \\ &= \left(u_3, u_2, u_1, u_0, u_{-1} \right) \begin{pmatrix} \phi\left(\frac{0}{4}\right) & \phi\left(\frac{1}{4}\right) & \phi\left(\frac{2}{4}\right) & \phi\left(\frac{3}{4}\right) \\ \phi\left(\frac{4}{4}\right) & \phi\left(\frac{5}{4}\right) & \phi\left(\frac{6}{4}\right) & \phi\left(\frac{7}{4}\right) \\ \phi\left(\frac{8}{4}\right) & \phi\left(\frac{9}{4}\right) & \phi\left(\frac{10}{4}\right) & \phi\left(\frac{11}{4}\right) \\ \phi\left(\frac{12}{4}\right) & \phi\left(\frac{13}{4}\right) & \phi\left(\frac{14}{4}\right) & \phi\left(\frac{15}{4}\right) \\ \phi\left(\frac{16}{4}\right) & \phi\left(\frac{17}{4}\right) & \phi\left(\frac{18}{4}\right) & \phi\left(\frac{19}{4}\right) \end{pmatrix} \end{aligned}$$

上面那個矩陣的第一列 (row) 就是 $\phi(x)$ 在 $[0, 1)$ 中的 $\frac{k}{2^{\ell-j}}$ 點值，第二列就是 $\phi(x)$ 在 $[1, 2)$ 中的 $\frac{k}{2^{\ell-j}}$ 點值，依此類推。所以那個矩陣的維度是 $(2p-1) \times 2^{\ell-j}$ 。這就是為什麼我們先將序列 \mathbf{v} 分列改成矩陣 P ，然後再做向量與矩陣的乘法 $\mathbf{u}(m+ns:-1:m)*P$ 。

5.2 函數插值與投影

在這一節裡，我們仍然討論像 (5.2) 中 $u(x)$ 的製圖問題。但是現在我們假設係數 u_n 來自於一個已知的函數 $f(x)$ 。考慮兩種情形： u_n 是 $f(x)$ 的投影係數或是 u_n 就是 $f(x)$ 的插值。如果 j 夠大，這兩種情形相差不多。假設 $f(x)$ 具有連續的一次導函數，則我們可以推導這兩者之間的差異。我們舉一個例子表現，當 j 夠大的時候， $f(x)$ 的投影函數和插值函數相去不遠；再舉一個例子表現，當 j 不很大的時候，插值函數的偏差。最後討論造成偏差的原因；我們談到插值與對稱兩個性質。

爲了符號的方便起見，令 $f_j(x) = \mathcal{P}_j f(x)$ 則

idx:Pj

$$f_j(x) = \sum_k \langle f, \phi_{jk} \rangle \phi_{jk}(x) = \sum_k \sqrt{2^j} \langle f, \phi_{jk} \rangle \phi(2^j x - k)$$

嚴格來說，我們必須計算 $\langle f, \phi_{jk} \rangle$ 以求得 $f_j(x)$ 。但是這個計算乃是一個積分問題，通常是不能得到真解的，而必須訴諸於數值積分。稍後我們會講到有關自格與凌波函數的數值積分。此處要討論的是，如果解析的層次夠高，也就是 j 足夠大，那麼 $\sqrt{2^j} \langle f, \phi_{jk} \rangle \approx f(\frac{k}{2^j})$ ，所以

$$\mathcal{P}_j f(x) \approx \sum_k f\left(\frac{k}{2^j}\right) \phi(2^j x - k)$$

這個敘述很類似於前一節的定性描述，但是在此節中我們假設一個已知的 $f(x)$ ，所以可以做比較詳細的分析。為了符號上的方便，我們令

$$\tilde{f}_j(x) = \sum_k f\left(\frac{k}{2^j}\right) \phi(2^j x - k)$$

如果函數 $f(x)$ 至少一次可微而且其導函數連續，那麼我們可以利用泰勒展開得到以下的估計。

idx:Taylor

$$\begin{aligned} \langle f(x), \phi_{jk}(x) \rangle &= \int f(x) \phi_{jk}(x) dx \\ &= \sqrt{2^j} \int_{\frac{k}{2^j}}^{\frac{k+2p-1}{2^j}} f(x) \phi(2^j x - k) dx \\ &= \sqrt{2^j} \int_{\frac{k}{2^j}}^{\frac{k+2p-1}{2^j}} \left[f\left(\frac{k}{2^j}\right) + f'(\xi) \left(x - \frac{k}{2^j}\right) \right] \phi(2^j x - k) dx \\ &= \frac{1}{\sqrt{2^j}} f\left(\frac{k}{2^j}\right) + \sqrt{2^j} \int_{\frac{k}{2^j}}^{\frac{k+2p-1}{2^j}} f'(\xi) \left(x - \frac{k}{2^j}\right) \phi(2^j x - k) dx \end{aligned} \quad (5.10)$$

eq:xat

其中 ξ 是介於 $\frac{k}{2^j}$ 與 $\frac{k+2p-1}{2^j}$ 之間的某個數，我們不知道它是誰。但是，由於 $f'(x)$ 是連續函數，存在

$$M = \max_{\frac{k}{2^j} \leq \xi \leq \frac{k+2p-1}{2^j}} |f'(\xi)|$$

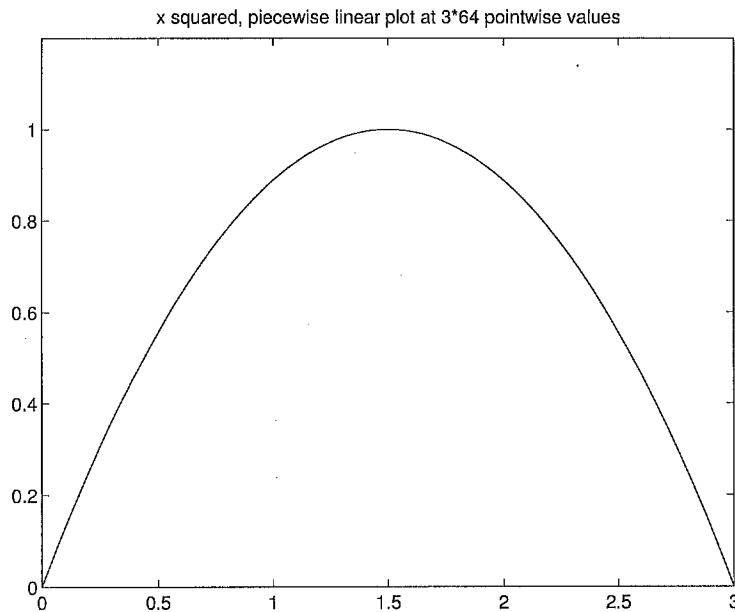


fig:xtwo 圖 56 $\frac{4}{9}x(3-x)$ 在 $[0, 3]$ 以 $\frac{1}{64}$ 為間隔的折線圖

所以 (5.10) 中最後的積分式之上界為

$$\sqrt{2^j} \int_{\frac{k}{2^j}}^{\frac{k+2^p-1}{2^j}} f'(\xi) \left(x - \frac{k}{2^j}\right) \phi(2^j x - k) dx \leq \sqrt{2^j} M \int_0^{2^p-1} \frac{x}{2^j} \phi(x) \frac{1}{2^j} dx$$

取 $M_1 = \int_S x \phi dx$ ，則推導出來

$$\langle f(x), \phi_{jk}(x) \rangle \leq \frac{1}{\sqrt{2^j}} f\left(\frac{k}{2^j}\right) + \frac{M M_1}{2^j \sqrt{2^j}} S \quad (5.11) \quad \text{eq:xag}$$

使用大 O 符號，我們將上式簡寫做

$$\langle f(x), \phi_{jk}(x) \rangle = \frac{1}{\sqrt{2^j}} f\left(\frac{k}{2^j}\right) + O\left(\frac{1}{2^j \sqrt{2^j}}\right) \quad (5.12) \quad \text{eq:xah}$$

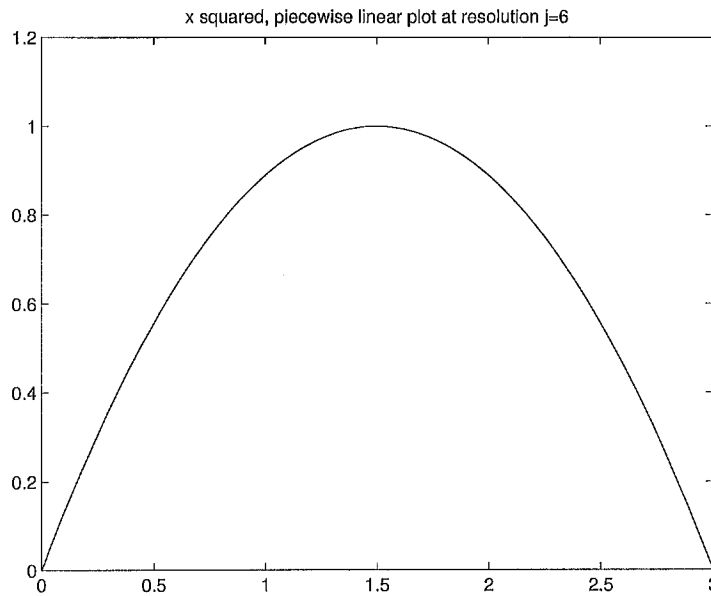


fig:xtwoproj 圖 57 $\frac{4}{9}x(3-x)$ 在 V_6 上做投影後的折線圖

因此

$$\begin{aligned}
 \mathcal{P}_j f(x) &= \sum_k \langle f, \phi_{jk} \rangle \phi_{jk}(x) \\
 &= \sum_k \sqrt{2^j} \langle f, \phi_{jk} \rangle \phi(2^j x - k) \\
 &= \sum_k \left(f\left(\frac{k}{2^j}\right) + O\left(\frac{1}{2^j}\right) \right) \phi(2^j x - k) \\
 &= \sum_k f\left(\frac{k}{2^j}\right) \phi(2^j x - k) + O\left(\frac{1}{2^j}\right)
 \end{aligned}$$

結論是

$$\mathcal{P}_j f(x) = \tilde{f}_j(x) + O\left(\frac{1}{2^j}\right)$$

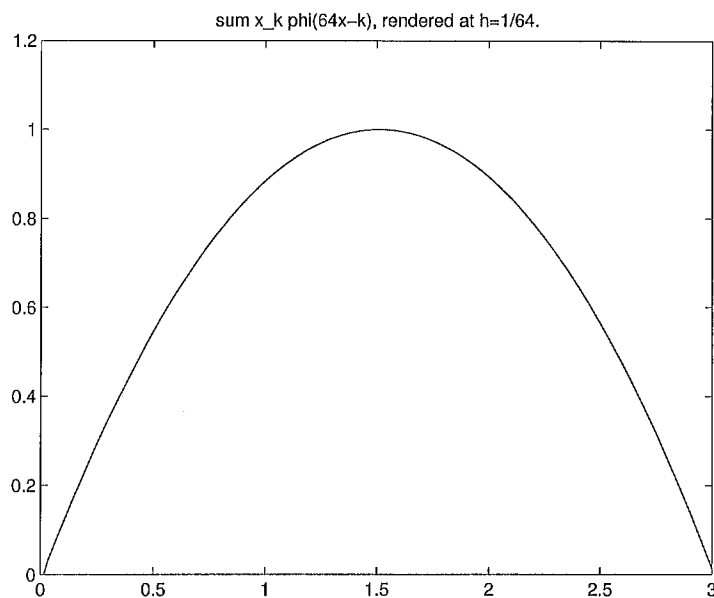


fig:xtwophi 圖 58 $\sum f(\frac{k}{64})\phi(64x-k)$ 以 $\frac{1}{64}$ 為間隔的折線圖

簡單地說，真正求積分做投影的 $\mathcal{P}_j f(x)$ 和拿插值做合成係數的 $\tilde{f}_j(x)$ 之間的誤差大約是 2^{-j} 。因此當 j 夠大的時候，這兩個函數差不多一樣。 idx:interp

我們做一個簡單的實驗。令 $f(x) = \frac{4}{9}x(3-x)$ ，我們只截取 $x \in [0, 3]$ 的部份。我們在 x -軸上以 $\frac{1}{64}$ 為間隔取均勻節點來製圖（對照前一節的符號，也就是取 $\ell = 6$ ）。圖 56 是 $f(x)$ 的圖形，圖 57 是投影函數

$$\mathcal{P}_6 f(x) = f_6(x) = \sum_{k=0}^{192} \langle f, \phi_{6,k} \rangle \phi_{6,k}(x)$$

的圖形。此處選用了 $p = 2$ 的自格函數。由於 $f(x)$ 和 $f_6(x)$ 之間只有 0.01% 的相對誤差：

$$\frac{\|f(x) - f_6(x)\|}{\|f(x)\|} \approx 10^{-4}$$

所以讀者們看不出來圖 56 和圖 57 的差別。

最後，圖 58 是用 $f(x)$ 的插值所製造的合成函數

$$\tilde{f}_6(x) = \sum_{k=0}^{192} f\left(\frac{k}{64}\right)\phi(64x - k)$$

的圖形。注意，如果您要使用前一節的 Matlab 程序來製作 $\tilde{f}_6(x)$ 的圖形，必須添上 $k = -1, -2$ 的係數；我們令這兩個係數是 0。雖然 $f(x)$ 和 $\tilde{f}_6(x)$ 之間有 1% 的相對誤差，但是仍然很難看出來圖 58 和前兩張圖的差別。這個實驗結果符合我們的理論：當 j 夠大而且 $f(x)$ 一次可微連續的時候， $f(x)$, $f_j(x)$ 和 $\tilde{f}_j(x)$ 其實都差不多一樣。

我們要追述一點。所謂 j 夠不夠大，是個性質上的描述。我們並沒有一個明確的規則。讀者應該回顧 (5.11)，如果 M 和 M_1 都比較大，那麼當然 j 就要比較大才能使得誤差項變得夠小。

注意，如果 j 不夠大，那麼前述的幾個函數就不會相近。底下我引用陳聲偉先生碩士論文 [1] 中的一個實驗，來表現當 j 不夠大的時候， $f(x)$ 和 $\tilde{f}_j(x)$ 這兩個函數圖形有顯著差異。在這個實驗中，他選擇了 $f(x) = x(1-x)$ ，並且局限在 $[0, 1]$ 區間中討論。他選擇了 $p = 3$ 的自格函數。在圖 59 中的實線都是 $f(x)$ 的折線圖（節點間隔是 $\frac{1}{32}$ ）。圖 59 上圖中的虛線是 $\tilde{f}_4(x)$ ，下圖中的虛線是 $\tilde{f}_5(x)$ 。

我們可以分幾點來說明這個實驗的意義。首先，注意 $\tilde{f}_4(x)$ 在靠近左端點的地方最不準確，這是因為沒有添上 $k = -1, -2, \dots, 2 - 2p$ 這些節點的插值，所以在靠近左端點的地方缺少了資料。

其次，我們發現 $\tilde{f}_4(x)$ 明顯地向左偏移。這是因為 $\phi(x)$ 的能量都集中在左邊（參考圖 61 和圖 62）。如果 $\phi(x)$ 是個對稱函數（對稱於 $\text{supp } \phi(x)$ 的中央）那麼就不會發生偏移的現象。這就是做影像處理的人喜歡對稱基底函數的原因之一。可惜 Daubechies 的正交自格與凌波函數是不可能對稱的一並不是還沒發現，而是數學上已經證明那是不可

idx:interp

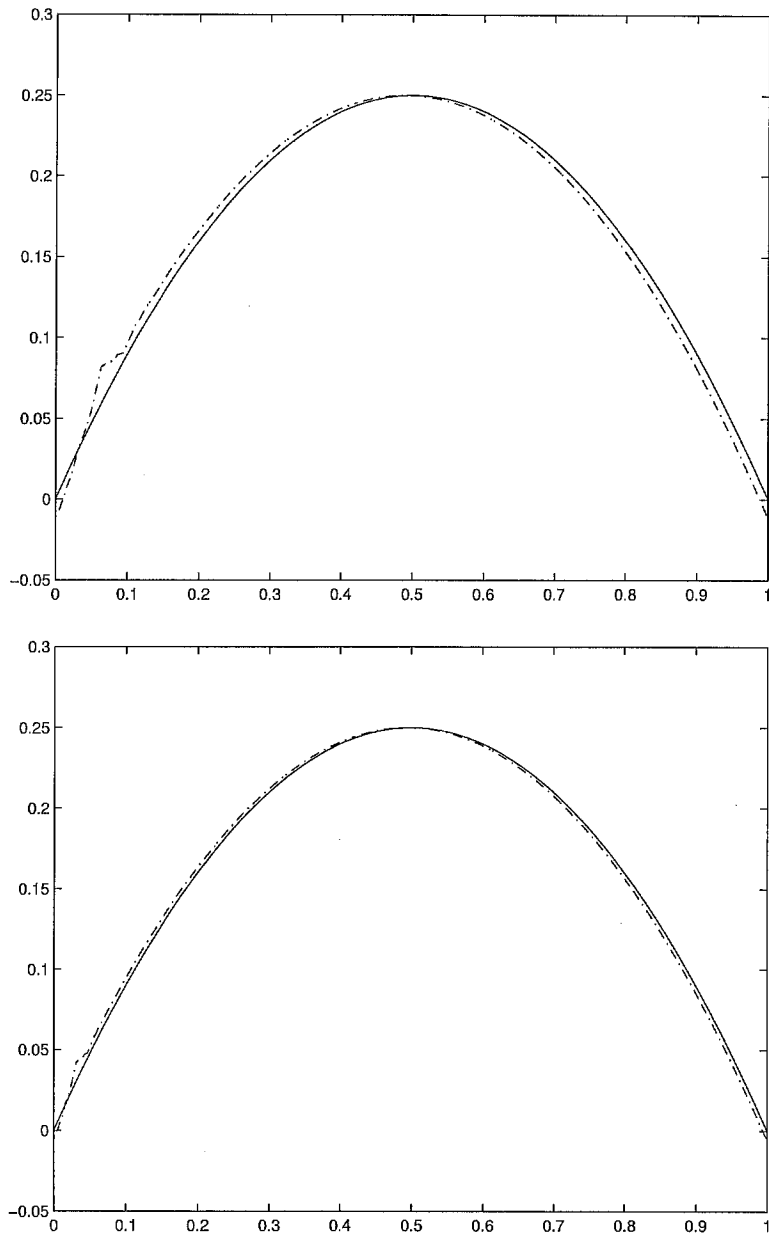


fig:chen 圖 59 $f(x) = x(1-x)$ 和 $\tilde{f}_4(x)$ 、 $\tilde{f}_5(x)$

可能的。更準確地說，自格函數與凌波函數的以下三個性質

$\underbrace{\text{正交、有限函蓋、圖形對稱}}_{\text{不共戴天}}$

最後，我們觀察 $\tilde{f}_4(x)$ 與 $\tilde{f}_5(x)$ 是整個偏開，除了靠近 $\frac{1}{2}$ 附近與 $f(x)$ 有一個交點之外，它們兩者完全不相交。也就是說幾乎對所有的點 $x \in [0, 1]$ 都 $\tilde{f}_j(x) \neq f(x)$ 。當 j 越來越大， $\tilde{f}_j(x)$ 會越來越靠近 $f(x)$ ，但是這兩條曲線幾乎沒有交點。這並不是一個討人喜歡的性質。這是因為 $\phi(x)$ 不是所謂的插值函數。如果存在某組節點 x_k ，使得

idx:interp

$$\phi(x_k) = \begin{cases} 1 & \text{if } k = 0 \\ 0 & \text{otherwise} \end{cases}$$

那麼這時候 $\phi(x)$ 就是一個插值函數。如果 $\phi(x)$ 是一個插值函數，那麼 $f(x)$ 和 $\tilde{f}_j(x)$ 在某些節點上就總是相等的。可惜 Daubechies 的正交自格與凌波函數不可能是插值函數。有趣的是， $\phi(x)$ 在頻率域上的傅立葉轉換 $\hat{\phi}(\omega)$ 卻是個插值函數，請回顧 (3.70)。更準確地說，自格函數與凌波函數的以下三個性質

$\underbrace{\text{正交、有限函蓋、插值函數}}_{\text{不共戴天}}$

爲了這些不共戴天的限制，人們有需要設計其他種類的自格與凌波函數，使得它們犧牲上述三件性質的其中一兩件，換取其他在某些目的上更重要的性質。魚與熊掌不可兼得，在這裡出現最好的寫照。我們將在後面略述其他種類的凌波函數。

5.3 自格與凌波函數的數值

這一節的目的是計算 $\phi(x)$ 和 $\psi(x)$ 在所謂的二次分點的值。根據自格等式，我們只需要知道 $\phi(x)$ 在整數點的值，就可以算出所有二次分點的值。而整數點的值可以由一個矩陣的特徵向量求得。

根據前兩節的討論，我們得以推論， $\phi(x)$ 和 $\psi(x)$ 本身的函數值，在大部分的應用問題中都不重要。因為數位訊號之取樣頻率通常都夠高，所以解析的層次夠高，因此我們並不需要根據 $\phi(2^j x - k)$ 的數值來合成 $\sum u_k \phi(2^j x - k)$ 的數值或圖形。但是在前面我們看到，一旦我們真的需要合成一個 V_j 中的函數時，我們就需要知道 $\phi(x)$ 在某些點上的值。還有，在有關求解方程的應用上，我們經常需要做數值積分；那時候也需要 $\phi(x)$ 的點值。

回顧 (5.6)，我們需要的是

$$\phi\left(\frac{k}{2^j}\right), \quad (\text{where } j \geq 0, 0 \leq k \leq 2^j(2p-1))$$

這些數值。所有 $\frac{k}{2^j}$ 這種形式的點稱作二次分點 (*dyadic point*)。固定一個 j 之後， $\frac{k}{2^j}$ 這種形式的節點稱作 2^j 分點。譬如說 $\frac{k}{64}$ 這種形式的節點稱作 64 分點。如果我們知道 $\phi(x)$ 和 $\psi(x)$ 在 2^ℓ 分點的值 (通常 ℓ 至少是 6)，也就可以製作它們的圖形。注意，只要知道了在 2^ℓ 分點的值，自然也就知道了所有次數小於 ℓ 的二次分點值。比如說知道了 $\phi(x)$ 在 $\frac{k}{64}$ 的值，當然也就知道了 $\phi(x)$ 在 $\frac{k}{8}$ 的值，因為 $\frac{k}{8} = \frac{8k}{64}$ 。

idx:dyadic

idx:2j-adic

除了 $p = 1$ 的情況之外，其他階數的 $\phi(x)$ 都不是基本函數。所以需要特殊的方法來計算它們在二次分點的值。在這一節，我們要探討如何計算 $\phi(x)$ 在二次分點的數值。在此我們引用 G. Strang [14] 的方法。

由於 $\phi(x)$ 的函蓋是 $[0, 2p-1]$ ，所以它在整數點的非零函數值只有

$$\phi(1), \phi(2), \dots, \phi(2p-2)$$

共 $2p - 2$ 個。如果我們知道這些整數點值，由於 $\phi(x)$ 符合自格等式 (3.4)，所以就知道 $\phi(x)$ 在二分點的值：

$$\phi\left(\frac{k}{2}\right) = \sum_n c_n \phi\left(2\frac{k}{2} - n\right) = \sum_n c_n \phi(k - n)$$

如果知道二分點的值，再利用自格等式，就知道四分點的值：

$$\phi\left(\frac{k}{4}\right) = \sum_n c_n \phi\left(2\frac{k}{4} - n\right) = \sum_n c_n \phi\left(\frac{k}{2} - n\right)$$

依此類推，我們就可以一層一層地計算上去，得到越來越高次的二次分點值。

所以，我們的要務就是求得 $\phi(x)$ 在 $2p - 2$ 個整數點的值。回憶非零的自格係數只有 $c_0, c_1, \dots, c_{2p-1}$ 。根據自格等式 (不寫出所有零項)

$$\phi(1) = c_0 \phi(2) + c_1 \phi(1)$$

$$\phi(2) = c_0 \phi(4) + c_1 \phi(3) + c_2 \phi(2) + c_3 \phi(1)$$

⋮

$$\phi(2p - 2) = c_{2p-2} \phi(2p - 2) + c_{2p-1} \phi(2p - 3)$$

用矩陣與向量的形式表達，就成了

$$\begin{pmatrix} \phi(1) \\ \phi(2) \\ \vdots \\ \phi(2p - 2) \end{pmatrix} = \begin{pmatrix} c_1 & c_0 & & & \\ c_3 & c_2 & c_1 & c_0 & \\ & & & \dots & \\ & & & & c_{2p-1} & c_{2p-2} \end{pmatrix} \begin{pmatrix} \phi(1) \\ \phi(2) \\ \vdots \\ \phi(2p - 2) \end{pmatrix} \quad (5.13)$$

eq:matA

將上式中的矩陣記做 $\mathbf{A} = (a_{ij})$ ，我們稱此矩陣為 *Strang* 矩陣。則 \mathbf{A} 是一個 $(2p - 2) \times (2p - 2)$ 的方陣，其一般項是

idx:Strang-

$$a_{ij} = c_{2i-j} \quad (5.14) \quad \text{eq:StrangA}$$

則 (5.13) 就是說 \mathbf{A} 具有一個特徵值是 $\lambda = 1$ ，而它對應於一個特徵向量 $(\phi(1), \phi(2), \dots, \phi(2p-2))^T$ 。

特徵值和特徵向量是整個線性代數理論中最重要的一課題之一。我們無法在此詳細複習這批觀念。簡單地說，如果 \mathbf{A} 是一個的方陣，若有一個非零向量 $\mathbf{x} \neq \mathbf{0}$ ，和某個數字 $\lambda \in \mathbb{C}$ (λ 可能是 0 或實數) 使得

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$$

則我們稱 \mathbf{x} 是一個特徵向量 (*characteristic vector or eigenvector*)，而 λ 是對應 \mathbf{x} 方向的特徵值 (*characteristic value or eigenvalue*)。我們理解 \mathbf{A} 是一個線性映射， $\mathbf{A}\mathbf{x}$ 的作用就是把向量 \mathbf{x} 映射到另外一個向量。而特徵向量的意思就是經過 \mathbf{A} 的映射之後方向不變的向量，特徵值的意思差不多就是特徵向量在被映射前後的長度比例：

idx:eig-val

$$|\lambda| = \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|}$$

在圖 60 中，我們呈現一個簡單的例子。令 \mathbf{A} 為 (2.11) 所定義的對稱正定矩陣。有矩陣理論告訴我們， \mathbf{A} 的特徵向量互相垂直，而且特徵值都是正數。圖中看到的圓是平面上的單位圓。用星星畫出的橢圓是單位圓被 \mathbf{A} 映射之後的結果。單位圓內的兩條短線是特徵向量的方向： $\mathbf{v}_1 = \begin{bmatrix} 0.9239 \\ 0.3827 \end{bmatrix}$ 和 $\mathbf{v}_2 = \begin{bmatrix} -0.3827 \\ 0.9239 \end{bmatrix}$ ；它們的長度是 1 所以叫做單位特徵向量。它們對應的特徵值是 $\lambda_1 = 0.7929$ ， $\lambda_2 = 2.2071$ 。觀察 $\lambda_2\mathbf{v}_2$ 和 $\lambda_1\mathbf{v}_1$ 就是橢圓的長短軸。在第二章五節中，我們提到矩陣的長度。在圖 60 中橢圓的長軸長度就是 $\|\mathbf{A}\|$ ，也就是 λ_1 和 λ_2 之中比較大的那個：2.2071。注意，這裡說的理論只配對稱正定的矩陣。一般的矩陣沒有這些好結果。詳細的情形通常在線性代數或矩陣計算的課程中交代。

