

東海大學
應用數學研究所
碩士論文

計算偏微分方程之頻譜配點法

On Spectral Collocation Method Applied in Solving
Partial Differential Equations



研究生：陳駿逸

指導教授：李天佑博士

中華民國九十三年六月

誌 謝

首先要感謝我的指導教授李天佑教授，在這兩年半來不厭其煩的指導與細心照顧，帶領我認識Linux系統和C程式語言，使我能順利完成本論文。另外也要感謝口試委員：簡澄陞教授、洪子倫教授及鄧宗禹教授對於本論文的批評與指導，有您們的寶貴意見，本論文才能更臻完善。也謝謝陪我一起成長努力的同窗好友文儀、詩舫、偉辰、小兵、淑燕、威志、佳潢、193、阿達、靜慧等人，感謝你們的鼓勵與砥礪，是你們讓我的研究所生活能多采多姿。最要感謝的是我最摯愛的家人們，謝謝您們的栽培與照顧，有了您們，我才能全心的完成我的學業。也感謝女友紋惠的加油與打氣，使在我遇到挫折時能繼續努力。最後，希望把這篇論文與所有我最摯愛的人一起分享。

陳駿逸 謹誌

東海大學

2004年7月

摘要

偏微分方程(PDE)無論在數學或物理上都佔有十分重要的地位。通常其解不易求出，一般常使用數值方法求其近似解。傳統多採用有限差分法(Finite difference method)或有限元素法(Finite element method)。近年來由於正交多項式(orthogonal polynomial)的廣泛應用，頻譜法(Spectral method)逐漸受到重視。本論文是採用頻譜法搭配配點法(Collocation Method)，使用C程式語言，研究求解一維、二維、三維偏微分方程式。

第一章介紹頻譜配點法的背景；第二章介紹頻譜配點法的基本原理、Fourier級數、Chebyshev多項式、邊界處理、座標轉換與相關數值方法；第三章以Poisson方程式測試頻譜配點法求解不同維度的邊界條件問題(Boundary Value Problem)；第四章以wave方程式測試頻譜配點法求解線性波方程式；第五章以Burgers方程式測試頻譜配點法求解不同維度的非線性波方程式；第六章為結論。

目錄

誌謝	i
摘要	ii
目錄	iii
圖目錄	vi
表目錄	vii
1 緒論	1
1.1 前言	1
1.2 研究方法與目的	2
2 頻譜配點法之介紹	3
2.1 基本原理	3
2.2 Fourier級數	5
2.3 Chebyshev級數	9
2.4 邊界處理	17
2.5 相關數值方法	22

2.6	座標轉換	24
3	頻譜配點法之應用(一) 邊界值問題	26
3.1	1D Poisson equation	27
3.1.1	Dirichlet Boundary Condition	28
3.1.2	Robin Boundary Condition	31
3.1.3	Neumann Boundary Condition	34
3.2	2D Poisson equation in a square	37
3.2.1	Dirichlet Boundary Condition	38
3.2.2	Robin Boundary Condition	41
3.2.3	Neumann Boundary Condition	44
3.3	2D Poisson equation in a circle	47
3.3.1	Dirichlet Boundary Condition	48
3.3.2	Robin Boundary Condition	51
3.3.3	Neumann Boundary Condition	54
3.4	3D Poisson equation in a cube	57
3.4.1	Dirichlet Boundary Condition	58
3.4.2	Robin Boundary Condition	61

3.4.3	Neumann Boundary Condition	64
3.5	3D Poisson equation in a cylinder	67
3.5.1	Dirichlet Boundary Condition	68
3.5.2	Robin Boundary Condition	73
3.5.3	Neumann Boundary Condition	78
4	頻譜配點法之應用(二) 線性波	83
5	頻譜配點法之應用(三) 非線性波	86
5.1	1D Burgers equation	87
5.2	2D Burgers equation in a square	94
5.3	2D Burgers equation in a circle	101
5.4	3D Burgers equation in a cube	108
5.5	3D Burgers equation in a cylinder	115
6	結論	122
	參考文獻	123