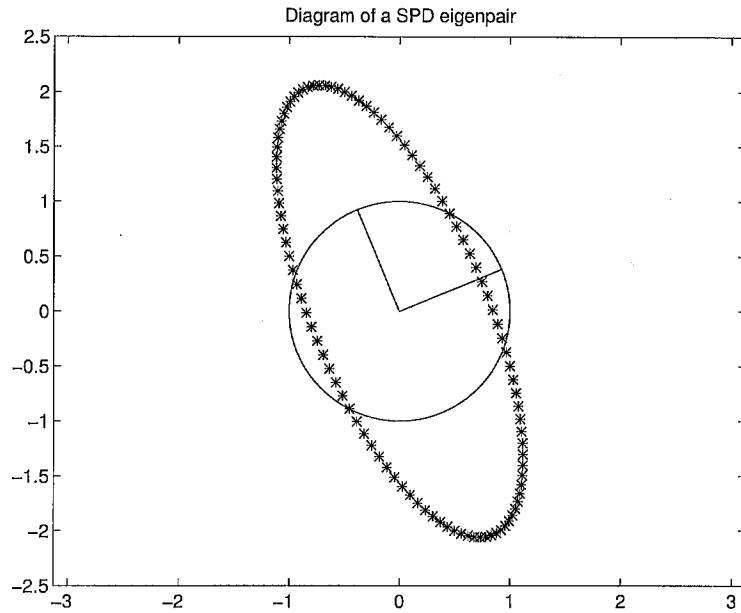


fig:SPDeig 圖 60 一個對稱正定 2×2 矩陣的特徵向量圖示

如果您有 5.0 版以上的 Matlab，可以試著執行

```
eigshow.m
```

這個程式。您將可以洞察特徵向量與特徵值的一些性質。讀者可以使用

```
makeA.m
```

製造 (5.13) 中的方陣 A。例如

```
A = makeA(3,getdaubc(3));
```

會得到 $p = 3$ 的矩陣 A 。

如果 A 是一個方陣，那麼使用 Matlab 指令

```
[V, E] = eig(A);
```

將會獲得兩個矩陣 V 和 E ，它們的維度與 A 相同。其中 V 的每一個行向量 (column vector) 就是一個特徵向量， E 是一個對角線矩陣。而對應第 n 個特徵向量的特徵值就是 E 的第 n 個對角線元素。 idx:eig

根據實驗，在 (5.13) 中的那個矩陣 A 的最大特徵值總是 1 (不論 p 是多少)，其他的 $2p - 3$ 個特徵值有可能出現 $\frac{1}{2}$ ， $\frac{1}{4}$ 這些數字，但也不盡然如此；還有可能出現負數。其實，關於方陣 A 的特徵值，有一個很好很簡單的描述：如果 A 是由 p 階自格係數製造出來的，則 A 有 $2p - 2$ 個特徵值，而且包括以下 p 個值

$$1, \frac{1}{2}, \frac{1}{4}, \dots, \frac{1}{2^{p-1}}$$

回顧 $\phi(x)$ 的性質之一是 $\phi(x - k)$ 可以組成多項式 idx:Strang-ma

$$1, x, x^2, \dots, x^{p-1}$$

這兩個性質有相互對照的關係。

總之，我們利用 Matlab 很容易得到特徵值是 1 的那個特徵向量，暫且稱之為 u 。

可是特徵向量並不是唯一的。比如說 $Ax = \lambda x$ ，則任給一個常數 $c \neq 0$ ，因為 $A(cx) = \lambda(cx)$ ，所以 cx 也是一個特徵向量，而且它的特徵值也是同樣的 λ 。Matlab 自動計算一個長度是 1 的特徵向量給我們。所以 $\|u\| = 1$ 。但是這個特徵向量未必是我們想要的。我們只能肯

定，Matlab 計算出來的特徵向量 u 與我們想要的特徵向量成比例。也就是說

$$\begin{pmatrix} \phi(1) \\ \phi(2) \\ \vdots \\ \phi(2p-2) \end{pmatrix} = c \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{2p-2} \end{pmatrix}$$

因此我們要設法決定 c 。利用 (3.71)，我們知道

$$\sum_n \phi(n) = 1$$

所以只要取

$$c = \frac{1}{u_1 + u_2 + \cdots + u_{2p-2}}$$

在 Matlab 裡面，如果 $E(i,i)=1$ 則 V 的第 i 行 (column) 是 u 。使用以下的程序，給定一個階數 p ， val 將會是 p 階 $\phi(x)$ 的整數點值。

code:phint

Matlab Code-18

```
A = makeA(p,getdaubc(p));
[V,E] = eig(A);
j = find(abs(diag(E)-1) < .1e-10);
u = V(:,j);
val = u/sum(u);
```

得到的結果是 $val(1) = \phi(1)$ ， \dots ， $val(2p-2) = \phi(2p-2)$

程序 Code-18 之中大概有兩行值得解釋。我們知道矩陣 E 的對角線上應該有一個元素是 1。如果那個元素是在第 j 列，那麼 V 的第 j 列向量就是對應 $\lambda = 1$ 的特徵向量 u 。因此 $j=find\dots$ 那一行的目的就是取得 j ，而 $V(:,j)$ 就取出了特徵向量 u 。

現在我們解釋 `find` 那一行在做什麼？首先 `diag(E)` 取出來 E 矩陣中的對角線向量。理論上這個向量內有一個 $= 1$ 的元素，但是因為浮點數計算誤差，所得的元素可能不恰好是 1，但應相去不遠。所以我們尋找一個和 1 距離 10^{-10} 以內的元素。`(abs(diag(E)-1) < .1e-10)` 就是將對角線元素每個都減 1 然後取絕對值。而邏輯運算符號 `< .1e-10` 將一個向量 (或矩陣) 中符合這個邏輯關係的元素寫成 1，否則寫 0。例如

idx:diag

idx:matlab

idx:rel

$$\left(\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \geq 5 \right) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

所以在執行

```
abs(diag(E)-1) < .1e-10
```

之後就應該得到一個向量，其第 j 個元素是 1，其他都是 0。而 `find` 指令就是把向量中非零元素的足標寫出來。因此就得到 j 了。

idx:find

我們曾經將 $\phi(x)$ 的整數點值都計算出來，儲存在 Matlab 的 `mat` 格式檔案內。

phint.mat

當您在 Matlab 中 `load phint` 將會讀入 13 個向量，名為 `phi2`， \dots ，`phi14`。它們分別是 $p = 2, \dots, 14$ 階 $\phi(x)$ 的整數點值。

自格函數 $\phi(x)$ 的某層 2^ℓ 分點值也已經計算出來，放在

phifix.mat

當您在 Matlab 中 `load phifix` 將會讀入 13 個向量，名為 `phi2`， \dots ，`phi14`。它們分別是 $p = 2, \dots, 14$ 階 $\phi(x)$ 的 2^ℓ 分點值。而 ℓ 的選擇與 p 有關，例如當 $p = 4$ 時 ℓ 就是：

```
log2(length(phi4)/(2*4-1))
```

一旦我們知道了 $\phi(x)$ 在 2^ℓ 分點的值，根據

$$\psi(x) = \sum_n (-1)^n c_{1-n} \phi(2x - n)$$

就可以知道 $\psi(x)$ 在 $2^{\ell+1}$ 分點的值。因為

$$\psi\left(\frac{k}{2^{\ell+1}}\right) = \sum_n (-1)^n c_{1-n} \phi\left(\frac{k - n \cdot 2^\ell}{2^\ell}\right)$$

我們可以利用 Code-15 從 $\phi(x)$ 的 2^ℓ 分點值計算 $\psi(x)$ 的 $2^{\ell+1}$ 分點值。凌波函數 $\psi(x)$ 的某層 2^ℓ 分點值也已經計算出來，放在

psifix.mat

檔案的內容和格式與 phifix.mat 一律相同。

最後，讓我們再談談 Strang 矩陣 \mathbf{A} 。如果 $\phi(x)$ 是一次可微，那麼由微分的連鎖律得到

$$\phi'(x) = \sum_k 2c_k \phi'(2x - k) \tag{5.15} \quad \text{eq:scaling}$$

可見 $\phi'(x)$ 仍然符合一個自格等式，而其自格係數是 $2c_k$ 。因為 $\phi'(x)$ 的函蓋還是 $[0, 2p - 1]$ ，所以造成 (5.13) 的所有條件都成立。但是此刻的方陣變成了 $2\mathbf{A}$ 。而整個線性系統是

$$2\mathbf{A} \begin{pmatrix} \phi'(1) \\ \phi'(2) \\ \vdots \\ \phi'(2p - 2) \end{pmatrix} = \begin{pmatrix} \phi'(1) \\ \phi'(2) \\ \vdots \\ \phi'(2p - 2) \end{pmatrix}$$

可見向量 $(\phi'(1), \phi'(2), \dots, \phi'(2p-2))^T$ 是 \mathbf{A} 的另外一個特徵向量，而它對應的特徵值是 $\frac{1}{2}$ 。

利用 `makeA.m` 和 `eig` 指令，我們還是可以求得 \mathbf{A} 的一個對應 $\lambda = \frac{1}{2}$ 的特徵向量 \mathbf{v} 。而 \mathbf{v} 和所求的特徵向量成比例，亦即

$$\begin{pmatrix} \phi'(1) \\ \phi'(2) \\ \vdots \\ \phi'(2p-2) \end{pmatrix} = c \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{2p-2} \end{pmatrix}$$

但是此時該如何確定那個 (非零的) 比例常數 c 呢？因為 (3.71) 所以

$$\sum_n \phi'(n) = \frac{d}{dx} \left(\sum_n \phi(x-n) \right) \Big|_{x=0} = \frac{d}{dx} 1 = 0$$

故不能用這個方法來確定比例常數。這時要利用性質 3.6。當 $\phi(x)$ 一次可微時，必定 $p \geq 3$ ，所以

$$\sum_n n \phi(x-n) = x + \alpha$$

其中 α 是個常數；由後面的 (5.26) 知道其實 $\alpha = -\int x\phi dx$ 。因此

$$\sum_n n \phi'(n) = \frac{d}{dx} (x + \alpha) \Big|_{x=0} = 1$$

就是一個確定比例常數的方法。所以要取

$$c = \frac{1}{\sum_{n=1}^{2p-2} n v_n}$$

可見如果 $\phi(x)$ 具有一次導函數，則 Strang 矩陣 \mathbf{A} 具有特徵值 $\frac{1}{2}$ 依理類推，如果 $\phi(x)$ 具有 m 次導函數，則 \mathbf{A} 具有特徵值 $\frac{1}{2^m}$ 。注

意，前面這句話逆過來說並不正確。如果 \mathbf{A} 具有特徵值 $\frac{1}{2^m}$ ，並不表示 $\phi(x)$ 具有 m 次導函數。例如當 $p = 2$ 的時候， \mathbf{A} 具有特徵值 $\frac{1}{2}$ ，但是 $\phi(x)$ 並沒有一次導數。究竟 $\phi(x)$ 會具備多少次的導函數？這不是一個簡單的問題，我們留到下節說明。

習題 5.1 寫一個程式，從 $\phi(x)$ 的整數點值計算 $\psi(x)$ 的整數點值。

習題 5.2 仿照 Code-18 寫一個算出 $\phi'(1), \phi'(2), \dots, \phi'(2p-2)$ 的 Matlab 程序。

5.4 自格與凌波函數的圖形

這一節的目的是觀察 $\phi(x)$ 與 $\psi(x)$ 的圖形。我們先用取自 `phifix.mat` 與 `psifix.mat` 的 $\phi(x)$ 與 $\psi(x)$ 的值來製圖。然後討論它們的平滑程度。最後再介紹一種利用疊積的製圖方法。

根據前面的討論，我們得以推論， $\phi(x)$ 和 $\psi(x)$ 本身的圖形，在大部分的應用問題中並不重要。但我們還是希望看到 $\phi(x)$ 和 $\psi(x)$ 的圖形。一方面可以說是出於好奇，另一方面在某些場合還是有其必要性。圖形可以增進我們的直覺。何況，我們現在整天與 $\phi(x)$ 和 $\psi(x)$ 為伍，總不好連它們的廬山真面目都沒有見過吧？！

有了 $\phi(x)$ 和 $\psi(x)$ 的二次分點值之後，我們就可以製圖。在圖 61-idx:dyadic 圖 63 中，我們畫出三對 ($p = 2, 3, 4$) $\phi(x)$ 和 $\psi(x)$ 的圖。製圖的節點間隔選的是 $\frac{1}{256}$ 。理論上它們的函蓋是 $[0, 2p - 1]$ ，但是明顯可見它們實際上非零的部份並沒有那麼長。例如，注意 $p = 4$ 的 $\phi(x)$ ，它在 $x > 5$ 之後就幾乎是零；但是在數值上它並不是零，而是很小的數（大約 10^{-3} 到 10^{-7} 之間）。

已知 $p = 1$ 的 $\phi(x)$ 不連續。 $p = 1$ 的 $\phi(x)$ 就是零階基本樣條函數，俗稱方盒函數； $p = 1$ 的 $\psi(x)$ 就是 Haar 函數。它們的圖形很簡

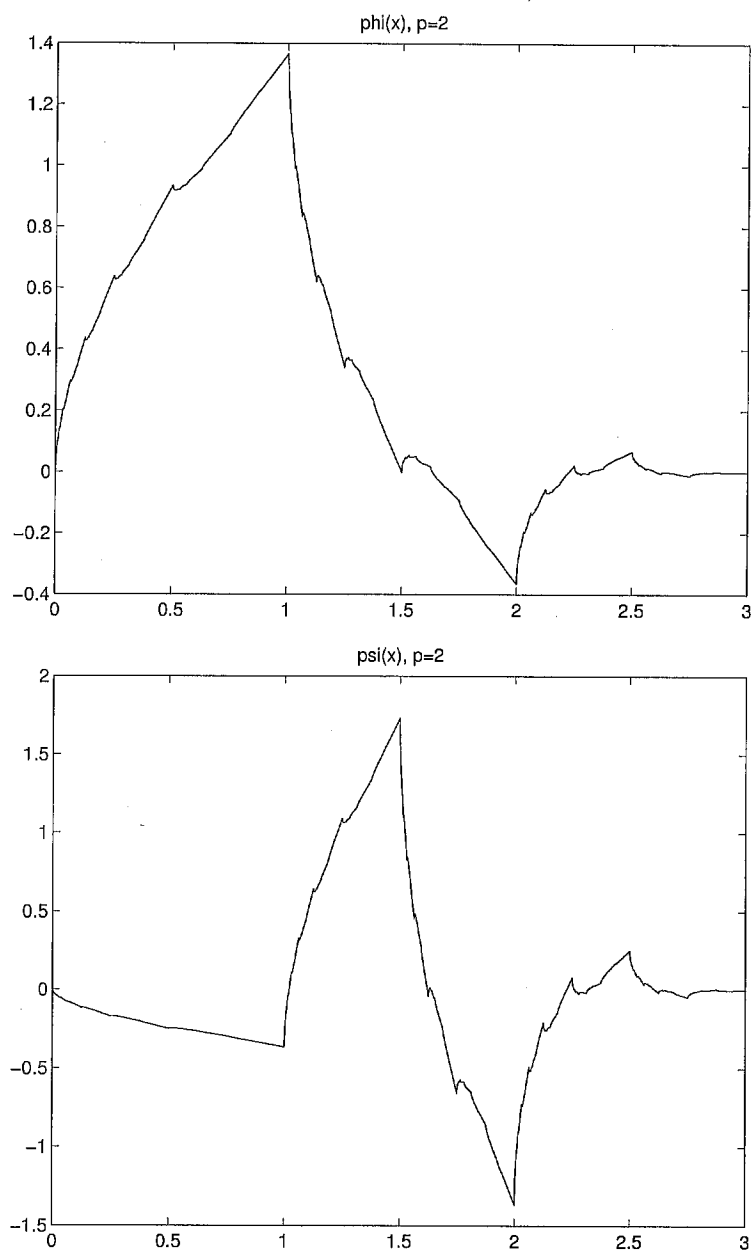


fig:psitwo 圖 61 Daubechies $\phi(x)$ (上) 和 $\psi(x)$ (下) $p = 2$

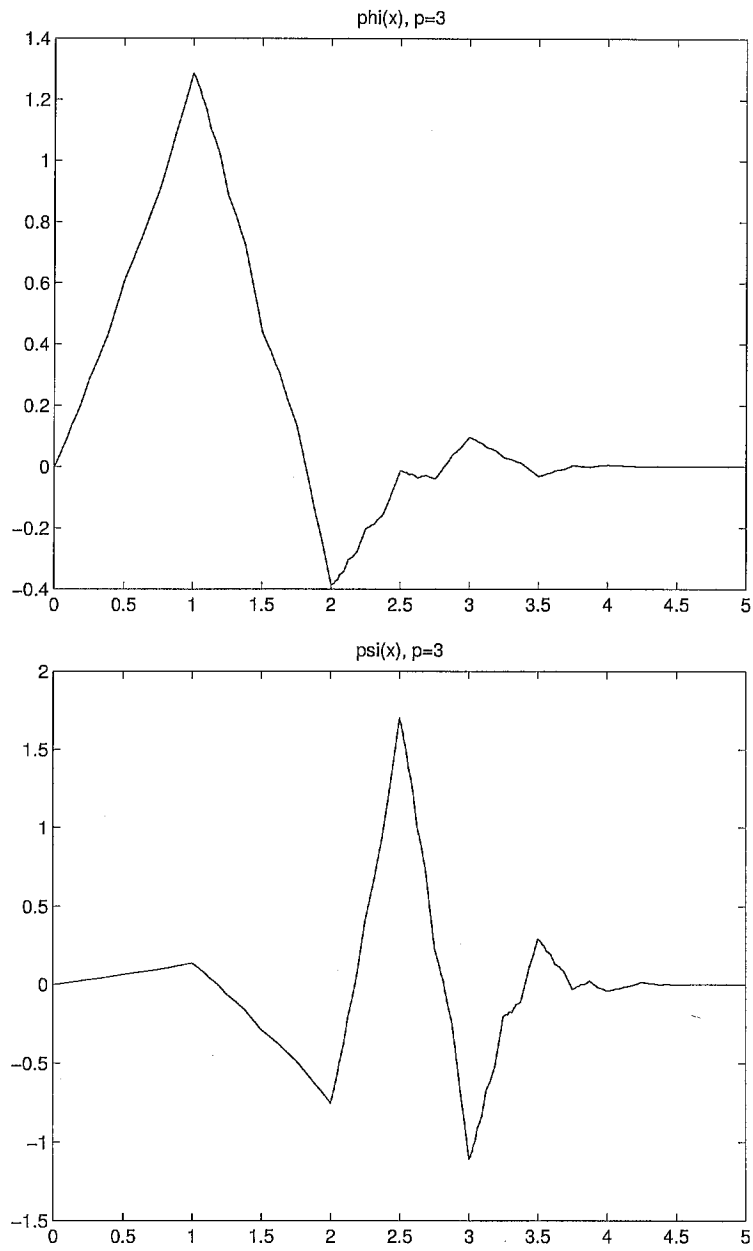


fig:psithree 圖 62 Daubechies $\phi(x)$ (上) 和 $\psi(x)$ (下) $p = 3$

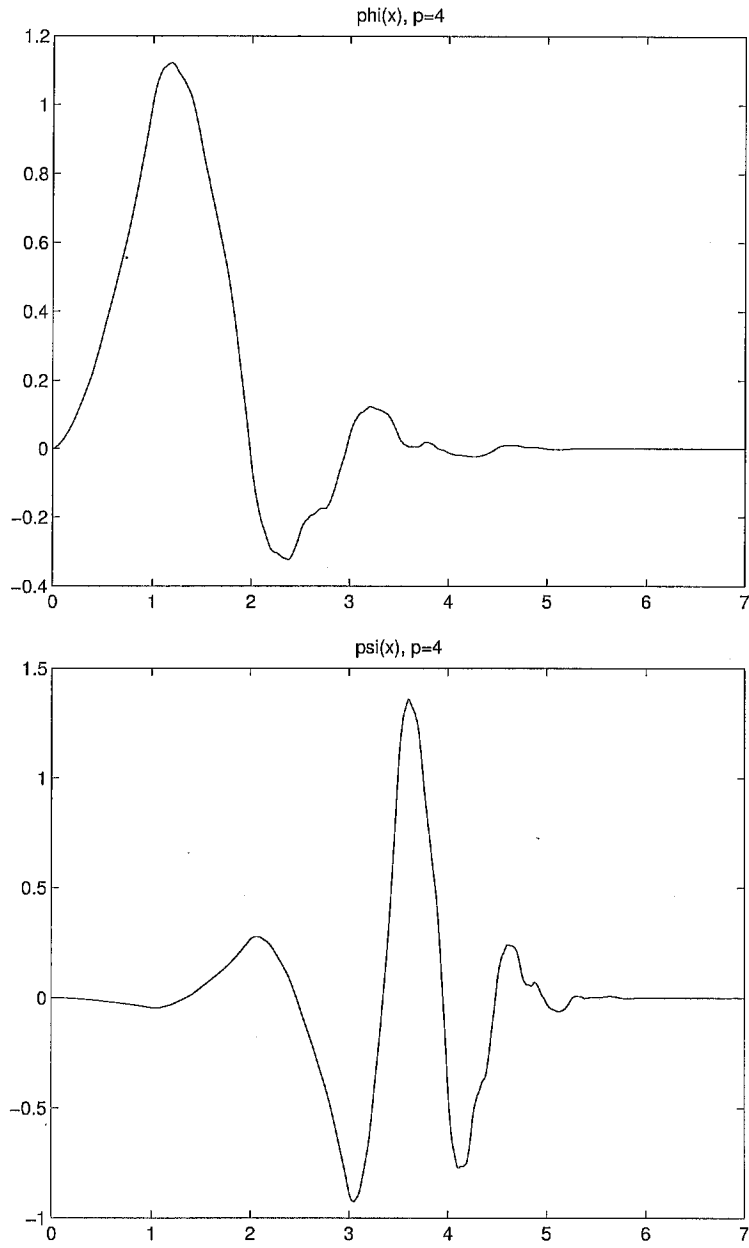


fig:psifour 圖 63 Daubechies $\phi(x)$ (上) 和 $\psi(x)$ (下) $p = 4$

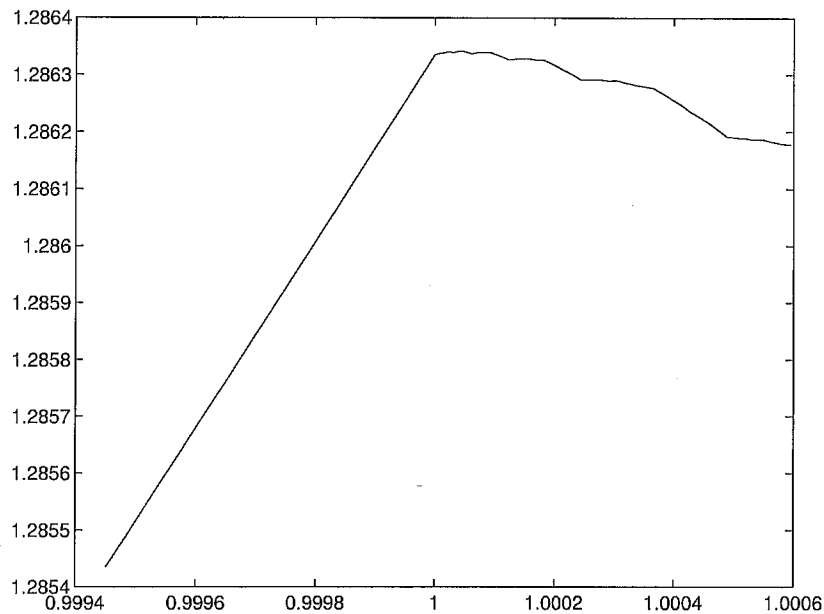


fig:phithreea 圖 64 $p = 3$ 的 $\phi(x)$ 在 $x \in [1 - 0.0006, 1 + 0.0006]$ 之間的函數圖形

單，參看圖 1 和圖 5。而 $p \geq 2$ 時 $\phi(x)$ 都是連續函數，而且隨著 p 的升高而逐漸平滑。從圖 61 可以明顯看到 $p = 2$ 的 $\phi(x)$ 是不可微分的，因為它的曲線上有許多尖銳的點。有趣的是，從圖 62 上面觀察，似乎 $p = 3$ 的 $\phi(x)$ 也是不可微分。但是數學的分析卻發現它是可微的。從圖 62 的尺度來看，似乎 $\phi(x)$ 的函數曲線在 $x = 1$ 附近有一個尖銳的高峰，所以看似不可微。就好像 $|x|$ 在 $x = 0$ 處不可微一樣。其實如果將尺度縮小，也就是在 $x = 1$ 附近放大來看，我們會發現其實它沒有那麼尖銳。參見圖 64。

爲了不過分浪費篇幅，我們在圖 65 到圖 67 的三張圖中畫出 $p = 1, 2, \dots, 12$ 的 $\phi(x)$ 與 $\psi(x)$ 圖形。而且爲了做整體的比較，在這三張圖中我們將圖框的尺度定爲一致：所有 $\phi(x)$ 的圖框都是 $[0, 12] \times$

342 凌波初步

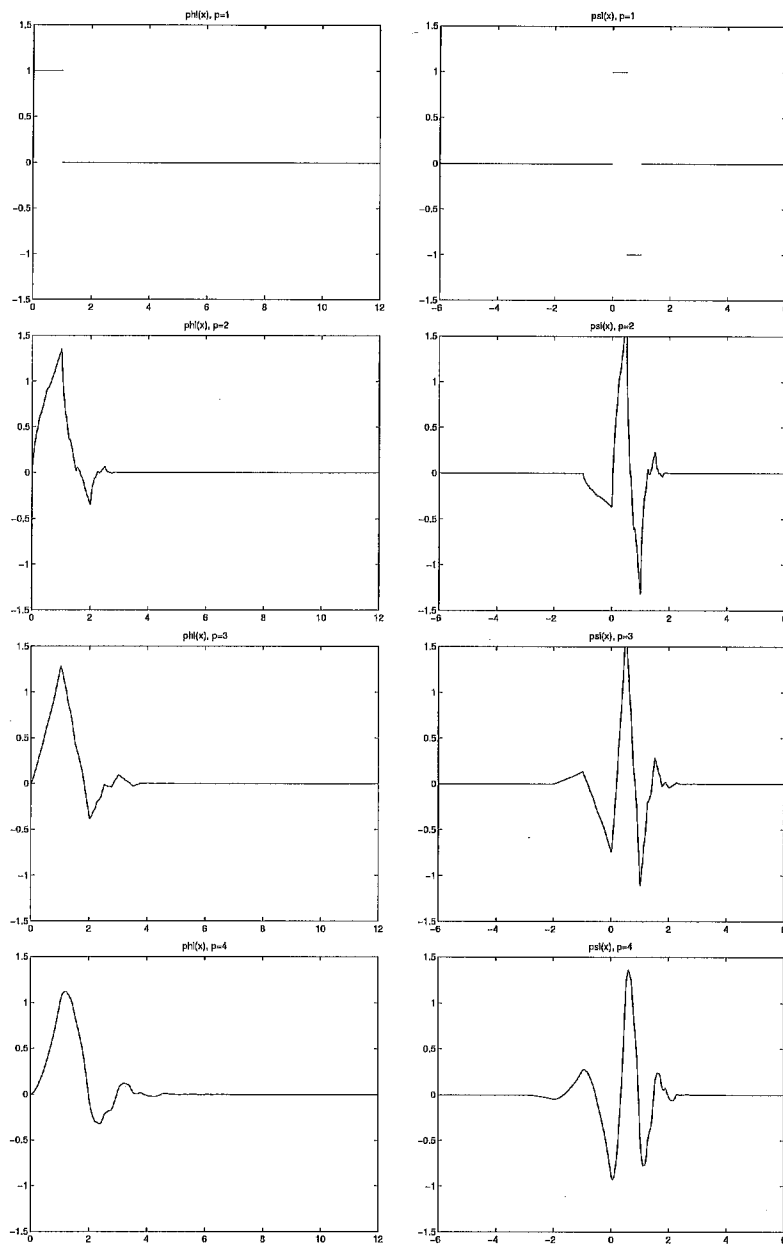


fig:phionefour 圖 65 左邊是 $p = 1, 2, 3, 4$ 的 Daubechies $\phi(x)$ ，右邊是對應的 $\psi(x - p + 1)$

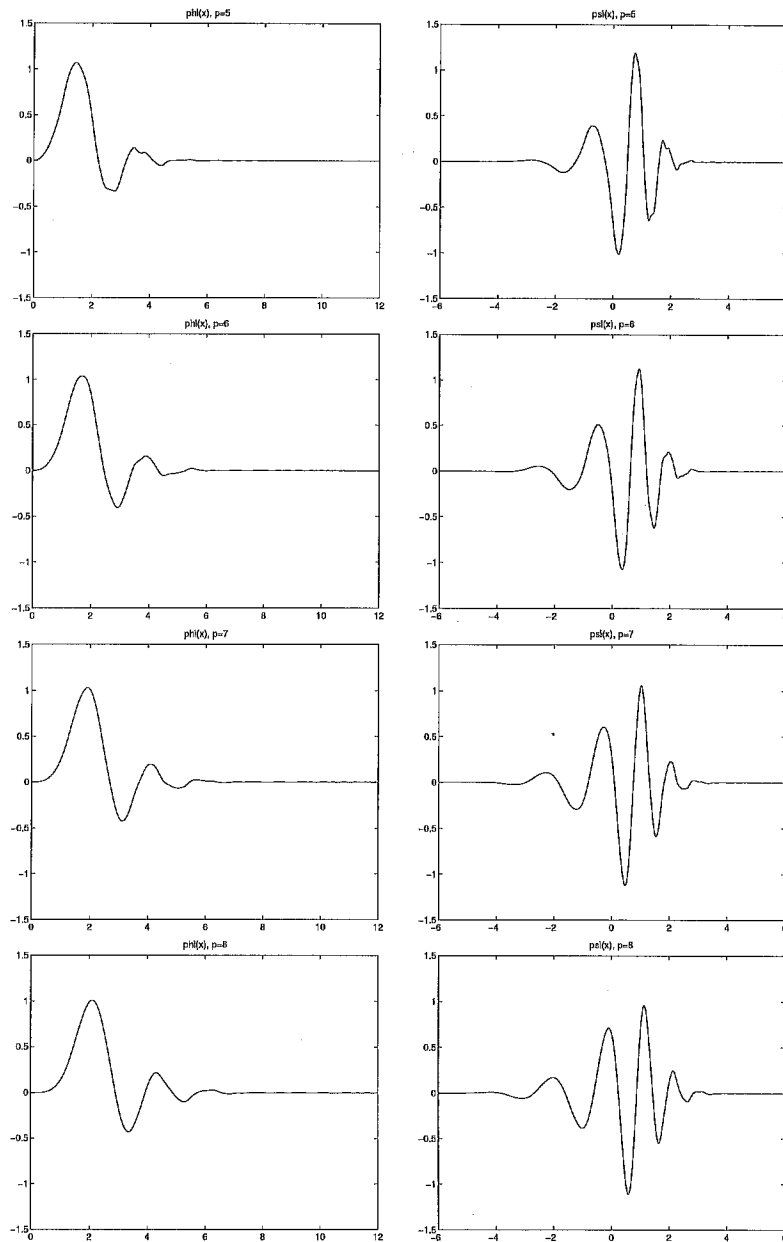


fig:phifiveight 圖 66 左邊是 $p = 5, 6, 7, 8$ 的 Daubechies $\phi(x)$ ，右邊是對應的 $\psi(x - p + 1)$

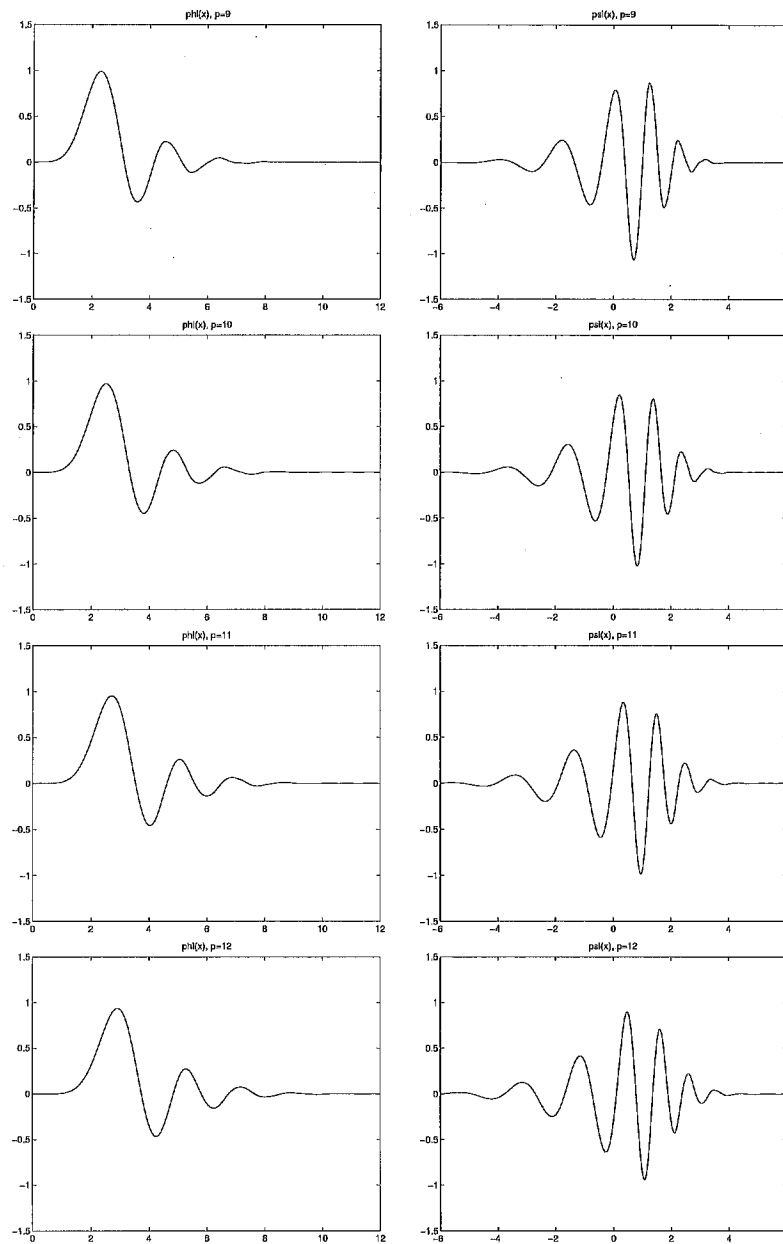


fig:phinineC 圖 67 左邊是 $p = 9, 10, 11, 12$ 的 Daubechies $\phi(x)$ ，右邊是對應的 $\psi(x - p + 1)$

$[-1.5, 1.5]$ ，所有 $\psi(x)$ 的圖框都是 $[-6, 6] \times [-1.5, 1.5]$ 。此時我們是根據 (3.20) 來定義 $\psi(x)$ ，而不是 (3.19)。因此 $\text{supp } \psi = [1 - p, p]$ ，它們以 $\frac{1}{2}$ 為中心。這些圖的節點間隔是 $\frac{1}{64}$ 。

從這些圖形中，我們看出來 $\psi(x)$ 的確外形如波，原因是

$$\int_0^{2p-1} \psi(x) dx = 0$$

所以曲線在 x -軸上方的面積必須與 x -軸下方的面積互相抵消。我們也可以從這些函數圖形中觀察 $\phi(x)$ 的能量比較集中在函蓋的左邊，而凌波的能量比較集中在函蓋的中間。事實上， $\psi^2(x)$ 的質量中心就是在其函蓋的中央 (性質 5.8)。隨著階數 p 的增加， $\phi(x)$ 和 $\psi(x)$ 的平滑度逐漸提高，波數逐漸增加，並且整個波形逐漸擴散。

idx:energy

回顧 Hölder 連續指數的意義 (3.29)。圖 61、圖 62、圖 64 和圖 68 是 Hölder 連續指數的絕妙圖例。當 $p = 2$ 時大約 $\phi(x) \in C^{0,0.55}$ ，所以是連續但不可微。當 $p = 3$ 時大約 $\phi(x) \in C^{1,0.09}$ ，所以它是一次可微，而其一次導函數非常得粗糙，因為它的 Hölder 指數只有 0.09；也就是 $\phi'(x) \in C^{0,0.09}$ 。當 $\phi(x)$ 可微但是其導函數的 Hölder 指數很小的時候， $\phi'(x)$ 的數值導數可能計算得很不準。參見圖 68。

idx:Holder

凌波 $\psi(x)$ 的可微程度和自格函數 $\phi(x)$ 是一樣的，因為它只是 $\phi(2x - k)$ 的一個線性組合。所以不必另外討論 $\psi(x)$ 的可微程度。

當 $p = 4, 5, 6$ 時， $\phi(x)$ 都是一次可微，而且它們的導函數越來越平滑，參見圖 69。當 $p \geq 7$ 時 $\phi(x)$ 是二次可微。

idx:phi-dif

在這以前，所有的 $\phi(x)$ 與 $\psi(x)$ 的函數圖形是利用已經儲存在 `phifix.mat` 與 `psifix.mat` 裡面的二次分點值所畫出來的。要計算這些二次分點值，必須先計算整數點值。其實如果只是要繪製一個自格函數的圖形，可以不必經過這個步驟。在 $L^2(\mathbb{R})$ 空間內的自格函數都

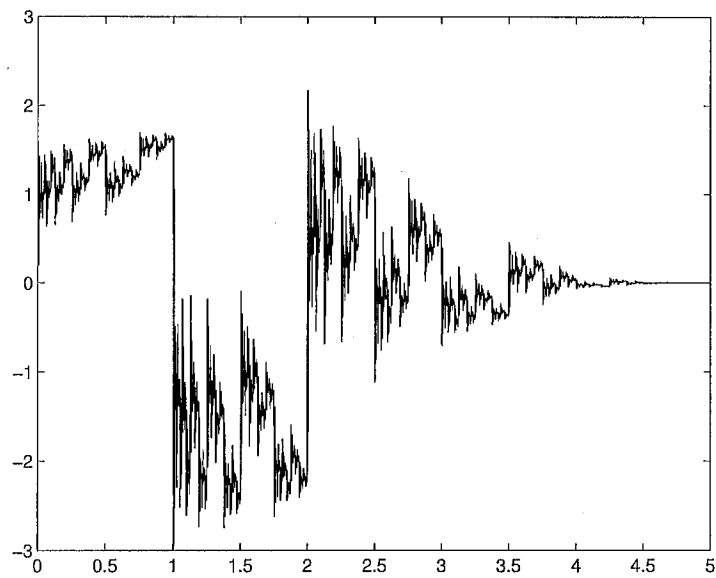


fig:phithreed 圖 68 $p = 3$ 的 $\phi'(x)$

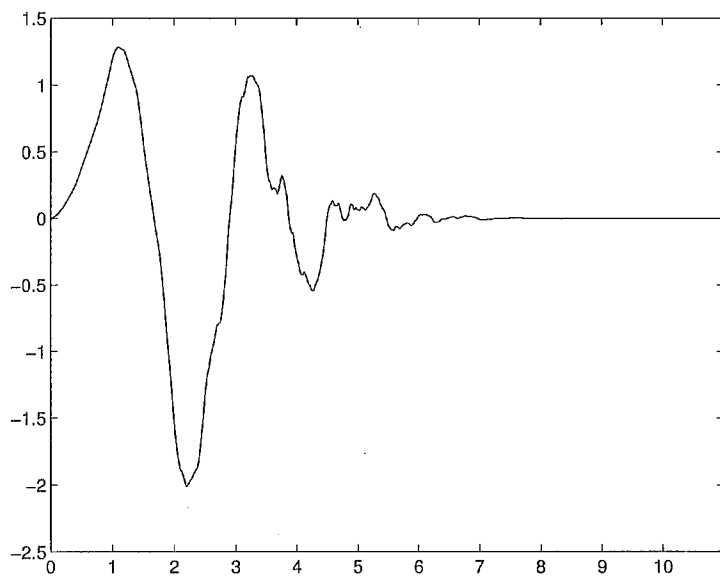


fig:phisixd 圖 69 $p = 6$ 的 $\phi'(x)$

可以藉由自格等式做遞迴式的定義：

$$\begin{aligned} \phi(x) &= \lim_{n \rightarrow \infty} \phi^{(n)}(x), \quad \text{where} \\ \phi^{(n+1)}(x) &= \sum_k c_k \phi^{(n)}(2x - k), \quad \text{for } n = 0, 1, 2, \dots \quad (5.16) \\ \text{with } \phi^{(0)}(x) &= B_1(x) \end{aligned} \quad \text{eq:recur}$$

回憶 $B_1(x)$ 就是一階基本樣條函數，它在 $[0, 1)$ 之間是 1，其他的地方是 0。通常 (5.16) 中的函數極限會存在（詳情不討論了），而極限就是一個自格函數。我們理解 $\frac{1}{2}c_k$ 是一組低頻濾波器。所以 (5.16) 中遞迴式子的意思差不多就是說將 $B_1(x)$ 做一次低頻濾波，再做一次，然後再做一次，... 重複一直做，那函數圖形就會漸漸地不變，最後趨近於一個函數。那個函數在經過低頻濾波之後，還是他自己。那就是自格函數了。大致地說，自格函數就是對一組低頻濾波器來說的不變函數。

先舉個例子來看看 (5.16) 要怎麼計算。比如說我們想要知道 $\phi(\frac{k}{16})$ 的值，而且已經知道 $\phi^{(n)}(\frac{k}{16})$ 的值。那麼

$$\begin{aligned} \phi^{(n+1)}\left(\frac{0}{16}\right) &= c_0 \phi^{(n)}\left(\frac{0}{8}\right), \quad \phi^{(n+1)}\left(\frac{1}{16}\right) = c_0 \phi^{(n)}\left(\frac{1}{8}\right), \\ \dots \quad \phi^{(n+1)}\left(\frac{7}{16}\right) &= c_0 \phi^{(n)}\left(\frac{7}{8}\right) \end{aligned}$$

然後

$$\begin{aligned} \phi^{(n+1)}\left(\frac{8}{16}\right) &= c_1 \phi^{(n)}\left(\frac{0}{8}\right) + c_0 \phi^{(n)}\left(\frac{8}{8}\right) \\ \phi^{(n+1)}\left(\frac{9}{16}\right) &= c_1 \phi^{(n)}\left(\frac{1}{8}\right) + c_0 \phi^{(n)}\left(\frac{9}{8}\right) \\ &\vdots \\ \phi^{(n+1)}\left(\frac{15}{16}\right) &= c_1 \phi^{(n)}\left(\frac{7}{8}\right) + c_0 \phi^{(n)}\left(\frac{15}{8}\right) \end{aligned}$$

可見

$$\begin{aligned} & (\phi^{(n+1)}(\frac{0}{16}), \phi^{(n+1)}(\frac{1}{16}), \phi^{(n+1)}(\frac{2}{16}), \dots) \\ & = (\phi^{(n)}(\frac{0}{8}), \phi^{(n)}(\frac{1}{8}), \phi^{(n)}(\frac{2}{8}), \dots) * (c_0, 0, \dots, 0, c_1, 0, \dots, 0, c_2, \dots) \end{aligned}$$

其中 * 代表離散疊積，而 c_i 與 c_{i+1} 之間插入 7 個 0 (頭尾不必加 0)。 idx:dconvolv

依理類推，一般的情況應該是這樣的：若需要以 2^ℓ 為間隔繪製 p 階的 $\phi(x)$ 圖形，令 $N = 2p - 1$ ，我們要在 $[0, N]$ 內繪製圖形。先準備 $\phi^{(0)}(x)$ 在 $[0, N]$ 之間的點值。我們知道當 $\phi^{(0)}(x) = B_1(x)$ 。將這些點值寫成一個向量

$$\begin{aligned} \mathbf{u} &= (u_0, u_1, \dots, u_{2^\ell N}) \\ \text{where } u_n &= \begin{cases} 1 & \text{if } 0 \leq n \leq 2^\ell - 1 \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

然後製造一個長度為 $2^{\ell-1}N + 1$ 的向量

$$\begin{aligned} \mathbf{v} &= (v_0, v_1, \dots, v_{2^{\ell-1}N}) \\ \text{where } v_n &= \begin{cases} c_k & \text{if } n = 2^{\ell-1}k \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

然後我們將 \mathbf{v} 與 \mathbf{u} 的偶數項做疊積。疊積的結果恰好就是 $\phi^{(1)}(x)$ 在 2^ℓ 分點的值。然後再將 $\phi^{(1)}(x)$ 的值寫入 \mathbf{u} ，同樣取其偶數項再與 \mathbf{v} 做疊積，就得到 $\phi^{(2)}(x)$ 在 2^ℓ 分點的值。依此類推。一直迭代到您滿意為止。 idx:recur

如果使用者給定了 $\phi(x)$ 的階數 p ，間隔的幂數 $\ell = \text{level}$ 和迭代的次數 $n = \text{num}$ 。則下面的 Matlab 程序會從 $\phi^{(0)}(x)$ 製造 $\phi^{(n)}(x)$ 。以 $p = 2$ 為例，我們在圖 70 中展現迭代過程中的 $\phi^{(n)}(x)$ 。在這個實驗中，我們定 $\ell = \text{level} = 6$ 。從這個尺度看來，當 $n = 6$ 的時候 (迭代

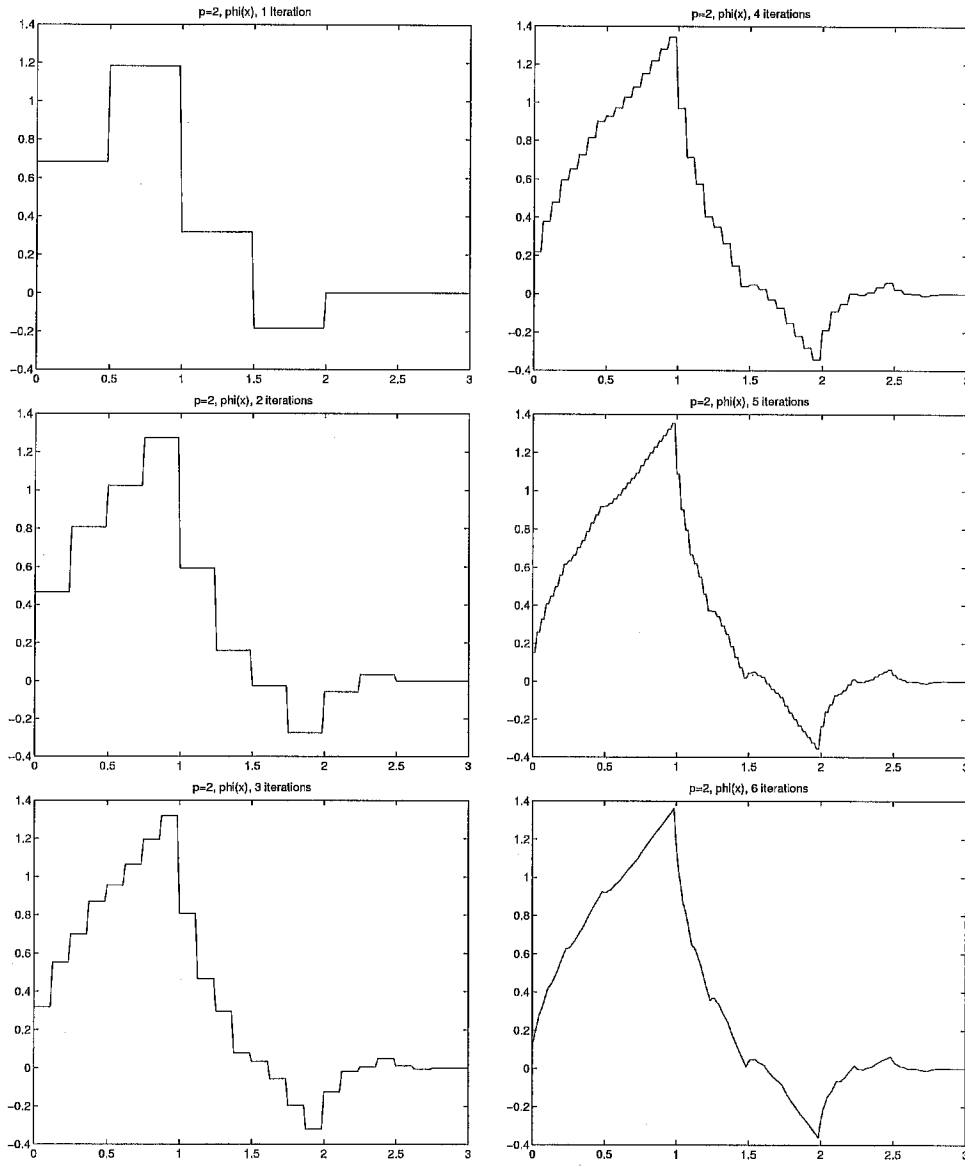


fig:casade 圖 70 $p = 2$ 的 $\phi^{(1)}(x) \cdots \phi^{(6)}(x)$ 固定六十四分點

六次) 函數的圖形就看起來很像圖 61 了。如果繼續迭代下去, $\phi^{(n)}(x)$ 和 $\phi^{(n+1)}(x)$ 之間會越來越靠近而很難看出差異。

code:cascade

Matlab Code-19

```

c=getdaubc(p);
N=2*p-1;
nu=(2^level)*N+1;
u=[ones(1,2^level) zeros(1,nu-2^level)];
nv=(2^(level-1))*N+1;
v=zeros(1,nv);
nv=2^(level-1);
i=1;
for k=1:N+1,
    v(i)=c(k);
    i=i+nv;
end
for i=1:num,
    u=conv(v,u(1:2:nu));
end

```

注意, 這個迭代的方法與前面使用二次分點值的方法不同。前一節所述的 Code-18 程序, 所求得 $\phi(x)$ 在整數點的值是理論上的真確。意思是說, 除了在使用電腦做浮點數運算所造成的誤差外, 所求得的值是正確的。然後我們利用自格等式求得二次分點的值, 這些值也是理論上的真確。反觀迭代法, 即使所有的計算步驟的是真確的; 亦即沒有任何浮點數運輸誤差發生。在理論上仍然需要 $n \rightarrow \infty$ 才能獲得真正的值。這是所有迭代法的特性。通常我們稱理論上真確的計算方法

idx:thm-exact

為直接法。在實用上，浮點數的運算誤差必然發生。所以直接法未必求得真解。但是，一般來說，如果直接法的計算步驟不是非常複雜，而且算法沒有不穩定，那麼直接法總是比迭代法有效率。

程序 Code-19 的設計方式，使得不管迭代多少次，函數的插值總是固定在 2^ℓ 分點上。所以插值的數目並沒有隨著迭代的次數提高。對於製圖而言，這或許是有意義的；因為如果是要在螢幕或紙張上製圖，沒有必要太多的插值點（參閱第一節）。但是在某些情況我們或許希望插值的點數隨著迭代的次數而提高。其實只要稍微修改 Code-19 就可以辦到。基本的想法如下：以 $p = 2$ 為例， $\text{supp } \phi(x) = [0, 3]$ 。做第一次迭代的時候，我們可以只用 $\phi^{(0)}(x)$ 的 2^0 分點值（整數點值）。

$$\mathbf{u} = (1, 0, 0, 0), \quad \mathbf{v} = (c_0, c_1, c_2, c_3)$$

經過一次疊積，得到 $\phi^{(1)}(x)$ 在 2^1 分點的值。因此我們將 \mathbf{v} 插入零而得到

$$\mathbf{v} = (c_0, 0, c_1, 0, c_2, 0, c_3)$$

與 $\phi^{(1)}(x)$ 的二分點值做疊積，就得到 $\phi^{(2)}(x)$ 在 2^2 分點的值。如果想要繼續做 $\phi^{(3)}(x)$ 在 2^3 分點的值，就重組 \mathbf{v} 向量

$$\mathbf{v} = (c_0, 0, 0, 0, c_1, 0, 0, 0, c_2, 0, 0, 0, c_3)$$

然後再做疊積。依此類推。觀察圖 71，那是 $\phi^{(1)}(x)$ 用二分點製圖、 $\phi^{(2)}(x)$ 用四分點製圖、... 到 $\phi^{(6)}(x)$ 用六十四分點製圖的圖形。可以看到 $\phi(x)$ 從粗略到詳細的演進情形。

底下這個程序，要求使用者輸入階數 p 和輸入序列 \mathbf{u} 。注意 \mathbf{u} 必須是 $\phi^{(\ell)}(x)$ 的 2^ℓ 分點值。那麼這個程序會製造 $\phi^{(\ell+1)}(x)$ 的 $2^{\ell+1}$ 分點值。例如在剛開始的時候定義 $\mathbf{u} = [1, \text{zeros}(1, 2*p-1)]$ ；。

code: casca

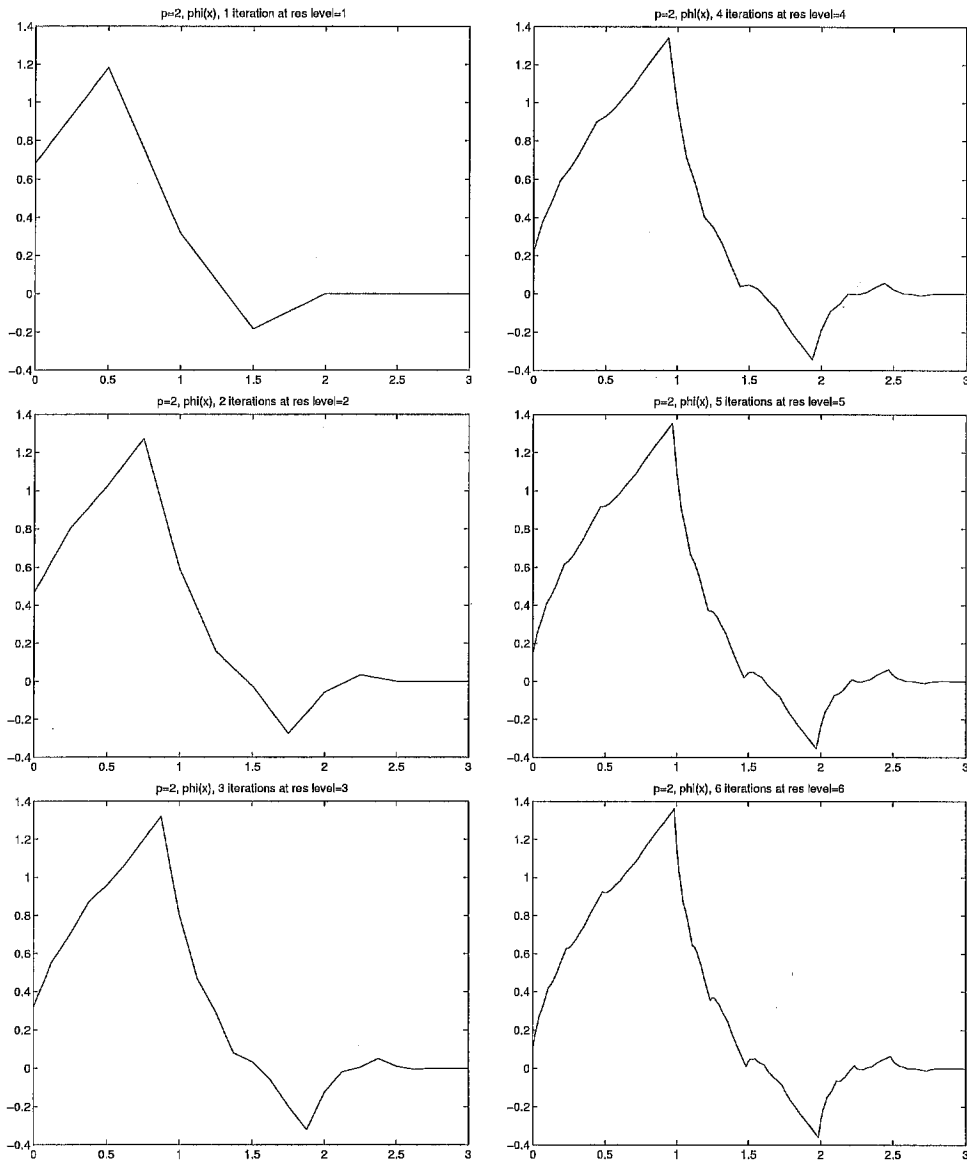


fig:cascad 圖 71 $p = 2$ 的 $\phi^{(1)}(x) \cdots \phi^{(6)}(x)$ 對 $\phi^{(n)}(x)$ 用 2^n 分點