圖目錄

1	一維Chebyshev格點示意圖	17
2	二維Chebyshev格點矩形定義域示意圖	18
3	二維Chebyshev格點圓形定義域示意圖	19
4	三維Chebyshev格點立方體定義域示意圖	20
5	三維Chebyshev格點圓柱定義域示意圖	21
6	tensor product 格點	22
7	lexicographic順序	23
8	極座標所對應之矩陣示意圖	24
9	非線性波傳頻譜配點法誤差圖	85

表目錄

1	1D Poisson with Dirichlet B.C.	28
2	1D Poisson with Dirichlet B.C. and LU Method	29
3	1D Poisson with Dirichlet B.C. and BCR Method	30
4	1D Poisson with Robin B.C.	31
5	1D Poisson with Robin B.C. and LU Method	32
6	1D Poisson with Robin B.C. and BCR Method	33
7	1D Poisson with Neumann B.C.	34
8	1D Poisson with Neumann B.C. and LU Method	35
9	1D Poisson with Neumann B.C. and BCR Method	36
10	2D Poisson in a square with Dirichlet B.C.	38
11	2D Poisson in a square with Dirichlet B.C. and LU Method	39
12	2D Poisson in a square with Dirichlet B.C. and BCR Method	40
13	2D Poisson in a square with Robin B.C.	41
14	2D Poisson in a square with Robin B.C. and LU Method	42
15	2D Poisson in a square with Robin B.C. and BCR Method	43
16	2D Poisson in a square with Neumann B.C.	44
17	2D Poisson in a square with Neumann B.C. and LU Method	45
18	2D Poisson in a square with Neumann B.C. and BCR Method	46
19	2D Poisson in a circle with Dirichlet B.C.	48
20	2D Poisson in a circle with Dirichlet B.C. and LU Method	49
21	2D Poisson in a circle with Dirichlet B.C. and BCR Method	50
22	2D Poisson in a circle with Robin B.C.	51
23	2D Poisson in a circle with Robin B.C. and LU Method	52
24	2D Poisson in a circle with Robin B.C. and BCR Method	53
25	2D Poisson in a circle with Neumann B.C.	54

26	2D Poisson in a circle with Neumann B.C. and LU Method	55
27	2D Poisson in a circle with Neumann B.C. and BCR Method	56
28	3D Poisson in a cube with Dirichlet B.C.	58
29	3D Poisson in a cube with Dirichlet B.C. and LU Method	59
30	3D Poisson in a cube with Dirichlet B.C. and BCR Method	60
31	3D Poisson in a cube with Robin B.C.	61
32	3D Poisson in a cube with Robin B.C. and LU Method	62
33	3D Poisson in a cube with Robin B.C. and BCR Method	63
34	3D Poisson in a cube with Neumann B.C.	64
35	3D Poisson in a cube with Neumann B.C. and LU Method	65
36	3D Poisson 3D in a cube with Neumann B.C. and BCR Method	66
37	3D Poisson in a cylinder with Dirichlet B.C.	68
38	3D Poisson in a cylinder with Dirichlet B.C. (1) and LU Method	69
39	3D Poisson in a cylinder with Dirichlet B.C. (2) and LU Method	70
40	3D Poisson in a cylinder with Dirichlet B.C. (3) and LU Method	71
41	3D Poisson in a cylinder with Dirichlet B.C. and BCR Method	72
42	3D Poisson in a cylinder with Robin B.C.	73
43	3D Poisson in a cylinder with Robin B.C. (1) and LU Method	74
44	3D Poisson in a cylinder with Robin B.C. (2) and LU Method	75
45	3D Poisson in a cylinder with Robin B.C. (3) and LU Method	76
46	3D Poisson in a cylinder with Robin B.C. and BCR Method	77
47	3D Poisson in a cylinder with Neumann B.C.	78
48	3D Poisson in a cylinder with Neumann B.C. (1) and LU Method	79
49	3D Poisson in a cylinder with Neumann B.C. (2) and LU Method	80
50	3D Poisson in a cylinder with Neumann B.C. (3) and LU	81
51	3D Poisson in a cylinder with Neumann B.C. and BCR Method	82
52	comparison of finite difference and spectral collocation	84