Matlab Code-20

```
function phi=cascad(p,u)
c=getdaubc(p);
N=2*p-1;
nu=length(u);
ni=(nu-1)/N;
v=zeros(1,nu);
for k=0:N,
    v(k*ni+1)=c(k+1);
end
phi=conv(v,u);
```

5.5 數值積分

從下一節開始，我們要探討一系列有關 $\phi(x)$ 與 $\psi(x)$ 的積分問題。除了在求投影係數時需要內積 $\langle f(x), \phi(x) \rangle$ 之外，當我們應用凌波來求微分或積分方程的數值解，也牽涉到積分的步驟。但是一般我們在數值分析課本裡面學到的傳統方法，比如梯形法、辛普森法、高斯法，都不能有效地直接在這裡應用。而且這些普通的積分法並沒有利用到自格函數本身的特性。

在這一節裡，我們先簡短地複習傳統的數值積分法，並且舉例說明這些方法並不適用於凌波函數的積分。從下一節開始，我們將呈現某些技巧，用以開發幾種積分方法。其中一些是理論上真確的積分法，另一些是數值積分法。

我們先簡短地複習數值積分法 (*numerical quadrature rules*)。在 idx:num-quar這本書裡，我們只探討所謂 Newton-Cotes 類型的數值積分法。徐敏堂

先生的碩士論文 [13] 中還探討了高斯類型 (Gaussian quadrature rules) 和 調適性的積分法 (adaptive quadrature rules)。

所謂 Newton-Cotes 類型的積分法，就是以一個算術公式

$$I_n f \Big|_a^b = \sum_{i=0}^n \alpha_i f(x_i) \quad (5.17)$$

eq:quadNC

來估計積分值。

idx:quad-NC

$$I_n f \Big|_a^b \approx \int_a^b f(x) dx$$

其中 I 代表某種數值積分法， a, b 表示求積分的區間，而 n 表示將 $[a, b]$ 均勻切成 n 等分，因此就有 $n+1$ 個節點 x_i 。令 $h = \frac{b-a}{n}$ 則 $x_0 = a, x_n = b, x_i = a + ih$ 。而 α_i 就是數值積分法的積分係數 (*quadrature coefficients*)。

idx:quad-coef

大部分數值分析的課本，都會推導一個以下型式的定理：

令 $s \geq t$ 為正整數。若 $f \in C^s(a, b)$ 且積分法 $I_n f \Big|_a^b$ 對小於 t 階的多項式是真確的，則

$$I_n f \Big|_a^b = \int_a^b f(x) dx + O(h^t)$$

我們稱此積分法為 t 階收斂。

前面所謂的對小於 t 階的多項式是真確，意思是說

idx:poly-exac

$$\int_a^b x^m dx = I_n x^m \Big|_a^b, \quad \text{for } 0 \leq m \leq t-1$$

所謂 t 階收斂的另外一個看法是

$$\left| \int_a^b f(x) dx - I_{2n} f \Big|_a^b \right| \approx \frac{1}{2^t} \left| \int_a^b f(x) dx - I_n f \Big|_a^b \right| \quad (5.18)$$

eq:quadnt

我們不打算在此證明上述的定理。它的意義大致如下：如果 $I_n f \Big|_a^b$ 對階數 $\leq t-1$ 的多項式是真確的，而 $s > t-1$ ，則 $f(x)$ 可以展開 $t-1$ 階的泰勒多項式，留下一個 $O(h^t)$ 的泰勒誤差項。而 $I_n f \Big|_a^b$ 對泰勒多項式是真確的，所以它和 $\int_a^b f dx$ 之間的誤差就只是泰勒誤差項。

收斂的階數 t 當然是隨方法而異。舉例來說，所謂梯形積分法就是

$$T_n f \Big|_a^b = h \cdot \left(\frac{1}{2} f(x_0) + f(x_1) + \cdots + f(x_{n-1}) + \frac{1}{2} f(x_n) \right) \quad (5.19)$$

使用 Matlab 來寫梯形積分法只需要一行：

```
y = (sum(u) - u(1)/2 - u(n)/2)*h;
```

其中 u 是向量 $f(x_i)$ ， n 是 u 的長度。我們提供一個小程序計算梯形數值積分

```
trapezoid.m
```

讀者可以檢查梯形積分法對於階數小於二的多項式是真確的，所以如果 $f \in C^2$ ，則梯形積分法是二階收斂。也就是

$$\left| \int_a^b f(x) dx - T_{2n} f \Big|_a^b \right| \approx \frac{1}{4} \left| \int_a^b f(x) dx - T_n f \Big|_a^b \right| \quad (5.20)$$

做個實驗來看看。令 $f = \sin x$ ， $[a, b] = [0, \pi]$ 。我們知道積分值是 2。以下這個程序可以做出 $n = 64$ 的 $T_{64} \sin x \Big|_0^\pi$ 。

```
h=pi/64;
x=0:h:pi;
trapezoid(sin(x),h)
```

idx:quad-e1

idx:Taylor

idx:trapez

eq:trapez

eq:traperr

下面是使用梯形積分法算出來的數值和其絕對誤差。

n	梯形積分值	絕對誤差
16	1.9935	.6430E-2
32	1.9984	.1606E-2
64	1.9996	.4016E-3
128	1.9999	.1004E-3

明顯可見其誤差符合 (5.20) 的預測。

另一個常用的數值積分法是所謂的辛普森法。

idx:Simpson

$$S_n f \Big|_a^b = \frac{h}{3} \cdot (f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \cdots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)) \quad (5.21)$$

eq:simp

注意此時的 n 必須是偶數。使用 Matlab 來寫辛普森積分法基本上也只要一行：

$$y = (\text{sum}(x) + \text{sum}(x(2:n-1)) + 2*\text{sum}(x(2:2:n))) * h / 3;$$

其中 u 是向量 $f(x_i)$ ， n 是 u 的長度。我們提供一個小程序計算辛普森數值積分

simpson.m

讀者可以檢查辛普森積分法對於階數小於四的多項式是真確的，所以如果 $f \in C^4$ ，則辛普森積分法是四階收斂。也就是

$$\left| \int_a^b f(x) dx - S_{2n} f \Big|_a^b \right| \approx \frac{1}{16} \left| \int_a^b f(x) dx - S_n f \Big|_a^b \right| \quad (5.22)$$

eq:simperr

我們再對 $\int_0^\pi \sin x \, dx$ 做一次實驗，結果如下。

n	辛普森積分值	絕對誤差
16	2.000016591	.1659E-4
32	2.000001033	.1033E-5
64	2.000000064	.6453E-7
128	2.000000004	.4032E-8

檢查

$$\frac{S_{16} \sin x \Big|_0^\pi}{S_{32} \sin x \Big|_0^\pi} \approx 16.05, \quad \frac{S_{32} \sin x \Big|_0^\pi}{S_{64} \sin x \Big|_0^\pi} \approx 16.01, \quad \frac{S_{64} \sin x \Big|_0^\pi}{S_{128} \sin x \Big|_0^\pi} \approx 16.00$$

可見其誤差符合 (5.22) 的預測，而且使用辛普森法的積分值比使用梯形法來得準確。

以上的實驗顯示，當被積分函數 $f(x)$ 夠平滑的時候，數值積分的表現一如理論所描述。但是如果 $f(x)$ 不夠平滑的時候會怎樣呢？並非數值積分就不能使用，而是 (5.20) 和 (5.22) 之中的理論不能派上用場，而且積分的結果通常會變差。

idx:quad-f

以 $p = 2$ 階的 $\phi(x)$ 為例，我們知道

$$\int_0^3 \phi^2(x) \, dx = 1$$

但是這個 $\phi(x)$ 只是個連續函數，它連一次導函數也沒有。所以 $\phi(x) \in C^0$ ，前面的誤差估計理論通通失效。下面是 $T_n \phi^2(x) \Big|_0^3$ 和 $S_n \phi^2(x) \Big|_0^3$ 的實驗結果。

n	梯形算法	辛普森算法	梯形絕對誤差	辛普森絕對誤差
16	0.9872	0.9423	0.01283	0.05771
32	0.9811	0.9831	0.01892	0.01689
64	1.0004	0.9939	0.00036	0.00607
128	0.9986	0.9992	0.00142	0.00082

由上面的數據可以看得出來，對一個不平滑的函數，梯形和辛普森數值積分算法都沒有一個明確的收斂階數，而且辛普森法的計算結果也未必比梯形法的計算結果更準確。

習題 5.3 證明梯形積分法對一階多項式 x 是準確的。

習題 5.4 證明辛普森積分法對三階多項式 x^3 是準確的。

5.6 自格函數與多項式的積分

由前一節的討論，我們知道由於自格函數不夠平滑，所以傳統的數值積分方法會變得比較沒有效率。但是，其實可以根據自格函數本身的特性，設計一些特殊的方法，使得有關自格函數的數值積分變得更準確。在這一節裡，我們將 $\phi(x)$ 與多項式乘積的積分問題轉換成矩陣問題，所以其積分值是理論上的準確。我們從最簡單的問題形態開始討論，然後介紹兩種變形。我們並不能窮舉所有可能的積分問題；當然還有許多種不同形態的問題有待探討。我們在此的目的只是示範幾種原則性的技巧，而希望讀者能夠掌握這些技巧，以利用來解決其他種類的計算問題。

我們先考慮 $\langle x^m, \phi(x) \rangle$ 形式的積分問題。假設 $\phi(x)$ 是階數為 p 的自格函數， m 是一個非負整數。定義 $\phi(x)$ 的 m 次動量 (*math moment*) 為

$$M_m = \int x^m \phi(x) dx \quad (5.23)$$

idx:moment-m

eq:Mm

由於 $\int \phi dx = 1$ ，故 $M_0 = 1$ ；當 $m \geq 1$ 時，由自格等式 (3.4) 和 $\sum c_k = 2$ 推得

$$M_m = \sum_k c_k \int x^m \phi(2x - k) dx = \frac{1}{2 \cdot 2^m} \sum_k c_k \int (x + k)^m \phi(x) dx$$

將 $(x+k)^m$ 做二項展開得

$$x^m + \sum_{\ell=1}^m \binom{m}{\ell} k^\ell x^{m-\ell}$$

因此整理後得到

$$M_m = \frac{1}{2^m} M_m + \frac{1}{2 \cdot 2^m} \sum_k c_k \sum_{\ell=1}^m \binom{m}{\ell} k^\ell M_{m-\ell}$$

因此

$$M_m = \frac{1}{2(2^m - 1)} \sum_k c_k \sum_{\ell=1}^m \binom{m}{\ell} k^\ell M_{m-\ell} \quad (5.24)$$

eq:formula

這是一個遞迴的公式，由 $M_0 = 1$ 開始。所以利用這遞迴公式可以正確地算出 M_1, M_2, \dots 的值。

爲了符號上的方便，我們再定義

$$M_{m,n} = \int x^m \phi(x-n) dx \quad (5.25)$$

eq:Mmn

顯然 $M_{m,0} = M_m$ 。

由 prep. polys. 回顧第三章， $\{\phi(x-k)\}$ 可以在任何區間中展開階數 $\leq p-1$ 的多項式。所以在任何區間內

$$\sum_n \langle x^m, \phi(x-k) \rangle \phi(x-k) = \begin{cases} x^m & \text{for } 0 \leq m \leq p-1 \\ \mathcal{P}_0 x^m \neq x^m & \text{for } m \geq p \end{cases}$$

而

$$\begin{aligned} \langle x^m, \phi(x-k) \rangle &= \int x^m \phi(x-k) dx = \int (x+k)^m \phi(x) dx \\ &= \sum_{\ell=0}^m \binom{m}{\ell} k^\ell M_{m-\ell} \end{aligned}$$

比如說，當 $p \geq 2$

$$\sum_k (k + M_1) \phi(x - k) = x \quad (5.26)$$

eq:xone

一般來說，我們得到以下的性質；讀者應將此性質與性質 3.6 對照。

性質 5.1 若 $\phi(x)$ 為 p 階自格函數且 $0 \leq m \leq p - 1$ ，則存在一個 k 的 m 階多項式 prop:sumpoly

$$\tau(k) = k^m + a_{m-1}k^{m-1} + \cdots + a_1k + a_0$$

使得

$$\sum_k \tau(k) \phi(x - k) = x^m$$

注意性質 5.1 中 $\tau(k)$ 的首項係數必為 1，而低階項的係數 a_i 當然都應該是整數。當 $m = 1$ 時由 (5.26) 得知 $\tau(k) = k + M_1$ ，對更大的 m 我們也可以用類似的方法求得 $\tau(k)$ ，但是結果比較複雜。根據 (5.26) 這個特殊的結果，Sweldens 和 Piessens [15] 得到一個有趣的性質。嚴健彰先生也得到同樣的結果，但是他以傅立葉技巧來證明。請參考他的碩士論文 [19]。

性質 5.2 若 $p \geq 2$ ， $\phi(x)$ 為 p 階自格函數，則

prop:Mone

$$\int x^2 \phi(x) dx = \left(\int x \phi(x) dx \right)^2$$

證明：對任意 $k \in \mathbb{Z}$ ，考慮以下的積分

$$\int x \phi(x) \phi(x - k) dx$$

則

$$\begin{aligned} \int x\phi(x)\phi(x+k) dx &= \int (x-k)\phi(x-k)\phi(x) dx \\ &= \int x\phi(x)\phi(x-k) dx - k \int \phi(x)\phi(x-k) dx = \int x\phi(x)\phi(x-k) dx \end{aligned}$$

其中最後一個等式是因為 $\{\phi(x-k)\}$ 的正交性質。因此

$$\begin{aligned} \sum_{k=-\infty}^{\infty} k \int x\phi(x)\phi(x-k) dx \\ &= \sum_{k=1}^{\infty} k \int x\phi(x)\phi(x-k) dx + \sum_{k=1}^{\infty} (-k) \int x\phi(x)\phi(x+k) dx \\ &= \sum_{k=1}^{\infty} k \int x\phi(x)\phi(x-k) dx - \sum_{k=1}^{\infty} k \int x\phi(x)\phi(x-k) dx \\ &= 0 \end{aligned}$$

另一方面

$$\sum_{k=-\infty}^{\infty} k \int x\phi(x)\phi(x-k) dx = \int x\phi(x) \left(\sum_{k=-\infty}^{\infty} k\phi(x-k) \right) dx$$

由 (5.26) 和 (3.71) 得知

$$\begin{aligned} x &= \sum_k (k + M_1)\phi(x-k) = \sum_k k\phi(x-k) + M_1 \sum_k \phi(x-k) \\ &= \sum_k k\phi(x-k) + M_1 \end{aligned}$$

結合前面三條式子，得到

$$\begin{aligned} 0 &= \int x\phi(x)(x - M_1) dx = \int x^2\phi(x) dx - M_1 \int x\phi(x) dx \\ &= \int x^2\phi(x) dx - \left(\int x\phi(x) dx \right)^2 \end{aligned}$$

故得證。

□

經過變數變換和凌波函數的定義 (3.20)，我們可以求得以下形式積分的值。

$$\int x^m \phi_{jk}(x) dx, \quad \int x^m \psi_{jk}(x) dx$$

但是要注意當 m 過大時 (以雙精度浮點數計算而言, 八階多項式就算很高階的了), 雖然 M_m 的值都是理論上的真確, 但由於多項式計算的不穩定性, 浮點數計算的誤差可能使得整個計算的數值不準。一個數值計算的基本原則是: 盡量避免高階多項式的運算。

idx:thm-exact

如果 $f(x) \in C^s$, 也就是說 $f(x)$ 有 s 次導函數, 而且第 s 次的導函數是連續函數。則 $f(x)$ 可以對某個參考點 x_0 做泰勒展開

$$f(x) = \sum_{i=0}^{s-1} \frac{f^{(i)}(x_0)}{i!} (x - x_0)^i + \frac{f^{(s)}(\xi)}{s!} (x - x_0)^s$$

其中 ξ 是某個在 x 和 x_0 之間的數, 我們不能確定它是誰。參照 (5.10) 和 (5.12), 我們推論

idx:Taylor

$$\begin{aligned} \langle f(x), \phi_{jk}(x) \rangle &= \sum_{i=0}^{s-1} \frac{f^{(i)}(x_0)}{i!} \langle (x - x_0)^i, \phi_{jk}(x) \rangle \\ &\quad + \frac{1}{s!} \langle f^{(s)}(\xi)(x - x_0)^s, \phi_{jk}(x) \rangle \\ &= \sum_{i=0}^{s-1} \frac{f^{(i)}(x_0)}{i!} \langle (x - x_0)^i, \phi_{jk}(x) \rangle + O\left(\frac{1}{\sqrt{2^j} \cdot 2^{sj}}\right) \end{aligned}$$

由此可見, 即使我們不知道任何的有關自格函數的特殊積分法, 還是可以利用 $f(x)$ 的泰勒展開式, 將 $\langle f, \phi_{jk} \rangle$ 的積分問題換成 $f(x)$ 的泰勒多項式與 $\phi_{jk}(x)$ 之間的積分問題。而多項式與 $\phi(x)$ 的積分值是可以真確求得的。

再補充兩點。第一，如果不便使用 $f(x)$ 的泰勒展開式，還有數值積分法可以使用，這是下一節的主題。第二，如果 $f(x)$ 不可微，因此沒有泰勒展開式可言，那麼我們說 $f(x)$ 本身的性質不良，因此根本就沒有一個可靠的數值積分方法可用。這種時候，最簡單的對策就是使用單純的梯形積分法，而且取非常大的點數 n ；當然這表示要消耗大量的計算資源，但這是 $f(x)$ 的本質問題，我們也無計可施。

我們已經計算過 M_m 的數值 (與 p 有關)，儲存在資料檔中：

```
Mm.mat
```

在 load Mm.mat 之後，會取出一個 11×10 的矩陣 M 。而 $M(i, j)$ 是階數 $p = j$ 的 M_{i-1} 值。但是請注意，因為我們使用的是雙精度的浮點數計算，所以當 $i \geq 7$ 時，我們不能肯定其數值的可靠性。如果要求更可靠的值，應該用四倍精度浮點數，或是利用像 Maple 之類的軟體，使用更高位數的計算。

此外，還有一個 Matlab 程序

```
xphi.m
```

目的是計算 $M_{m,n}$ 的值。使用方法是 `xphi(p,m,n)`，其中 p 就是自格函數的階數， m 是 x^m 的冪數， n 是 $\phi(x-n)$ 的平移量，參見 (5.25)。參數 p 、 m 、 n 都應該是整數，但 $1 \leq p \leq 10$ 、 $0 \leq m \leq 10$ 。這個程式呼叫一些副程式：`getb`、`xphi_1` 和 `choose`。其中

```
choose.m
```

就是計算二項係數 $\binom{n}{k}$ ，使用方法是 `choose(n,k)`。這個程式寫得比較小心，比較不容易出錯。

再一個 Matlab 程序

polyphi.m

目的是計算

$$\int q(x)\phi(2^j x - k) dx$$

呼叫的指令是 `polyphi(p,q,j,k)`。其中 $\phi(x)$ 的階數 p 仍然限制在 $1 \leq p \leq 10$ 。若

$$q(x) = q_m x^m + q_{m-1} x^{m-1} + \cdots + q_1 x + q_0$$

是一個 m 階多項式，其中 $0 \leq m \leq 10$ ，則

$$q = (q_m, q_{m-1}, \dots, q_1, q_0)$$

是一個長度 $m+1$ 的序列。

前面考慮的都是整個實數上的積分，現在我們考慮一個變形：在有限區間內的積分。一般形式的問題是

$$\int_a^b f(x)\phi(2^j x - k) dx \tag{5.27} \quad \text{eq:yaa}$$

就像前面的處理方式， $f(x)$ 可以先假定為多項式；當它不是多項式的時候，利用其泰勒展開來處理。

如果 $\text{supp } \phi_{jk} \subset [a, b]$ ，那麼在 $[a, b]$ 內的積分其實就等於在 \mathbb{R} 上的積分。所以可以用前面的方法來計算。但是如果 $\text{supp } \phi_{jk}$ 跨過邊界點，那就是新的情況了。現在我們就是要探討這種情況的積分問題。

我們先將問題簡化。令 j 足夠大，以至於 $|\text{supp } \phi_{jk}| < b - a$ ，所以 $\phi_{jk}(x)$ 一次最多只跨過一個邊界點。我們再假設 $a = \frac{n}{2^j}$ 而 $\phi_{jk}(x)$ 跨過了 a 點。所以 a 就在 $\text{supp } \phi_{jk}$ 的內部：

$$k < n < k + 2p - 1$$

經過變數變換和 $\text{supp } \phi = [0, 2p - 1]$ ，(5.27) 可以改寫成

$$\begin{aligned} \frac{1}{2^j} \int_{2^j a - k}^{2^j b - k} f\left(\frac{x+k}{2^j}\right) \phi(x) dx &= \frac{1}{2^j} \int_{n-k}^{2p-1} f\left(\frac{x+k}{2^j}\right) \phi(x) dx \\ &= \frac{1}{2^j} \int_{n-k}^{\infty} f\left(\frac{x+k}{2^j}\right) \phi(x) dx = \frac{1}{2^j} \int_0^{\infty} f\left(\frac{x+n}{2^j}\right) \phi(x + (n-k)) dx \end{aligned}$$

由於被積分函數 $f(\frac{x+k}{2^j})$ 是個多項式，因此可以只考慮最基本的形態的問題：

$$N_{m,n} = \int_0^{\infty} x^m \phi(x-n) dx \tag{5.28} \quad \text{eq:Nmn}$$

若 $n \geq 0$ 則 $\phi(x-n)$ 的函蓋全部落在 $[0, \infty)$ 裡面，也就是說 $N_{m,n} = M_{m,n}$ ；所以不須考慮 $n \geq 0$ 的情形。又因為當 $n \leq 1-2p$ ，則 $\phi(x-n)$ 的函蓋全部落在 $[0, \infty)$ 外面，也就是說 $N_{m,n} = 0$ ；所以也不須考慮 $n \leq 1-2p$ 的情形。現在只需考慮 $2-2p \leq n \leq -1$ 的情形。

利用自格等式 (3.4) 與變數變換可得

$$\int_0^{\infty} x^m \phi(x-n) dx = \frac{1}{2^{m+1}} \sum_{k=0}^{2p-1} c_k \int_0^{\infty} x^m \phi(x - (2n+k)) dx$$

若 $1-p \leq n \leq -1$ ，有一部分的 $\phi(x - (2n+k))$ 完全落在 $[0, \infty)$ 之內。也就是當 $k \geq -2n$ 的時候。綜合以上結果，我們得到

$$N_{m,n} = \frac{1}{2^{m+1}} \left(\sum_{k < -2n} c_k N_{m,2n+k} + \sum_{k \geq -2n} c_k M_{m,2n+k} \right) \tag{5.29} \quad \text{eq:yab}$$

其中 $n = -1, -2, \dots, 2 - 2p$ 。假設 $M_{m,*}$ 為已知，則 (5.29) 可以寫成一個線性系統。

令 $p \geq 2$ 且 m 為一非負整數， $\mathbf{x} = (x_i)$ 與 $\mathbf{b} = (b_i)$ 為 $2p - 2$ 維之向量

$$x_i = N_{m,-i} \quad \text{and} \quad b_i = \frac{1}{2^{m+1}} \sum_{k=2i}^{2p-1} c_k M_{m,2i+n}$$

其中 $i = 1, 2, \dots, 2p - 2$ 。注意，當 $i \geq p$ 的時候， $b_i = 0$ 。則 (5.29) 可以改寫成

$$\left(I - \frac{1}{2^{m+1}}\mathbf{A}\right)\mathbf{x} = \mathbf{b} \tag{5.30} \quad \text{eq:eqA}$$

其中 I 是 $(2p - 2) \times (2p - 2)$ 的單位矩陣、 \mathbf{A} 是一個 $(2p - 2) \times (2p - 2)$ 的方陣，而 $a_{ij} = c_{2i-j}$ 。回顧 (5.13)，發現其實這個矩陣 \mathbf{A} 就是 Strang 矩陣。只是在第三節之時，我們探討 Strang 矩陣的特徵值問題，現在我們探討它的線性方程組問題。 idx:Strang-ma

根據一個線性代數的定理，如果 $\|\frac{1}{2^{m+1}}\mathbf{A}\| < 1$ ，則 (5.30) 有解。但是根據性質 5.3，而且 $m \geq 0$ ，所以 $\|\frac{1}{2^{m+1}}\mathbf{A}\| \leq \frac{\sqrt{2}}{2} < 1$ ，故 (5.30) 有解。而這個解就是 $N_{m,-1}, \dots, N_{m,2-2p}$ 在理論上的真確值。但是在求解線性系統的過程中難免產生誤差，所以實際上當然未必是真確的。這裡看到的 $\|\mathbf{A}\|$ 是矩陣 \mathbf{A} 的長度，回顧 (2.7)。

性質 5.3 令 $p \geq 2$ 而 c_k 是 p 階自格係數，定義 Strang 矩陣 \mathbf{A} 為 $(2p - 2) \times (2p - 2)$ 之方陣，且 $a_{ij} = c_{2i-j}$ 。則 prop:norma

$$\|\mathbf{A}\| \leq \sqrt{2} \quad (\text{對任意的 } p \geq 2)$$

idx:Strang-ma

性質 5.3 的證明可能是超出了這本書想要函蓋的範圍。但是因為它很短，而且又對自格係數之性質做了有趣的應用。所以我們將它附在這裡。

證明：令

$$\mathbf{S} = \begin{pmatrix} c_1 & c_0 & & 0 \\ & & \ddots & \\ c_{2p-3} & c_{2p-4} & \cdots & c_0 \end{pmatrix}$$

和

$$\mathbf{T} = \begin{pmatrix} c_{2p-1} & c_{2p-2} & \cdots & c_2 \\ & \ddots & & \\ 0 & & c_{2p-1} & c_{2p-2} \end{pmatrix}$$

則

$$\mathbf{A} = \begin{pmatrix} \mathbf{S} \\ \mathbf{T} \end{pmatrix} \quad (5.31) \quad \text{eq:yac}$$

由 (3.6) 可知

$$\mathbf{S}\mathbf{T}^T = \mathbf{T}\mathbf{S}^T = \mathbf{0} \quad (5.32) \quad \text{eq:yad}$$

因此

$$\mathbf{A}\mathbf{A}^T = \begin{pmatrix} \mathbf{S}\mathbf{S}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{T}\mathbf{T}^T \end{pmatrix}$$

所以，假如 (λ, \mathbf{v}) 是 $\mathbf{S}\mathbf{S}^T$ 的一組特徵值和其相對的特徵向量，則 λ 必定也是 $\mathbf{A}\mathbf{A}^T$ 的特徵值，而其對應的特徵向量是 $\begin{pmatrix} \mathbf{v} \\ \mathbf{0} \end{pmatrix}$ 。再由 (3.6) 可知

$$\mathbf{S}\mathbf{S}^T + \mathbf{T}\mathbf{T}^T = 2\mathbf{I} \quad (5.33) \quad \text{eq:yae}$$

因此 $2 - \lambda$ 也是 $\mathbf{A}\mathbf{A}^T$ 的特徵值，而其對應的特徵向量是 $\begin{pmatrix} \mathbf{0} \\ \mathbf{v} \end{pmatrix}$ 。但 $\mathbf{A}\mathbf{A}^T$ 是對稱半正定矩陣，所以

$$\lambda \geq 0 \quad \text{而且} \quad 2 - \lambda \geq 0$$

也就是 $0 \leq \lambda \leq 2$ ，故 $\|\mathbf{A}\mathbf{A}^T\| \leq 2$ 或者 $\|\mathbf{A}\| \leq \sqrt{2}$ 。 □

根據性質 5.3，因為對任何 $m \geq 0$ 都有 $\|\frac{1}{2^{m+1}}\mathbf{A}\| < 1$ ，所以 Neumann 級數

$$\left(\mathbf{I} - \frac{1}{2^{m+1}}\mathbf{A}\right)^{-1} = \sum_{n=0}^{\infty} \left(\frac{1}{2^{m+1}}\mathbf{A}\right)^n$$

將會收斂。這是一個求解 (5.30) 的好方法。

我們曾經計算出來 $N_{m,n}$ 的數值，儲存在資料檔

Nm.n.mat

當您 load Nm.n.mat 之後，會取得 p2、...、p14 十三個矩陣。每個矩陣的維度是 $(2p - 2) \times 15$ 。而 pp(n,m) 就是階數 p 的 $N_{m-1,-n}$ ，參照 (5.28)。譬如說，p = 7 的 $N_{3,-12}$ 就是 p7(12,4)。

對 Matlab 不夠熟悉的讀者們或許會發現資料檔中的數字只有五位有效數字。其實 Matlab 內部一律用雙精度儲存數字，也就是大約有十六位有效數字。為了呈現上的方便，Matlab 內定以五位有效數字的方式輸出在螢幕上。如果您要看更多的數目字，可以說 format long；如果您要看五位有效數字的科學記號，可以說 format short e；如果要還原，說 format short。

習題 5.5 證明

$$M_{m,n} = \sum_{\ell=0}^m \binom{m}{\ell} n^\ell M_{m-\ell}$$

習題 5.6 令 $p = 2$ ，利用 Mm.mat 裡面的數據和 Code-15 繪製 $\sum(k + M_1)\phi(x - k)$ 在 $[0, 3]$ 內的圖形。

習題 5.7 利用 Mm.mat 裡面的數據，驗證性質 5.2。

習題 5.8 針對 $1 \leq p \leq 4$ 和 $0 \leq m \leq 6$ ，計算 $\int x^m \psi(x) dx$ 的數值。

idx:Neumann

idx:matlab

習題 5.9 令 $p \geq 3$ ，找到 $\tau(k)$ 使得 $\sum \tau(k)\phi(x-k) = x^2$ 。

習題 5.10 證明 (5.32)。

習題 5.11 證明 (5.33)。

5.7 自格函數的連乘積分問題

這一節的主題是探討兩個自格函數相乘後的積分問題：

$$\int f(x)\phi^{(r)}(x)\phi^{(s)}(x-n) dx$$

其中 r 和 s 是非負整數，代表 $\phi(x)$ 的導函數次數。我們在此只考慮 $0 \leq r, s \leq 1$ 的情況。當係數函數 $f(x) = 1$ 時，這種積分值又稱做連接係數 (*connection coefficients*)。當我們使用有限元素法或廣義的 Galerkin 方法計算積分或微分方程的數值解，就需要做這種積分運算。在有限元的術語中，稱作剛度矩陣 (*stiffness matrix*) 或質量矩陣 (*mass matrix*) 的計算。

idx:conn-co

idx:massf

先考慮連接係數問題。若 $r = s = 0$ ，那麼就是計算

$$T_n = \int \phi(x)\phi(x-n) dx \quad (5.34)$$

eq:Tn

到目前為止我們所考慮的自格函數都是 Daubechies 的有限函蓋正交函數，所以上述問題根本不需計算，就是 $T_0 = 1$ 而當 $n \neq 0$ 時 $T_n = 0$ 。但是如果 $\phi(x)$ 不是正交類型的自格函數，那麼 (5.34) 就有戲可看了。以後讀者們會遇到非正交的自格函數，而且在稍後的討論，我們期盼用到一種普遍性的方法。所以我們還是認真地來探討 (5.34) 的計算方法。

如果令

$$T(x) = \int \phi(\xi)\phi(\xi-x) d\xi, \quad x \in \mathbb{R} \quad (5.35)$$

eq:Tx

(注意，這個積分形式很像疊積 (2.23)，其實不是。) 由於 ϕ 的函蓋是 $[0, 2p - 1]$ ，所以 $T(x)$ 也是個有限函蓋的函數，而且 $\text{supp } T = [-(2p - 1), 2p - 1]$ 。那麼 (5.34) 就相當於求函數 $T(x)$ 在整數點的值。根據下面的性質 5.4，我們知道 $T(x)$ 自己也是個自格函數。事實上 $T(x)$ 的函蓋寬度是 $4p - 2$ ，而它有 $4p - 1$ 個非零的自格係數。因此，回顧第三節中求自格函數整數點值的方法，我們可以利用 $T(x)$ 的自格係數製造 Strang 矩陣，而 $T(x)$ 的整數點值就形成 Strang 矩陣對應特徵值 $\lambda = 1$ 的一個特徵向量。

idx:Strang-ma

性質 5.4 設若有兩個自格函數

prop:Tx

$$u(x) = \sum_k a_k u(2x - k), \quad v(x) = \sum_k b_k v(2x - k)$$

定義新函數

$$T(x) = \int u(\xi) v(\xi - x) d\xi$$

則 $T(x)$ 也是一個自格函數

$$T(x) = \sum_k h_k T(2x - k)$$

其中

$$h_k = \frac{1}{2} \sum_n a_{k+n} b_n$$

證明：只是直接的計算與變數變換或足標變換而已：

$$\begin{aligned} T(x) &= \int u(\xi) v(\xi - x) d\xi \\ &= \int \sum_m a_m u(2\xi - m) \sum_n b_n v(2(\xi - x) - n) d\xi \end{aligned}$$

$$\begin{aligned}
 &= \frac{1}{2} \sum_m \sum_n a_m b_n \int u(\xi) v(\xi - 2x + m - n) d\xi \\
 &= \frac{1}{2} \sum_k \sum_n a_{k+n} b_n \int u(\xi) v(\xi - 2x + k) d\xi \\
 &= \sum_k \left(\frac{1}{2} \sum_n a_{k+n} b_n \right) T(2x - k) \quad \square
 \end{aligned}$$

令 c_k 是原來 $\phi(x)$ 的自格係數。回顧 Strang 矩陣的構造方式 (5.14) 並對照性質 5.4, $\mathbf{T} = (t_{ij})$ 的元素是

$$t_{ij} = h_{2i-j} = \frac{1}{2} \sum_n c_{2i-j+n} c_n = \frac{1}{2} \sum_k c_k c_{k+j-2i} \quad (5.36)$$

對照 (3.14), 我們發現這時候的矩陣 \mathbf{T} 其實就是 *Lawton* 矩陣。

矩陣 \mathbf{T} 看似複雜, 但其實有一個簡單的形式。考慮 \mathbf{T} 的第 j 行 (column) 的行向量 \mathbf{T}_j 。由 (3.6) 得到, 若 j 為偶數, 則 \mathbf{T}_j 的元素 t_{ij} 全都是 0, 除了當 $i = \frac{j}{2}$ 時 $t_{ij} = 2$; 若 j 為奇數, 則 \mathbf{T}_j 的元素 t_{ij} 全都是 0, 除了當

$$\lceil \frac{j-2p}{2} \rceil \leq i \leq \lfloor \frac{j+2p}{2} \rfloor$$

這時的 (t_{ij}) 是長度為 $2p$ 的行向量, 其元素如下:

$$\frac{1}{2} \begin{pmatrix} \sum c_k c_{k+(2p-1)} \\ \vdots \\ \sum c_k c_{k+3} \\ \sum c_k c_{k+1} \\ \sum c_k c_{k+1} \\ \sum c_k c_{k+3} \\ \vdots \\ \sum c_k c_{k+(2p-1)} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \int \phi(x) \phi(x - \frac{2p-1}{2}) dx \\ \vdots \\ \int \phi(x) \phi(x - \frac{3}{2}) dx \\ \int \phi(x) \phi(x - \frac{1}{2}) dx \\ \int \phi(x) \phi(x - \frac{1}{2}) dx \\ \int \phi(x) \phi(x - \frac{3}{2}) dx \\ \vdots \\ \int \phi(x) \phi(x - \frac{2p-1}{2}) dx \end{pmatrix}$$

idx:Strang-

eq:tij

idx:Lawton-

階數 $2 \leq p \leq 14$ 的 Lawton 矩陣都儲存在 Matlab 的資料檔

T.mat

當您 load T.mat，會得到 T2、...、T14 十三個矩陣，其中 T_p 就是由 p 階自格係數 c_k 所製造的 Lawton 矩陣 T。

要求 (5.34) 中的積分值 T_n，就等於求 T(x) 的整數點值，而這又等於求 Lawton 矩陣 T 對應於特徵值 $\lambda = 1$ 的特徵向量。如果 $\phi(x)$ 是正交自格函數，則這個特徵向量將會是一個單位向量。如果 $\phi(x)$ 不是正交自格函數，則會出現其他的結果。回顧第四節中的討論，特徵向量並不唯一，我們需要某種方法來決定一個正確的特徵向量。即使非正交的自格函數，通常也會具備以下性質：

idx:eig-val

$$\sum_n \phi(x - n) = 1$$

還是可以利用這個性質來確定一個特徵向量：

$$\sum_n T_n = \int \phi(x) \sum_n \phi(x - n) dx = \int \phi(x) dx = 1$$

接著，如果 $\phi(x)$ 一次可微，我們考慮兩種連接係數的問題：

$$T_n^{(1)} = \int \phi(x) \phi'(x - n) dx \quad (5.37) \quad \text{eq:Tnone}$$

和

$$T_n^{(2)} = \int \phi'(x) \phi'(x - n) dx \quad (5.38) \quad \text{eq:Tntwo}$$

與前面 T_n 的情況相同，只有當 $2 - 2p \leq n \leq 2p - 2$ 時，T_n⁽¹⁾ 和 T_n⁽²⁾ 才有非零的值。回顧 (5.15)， $\phi'(x)$ 還是一個自格函數，其自格係數為

$2c_k$ ，所以可以套用性質 5.4。我們發現 Lawton 矩陣 \mathbf{T} 還是會出現，而且

$$2\mathbf{T} \begin{pmatrix} \Gamma_{2-2p}^{(1)} \\ \vdots \\ \Gamma_0^{(1)} \\ \vdots \\ \Gamma_{2p-2}^{(1)} \end{pmatrix} = \begin{pmatrix} \Gamma_{2-2p}^{(1)} \\ \vdots \\ \Gamma_0^{(1)} \\ \vdots \\ \Gamma_{2p-2}^{(1)} \end{pmatrix}, \quad 4\mathbf{T} \begin{pmatrix} \Gamma_{2-2p}^{(2)} \\ \vdots \\ \Gamma_0^{(2)} \\ \vdots \\ \Gamma_{2p-2}^{(2)} \end{pmatrix} = \begin{pmatrix} \Gamma_{2-2p}^{(2)} \\ \vdots \\ \Gamma_0^{(2)} \\ \vdots \\ \Gamma_{2p-2}^{(2)} \end{pmatrix}$$

也就是說 \mathbf{T} 有特徵值 $\frac{1}{2}$ 和 $\frac{1}{4}$ 。令 \mathbf{x} 和 \mathbf{y} 分別是對應 $\frac{1}{2}$ 和 $\frac{1}{4}$ 的特徵向量，則 $(\Gamma_n^{(1)})$ 和 \mathbf{x} 成比例， $(\Gamma_n^{(2)})$ 和 \mathbf{y} 成比例。以下要探討如何確定比例常數。

當 $\phi(x)$ 一次可微，都是 $p \geq 3$ ，所以 V_0 可以展開 $\{1, x, x^2\}$ 。根據性質 3.6 和 (5.26)，我們知道

$$\sum_n n\phi(x-n) = x - M_1$$

所以

$$\begin{aligned} \sum_n n\Gamma_n^{(1)} &= \int \phi(x) \sum_n n\phi'(x-n) dx = \int \phi(x)(x - M_1)' dx \\ &= \int \phi(x) dx = 1 \end{aligned}$$

這就是決定 $(\Gamma_n^{(1)})$ 和 \mathbf{x} 比例常數的辦法。再由

$$\sum_n n^2\phi(x-n) = x^2 + \alpha x + \beta$$

運用分部積分 (integration by parts) 和

$$\int \phi'(x) dx = \phi(x) \Big|_{-\infty}^{\infty} = \phi(x) \Big|_0^{2p-1} = 0$$

我們推導 (這一段取自蘇智穎先生的碩士論文 [16])

$$\begin{aligned}
 1 &= \int \phi(x) dx = - \int x \phi'(x) dx \\
 &= - \int (x + \frac{\alpha}{2}) \phi'(x) dx \\
 &= -\frac{1}{2} \int (x^2 + \alpha x + \beta)' \phi'(x) dx \\
 &= -\frac{1}{2} \int \left(\sum_n n^2 \phi(x-n) \right)' \phi'(x) dx \\
 &= -\frac{1}{2} \sum_n n^2 \int \phi'(x) \phi'(x-n) dx = -\frac{1}{2} \sum_n n^2 T_n^{(2)}
 \end{aligned}$$

所以

$$\sum_n n^2 T_n^{(2)} = -2$$

這就是決定 $(T_n^{(2)})$ 和 y 比例常數的辦法。

最後，我們探討多項式與 $\phi(x)$ 連乘積分的算法。令

$$T_{m,n} = \int x^m \phi(x) \phi(x-n) dx \quad (5.39) \quad \text{eq: Tmn}$$

顯然 $T_{0,n} = T_n$ ，所以我們只考慮 $m \geq 1$ 的情形。如果 $\phi(x)$ 是 p 階的自格函數，給定一個 m ，只有 $2 - 2p \leq n \leq 2p - 2$ 的時候 $T_{m,n}$ 才有非零值。

當 $2 - 2p \leq n \leq 2p - 2$ ，

$$\begin{aligned}
 T_{m,n} &= \int x^m \sum_k c_k \phi(2x-k) \sum_\ell c_\ell \phi(2x-2n-\ell) dx \\
 &= \frac{1}{2} \sum_k \sum_\ell c_k c_\ell \int \left(\frac{x+k}{2} \right)^m \phi(x) \phi(x+k-2n-\ell) dx
 \end{aligned}$$

$$= \frac{1}{2^{m+1}} \left(\sum_k \sum_\ell c_k c_\ell T_{m,2n+\ell-k} + \sum_k \sum_\ell c_k c_\ell \sum_{r=1}^m \binom{m}{r} k^r T_{m-r,2n+\ell-k} \right)$$

觀察 $T_{m,2n+\ell-k}$ 那一項，經過足標變換，整理得到

$$\begin{aligned} \sum_k \sum_\ell c_k c_\ell T_{m,2n+\ell-k} &= \sum_k \sum_i c_k c_{i+k-2n} T_{m,i} \\ &= \sum_i \left(\sum_k c_k c_{k+i-2n} \right) T_{m,i} \end{aligned}$$

若令

$$t_{ni} = \frac{1}{2} \sum_k c_k c_{k+i-2n}$$

則上式變成矩陣與向量的乘積

$$2 \sum_i t_{ni} T_{m,i}$$

令 $\mathbf{T} = (t_{ni})$ ，對照 (5.36) 我們發現 \mathbf{T} 又是 Lawton 矩陣。因此，我們 idx:Lawton- 得到一個線性系統 (可與 (5.30) 對照)

$$\left(\mathbf{I} - \frac{1}{2^m} \mathbf{T} \right) \mathbf{x} = \mathbf{b} \tag{5.40} \quad \text{eq:eqT}$$

其中的向量

$$\mathbf{x} = (T_{m,n}), \quad \mathbf{b} = (b_n) \quad \text{其中 } 2 - 2p \leq n \leq 2p - 2$$

而 \mathbf{b} 的元素是

$$b_n = \frac{1}{2^{m+1}} \sum_{k=1}^{2p-1} c_k \sum_{\ell=\ell_1}^{\ell_2} c_\ell \sum_{r=1}^m \binom{m}{r} k^r T_{m-r,2n+\ell-k}$$

其中

$$\ell_1 = \max\{k - 2n - 2p + 2, 0\}, \quad \ell_2 = \min\{k - 2n + 2p - 2, 2p - 1\}$$

性質 5.5 令 $p \geq 2$ 而 c_k 是 p 階自格係數，令 Lawton 矩陣 \mathbf{T} 如 (5.36) 所定義。則

prop: normT
idx: Lawton-ma

$$\|\mathbf{T}\| \leq \sqrt{2} \quad (\text{對任意的 } p \geq 2)$$

由性質 5.5 可知 $\|I - \frac{1}{2^m} \mathbf{T}\| < 1$ ，故 (5.40) 中的線性系統有唯一解。而性質 5.5 的證明已超出本書意圖函蓋的範圍，故節略。

由於計算 $T_{m,n}$ 的值需要比較高的精度，尤其是當多項式的階數稍高的時候。所以這些數據的計算，當初並非在 Matlab 中運作，而是寫了 FORTRAN 程式，在一台 VAX 機器上以四倍精度計算而成。這些數據以純文字檔的格式儲存在

data/Tmn.dat

若 $\phi(x)$ 是階數 p 的自格函數， $2 \leq p \leq 14$ ，則 $T_{m,n}$ 放在 $T_p(n,m)$ ，其中 $0 \leq m \leq 14$ 而 $2 - 2p \leq n \leq 2p - 2$ 。

顯然我們還應該探討以下幾種積分問題：

$$\begin{aligned} & \int_0^\infty x^m \phi(x) \phi(x-n) dx \\ & \int x^m \phi(x) \phi'(x-n) dx \quad \int_0^\infty x^m \phi(x) \phi'(x-n) dx \\ & \int x^m \phi'(x) \phi'(x-n) dx \quad \int_0^\infty x^m \phi'(x) \phi'(x-n) dx \end{aligned}$$

但是它們都可以用類似的技巧計算出來，所以將細節全部省略了。只要讀者瞭解了本節和前一節的原則性想法，就應該可以自行推導上述積分的算法。

習題 5.12 當 (5.35) 中的 $T(x)$ 是由 p 階的 $\phi(x)$ 製造出來的，請問 $T(x)$ 有幾個非零的整數點值，而 Lawton 矩陣 T 的維度是多少？

習題 5.13 針對 $p = 3, 4, 5$ 的 $\phi(x)$ ，計算 $(T_n^{(1)})$ 和 $(T_n^{(2)})$ 的值。您可以利用 `T.mat` 裡面的 Lawton 矩陣。

習題 5.14 利用 T_n 的值，計算 $\int \psi(x)\psi(x-n) dx$ 的值。

習題 5.15 設計 $\int_0^\infty x^m \phi(x)\phi(x-n) dx$ 的算法。

習題 5.16 設計 $\int x^m \phi'(x)\phi'(x-n) dx$ 的算法。

5.8 自格函數的數值積分法

在前兩節中我們呈現了一些有關自格函數在理論上真確的積分法。這一節的主題是開發一些有關自格函數的數值積分法。這是一個頗為煩瑣 (但不困難) 的課題，我們不能在此列舉所有已知的方法，只能做原則性的介紹。在瞭解了這些設計數值積分法的原則之後，讀者們應該有能力自行開發其他的方法。

我們先探討 Newton-Cotes 類型之數值積分法的設計。這是最基本的類型。然後我們介紹一個很高效率的單點積分法，它應該屬於高斯積分法的類型。但是我們並不花費篇幅介紹一般的高斯積分法，有興趣的讀者可以參考徐敏堂先生的碩士論文 [13]。最後，我們利用單點積分法設計一種後驗誤差估計的方法，然後可以利用它來設計調適性的積分法。

idx:quad-N

回顧第六節，我們發現傳統的積分法倚賴於兩件原則：(1) 積分法要對某些多項式是真確的，(2) 被積分函數要足夠平滑。由於 $\phi(x)$ 可能不夠平滑，所以即使 $f(x)$ 平滑，也使得 $f(x)\phi(x)$ 不平滑，所以拿傳統的積分法來處理 $\int f(x)\phi(x) dx$ 的時候會沒有效率。我們現在要發展一種積分法，使得只要 $f(x)$ 夠平滑，而不必理會 $f(x)\phi(x)$ 的平滑程

度，就可以得到高效率的積分結果。其秘訣就在於，我們的積分法要對 $x^m \phi(x)$ 這樣的積分是真確的。

考慮

$$\int_a^b f(x)\phi(x) dx \quad (5.41)$$

eq:xnc

的數值積分問題。由於 $\text{supp } \phi = [0, 2p - 1]$ ，所以我們總是選 $[a, b] = [0, 2p - 1]$ 。我們要建構 $r + 1$ 點數值積分算法

idx:quad-rpt

$$\mathcal{U}_r f \Big|_0^{2p-1} \approx \int_0^{2p-1} f(x)\phi(x) dx$$

參照第六節的 (5.17)，令

$$\mathcal{U}_r f \Big|_0^{2p-1} = \sum_{i=0}^r \alpha_i f(x_i) \quad (5.42)$$

eq:quadUr

其中 x_i 是 $[0, 2p - 1]$ 的 $r + 1$ 個等分節點，所以需要決定的是積分係數 α_i 。而決定 α_i 的條件是，要能對多項式 $1, x, \dots, x^m$ 得到真確的積分值，而且 m 越大越好。也就是希望

idx:quad-coef

$$\mathcal{U}_r x^m \Big|_0^{2p-1} = \int_0^{2p-1} x^m \phi(x) dx$$

由於有 $r + 1$ 個未知數 α_i 待決定，所以我們最多能給 $r + 1$ 個條件。因此最多能對階數 $\leq r$ 的多項式是真確的。因此我們的 $r + 1$ 個條件就是

$$M_m = \mathcal{U}_r x^m \Big|_0^{2p-1} = \sum_{i=0}^r \alpha_i (x_i)^m, \quad 0 \leq m \leq r \quad (5.43)$$

eq:xna

其中 M_m 就是 (5.23) 中定義的積分，我們知道它的真確值。

根據 (5.43)，我們可以寫成一個線性系統

$$\mathbf{H} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_r \end{pmatrix} = \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ M_r \end{pmatrix}$$

其中矩陣 \mathbf{H} 的元素 $h_{ij} = x_{j-1}^{i-1}$ 。求解上面的線性等式，即得 α_i 。但是要注意 \mathbf{H} 是一個很不平衡的矩陣，因此不要設定太大的 r 值。

比如說 $r = 4$ ，那麼 $h = \frac{2^p-1}{4}$ 。這個 h 並不是個小的數字，所以此數值積分的誤差就不小。實際應用的時候，可以將 (5.41) 中的 $\phi(x)$ 利用自格等式 (3.4) 提高解析度到 $\phi_{jk}(x)$ 。然後考慮 $\int f(x)\phi_{jk}(x) dx$ 。這時候的節點是取在 $[\frac{k}{2^j}, \frac{k+2^p-1}{2^j}]$ 之內，所以 $h \approx 2^{-j}$ 就是個比較小的數了。

idx:nodes

如果在 (5.42) 中只取一點，亦即 $r = 0$ ，則 $U_r f|_0^{2^p-1}$ 只對常數多項式真確，而若 $f \in C^1$ 則此方法可以達到一階收斂的效果。底下介紹一種特殊的單點積分法，它也是只用一個積分節點，但是有三階收斂的效果。

考慮

$$\int f(x)\phi_{jk}(x) dx = \frac{1}{\sqrt{2^j}} \int f\left(\frac{x+k}{2^j}\right)\phi(x) dx$$

若 $f \in C^s$ ，取 $f(x)$ 相對於參考點 $\frac{k+\tau}{2^j}$ 的泰勒展開：

$$f\left(\frac{x+k}{2^j}\right) = f\left(\frac{k+\tau}{2^j}\right) + \sum_{n=1}^{s-1} \frac{1}{n!} f^{(n)}\left(\frac{k+\tau}{2^j}\right) \left(\frac{x-\tau}{2^j}\right)^n + \frac{1}{s!} f^{(s)}(\xi) \left(\frac{x-\tau}{2^j}\right)^s$$

其中 τ 是個還沒決定的值。若可以選擇 τ 使得

$$\int (x-\tau)^n \phi(x) dx = 0, \quad \text{for } 1 \leq n \leq s-1 \quad (5.44) \quad \text{eq:xnd}$$

則我們可以設計一個單點積分法：

idx:quad-1pt

$$\frac{1}{\sqrt{2^j}} f\left(\frac{k+\tau}{2^j}\right) \approx \int f(x) \phi_{jk}(x) dx \quad (5.45)$$

eq:onep

而其誤差是

$$\frac{1}{s!} 2^{-j(s+1/2)} \int f^{(s)}(\xi) (x-\tau)^s \phi(x) dx \quad (5.46)$$

eq:oneperr

對於 Daubechies 的自格函數 $\phi(x)$ ，最自然的選擇就是 $\tau = M_1$ 。則從 (5.44) 中確定 $s = 2$ ，代入 (5.46) 而將 2^{-j} 看作 h ，故知道若 $f \in C^2$ 則單點積分法 (5.45) 是二階收斂。但是，根據性質 5.2， $M_2 = (M_1)^2$ 所以很幸運地

$$\int (x - M_1)^2 \phi(x) dx = M_2 - 2(M_1)^2 + (M_1)^2 = 0$$

所以其實在 (5.44) 中可以確定 $s = 3$ ，而代入 (5.46) 發現其實單點積分法 (5.45) 當 $f \in C^3$ 的時候是三階收斂的方法。我們在研究高斯類型的積分法時，發現這就是一個點情況下的高斯積分法。

根據 Mm.mat 中的數據，我們發現 $M_1 \approx 0.2(p+1)$ ，所以 $\frac{k+M_1}{2^j}$ 總是會落在 $\text{supp } \phi_{jk}$ 的內部。

當函數 $f(x)$ 在某一個特定部分發生比較大的變化，而其他部分則比較平緩，則 $\int f(x) \phi(x) dx$ 會在劇烈變化的部分發生較大的誤差，而在平緩的部分發生比較小的誤差。但是因為誤差的計算是累加的，所以那劇烈變化部分所發生的大誤差就使得整個數值積分的誤差變大。為了節省計算量，我們可以只把發生劇烈變化的部分加以細分成更小的區間，而保留誤差已經夠小的部分。這就是調適性 (*adaptive*) 數值積分法的想法。

idx:quad-adap

針對自格函數所設計的調適性積分法，和一般傳統的調適性積分法並不完全相同。讀者可能已經在初級的數值分析課程中學習到調適

性積分法。通常是配合梯形法或低階的高斯積分法使用。其基本做法是在某段區間中做數值積分，然後估計其誤差。如果誤差夠小了，就留下答案。如果不夠小，就將這段區間切半。在其中一半裡面做數值積分，然後估計其誤差。如果誤差夠小了，就留下答案，前去做另外一半；如果不夠小，就將這一半的區間再割一半。如此持續下去，一直到每一個小段區間內的數值積分誤差都達到夠小的標準為止。

而我們是要利用自格等式，使積分 $\int f(x)\phi(x) dx$ 可以上升解析度變成

$$\sum_k c_k \int f(x)\phi(2x - k) dx$$

原本的積分範圍是 $[0, 2p - 1]$ ，提高解析度之後變成 $2p$ 個積分，但每個積分的區域縮小成 $[\frac{k}{2}, \frac{2p-1+k}{2}]$ 。對每一個 k ，我們做 $\int f(x)\phi(2x - k) dx$ 的數值積分，然後檢查其誤差是否夠小。如果夠了就留著它的積分值，如果不夠就將它再上升一層，變成

$$\sum_n c_n \int f(x)\phi(4x - 2k - n) dx$$

如此持續下去，一直到每一小段數值積分 $\int f(x)\phi(2^l x - m) dx$ 的誤差都足夠小，才將它們重組降層到原來的積分問題。

根據前面的描述，讀者應該敏銳地問：在實際使用的時候，我們又不知道真正的積分值是多少，如何知道誤差是多少呢？這就是設計一個調適性積分方法的核心部分了。實際上我們當然不知道真正的積分值（否則還要數值積分幹嘛？），所以我們要利用計算出來的積分數值，來估計它自己的誤差。所以我們並不知道真正的誤差，我們只能估計其誤差。利用已經計算出來的數字，來估計它自己的誤差，這種誤差的估計方式稱為後驗誤差估計 (*a posteriori error estimates*)。下面我們針對單點積分法來設計一個後驗誤差的估計方法。