53	1D Burgers with $Re = 1.0$ and LU Method	88
54	1D Burgers with $Re = 10.0$ and LU Method	89
55	1D Burgers with $Re = 100$ and LU Method	90
56	1D Burgers with $Re = 1.0$ and GMRes Method	91
57	1D Burgers with $Re = 10.0$ and GMRes Method	92
58	1D Burgers with $Re = 100$ and GMRes Method	93
59	2D Burgers in a square with $Re = 1.0$ and LU Method	95
60	2D Burgers in a square with $Re = 10.0$ and LU Method	96
61	2D Burgers in a square with $Re = 100$ and LU Method	97
62	2D Burgers in a square with $Re = 1.0$ and GMRes Method	98
63	2D Burgers in a square with $Re = 10.0$ and GMRes Method	99
64	2D Burgers in a square with $Re = 100$ and GMRes Method	100
65	2D Burgers in a circle with $Re = 1.0$ and LU Method	102
66	2D Burgers in a circle with $Re = 10.0$ and LU Method	103
67	2D Burgers in a circle with $Re = 100$ and LU Method	104
68	2D Burgers in a circle with $Re = 1.0$ and GMRes Method	105
69	2D Burgers in a circle with $Re = 10.0$ and GMRes Method	106
70	2D Burgers in a circle with $Re = 100$ and GMRes Method	107
71	3D Burgers in a cube with $Re = 1.0$ and LU Method	109
72	3D Burgers in a cube with $Re = 10.0$ and LU Method	110
73	3D Burgers in a cube with $Re = 100$ and LU Method	111
74	3D Burgers in a cube with $Re = 1.0$ and GMRes Method	112
75	3D Burgers in a cube with $Re = 10.0$ and GMRes Method	113
76	3D Burgers in a cube with $Re = 100$ and GMRes Method	114
77	3D Burgers in a cylinder with $Re = 1.0$ and LU Method	116
78	3D Burgers in a cylinder with $Re = 10.0$ and LU Method	117
79	3D Burgers in a cylinder with $Re = 100$ and LU Method	118

80	3D Burgers in a cylinder with $Re = 1.0$ and GMRes Method	119
81	3D Burgers in a cylinder with $Re = 10.0$ and GMRes Method	120
82	3D Burgers in a cylinder with $Re = 100$ and GMRes Method	121

1 緒論

1.1 前言

自然界有許多現象可藉由偏微分方程式(PDE)來表示，如熱傳方程式，波動方程式。有意義而且影響深遠的偏微分方程式來源，主要是物理與幾何。除了前面所列舉的方程式外，舉例來說，還有 Euler 以及 Navier-Stokes 的流體力學方程式；愛因斯坦廣義相對論的愛因斯坦方程式；量子力學中的 Schrödinger 方程式、Dirac 方程式；幾何上的測地線方程式、最小曲面方程式等。因此偏微分方程式廣泛的被應用在科學、工程、數值模擬和天氣預測等領域。然而，高階偏微分方程式的解一般不容易求出，除非是很特殊的例子。偏微分方程式不易解出的原因乃在於其解如黑盒子般，以人工方式幾乎不可能找出其解。由於廿世紀中期之後電子計算機的發達，運用數值方法來求偏微分方程式的解，已經是一門相當專門的學科。

在數值方法求解偏微分方程式的過程中，常使用到不同的技巧選擇展開函數，包括有限元素法(finite element)、有限體積法(finite volume)、有限差分法(finite difference)和近年來常用的頻譜法(spectral method)。頻譜法、有限元素法和有限體積法，都與有限差分法有著密切的關係。但以空間分割的觀念來分，有限差分法和有限體積法採取的是局部近似(local approximation)的觀念，即均須將計算區域切分成許多小單元，再進行空間微分的運算；而頻譜法和有限元素法則是採取整體近似(global approximation)的觀念。頻譜法的收斂速度是呈指數型的收斂，具有誤差收斂速度快的好處。其誤差的大小為 $O(e^{-\alpha N})$ ， α 為一正數， N 為基函數自由度。