令 $\phi(x)$ 的階數是 $p \geq 2$ 以及 $f(x) \in C^3$ 。令 M_1 是 $\phi(x)$ 的一次動量，如 (5.23) 所定義。根據單點積分法 (5.45)，將 $f(x)$ 對 $\frac{k+M_1}{2^j}$ 作泰勒展開，得

$$\begin{aligned} f\left(\frac{x+k}{2^j}\right) &= f\left(\frac{k+M_1}{2^j}\right) + f'\left(\frac{k+M_1}{2^j}\right) \left(\frac{x-M_1}{2^j}\right) \\ &\quad + \frac{1}{2!} f^{(2)}\left(\frac{k+M_1}{2^j}\right) \left(\frac{x-M_1}{2^j}\right)^2 + \frac{1}{3!} f^{(3)}(\xi) \left(\frac{x-M_1}{2^j}\right)^3 \end{aligned}$$

其中 ξ 是介於 $\frac{x+k}{2^j}$ 和參考點 $\frac{k+M_1}{2^j}$ 之間的某數。則對於任意的 j, k ，由於單點積分法 (5.45) 及其誤差項 (代入 $s=3$)，得到

$$\begin{aligned} \int f(x) \phi_{jk}(x) dx & \\ &= \frac{1}{\sqrt{2^j}} f\left(\frac{k+M_1}{2^j}\right) + \frac{1}{3!} 2^{-\frac{7}{2}j} \int f^{(3)}(\xi) (x-M_1)^3 \phi(x) dx \end{aligned} \quad (5.47)$$

eq:xnu

如果我們利用自格等式 (3.4) 提升一層解析度之後，就會得到以下的積分問題

$$\begin{aligned} \int \phi_{j,k} f(x) dx &= \sqrt{2^j} \int f(x) \phi(2^j x - k) dx \\ &= \sqrt{2^j} \sum_n c_n \int f(x) \phi(2^{j+1} x - 2k - n) dx \\ &= \frac{1}{\sqrt{2}} \sum_n c_n \int f(x) \phi_{j+1, 2k+n}(x) dx \end{aligned}$$

如果對每一項積分使用單點積分法，則

$$\begin{aligned} \int \phi_{j,k} f(x) dx &= \frac{1}{\sqrt{2}} \sum_n c_n \left[\frac{1}{\sqrt{2^{j+1}}} f\left(\frac{2k+n+M_1}{2^{j+1}}\right) \right. \\ &\quad \left. + \frac{1}{3!} 2^{-\frac{7}{2}(j+1)} \int f^{(3)}(\xi_n) (x-M_1)^3 \phi(x) dx \right] \end{aligned}$$

其中 ξ_n 是介於 $\frac{x+2k+n}{2^{j+1}}$ 和參考點 $\frac{2k+n+M_1}{2^{j+1}}$ 之間的某數。這些數當然未必相同，但是當 j 夠大而且 $f^{(3)}(x)$ 夠平滑的時候，我們假設它們差不多都一樣，而且都差不多等於 (5.47) 中的那個 ξ 。(當然這是不準的。所以我們才說，這是估計誤差的方法，而不是真正計算誤差的方法。) 基於這個假設，則

$$\begin{aligned} & \int \phi_{j,k} f(x) dx \\ & \approx \frac{1}{2\sqrt{2^j}} \sum_n c_n f\left(\frac{2k+n+M_1}{2^{j+1}}\right) \\ & \quad + \left[\sum_n c_n \right] \frac{1}{3!} 2^{-\frac{7}{2}j-4} \int f^{(3)}(\xi)(x-M_1)^3 \phi(x) dx \quad (5.48) \\ & = \frac{1}{2\sqrt{2^j}} \sum_n c_n f\left(\frac{2k+n+M_1}{2^{j+1}}\right) \\ & \quad + \frac{1}{3!} 2^{-\frac{7}{2}j-3} \int f^{(3)}(\xi)(x-M_1)^3 \phi(x) dx \end{aligned}$$

eq:xnv

因為假設了 (5.47) 和 (5.48) 中的 ξ 是同一個數，所以可以用

$$8 \times (5.48) - (5.47)$$

消去有 ξ 的項，得到

$$\begin{aligned} & 7 \int f(x) \phi_{j,k}(x) dx \\ & \approx \frac{4}{\sqrt{2^j}} \sum_n c_n f\left(\frac{2k+n+M_1}{2^{j+1}}\right) - \frac{1}{\sqrt{2^j}} f\left(\frac{k+M_1}{2^j}\right) \end{aligned}$$

也就是說

$$\begin{aligned} & \int f(x) \phi_{j,k}(x) dx \\ & \approx \frac{1}{7} \left[\frac{4}{\sqrt{2^j}} \sum_n c_n f\left(\frac{2k+n+M_1}{2^{j+1}}\right) - \frac{1}{\sqrt{2^j}} f\left(\frac{k+M_1}{2^j}\right) \right] \quad (5.49) \end{aligned}$$

eq:xnt

我們假設 (5.49) 的數值是正確的，就可以將它代回 (5.47) 中去估計其誤差項。這就是一個做後驗誤差估計的方法。

剛開始執行調適性積分法的時候，使用者必須輸入一個誤差審核標準 ϵ 。比如說 $\epsilon = 10^{-4}$ 。這個意思是，根據其後驗誤差估計的結果，要求程式必須保證所得的積分值與真確值之誤差在 ϵ 之內。為了確保這個要求，我們在第一層使用 ϵ 為誤差審核標準 (*criteria*)。但是如果上升了一層，就不能使用同一個標準。因為上升一層之後每個積分的值應該比較小，但是一共有 2^p 個積分要算，所以通常每上升一層，其誤差審核標準應該縮小。我們用以下的程序決定審核標準的變化：若 ϵ 是對 $\int f \phi_{jk} dx$ 的審核標準，則 $\int f \phi_{j+1, 2k+n} dx$ 的審核標準是。

idx:quad-crit

$$\epsilon_n = \frac{\sqrt{2}}{2^p |c_n|} \epsilon$$

那是因為

$$\int \phi_{j,k} f(x) dx = \frac{1}{\sqrt{2}} \sum_n c_n \int f(x) \phi_{j+1, 2k+n}(x) dx$$

如果上式中的每一個積分都符合了誤差審核標準 ϵ_n ，那麼整體的誤差就會

$$\frac{1}{\sqrt{2}} \sum_n |c_n| \epsilon_n \leq \epsilon$$

符合原來那一層的誤差審核標準。

後驗誤差估計的目的是要保證計算結果的準確性。如果後驗誤差估計的值小於誤差審核標準，電腦程式就認為計算得夠精確了。所以後驗誤差估計一定要比真正誤差稍微大一點，否則會出錯。但是又不能大太多，否則就沒有效率。

下面的表格取自徐敏堂先生碩士論文 [13] 的一個片段。他用 $\epsilon = 10^{-3}$ 測試 $\int x^5 \phi dx$ 的積分問題。其中 $\phi(x)$ 的階數是 $p = 2$ 。我們知道如何計算那個積分的真確值。

(j, k)	真的誤差	後驗誤差
(0,0)	9.7985091868044E-02	1.5290024417622E-01
(1,3)	4.9026624362130E-02	5.3791406794917E-02
(2,9)	7.2319649939985E-03	7.4942003501763E-03
(3,21)	7.9195232741114E-04	8.0468313193600E-04
(3,20)	7.2091900622695E-04	7.3307179682506E-04
(3,19)	6.5322284997649E-04	6.6479762665444E-04
(3,18)	5.8886385865975E-04	5.9986062141805E-04
(2,8)	5.8296379784615E-03	6.0657154105413E-03

從這個表格可以看出來，我們設計的后驗誤差估計方法只比真的誤差稍微大一點點，這是一個好的現象。

調適性計算法並不會比其他的計算法更快，其計算複雜度通常是比較高的。使用調適性計算法的目的並非要快速解決問題，而是要一個誤差的保證。如果計算問題本身不夠好，比如說被積分函數 $f(x)$ 不夠平滑，而誤差審核標準又設得太苛，比如說 $\epsilon = 10^{-12}$ ，那麼調適性的計算法可能永遠無法停止。

idx:quad-ac

5.9 傅立葉技巧

前面幾節都是在物理域 (physical domain) 內開發積分算法。其實在頻率域 (frequency domain) 內也有機會發現好的算法。我們在此示範三種利用頻率函數 (傅立葉轉換) 所得到的積分算法。

在求積分方程的數值解時，我們常需要計算二重積分

$$a_{lm} = \iint K(x, y) \phi_{j\ell}(x) \phi_{jm}(y) dx dy \tag{5.50}$$

eq:dint

其中 $K(x, y)$ 叫做核心 (kernel) 函數，一般來說是雙變數函數。但是經常 $K(x, y)$ 其實是一個單變數函數 $K(s)$ 代入 $s = x - y$ 而得。經過變數變換與疊積的定義 (2.23) 推導出

idx:kernel

$$\begin{aligned} & \int \int K(x - y) \phi_{j\ell}(x) \phi_{jm}(y) dx dy \\ &= \frac{1}{2^j} \int \int K\left(\frac{x + \ell - y - m}{2^j}\right) \phi(x) \phi(y) dx dy \\ &= \frac{1}{2^j} \int \phi(x) \int K\left(\frac{(x - y) + \ell - m}{2^j}\right) \phi(y) dy dx \\ &= \frac{1}{2^j} \int \phi(x) \left[K\left(\frac{x - (m - \ell)}{2^j}\right) * \phi(x) \right] (x) dx \end{aligned}$$

根據 Parseval 等式 (2.43)，上式可以寫成在頻率域的內積

idx:Parseval

$$\frac{1}{2^j} \frac{1}{2\pi} \int \hat{\phi}(\omega) \overline{\mathcal{F}\left[K\left(\frac{x - (m - \ell)}{2^j}\right)\right](\omega)} \hat{\phi}(\omega) d\omega$$

其中 $\mathcal{F}[\cdot]$ 表示傅立葉轉換。若 $\hat{K}(\omega)$ 為 $K(s)$ 的傅立葉轉換，則

idx:Ftrans

$$\mathcal{F}\left[K\left(\frac{x - (m - \ell)}{2^j}\right)\right](\omega) = e^{-i(m-\ell)\omega} \hat{K}(2^j\omega) \quad (5.51)$$

eq:yca

所以 (5.50) 可以寫成

$$\begin{aligned} a_{\ell m} &= \frac{1}{2^j} \frac{1}{2\pi} \int |\hat{\phi}(\omega)|^2 \overline{\hat{K}(2^j\omega)} e^{i(m-\ell)\omega} d\omega \\ &= \frac{1}{2^j} \mathcal{F}^{-1}\left[|\hat{\phi}(\omega)|^2 \overline{\hat{K}(2^j\omega)}\right](m - \ell) \end{aligned} \quad (5.52)$$

eq:ycb

其中 $\mathcal{F}^{-1}[\cdot]$ 表示傅立葉逆轉換。

idx:Fback

至於傅立葉轉換與逆轉換的數值計算，已經有許多既成的算法。那是另一門學問，我們不能在此深究。即使不使用特殊的數值傅立葉

轉換計算法，採用一種普通的數值積分方法來計算 (5.52)，也比直接計算 (5.50) 要有效率得多。

接著，我們談談有關 $\psi(x)$ 的连接係數問題。當我們有了 $T_n = \int \phi(x)\phi(x-n) dx$ 的值 (參考第七節)，我們可以根據 $\psi(x)$ 的定義 (3.20) 來計算 $\int \psi(x)\psi(x-n) dx$ 的值，而且那是理論上真確的積分值。但是使用傅立葉技巧，我們發現

$$\int \psi(x)\psi(x-n) dx = 2T_{2n} - T_n \quad (5.53) \quad \text{eq:xpa}$$

所以，其實我們根本不需要使用 (3.20)。但是請注意，這個性質只有當 $\psi(x)$ 是正交凌波函數時才成立。對於以後會遇到的非正交凌波函數，這個關係未必成立。詳情請看底下的證明過程。

性質 5.6 若 $\phi(x)$ 和 $\psi(x)$ 是 Daubechies 之正交自格與凌波函數。若 $\phi(x) \in C^k$ ，則對任意整數 $0 \leq r \leq k$ ， $0 \leq s \leq k$ ，

$$\begin{aligned} \int \psi^{(r)}(x)\psi^{(s)}(x-n) dx = \\ 2^{r+s+1} \int \phi^{(r)}(x)\phi^{(s)}(x-2n) dx - \int \phi^{(r)}(x)\phi^{(s)}(x-n) dx \end{aligned}$$

其實 (5.53) 是性質 5.6 的一個特例 ($r = s = 0$)。爲了要證明性質 5.6，我們需要介紹一些傅立葉轉換的性質。首先，由分部積分，得到

$$\begin{aligned} \mathcal{F}[\psi'(x)](\omega) &= \int e^{-i\omega x} d\psi \\ &= e^{-i\omega x}\psi(x) \Big|_{-\infty}^{\infty} - \int \psi(x)(-i\omega)e^{-i\omega x} dx = i\omega\hat{\psi}(\omega) \end{aligned}$$

其實，我們會發現一個普遍性的結果。

性質 5.7 若 $f(x)$ 為一有限函蓋函數，若 $f(x)$ 及其 r 次導函數均是 L^2 函數，則 prop:fdiff
idx:Ftrans

$$\mathcal{F}[f^{(r)}(x)](\omega) = (i\omega)^r \hat{f}(\omega)$$

根據 (3.48)、(3.76)、(3.78) 和 Parseval 等式 (2.43)，我們知道 idx:Parseval

$$\begin{aligned} \int \psi^{(r)}(x)\psi^{(s)}(x-n) dx &= \frac{1}{2\pi} \int (i\omega)^r \hat{\psi}(\omega) \overline{(i\omega)^s \hat{\psi}(\omega)} d\omega \\ &= \frac{1}{2\pi} \int (i\omega)^r |m_1(\frac{\omega}{2})|^2 |\hat{\phi}(\frac{\omega}{2})|^2 \overline{(i\omega)^s e^{-in\omega}} d\omega \\ &= \frac{1}{2\pi} \int (i\omega)^r (|\hat{\phi}(\frac{\omega}{2})|^2 - |m_0(\frac{\omega}{2})\hat{\phi}(\frac{\omega}{2})|^2) \overline{(i\omega)^s e^{-in\omega}} d\omega \end{aligned}$$

利用變數變換以及 (3.61)，上述的最後一條式子可以寫成兩項：

$$\begin{aligned} &2^{r+s+1} \frac{1}{2\pi} \int (i\omega)^r |\hat{\phi}(\omega)|^2 \overline{(i\omega)^s e^{-i2n\omega}} d\omega \\ &- \frac{1}{2\pi} \int (i\omega)^r |\hat{\phi}(\omega)|^2 \overline{(i\omega)^s e^{-in\omega}} d\omega \end{aligned}$$

代入 $|\hat{\phi}(\omega)|^2 = \hat{\phi}(\omega) \overline{\hat{\phi}(\omega)}$ ，再參照 (3.48) 和 Parseval 等式，就應該可以看出來 (留作習題) 前面的式子等於

$$2^{r+s+1} \int \phi^{(r)}(x)\phi^{(s)}(x-2n) dx - \int \phi^{(r)}(x)\phi^{(s)}(x-n) dx$$

故得證性質 5.6。

最後，我們要證明 $\psi^2(x)$ 的質量中心在其函蓋的中央。因為 $\psi^2(x)$ 是個非負函數，我們可以將 ψ^2 想成一個質量分佈函數，因此 $\int x\psi^2 dx$ 就是它的質量中心。又因為 $\int \psi^2 dx = 1$ ，所以其實也可以將 ψ^2 想成一個機率分佈函數 (p.d.f.: probability distribution function)。而這個 idx:pdf

p.d.f. 的平均值 (mean value) 也是 $\int x\psi^2 dx$ 。提出這個看法的是美國 Duke 大學統計系的 Vidakovic 教授。他經過數值積分的實驗，發現似乎 $\psi^2(x)$ 的平均值，也就是質量中心，是在其函蓋的中央。我們證明了這個觀察。 idx:mean

此時我們是根據 (3.20) 來定義 $\psi(x)$ ，而不是 (3.19)。因此 $\text{supp } \psi$ 是 $[1-p, p]$ ，它的中央就是 $\frac{1}{2}$ 。

性質 5.8 若 $\psi(x)$ 是根據 (3.20) 定義的 Daubechies 正交凌波函數，則 prop:psime

$$\int x\psi^2(x) dx = \frac{1}{2}$$

對任意階數 $p \geq 1$ 都成立。

證明：利用 Parseval 等式 (2.43)， idx:Parseval

$$\int x\psi^2(x) dx = \langle x\psi, \psi \rangle = \frac{1}{2\pi} \langle \mathcal{F}[x\psi](\omega), \hat{\psi}(\omega) \rangle \quad (5.54) \quad \text{eq:xqb}$$

但是 $\mathcal{F}[x\psi](\omega) = i \frac{d}{d\omega} \hat{\psi}(\omega)$ 。令 $\hat{\psi}' = \frac{d}{d\omega} \hat{\psi}(\omega)$ ，我們考慮 $\langle \hat{\psi}', \hat{\psi} \rangle$ 。

根據 (3.76) $\hat{\psi}(\omega) = m_1(\frac{\omega}{2}) \hat{\phi}(\frac{\omega}{2})$

$$\langle \hat{\psi}', \hat{\psi} \rangle = \frac{1}{2} \int (m_1'(\frac{\omega}{2}) \hat{\phi}(\frac{\omega}{2}) + m_1(\frac{\omega}{2}) \hat{\phi}'(\frac{\omega}{2})) \overline{m_1(\frac{\omega}{2}) \hat{\phi}(\frac{\omega}{2})} d\omega$$

經過變數變換 $\omega \mapsto 2\omega$ 得到

$$\begin{aligned} & \langle \hat{\psi}', \hat{\psi} \rangle \\ &= \int m_1'(\omega) \overline{m_1(\omega)} |\hat{\phi}(\omega)|^2 d\omega + \int |m_1(\omega)|^2 \hat{\phi}'(\omega) \overline{\hat{\phi}(\omega)} d\omega \\ &= \int_0^{2\pi} m_1'(\omega) \overline{m_1(\omega)} \sum_k |\hat{\phi}(\omega + 2k\pi)|^2 d\omega \\ & \quad + \int |m_1(\omega)|^2 \hat{\phi}'(\omega) \overline{\hat{\phi}(\omega)} d\omega \\ &= \int_0^{2\pi} m_1'(\omega) \overline{m_1(\omega)} d\omega + \int |m_1(\omega)|^2 \hat{\phi}'(\omega) \overline{\hat{\phi}(\omega)} d\omega \end{aligned}$$

倒數第二條等式是經常使用的技巧，參閱 (3.62)。倒數第一條等式是因為 $\phi(x)$ 是正交自格函數，故符合 (3.64)。

將前面的 $\hat{\psi}$ 換成 $\hat{\phi}$ ，做完全一樣的推導，只是使用 (3.61) 關係，所以 $m_1(\omega)$ 全部換成 $m_0(\omega)$ ，則得到

$$\langle \hat{\phi}', \hat{\phi} \rangle = \int_0^{2\pi} m_0'(\omega) \overline{m_0(\omega)} d\omega + \int |m_0(\omega)|^2 \hat{\phi}'(\omega) \overline{\hat{\phi}(\omega)} d\omega \quad (5.55) \quad \text{eq: xqa}$$

由於 $|m_1(\omega)|^2 = 1 - |m_0(\omega)|^2$ ，故利用 (5.55) 得到

$$\langle \hat{\psi}', \hat{\psi} \rangle = \int_0^{2\pi} m_1'(\omega) \overline{m_1(\omega)} d\omega + \int_0^{2\pi} m_0'(\omega) \overline{m_0(\omega)} d\omega$$

現在，根據 (3.77) 展開 $m_1(\omega)$ 得到

$$\begin{aligned} & \langle \hat{\psi}', \hat{\psi} \rangle \\ &= \int_0^{2\pi} (-i) |m_0(\omega + \pi)|^2 d\omega + \int_0^{2\pi} \overline{m_0'(\omega + \pi)} m_0(\omega + \pi) d\omega \\ & \quad + \int_0^{2\pi} m_0'(\omega) \overline{m_0(\omega)} d\omega \\ &= -i \int_0^{2\pi} |m_0(\omega)|^2 d\omega + \int_0^{2\pi} \overline{m_0'(\omega)} m_0(\omega) d\omega \\ & \quad + \int_0^{2\pi} m_0'(\omega) \overline{m_0(\omega)} d\omega \\ &= -i \int_0^{2\pi} |m_0(\omega)|^2 d\omega + \int_0^{2\pi} \frac{d}{d\omega} [\overline{m_0(\omega)} m_0(\omega)] d\omega \\ &= -i \int_0^{2\pi} |m_0(\omega)|^2 d\omega \end{aligned}$$

在上面的第二個等式中，我們幾次利用了 $m_0(\omega)$ 是 2π -週期函數的事實。最後，我們根據 (3.60) 展開 $m_0(\omega)$ ，並因為 $\{e^{-ik\omega}\}$ 是 $L^2[0, 2\pi]$

空間內的正交函數，所以

$$\begin{aligned}\langle \hat{\psi}', \hat{\psi} \rangle &= -i \int_0^{2\pi} \left| \frac{1}{2} \sum_k c_k e^{-ik\omega} \right|^2 d\omega \\ &= -\frac{i}{4} \int_0^{2\pi} \sum_k c_k^2 d\omega \\ &= -\frac{i}{4} 4\pi \\ &= -i\pi\end{aligned}$$

上面的第三個等式是因為 (3.6) $\sum c_k^2 = 2$ 。代回 (5.54)，

$$\int x\psi^2(x) dx = \frac{i}{2\pi} \langle \hat{\psi}', \hat{\psi} \rangle = \frac{i}{2\pi} (-i\pi) = \frac{1}{2}$$

故得證。 □

Vidakovic 教授的數值實驗還顯示，當 $p \geq 2$ 時， $\psi^2(x)$ 的標準方差 (standard deviation) 呈線性關係。明確地說，若 σ_p 代表 p.d.f. $\psi^2(x)$ 的標準方差，則大約

$$\sigma_p \approx 0.074p + 0.202$$

可惜這在理論上似乎是不對的，只是恰巧在實驗上非常接近如此。

習題 5.17 證明 (5.51)。

習題 5.18 證明性質 5.7。

習題 5.19 證明

$$\frac{1}{2\pi} \int (i\omega)^r |\hat{\phi}(\omega)|^2 \overline{(i\omega)^s e^{-i2n\omega}} d\omega = \int \phi^{(r)}(x) \phi^{(s)}(x - 2n) dx$$

習題 5.20 證明當 $f(x)$ 與 $xf(x)$ 皆有傅立葉轉換時，

$$\mathcal{F}[xf(x)](\omega) = i \frac{d}{d\omega} \hat{f}(\omega)$$

注意，這個結果告訴我們，如果 $f(x)$ 與 $xf(x)$ 都是 L^2 函數，則 $\hat{f}(\omega)$ 必可微。而 $xf(x) \in L^2$ 基本上的意思是，當 $x \rightarrow \pm\infty$ 時， $|f(x)| \rightarrow 0$ 的速度夠快，使得 $\|xf(x)\|^2 = \int |x|^2 |f(x)|^2 dx < \infty$ 。可見 $|f(x)| \rightarrow 0$ 的速度和 $\hat{f}(\omega)$ 的可微程度是一體的兩面。當 $f(x) \in L^2$ 而且 $f(x)$ 是有限函蓋函數的時候， $xf(x) \in L^2$ ，所以 $\hat{f}(\omega)$ 必可微。由此可以類推，任何 L^2 中的有限函蓋函數，其傅立葉轉換是無限可微。

附 錄

計 算 工 具

這本書裡的所有程式範例都是以 Matlab 指令寫成。而製作本書的過程當中，只要牽涉到製圖或計算，也大多是由 Matlab 完成。只有很少數的例外，我們用了 FORTRAN 或 C 程式語言做四倍精度的計算 (Matlab 只提供雙精度的浮點計算)，或是用了 Maple 做符號計算。

Matlab 是非常容易使用的一個科學計算環境。只要熟悉矩陣運算的規則就會使用 Matlab。這個附錄的目的，並非提供 Matlab 小手冊。而是協助讀者瞭解出現在此書中的 Matlab 範例程式。

我個人向來不建議，爲了學習一個軟體而學習。我認爲學習軟體時一定要伴隨一個計畫，或是一個工作的目標，這樣才不空洞。否則很容易流於浪費時間。所以，如果您是首次接觸 Matlab，那也很好。這本書裡的範例，雖然只佔 Matlab 功能的一小部分，但是見微知著，讀者應該可以從這裡學習使用 Matlab 的要領。

市面上逐漸出現以 Matlab 爲輔助工具或標準語言的數值分析教科書，甚至矩陣計算或微分方程數值解的教科書。這些書也是順便學習 Matlab 的好途徑。例如黃俊銘先生編譯的數值方法 [5] 即爲一本實用的教科書。

A.1 Matlab 概要

使用 Matlab 的第一個心理建設就是：不要把它當 C 或 FORTRAN 的代用品。它具有程式語言所該有的邏輯分支指令 if-elseif-else，也有迭代 (iteration) 指令 for；它支援副程式的呼叫，甚至副程式可以呼叫自己，所以可以寫遞迴 (recursion) 程式。因此 Matlab 可以成爲一套完整的程式語言 (它沒有明確的資料形態 (data types))。儘管如此，程式語言並非 Matlab 的設計理念；我們使用 Matlab 的心態是把它當做一個數學運算的工具，而非程式設計的工具。相信讀者在這本書裡到處可以看到如是例子。

Matlab 在內部一律用雙精度浮點數儲存 (64 bits)，即使整數也是利用浮點數的底數部分 (mantissa) 儲存。在 Matlab 裡面，所有物件都是矩陣。當 Moler 教授在 1984 年設計 Matlab 的時候，還沒有所謂的物件導向程式語言。但是他已經具備這樣的觀念。常數 (或純量) 就是 1×1 的矩陣，向量是 $n \times 1$ 的矩陣，序列是 $1 \times n$ 的矩陣。字串就是一個由整數組成的序列，而每個整數被解讀爲字碼 (ASCII 或中國字的 Big-5 碼)。Matlab 內建複數計算，當計算過程中該要出現複數的時候，就自然會出現，不必事先宣告。例如 `sqrt(-4)` 得到 $2i$ 。

由於所有的物件都是矩陣，故基本運算 $+$ 、 $-$ 、 $*$ 都須按照矩陣維度的規則。必須相同維度的矩陣 (要知道，廣義的矩陣包括向量、序列、和常數) 才能相加減，必須維度相銜接的矩陣才能相乘 (維度 $m \times n$ 的矩陣只能和維度 $n \times k$ 的矩陣相乘，結果是一個 $m \times k$ 的矩陣)。除了 1×1 矩陣是個例外，這是爲了習慣上的方便。例如

```
x=[1 2 3];
```

就設立一個序列 `x`，後面的 `;` 表示 Matlab 不回應，因此看不到螢幕上的輸出。Matlab 的矩陣編號一律從 1 開始 (這是我個人覺得它最不方便

便的地方)。所以 x 的元素是 $x(1)$ 、 $x(2)$ 和 $x(3)$ 。例如

```
A=[1 2 3; 4 5 6]
```

就設立一個 2×3 的矩陣，其中 $A(1,1)=1$ 、 $A(2,3)=6$ 。因為指令沒有 `idx:matlab` 用；結束，所以 Matlab 回應結果，您可以在螢幕上看到一個矩陣

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \quad (\text{A.1})$$

(但是在螢幕上沒有大括號。) $A+x$ 不能計算，因為維度不同。 $A*x$ 也不能，因為維度不銜接。但是

```
A*x'
```

就可以計算。因為 x' 就是 x 的轉置 (transpose)，所以 x' 是 3×1 的 `idx:matlab` 向量，故 $A*x'$ 就是 2×1 的向量。如果加數或乘數中有常數，則代表將常數加或乘入矩陣的每一項。例如

$$x + 1 = (2 \ 3 \ 4), \quad x * 2 = (2 \ 4 \ 6)$$

矩陣運算中本來沒有除法。如果 $Ax = b$ 而且 A 是可逆矩陣，則 $x = A^{-1}b$ 。因此 Matlab 定義左除法 $A \setminus b = I$ 。所以 $x = A \setminus b$ 。同理還有右除法，但是 b/A 不能算。因為右除法的意思是 $A/A = I$ ，所以 b/A 是 Ax/A 的意思，這是沒有意義的。但是 $b^T = x^T A^T$ ，所以 b'/A' 就可以算，而答案是 x^T 。Matlab 的除法計算不是很單純地先算逆矩陣再乘向量。除法的演算法比求逆矩陣更快速而且更穩定。比如說當 A 不是可逆矩陣的時候，Matlab 仍然能算除法，而除出來的結果是最靠近解。我們不再細說了，因為這本書的內容並不需要知道這些細節。有意知道更多的讀者可以參考 [5]。 `idx:matlab` `idx:matlab`

事實上，幾乎所有的矩陣計算過程中，我們都不會真的需要逆矩陣。求逆矩陣的動作常常是既不必要又不穩定的*。所以我故意不寫出來求逆矩陣的指令。

如果要設立一個複數，比如 $1 + 2i$ ，指令是 $1+2*i$ 。但是要小心，當矩陣 A 的任意一個元素是複數時，整個矩陣就被視為複數矩陣。而複數矩陣的 A' 是 Hermitian 的意思。亦即，A' 將每個元素取共軛複數，再轉置。例如

idx:matlab

```
[1+2*i 2; 3 4]'
```

的結果是

$$\begin{pmatrix} 1 - 2i & 3 \\ 2 & 4 \end{pmatrix}$$

注意 $1+2*i$ 的算式中不可有空格，因為 Matlab 以空格分隔元素。

前面我們看到如何設立一個矩陣。基本上我們可以逐項寫出來，例如 $[1 \ 2 \ 3]$ 或 $[1;2;3]$ 或 $[12;34]$ 。但是我們也有某些製造矩陣的快速方法。例如 $\text{eye}(n)$ 設立 $n \times n$ 的單位矩陣（試試看 $\text{eye}(2,3)$ 出來什麼？）， $\text{ones}(m,n)$ 設立 $m \times n$ 每項都是 1 的矩陣， $\text{zeros}(m,n)$ 設立 $m \times n$ 每項都是 0 的矩陣。當我們逐項定義矩陣元素的時候，一行可能寫不下。用 ... (連續三個點) 表示繼續到下一行。

idx:matlab

idx:matlab

指令 $n:m$ 設立一個長度為 $m - n + 1$ 的序列，其元素為從 n 到 m 的整數；如果 $n > m$ ，則出現一個空矩陣 $[]$ (0×0 矩陣)。空矩陣是個很有趣的東西，有時候會有妙用的。 $1:2:n$ 則列出 1 到 n 之間的所有奇數。注意 Matlab 的計算比較聰明，例如

idx:matlab

idx:matlab

* 每當我授課講到這裡的時候，都對聽眾說：「如果你在計算過程中，發現你要算逆矩陣。停下來，想想有沒有辦法避免；如果想了一下覺得還是需要 X 請停下來，再想一想。」

```

h = 1/3;
x(1)=0;
for i=1:3, x(i+1)=x(i)+h; end;

```

一般的程式語言，因為浮點數運算誤差的關係，所得的 $x(4)$ 經常不會是 1 而是 $0.999\dots$ 。但是 Matlab 可以得到 $x(4)=1$ 。在這裡例子裡面，我們順便展示了如何使用 Matlab 的迭代指令 for。同理，在 $[-\pi, \pi]$ 中取 101 個均勻分隔的節點，指令是 $-\pi:(2*\pi/100):\pi$ 。其中 pi 是內建常數，就是圓周率 π 。

size(A) 回應 A 的維度，而 length(A) 回應兩個維度數字中比較大的那個。所以如果 A 是個向量或序列，則 length(A) 就是它的長度。 idx:matlab

要將矩陣變換成向量或序列是簡單的。比如說 A 是 (A.1) 中的 2×3 矩陣。這個矩陣共有 6 個元素，如果將 A 的足標轉成序列，例如說 A(1:6)，就是要列出 A(1)、A(2)、 \dots 、A(6) 的值。得到

1 4 2 5 3 6

可見 Matlab 的內部以行導向 (column oriented) 儲存矩陣。如果要以列導向將矩陣轉成序列，可以說 $B=A'$; B(1:6)。 idx:matlab

但是若要將一個長度為 mn 的向量或序列轉成 $m \times n$ 的矩陣，就比較麻煩了。我個人只有一個笨辦法，就是將序列一段一段地切下來，再依序排在矩陣上。下面是示範的程序。其中 u 是長度 mn 的序列，我們將它以列導向排成一個 $m \times n$ 的矩陣 A。

```

A = [];
for i=1:m, A=[A; u((i-1)*n+1:i*n)]; end;

```

看，我們用到空矩陣！例如 $m = 3$ ， $n = 4$ ， $u = 1:12$ ，則 A 是

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix}$$

黃華民教授提供了兩個 Matlab 副程式，專門做矩陣與序列之間的轉換。我將它們附在磁片裡：

row2mat.m	mat2row.m
-----------	-----------

如果 u 是長度 n 的向量或序列，則 $\text{diag}(u)$ 產生一個 $n \times n$ 的方陣，其對角線上是 u ，其他元素都是零。如果說 $\text{diag}(u, k)$ ，其中 k 是個整數，則產生一個 $(n + |k|) \times (n + |k|)$ 的方陣，其第 k 次對角線是 u 其他元素都是零。所謂第 k 次對角線是從對角線向右上角（當 $k > 0$ ）或左下角（當 $k < 0$ ）算的第 $|k|$ 條對角線。例如 idx:matlab

```
diag(2*ones(1,4)) + diag(3*ones(1,3),1) + diag(ones(1,3),-1)
```

得到矩陣

$$\begin{pmatrix} 2 & 3 & 0 & 0 \\ 1 & 2 & 3 & 0 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

反過來，如果 A 是一個矩陣，則 $\text{diag}(A)$ 取出它的對角線向量，而 $\text{diag}(A, k)$ 取出它的第 k 次對角線向量。

符號 \wedge 表示乘方。但是只有方陣（包括常數）是可以做乘方運算的。至於 n 次方根，除了開平方根用 `sqrt` 之外，其他的整數方根都要用乘方符號，例如 $8^{(1/3)}$ 就是開 8 的 3 次方根，答案是 2。有兩 idx:matlab

點要提醒讀者注意。第一，因為在 Matlab 中 a^b 其實是用指數與對數計算的，也就是 $\exp(b \cdot \log(a))$ 。所以當 a 是負數的時候，會出現複數答案。例如我們知道 -8 的實數 3 次方根應該是 -2 ，但是當您說 $(-8)^{(1/3)}$ 卻得到答案 $1 + 1.7320i$ 。這個答案並沒有錯，它是

$$\sqrt[3]{8} e^{i\frac{\pi}{3}}$$

其實 $(-8)^{(1/3)}$ 相當於求解多項式 $x^3 + 8 = 0$ 的根。我們知道此三階多項式必有三個複數根，Matlab 告訴我們的是基本根 (primitive root, 它是幅角最小的根)。一般而言，若 $a > 0$ 則 $(-a)^{\frac{1}{n}}$ 是 $x^n + a = 0$ 的基本根：

$$\sqrt[n]{a} e^{i\frac{\pi}{n}}$$

如果要得到 $x^3 + 8 = 0$ 的所有根，下指令 `roots([1 0 0 -8])`。

第二，若 A 是一個方陣，則 `sqrt(A)` 是將 A 中的每個元素開平方根的意思，而不是求 A 的平方根矩陣。^{*} 所謂平方根矩陣是一個矩陣 M 使得 $M^2 = A$ 。求矩陣平方根的指令是 `sqrtm`。例如 $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ ，則 idx:matlab

$$\text{sqrt}(A) = \begin{pmatrix} 1.0000 & 1.4142 \\ 1.7321 & 2.0000 \end{pmatrix}$$

$$\text{sqrtm}(A) = \begin{pmatrix} 0.5537 + 0.4644i & 0.8070 - 0.2124i \\ 1.2104 - 0.3186i & 1.7641 + 0.1458i \end{pmatrix}$$

幾乎所有 Matlab 的運算指令都像 `sqrt` 一樣，如果輸入是一個矩陣，則將指令作用在矩陣的每一個元素上。而運算符號也可以逐項計算，只要在符號的前面加一個點即可。例如 $A.^2$ 是將 A 中的元素逐

^{*} 如果 A 的特徵值都是正數，則 A 具有一個實數的平方根矩陣，否則 A 的平方根矩陣是複數矩陣。這個課題與本書的內容無關。

項做平方； $u .* v$ 是將 u 與 v 的對應項元素兩兩相乘，所以 u 與 v 維度必須相同。例如求向量 x 中兩兩相鄰元素的比例，可以下指令 $x(1:n-1) ./ x(2:n)$ ，其中 n 是 x 的長度。再例如 u 與 v 是長度相同的序列，則 $\text{sum}(u .* v)$ 就是它們的內積。當然也可以用矩陣乘法來算，但是維度要銜接對，就像

$$(1 \ 2) \begin{pmatrix} 3 \\ 4 \end{pmatrix} = 11, \quad \begin{pmatrix} 1 \\ 2 \end{pmatrix} (3 \ 4) = \begin{pmatrix} 3 & 4 \\ 6 & 8 \end{pmatrix}$$

邏輯運算指令都設計成逐項計算。例如 $A > 1$ 得到一個與 A 同樣維度的矩陣，當 A 的元素 > 1 的時候，對應答案 1，否則對應答案 0。舉例來說， $\text{rand}(5,5)$ ；得到一個 5×5 的亂數矩陣，其中的每個元素都介於 $[0, 1]$ 之間。如果我們想知道這個矩陣中共有多少個 > 0.5 的元素，可以這樣做：

```
sum(sum(rand(5,5) > 0.5))
```

如果 A 是 $m \times n$ 矩陣，則 $\text{sum}(A)$ 就是各行的和所成的序列。例如當 A 是 (A.1) 的矩陣，則 $\text{sum}(A)$ 得到 (5, 7, 9)。而如果 x 是一個向量或序列，則 $\text{sum}(x)$ 就是各元素的和。例如 $\text{sum}(\text{sum}(A))$ 得到 21。由於 $\text{rand}(5,5)$ 是亂數矩陣，所以其中出現 > 0.5 的機率應該大約是 $\frac{1}{2}$ 。讀者可以執行上面那句指令若干次，看看是否符合機率的描述。

我們開宗明義地說，不要把 Matlab 當程序導向的程式語言來用，至少要把它當物件導向的程式語言。技術之一就是，我們要盡量避免使用迭代 `for` 指令，而盡量使用以矩陣為物件的計算指令。舉一個例子。下面這段程序，在 $[0, 1]$ 取 10001 個均勻分隔節點，然後一一算它們的 \sin 值。

```
h=10^(-4);
```

```
x=0:h:1;
tic; for i=1:10001, y(i)=sin(x(i));end; toc
```

其中的 tic 和 toc 就是時鐘滴答滴答走的滴和答。tic 的時候把馬錶歸零開始記時，toc 的時候讀取馬錶的時間。當我做實驗的時候，一共花了 7.12 秒。注意，這是所謂的真實時間，不是 CPU 時間。所以，除了不同的機器會花費不同的時間之外，同樣機器在多工狀態下，也因為系統的負荷而有不同的表現。但是重點不在上面的迭代指令花了多少時間，重點是要與下面的物件導向指令做比較。

idx:matlab

```
h=10(-4);
x=0:h:1;
tic; u=sin(x); toc
```

我做完前面的實驗後立刻做這個實驗，一共花了 0.0127 秒。比使用迭代的算法快了 560 倍。

最後，我們列出 row2mat.m，並藉此示範 Matlab 副程式的寫法。為了大家的方便，我們最好在一個檔案中只儲存一個副程式，而檔案名與副程式名相同（副檔名必須是 .m）。首先我們看到，檔案的一開頭就是一連串的說明文字（comment）。Matlab 的說明文字寫在 % 的後面。% 可以出現在任何位置，凡是在 % 之右的同行文字，都被視為說明文字，所以 Matlab 不會去處理它。

code:rowtor

```
———— Matlab Code-21 —————
% Convert from a vector or an array to a matrix.
% Provided to me by Prof Huang Hua-Min on July 29, 1998.
%
% Input: s is an array of length m*n
```

```

% Output: A matrix of dimension m by n, rows from s
%
% A = row2mat(s,m,n)
function A=row2mat(s,m,n)
N=length(s);
% Verify the input
if ( N ~= m*n )
    disp('Wrong size')
    break
end
[x,y]=meshgrid(1:m,1:n);
A=full(sparse(x,y,s));

```

Matlab 副程式中從第一行開始的連續說明文字 (連續以 % 開頭的文字)，還有特別的用途。如果使用者在 Matlab 環境中下指令 `help row2mat` 則可以看到這串說明文字。所以，這段說明文字是開放的說明文字。在開頭的連續說明文字之後的任何說明文字，就不會被 `help` 拿出來給使用者看。例如 `% Verify the input` 這一行就不會出現在 `help` 的輸出。

第一行 `function A=row2mat(s,m,n)` 是副程式的宣告。這個副程式輸入三個參數，在副程式內參數值的任何改變都是區域性的 (術語叫做 `pass by value`)。副程式的名字是 `row2mat`，它輸出一個物件 `A`。

在 Code-21 中，我們順便示範了 `if` 和 `break` 的用法。`meshgrid` 是 Matlab 內建的副程式。注意它輸出兩個物件。在這本書的其他地方，都不需要使用 `meshgrid`、`full` 和 `sparse`，所以我們就不多做介紹了。讀者可以自行用 `help` 或 `doc` 指令學習它們的用法。

有關 Matlab 的普遍性介紹，就到此為止。以下兩節，我們針對製圖和顯像的指令多做介紹。順便還會引出一些其他的 Matlab 用法。如果要更全面地學習 Matlab 軟體，可以參考歐崇明的專書 [8]。

A.2 Matlab 的製圖功能 plot

在這一節中，我們介紹 Matlab 的製圖指令。這本書內的幾乎所有函數曲線圖形，都是用這個指令製作的：`plot`。我們先說明 `plot` 的工作原理，然後交代一些使用上的經驗。最後，我們解釋一個出現在第一章的例子。

當我們有一個函數 $f(x)$ ， $x \in [a, b]$ ，怎樣製作 $y = f(x)$ 在 $[a, b]$ 區間中的函數圖形呢？在中學的時候，我們學會在 x - y 坐標平面上定出幾個點，然後就憑經驗地畫出函數曲線。基本上這是個正確的方法。有了電腦的幫助，我們可以在 x - y 坐標平面上定出非常多的點，因此描繪的曲線就比較準確。

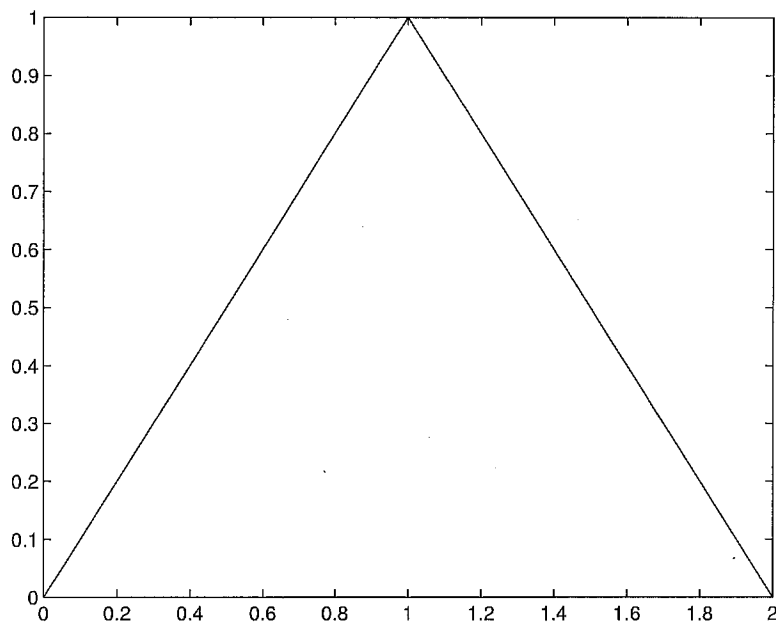
此處我們以 Matlab 的 `plot` 指令為例。令 $\mathbf{x} = (x_k)$ 和 $\mathbf{y} = (y_k)$ 是兩個長度同樣為 n 的向量。所謂 `plot(x,y)` 就是在 x - y 坐標平面上定出 n 個點 (x_1, y_1) ， (x_2, y_2) ， \dots ， (x_n, y_n) ，然後將這些點按照順序用直線段連起來。例如若 idx:plot

$$\mathbf{x} = (0, 1, 2) \quad \mathbf{y} = (0, 1, 0)$$

則 `plot(x,y)` 會依序將 $(0,0)$ ， $(1,1)$ 和 $(2,0)$ 用線段連起來，構成圖 72 也就是二階基本樣條函數 $B_2(x)$ 的圖 (參照 (2.53))。 idx:B2

注意標定點的順序是很重要的，否則畫出來的圖形就不會是函數圖形。通常我們都是先定一個固定間隔

$$h = \frac{b - a}{n - 1}$$

fig:Btwo 圖 72 $B_2(x)$

取節點 $x_k = a + (k-1)h$ (Matlab 指令是 `x=a:h:b`) 和函數插值 $y_k = f(x_k)$ 。如此一來，順序就一定正確而且 plot 的結果就大約是 $y = f(x)$ 的函數圖形。

如果標定的點數不多 (n 不夠大)，那麼看起來就是折線圖。如果標定的點數夠多，那些折線都很短，眼睛看起來就像是一條平滑曲線了。點數要多少才夠呢？這和電腦輸出設備的解析度有關。對人的眼睛而言 (如果沒有使用放大鏡或顯微鏡) 每一毫米 (1mm) 標定一個點就夠密了。比如說您要將圖形印在紙上，印出來的寬度是 15 公分 (150mm)，那麼只要標定 150 個點就夠了；除非您的函數圖形有急劇的振盪。太多的點對機器而言一點兒也沒有好處。比如說您的雷射印刷機的解析度是 600dpi，意思是它每英吋只能打出 600 個點。普通一

張 A4 紙的列印寬度是 6 英吋。所以至多只能打出 3600 個點。因此在製圖的時候，標定多於 3600 個點是沒有用的。

如果在 Matlab 的 plot 中只給一個向量，例如 plot(x)，則 Matlab 會自動以 $1, 2, \dots, n$ 作為橫軸坐標。所以畫出來的向量圖形就是以 (k, x_k) ， $k = 1, 2, \dots, n$ 為標定點的折線圖。

當我們說繪製一個函數圖形，或是說繪製一個向量圖形的時候，指的都是以 Matlab 的 plot 方式製圖。因此就不再分別贅述細節。

最後，我們以圖 2 中 $f(x) = -\frac{16}{9}x^2 + \frac{8}{3}x$ 與 $f_5(x)$ 的圖為例，解釋製作那張圖的 Matlab 指令。為了方便讀者，我們將那張圖在次頁再畫一次。

```
clear;
j=5;
hold on
x=0:0.01:1;
f=-(16/9)*x.^2+(8/3)*x;
plot(x,f,':');
x=0:1/2^j:1;
xk=(0:length(x)-2);
n=length(xk);
xk1=(1:length(x)-1);
v=2^j*(4/27*xk.^2.*(4*xk-(9*2^j)*ones(1,n))/(2^j)^3 ...
-4/27*xk1.^2.*(4*xk+(4-9*2^j)*ones(1,n))/(2^j)^3);
for i=1:2^j,
    plot(x(i:i+1),[v(i) v(i)])
end;
```

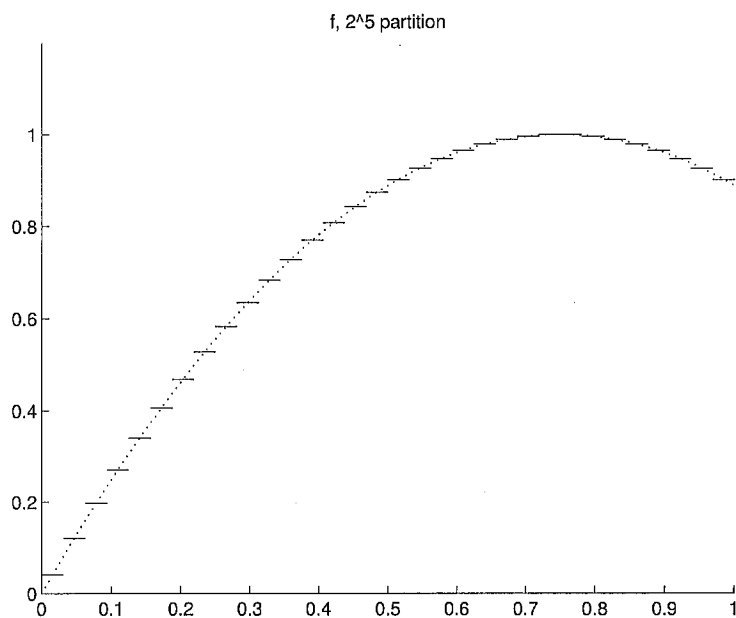


fig:tmpplot 圖 73 示範圖片

```
title('f, 25 partition')
axis([0 1 0 1.2])
hold off
```

首先我們用 `clear` 將 Matlab 記憶中的矩陣都清除。這樣做的原因是避免剛才定義了同樣名字的矩陣，而擾亂我們現在的計算。在 `hold on` 和 `hold off` 之間，我們可以使用 `plot` 指令製圖若干次，而每次作的圖都重複畫在同一個畫面上。如果沒有 `hold on`，則每次使用 `plot` 會清除上一次畫的圖。如果在畫任何圖之前就說 `hold on`，就不會出現外框，參照圖 2。如果先畫第一張圖，然後說 `hold on`，接著再畫其他的圖，那就會出現外框（這本書中所有其他的圖都是有外框的）。其實 `hold` 是開關類指令。也就是說，當它第一次被呼叫的時候，表示 `hold`

on，第二次被呼叫的時候，代表 hold off。

接著，`x=0:0.01:1;` 表示在 $[0, 1]$ 間每隔 0.01 設立一個節點。下一行就是計算函數在節點的值 $f(x_k)$ 。注意我們用到逐點平方運算 `.^2`。然後我們畫 $f(x)$ 的圖，用虛線。 $f(x)$ 是個平滑曲線，我們用 101 個點來畫它的折線圖。因為點數夠多，所以看起來夠平滑。

從 `x=0:1/2^j:1;` 開始，我們準備製作 $f_5(x)$ 的分片常數圖形。 $f_5(x)$ 在 $[0, 1]$ 內分成 2^5 片常數，所以我們先以 $\frac{1}{32}$ 定義節點。`xk`、`xk1` 和 `v` 那四行在計算 $f(x)$ 在每一個區間內的積分平均值。我們先用 Maple 算出來這些積分平均值的公式，然後再將公式剪貼到 Matlab 裡面。注意我們用了繼續符號 `...`。別理細節，總之，依第一章中的符號，

idx:matlab

$$v(k) = v_{j,k-1} = 2^j \int_{\frac{k-1}{2^j}}^{\frac{k}{2^j}} f(x) dx$$

我們解釋過，Matlab 的序列編號一律從 1 開始，我們的程式不能完全按照課文中的足標編號方式（從 0 開始）。

等到橫軸的節點向量 (`x`) 和縱軸的函數值 (`v`) 都準備好了，接著就可以畫圖。因為必須畫出來不連續的分片常數圖，所以要分別使用 `plot` 32 遍。這就是 `for i=1:2^j`，那三行所做的事。因為 `plot` 指令並不會在 Matlab 工作視窗內做任何回應，所以不寫 `;` 也無所謂。

最後，我們用 `title` 在圖片上方寫一行簡短的標題。在書本中，圖片的說明都寫在文章內容裡。所以，這一行隨著圖片本身的簡短標題，通常只是為了作者以後識別的方便。然後我們設定圖片的坐標範圍。Matlab 會自動給一個範圍，但我們可以用指令 `axis(s)` 來自定範圍。其中 `s` 必須是一個長度為 4 的向量。橫軸的範圍從 `s(1)` 到 `s(2)` 縱軸的範圍從 `s(3)` 到 `s(4)`。

idx:matlab

A.3 Matlab 的顯像功能 image

在這一節中，我們介紹 Matlab 的顯像指令。這本書內的所有影像，都是用這個指令製作的。基本上只需要一個指令：`image`。我們先說明 `image` 的工作原理，然後交代一些使用上的經驗，最後解釋一個出現在第一章例子。

一張數位影像，其實是由許多有顏色的點所組成。這些點我們稱為像素 (pixel)。這些像素聚在一起，被我們看到，在腦內產生印象，而被誤以為是某種物體的影像。好比下面這幅影像



我們的經驗反射，認為是一位女士的影像 (我們還替它取了名字，叫做 Tiffany)。其實它是 256×256 個像素，印在大約 $3 \times 2\frac{3}{4}$ 英吋的紙上。每一個像素大約被 7 個雷射印表機的碳粉所實現。

我們對於一張數位影像，所要知道的就是像素的相關位置和它們所代表的顏色。由於影像通常總是矩形，所以我們假設這些像素排列成矩陣。而它們的相關位置，就由矩陣的足標決定。至於顏色，我們

用一個色譜來記錄顏色，然後將這些顏色編號，那麼矩陣中的像素只要記錄顏色的編號就可以了。所以，一張數位影像的數學模型，就是一個整數矩陣外加一張色譜。

而顏色要怎麼記錄呢？我們在中學時候就學過所謂的三原色光：紅 (R) 綠 (G) 藍 (B)。將這三種光以各種強度投射在一起，就在人類的眼睛中呈現各種不同的顏色。我們想像自然界中三原色光的強度是連續變化的，也就是說有無限多種不同顏色的可能。但我們不知道是否人類眼睛能夠分辨無限多種顏色。在電子數位影像的領域裡，目前定義的不同顏色大約是一千六百萬色。這是將每個原色光的強度定於 0 到 255 之間，0 是最暗 255 是最亮。每個原色有 $256 = 2^8$ 種不同的強度，所以總共有 $(2^8)^3 = 2^{24}$ 種不同可能的顏色。理論上只要定義了三種原色光的強度，就決定了顏色。實際上沒有這麼單純。因為螢幕產生顏色的原理是發光，但是在紙或布上，顏色是由顏料造成的，而顏料產生顏色的原理不是發光，而是吸光。例如一種會吸收紅光與藍光的顏料，就呈現綠色。國際間有幾種標準，企圖使得一張彩色數位影像，在不同的媒體上呈現一樣的色彩。這是一件困難的工作。

一張可能包含了所有顏色的數位影像，稱為全彩影像。全彩影像不必對應色譜，只要每個像素能夠記錄三個介於 0 和 255 之間的整數就可以了。這意味著，每個像素必須要有三個字元 (byte)。所以一張具有 256×256 個像素的全彩影像，就需要使用 196608 bytes (192KB) 來儲存。但是大部分的影像不需要這麼多不同的色彩，人類的眼睛和輸出影像的裝置都未必能分辨這麼多不同的色彩。所以，有許多實用的影像不是全彩影像。比如說常見的 256 色影像，整張影像只有 256 種不同的顏色。每個顏色有個定義和編號；所謂定義就是它的 RGB 強度，所謂編號就是它在色譜上的位置。因為只有 256 個顏色，所以只需要一個字元就可以記錄顏色。因此每個像素只需要一個字元，而色

譜上的每個顏色需要三個字元來定義顏色 (RGB 各一字元)。一張具有 256×256 個像素的 256 色影像，就需要使用 66304 bytes ($64\frac{3}{4}$ KB) 來儲存；比全彩影像幾乎節省了三分之一的資料量。節省資料量不只是爲了儲存的效率，更是爲了傳輸的效率。

比如 Tiffany 那張影像，它是一張 256 階的灰階影像。當 RGB 三色的強度相等時，呈現的就是灰色。不同的強度呈現不同深淺的灰。所謂 256 階的灰階，就是將灰色等分成 256 種不同深淺的灰。讓我們將色譜也用矩陣表達。一份記錄 n 種顏色的色譜可以寫成 $n \times 3$ 的矩陣。例如 256 階的灰階色譜就是

$$\begin{pmatrix} 0 & 0 & 0 \\ \frac{1}{255} & \frac{1}{255} & \frac{1}{255} \\ \frac{2}{255} & \frac{2}{255} & \frac{2}{255} \\ \vdots & \vdots & \vdots \\ 1 & 1 & 1 \end{pmatrix}$$

矩陣中的每一列定義一種顏色，而列號就是顏色的編號。

在 Matlab 中，一張數位影像就是一個像素矩陣配上一個色譜矩陣。例如前面的 Tiffany 影像，儲存在

Tiffany.mat

裡面。像素矩陣叫做 `Origin`，色譜矩陣叫做 `g256`。顯像的指令如下。

```
image(Origin);
colormap(g256);
```

她的右眼影像，大約記錄在矩陣的 Origin(100:120, 115:145) 之中，讀者可以觀察那一小塊矩陣的影像：

```
image(Origin(100:120, 115:145))
```

由於這塊矩陣只有 21×32 個像素，所以影像會非常模糊，您可以將顯像的視窗縮小，影像會相對變得清楚。

有些版本的 Matlab 在像素矩陣不正常的時候無法容錯，以至於顯像錯誤。所謂的不正常可以分成兩種狀況：像素不是整數，像素的數值超出色譜的範圍。第一種不正常很容易解決。如果像素矩陣是 A，則說 `round(A)` 就是將 A 中的元素取四捨五入的近似整數。另外還有兩種取整數的方法，就是 `floor` 和 `ceil`。`floor` 取下界整數，就是無條件捨去；`ceil` 取上界整數，就是無條件進入。

idx:matlab

要解決第二種不正常就比較複雜了。讓我們假設以飽和性 (*saturation*) 計算來處理 A。意思是，把所有小於下界 `lowbd` 的數字都變成 `lowbd`，把所有大於上界 `highbd` 的數字都變成 `highbd`。

code:satur

Matlab Code-22

```
A = A .* (A >= lowbd) + lowbd * (A < lowbd);
A = A .* (A <= highbd) + highbd * (A > highbd);
```

我們只解釋第一行指令的意思。`A >= lowbd` 產生一個和 A 維度相同的矩陣，對應 A 的元素 $\geq \text{lowbd}$ 的元素是 1，否則是 0。所以逐項與 A 相乘的結果，就是 A 中 $\geq \text{lowbd}$ 的元素都被保留下來，而 $< \text{lowbd}$ 的元素全部變成 0。另一部分，`lowbd*(A < lowbd)` 產生一個和 A 維度相同的矩陣，對應 A 的元素 $< \text{lowbd}$ 的元素是 `lowbd`，其他的是 0。所以，與前面的矩陣相加之後，就保留了原來 $\geq \text{lowbd}$ 的元素，而將 $< \text{lowbd}$ 的元素全部變成 `lowbd`。

比較新版本的 Matlab 應該沒有上述問題。但是在我自己的老機器上面，是有這些問題的。

最後，我們以圖 9 中的測驗影像為例，解釋製作那些影像的 Matlab 指令。為了方便讀者，我們將做過一次凌波分解的影像，也就是圖 9 左下角的影像，在次頁再畫一次。首先，我們解釋那張測驗影像的定義。

```
A=5*ones(128,128);
A(34:97,34:97)=60*ones(64,64);
map=gray;
map=map(64:-1:1, :);
image(A)
colormap(map);
```

這張圖有 128×128 個像素。我們先定義一個全是 5 的 128×128 方陣，然後將其中一塊 64×64 的部分定成 60。這張圖引用了 Matlab 內建的色譜之一：64 階灰階色譜 gray。但是原色譜定義黑色為 1 號，白色為 64 號。我們則希望數字越大顏色越暗，數字越小顏色越淺，所以我們要把色譜編號反過來。這就是 `map(64:-1:1, :)` 的用意。因此整張圖的周圍幾乎是白色 (5 號灰階)，而靠近中間的 64×64 部分幾乎是黑色 (60 號灰階)。

現在我們看看，做一次 Haar 凌波分解的程序，和顯影的程序。

```
clear B v u w
N=length(A);
B = A;
for i=1:N,
```

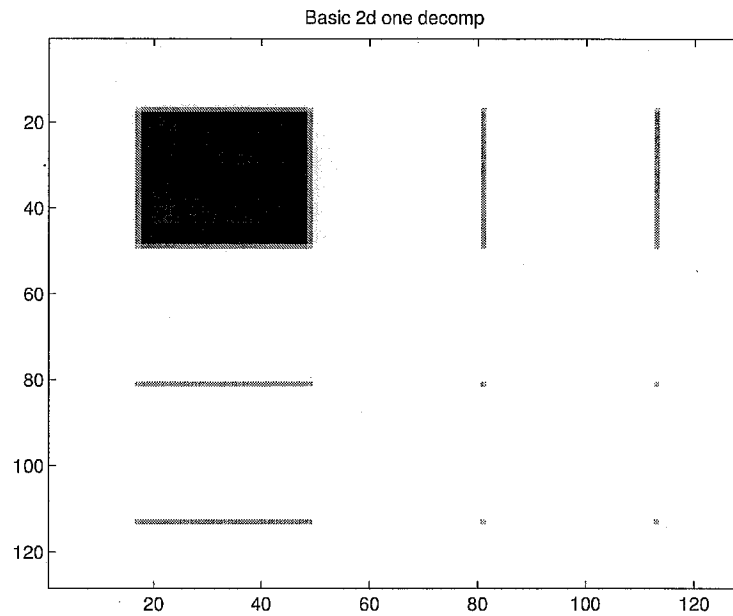



fig:tmpimage 圖 74 示範影像

```

v = B(:,i);
u(:,i) = (v(1:2:N) + v(2:2:N))/2.;
w(:,i) = (v(1:2:N) - v(2:2:N))/2.;
end;
B = [u;w]';
for i=1:N,
    v = B(:,i);
    u(:,i) = (v(1:2:N) + v(2:2:N))/2;
    w(:,i) = (v(1:2:N) - v(2:2:N))/2;
end;
B = [u;w]';

```

```

B1=round(B(1:N/2, 1:N/2));
B = abs(round(B));
B = B + (B==0);
B(1:N/2, 1:N/2) = B1;
image(B)
colormap(map)

```

首先，我們清除 Matlab 記憶中的變數 B, v, u, w ，這是為了避免剛才定義過的同名變數對此時的計算產生困擾。因為 A 是方陣，所以 $\text{length}(A)$ 就是它的維度。然後我們將 A 抄一份到 B ，這樣才不會破壞原來的像素矩陣 A 。接下來的第一個 for 迴圈，將 B 的每一個行向量做 Haar 凌波分解，並將結果儲存在 u 和 w 的行向量中。迴圈結束後， u 和 w 分別是 $\frac{N}{2} \times N$ 的矩陣。然後我們將 u 和 w 上下堆成一個 $N \times N$ 的方陣。至此做完了半個分解過程。接著要對每一個列向量做分解。我們將剛才堆成的矩陣轉置之後存回 B ，然後再對這個新的 B 做一次行向量的凌波分解。結束後再轉置回來，就是原來矩陣的一次凌波分解。儲存在 B 裡面。

接著就是準備顯影的工作。經過了分解的過程，可能出現非整數的像素，所以我們將矩陣四捨五入 (round)。因為我們要將低頻部分 ($B(1:N/2, 1:N/2)$) 保持原樣，卻要將高頻部分取絕對值並作飽和性處理。所以先把低頻部分保留在 $B1$ 。我們知道高頻不會有超過 64 的元素 (因為高頻是取差異，所以不會超過原來的最大值，也就是 60)。所以不必做上界飽和。而下界是 1，所以取了絕對值後低於下界的元素只可能是 0。因此我們只要把原來是 0 的部分變成 1 就好了。這就是 $B+(B==0)$ 的用意。將高頻部分處理好之後，再把低頻部分補進去，這就是 $B(1:N/2, 1:N/2)=B1$ 的用意。最後，就可以顯影了。

後記

1995 年春季我在中央大學數學系的大學部，首度嘗試開凌波函數這門課。由於沒有合適的課本，就自己編寫講義。後來我在網路上公開這批講義，得到許多工學院研究生的回響。因為感到國內的讀者有所需求，在黃俊銘先生的鼓勵之下，決定擴充這批講義，寫一本更完整的書。當初以為只要加枝添葉而已，不料最後竟然幾乎完全重寫，而我也因此渡過四個月毫無生活的日子。

我在寫作的過程當中，製造了大量的圖例，以期生動地呈現數學性質或凌波函數的特色。製作圖例所用到的圖檔，共有 167 個，佔地 5.2MB。絕大部分以 Matlab 製作，小部分以 xfig 或 Corel Draw 製作。它們都是 PS 或 EPS 格式的圖檔，應該可以很方便地放大、縮小或轉換成其他圖形格式。如果有老師要以這本書當做教材，而且希望在課堂中更清楚地呈現這些圖例，我可以提供所有的圖檔，供您製作投影片，或作其他教學用途。

我深信一本書不能太厚。太厚的書徒增讀者的負擔：比較厚的書比較貴、比較重、比較佔空間、比較讀不完。所以從一開始，我就預定這本書不超過三百頁。後來無可奈何地到達了四百頁，實在要向讀者致歉。儘管如此，這本書仍然遺漏了許多課題。例如非正交凌波函數、有限區間內的凌波函數、和它們在微分方程與積分方程數值解方面的應用範例。其實這些是比較專門的課題，本來就不該寫在初步裡面。這本書的使命之一，就是希望讀者在具備了初步的認識之後，可以進入各領域的專門課題。

在這個網路通訊時代，我希望這是一本會成長的書。它不應受限於

紙張的範圍，也不必從印刷出版的那一刻開始老化。簡單地說，就是讀者與作者、以及讀者與讀者之間，可以方便地在網路上溝通，共同來經營這本書、探討這個課題。沒有人從事過這種實驗，所以我並不知道確實該怎麼做。我在此邀請所有的讀者，一起來參與這個實驗。

我的^手法^的是，從一個簡單的地方開始。最重要的就是建立溝通的管道。網際網路上最普通管道是電子郵件。但是它並不能引起廣大的討論。如果您要送電子郵件給我，請寄

shann@math.ncu.edu.tw

但是請注意：(1) 不要夾帶 (attach) 非純文字檔案 (例如 MS-Word 文件、圖檔、可執行檔)，(2) 只使用 Big-5 碼或 BASE64 編碼，(3) 除非是私人性質的通信，我可能會在 BBS 上公佈您的來信。

最普通的公開雙向溝通管道是電子佈告欄 (BBS)。如果您要使用 BBS，我們的討論版設在中央大學數學系的 BBS 站：織夢天堂

bbs.math.ncu.edu.tw

版名就是書名。我自封為版主。因為沒有對參與討論者設限，而且也不在網路上賣書，所以我認為這應該不構成商業行為。歡迎各站轉載這個討論版。

BBS 雖然簡便，但不方便傳遞數學符號和聲音、影像。所以，我將另外經營一批 WWW 網頁，公布這本書的更正、修改、補充、甚至後續發展，以及所有讀者討論的精華結果。此外，也會呈現許多有關凌波函數的資訊。這批網頁的進入點叫做凌波小棧。它的網址是

<http://www.math.ncu.edu.tw/~shann/Teach/wavelet.html>

參考書目

1. 陳聲偉，〈一維 Poisson 方程凌波有限元素法〉中央大學數學研究所碩士論文，1998
2. I. Daubechies, *Orthonormal bases of compactly supported wavelets*, Comm. Pure Appl. Math. 41 (1988), 909–996.
3. I. Daubechies, *Ten lectures on wavelets*, SIAM, Philadelphia, 1992.
4. G. B. Folland, *Fourier analysis and its applications*, Wadsworth & Brooks/Cole, Pacific Grove, 1992.
5. 黃俊銘 (譯)，〈數值方法—使用 Matlab 程式語言〉Lindfield and Penny 原著，全華科技圖書公司，1998
6. D Kincaid and W Cheney, *Numerical Analysis*, Brooks/Cole, Pacific Grove, 1991.
7. W. Lawton, *Necessary and sufficient conditions for constructing orthonormal wavelet bases*, J. Math. Phys. 32 (1991), 57–61.
8. 歐崇明，〈MATLAB 使用入門手冊〉高立出版社，1998
9. W. Rudin, *Real and complex analysis*, 3rd ed., McGraw-Hill, New York, 1987.
10. J. J. Schoenberg, *Contributions to the problem of approximation of equidistant data by analysis functions*, Quart. Appl. Math. 4 (1946), 45–99.
11. 單維彰，〈傅立葉級數〉數學傳播 22(1998), 81–92

12. W.-C. Shann and C.-C. Yen, *Exact solutions for Daubechies orthonormal scaling coefficients*, 中央大學數學系技術報告 Technical Report 9704
13. 徐敏堂, 《凌波函數的積分法》中央大學數學研究所碩士論文 1995
14. G. Strang, *Wavelets and dilation equations: A brief introduction*, SIAM Review **31** (1989), 614–627.
15. W. Sweldens and R. Piessens, *Quadrature formulas and asymptotic error expansions for wavelet approximations of smooth functions*, SIAM J. Numer. Anal. **31** (1994), 1240–1264.
16. 蘇芻穎, 《Helmholtz 方程的凌波-有限元素數值解與穩定性分析》中央大學數學研究所碩士論文, 1998
17. 曾正男, 《一套提昇凌波函數逼近能力與平滑度的方法》中央大學數學研究所碩士論文, 1997
18. L. F. Villemoes, *Energy moments in time and frequency for two-scale difference equation solutions and wavelets*, SIAM J. Math Anal **23** (1992), 1519–1543.
19. 嚴健彰, 《凌波理論》中央大學數學研究所碩士論文, 1995
20. 鄭儘, 《區段上的凌波函數》中央大學數學研究所碩士論文, 1995

英漢名詞對照

<i>a posteriori</i> error estimates	後驗誤差估計		
analyze	分解	approximate	逼近
array	序列		
B-spline	基本樣條函數	band-limited	有限頻寬
basis	基底	best approximation	最佳估計
box function	方盒函數		
Cartesian product	直積	Cauchy sequence	柯西數列
characteristic value	特徵值	characteristic vector	特徵向量
column vector	行向量	convolution	疊積
compactly supported		有限函蓋	
computational complexity		計算複雜度	
connection coefficients		連接係數	
DSP (Digital Signal Processing)		數位訊號處理	
DWT (discrete wavelet transform)		離散凌波轉換	
data compression	資料壓縮	decompose	分解
denoising	除雜訊	dense	稠密
dilation	脹縮	dimension	維度
direct sum	直和	discretization	離散化
dual basis	對偶基底	dyadic point	二次分點
eigenvalue	特徵值	eigenvector	特徵向量
equivalence condition		等價條件	

FFT (Fast Fourier Transform)	快速傅立葉轉換
fixed-point	不動點
Fourier inverse transform	傅立葉逆轉換
Fourier series	傅立葉級數
frame bounds	框架上下界
global	全域
Heisenberg's Uncertainty Principle	海森堡測不準原理
high-pass filter	高頻濾波器
inner product	內積
inner product (linear) space	具有內積性質的向量空間
interpolation polynomial	內插多項式
kernel function	核心函數
Lebesgue	勒貝格
local	局部
mass matrix	質量矩陣
multiresolution space	多層解析空間
norm	模、長度
normed (linear) space	具有長度性質的向量空間
necessary condition	必要條件
numerical quadrature rule	數值積分法
orthogonal	正交
orthonormal basis	正則基底
pdf (probability distribution function)	機率分佈函數
periodic function	週期性函數
periodize	週期化
Fourier	傅立葉
Fourier transform	傅立葉轉換
frequency domain	頻率域
Hilbert	希爾伯特
interpolation	插值
linear combination	線性組合
low-pass filter	低頻濾波器
moment	動量

phase plain	相位平面		
quantization	數位化		
reconstruct	重建	resolution	解析度
Riemann	黎曼	row vector	列向量
scaling coefficient	自格係數	scaling function	自格函數
Schwarz inequality	席瓦茲不等式	Shannon	單農
spectrum	頻譜	spline	樣條函數
step functions	階梯函數	stiffness matrix	剛度矩陣
sufficient condition	充分條件	synthesis	合成
Taylor expansion	泰勒展開	time domain	時間域
trigonometric polynomial		三角多項式	
transpose	轉置		
unconditional basis	無條件基底		
vector	向量	vector space	向量空間
wavelet	凌波	wavelet transform	凌波轉換
wavelets shrinkage	凌波縮減		

索引

- $\|\cdot\|$, 86
- $\int \sum_k \sum$, 73
- \downarrow (down sampling), 70
- $\binom{p}{k}$ (二項式係數), ~~121~~, 158
- \oplus (直和), 33
- $0! = 1$, 75
- 1-週期函數, 150
- 2^j 分點, 328
- δ_{ij} (Dirac 符號), 97
- ϕ , ~~1-2~~, 157
- ψ , ~~31-32~~, 171
- Banach 空間, 111
- \mathbb{C} (複數集合), 72
- C^0, C^n , 72
- $C^{0,\sigma}$ (Hölder 連續), ~~345~~, 179
- C^1 , ~~7~~, 72
- CZO 形式測試矩陣, 299
- Dirac 符號 (δ_{ij}), 97
- DWT (離散凌波轉換), 45
 - 與 FFT 相比, 271
 - 分解、合成, 241
- Fourier, J. (1768-1830), 122
- FFT (快速傅立葉轉換), 271
- flop, 62, 259
- Gibbs, J. W. (1839-1903), 130
 - Gibbs 現象, ~~129~~, 271
- Haar 函數, 32
 - 凌波係數消散定理, ~~40~~, 182
- Heaviside 函數, 271
- Hilbert 空間, 111, 138
- Hölder 連續指數, ~~179~~, 345
- Lawton 矩陣, ~~166~~, 371, 375
 - 長度, 376
- Lawton 條件, 167
- Lebesgue, H. L. (1875-1941), 111
- L^1 傅立葉轉換定理, 139
- L^2 收斂, 106
- $L_p^2([-T, T])$ (週期函數空間), 125
- $L^2(\mathbb{R}^2)$, 279
- L^2 傅立葉轉換定理, 142

Matlab

- 行導向運算, 397
- 迭代效率, 319
- 定義矩陣, 395
- 副程式, 402
- 線上說明 (help), 402
- ↔ 算術運算, 334
- ↑ 精度, 368
- ↔ 邏輯運算, 334

Matlab 指令

- axis, 407
- break, 402
- ceil, 411
- clear, 406
- conv, 317
- diag, 333, 398
- eig, 332 *disp*
- eye, 396
- find, 334
- floor, 411
- for, 396, 400
- hold, 406
- image, 408* → if, 402
- length, 397
- ones, 396

- plot, 403
- rand, 400
- round, 290, 411
- sin, 400
- size, 397, 248
- sqrt, 398
- sqrtm, 399
- sum, 400
- tic, 401
- title, 407
- toc, 401
- zeros, 396

Matlab 運算符號

- a:h:b, 404
- ' (Hermitian), 396
- + - *, 395
- ^, 398
- /, 395
- \, 395
- .* .+ .^ ./ (逐項運算), 400
- ... (continuation), 396, 407
- :, 396
- >, 400
- Michelson, A. A. (1852-1931), 129
- min j₀, 252

log²

- Neumann 級數, 368
- $O(\cdot)$ (大 O 符號), 10
- P_j (投影算子), 328, 329
- P_n (多項式空間), 81
- Paley-Wiener 定理, 147
- Parseval 等式, 142, 205, 386, 388
- Plancherel 定理, 142
- Poisson 求和公式, 144, 236
- \mathbb{R} (實數集合), 72
- RMS, 212
- Shannon, C. E. (1916-), 152
- sinc 函數, 151
- span (空間展開), 82
- Strang 矩陣, 329, 335, 366, 370, 371
 - 長度, 366
 - 特徵值, 332
- supp (函蓋), 2, 73
- \mathbb{Z} (整數集合), 72
- 二次分點, 328, 327
- 二項式係數, 121, 158
- 三角多項式, 123 = 解字式 18, 86, 88
- 三角帽函數, 149
- 小波, 38
- 不動點, 226
- 內插多項式, 187
- 內積, 18, 91, 93
- 分配律, 121
- 尺度, 30
- 方盒函數, 1, 148
- 平均值, 388
- 必要條件, 165, 23
- 正交投影算子, 22, 24, 168
- 正交, 26
- 交換律, 121
- 充分條件, 165
- 收斂—逐點或 L^2 , 106, 107
- 有限能量函數, 見「函數」
- 有限頻寬, 152
- 有效函蓋, 212
- 自格函數, 157
 - 正交、正則, 164
 - 可微程度, 345
 - 存在之充分條件, 226-227
 - 能量集中, 345
 - 特徵函數, 222
 - 遞迴定義, 347, 348
 - 雙變數, 278

- 自格係數, 157
 - 存在之充分條件, 166-167
- 自格等式, 157
 - 雙變數, 279
- 向量空間, 80
- 向量, ~~79~~, 316
 - 平面向量, 81
 - 正交化, 99
 - 行向量, 57
 - 列向量, 57
 - 序列, 316
 - 長度, 86
 - 模, 87, 86
 - 線性相關, 82
 - 線性組合, 82
 - 線性無關, ~~25~~, 82, 96
 - 線性獨立, 見「線性無關」
- 多層解析空間, 22, 157, 157
- 完美地重建, 54, 70, 104, 104
- 局部性, 270, 274
- 希爾伯特空間, 111, 117, 138
- 序列, 見「向量」
- 折線圖, 404
- 函蓋, ~~2~~, 73

子 12

- 分片常數, 5
- 平移, 12
- 有限函蓋, 73
- 有限能量, 106
 - 距離, 3
 - 脹縮, 12
 - 週期性, 124
- 廣義函數, 226
- 直和, 33
- 直積, 278
- 表達陣列, 83
- 垂直子空間, ~~34~~, 93
- 後驗誤差估計, 381
- 相位平面, 210
- 相位差, 203
- 柯西數列, 111
- 計算複雜度, 62
- 矩陣
 - 可逆、非奇異, 89
 - 特徵向量, 330
 - 特徵值, 330, 372
 - 對稱正定, 94, 99
 - 轉置, 57
- 凌波, 見「凌波函數」

- 凌波函數, 38, 171¹⁷
 - 正交、正則, 178
 - 母函數, 178
 - 可微程度, 345
 - 非正交族類, 38, 155, 15
 - 能量集中, 345
- 凌波係數, 179
 - 消散性質, 39
 - 消散定理, 182
- 凌波縮減, 268
- 凌波轉換, 45
 - 分解合成, 35, 98
 - 分層轉換, 301
 - 全轉換, 301
 - 矩陣表達, 282⁵⁵
 - 矩陣, 56, 243
 - 連續轉換, 196
 - 轉換係數, 53
- 剛度矩陣, 369⁵¹
 - 時域, 212
 - 時間域, 136, 199
- 核心函數, 386
 - 框架, 98
 - 上下界, 98
- 泰勒多項式, 75
- 泰勒展開, 74, 321, 355, 362⁷⁷
- 泰勒級數, 132
- 消散速率, 153, 182
- 海森堡測不準定理, 213
- 席瓦茲不等式, 18, 92, 103
- 弱相等, 94
- 純量, 79, 116
- 能量頻譜, 201
- 除雜訊, 268
- 高斯分佈函數, 214
- 勒貝格收斂定理, 116
- 勒貝格積分, 114¹¹⁶
- 理論上的真確, 350, 362
- 動量— m 次, 189, 358
- 基本樣條函數, 158¹⁴⁸
 - 一階, 347
 - 二階, 403
 - p 階, 148
- 基底, 83
 - 正則基底, 27
 - 自然基底, 26, 44
 - 基底變換, 36, 84
 - 無條件基底, 97, 194
 - 對偶基底, 99
- 傅立葉級數, 123

- ↗ 傅立葉轉換, ~~137~~, 386
 - ↗ 快速, 271
 - ↗ 逆轉換, ~~137~~, 386
 - ↗ 導函數, 388
- ↗ 最佳估計, 184, 94
- ↗ 單農取樣定理, 152
- ↗ 連接係數, 369
 - ↗ 插值, ~~324~~
- 插值函數, 327
 - 偏移現象, 325
- 等價條件, 227
- 結合律, 79
- 階梯函數, 5
 - 逼近論, 15
 - ϵ_j 次, 15
- ↗ 週期化, ~~133~~, ~~143~~, 262, 293
 - ↗ 數位訊號, 244
 - ↗ $F_\phi(\omega)$, 224
 - ↗ pwd, 254
- ↗ 楊輝三角, 121
- ↗ 微分均值定理, 77
- 解析度, 22
- ↗ 資料壓縮, 264
 - ↗ 壓縮比, 295
- 稠密, 21, 41
- ↗ 節點, 312, 379
- ↗ 實數完備性, 76, 108
- 逼近, 6
- ↗ 逼近理論, ~~39~~, ~~187~~
- 189 ↗ 維度, ~~83~~, ~~283~~
 - ↗ n 維、無窮維, 83
- ↗ 數位化, 308
 - 數位訊號處理, 44
- ↗ 數位影像, ~~46~~, 308
- ↗ 數值積分法, 353
 - ↗ 平滑度與誤差, 357
 - ↗ 辛普森法, 356
 - ↗ 梯形法, 355
 - ↗ 單點法, 380
 - ↗ 階數, 355
 - 調適性計算, 380, 385
 - 積分係數, 378, 354
 - $r+1$ 點, 378
 - Newton-Cotes 類型, ~~354~~, 377
 - ↗ 多項式真確, ~~354~~
- ↗ 數值黏滯性, 274
- 廣義平均值, 314
 - 廣義函數, 見「函數」
- ↗ 影像辨識, ~~268~~, 293
- ↗ 質量矩陣, ~~99~~, 369

- └ 黎曼積分, 113
- ρ 機率分佈函數, 388
- 頻率反應範圍, 205
- ← 頻率域, ~~136~~, 199
- ↘ 頻寬, 212
- ⇒ 離散化, 308
- 離散凌波轉換, 見「凌波轉換」
- 濾波
 - └ 低頻濾波函數, 206
 - └ 低頻濾波係數, 207
 - ↗ 低頻濾波器, ~~55~~, ~~69~~, 347
 - ↗ 高頻濾波函數, 207
 - ↗ 高頻濾波係數, 208
 - ↗ 高頻濾波器, ~~56~~, ~~70~~
 - ↗ 邊框效應, ~~293~~, 304
 - 鐘形函數, 14
 - ↗ 疊積, 120
 - └ 反疊積, 121
 - ↗ 離散疊積, ~~69~~, ~~120~~, 315, 348