早期的頻譜法是應用於週期性的問題上[10]，而非週期性的問題則由 Orszag [4]於1971年，首先提出以 Chebyshev 多項式作為基函數

的各種頻譜法技巧後，逐漸發展才得以應用於非週期性的微分方程式。1977年，Gottliet 和 Orszag [5]建立起頻譜法在數學上的數值理論分析與應用方法，頻譜法才逐漸發展並應用於各類不同的問題。1980年由 P.Moin 和 J.Kim [6]應用擬頻譜法(Pseudospectral Method)於時變不可壓縮黏性流的分析。1984年 Patera [7]發展出頻譜法搭配有限元素法的頻譜元素法(Spectral Element Method)。1987年 Canuto [9]對各種頻譜法的數值方法以及在流體力學的應用有詳細的說明。

1.2 研究方法與目的

在數值模擬的應用上，有限元素法、有限體積法、有限差分法已經有相當悠久的歷史。頻譜法的應用，則開始於1970年代。頻譜法主要是利用正交多項式來近似求解微分方程。正交多項式很早就出現於數學領域之中，但一直未受矚目，直到最近三十年，多位學者以正交多項式的特性，提出有效的代數方法。正交多項式主要的特性，乃是它能以有限的項數，近似展開可連續微分的函數。另外，正交多項式具有的遞迴關係(1)係數遞迴關係(2)微分遞迴關係，特別利於電腦數值執行，故此種數值方法漸受國際學術界重視。利用上述遞迴關係，可推導出展開係數轉換矩陣、乘積運算矩陣、積分運算矩陣及微分運算矩陣。因此近年來正交多項式被廣泛地應用於不同領域。

本文乃是探討利用頻譜配點法，研究求解一維、二維、三維偏微分方程式，在不同定義域下的初值問題與邊界值問題。本文之數值處理程序，是先將偏微分方程及邊界配點後，得到在所選擇的點上，由未知數為一維展開係數所構成的聯立代數方程組，最後以不同的方法求解此聯立代數矩陣，即可得到配點上之近似值。

2 頻譜配點法之介紹

本研究所發展的方法是頻譜配點法(Spectral Collocation Method)。本章的重點在於介紹本研究的基本原理、Fourier 級數、Chebyshev 級數、邊界處理、座標轉換與相關數值方法。

2.1 基本原理

頻譜配點法處理的是空間微分項，時間導數仍使用有限差分法。頻譜法的基本原理是利用一組特徵函數作展開，求得微分方程式的近似解。使用函數展開的近似解並不會完全滿足原方程式，而存有誤差，稱為餘量(residual)。因此要再利用加權餘量法(method of weighted residual)使誤差達到最小。

假設 $u(x, t)$ 為一已知邊界條件的可微分平滑函數，其方程式表示為

$$\begin{cases} P.D.E. : & L(u(x, t)) = f(x, t) \quad x \in \Omega \subset \mathbb{R}^d. \\ B.C. : & B(u(y, t)) = 0 \quad y \in \partial\Omega. \end{cases}$$

L 與 B 為空間微分運算子。假設 $u(x, t)$ 的近似解 $v^N(x, t)$ 落在 Hilbert 空間 H 中的一個有限子空間 P_N 中(通常取 L^2 空間)，則 $v^N(x, t)$ 可以由 P_N 空間中的基底函數 $(\phi_0, \phi_1, \dots, \phi_N)$ 展開如下：

$$v^N(x, t) = \sum_{k=0}^N a_k(t) \phi_k(x).$$

其中， N 為基底函數的自由度； $\phi_k(x)$ 為在求解空間上相互正交的已知解析函數，一般稱為 trial functions； $a_k(t)$ 為待定的展開係數，是時間的函數。則餘量定義如下：

$$R(x; a_0, a_1, \dots, a_N) = L(v^N) - f(x, t)$$

$$= L\left(\sum_{k=0}^N a_k(t)\phi_k(x)\right) - f(x, t).$$

若函數 $(\chi_0, \chi_1, \dots, \chi_N)$ 可利用 Hilbert 空間係數積使得餘量最小，即

$$(\chi_k, R) = 0 \quad \forall k \in 0, 1, \dots, N$$

，則 χ_k 稱為 test functions 。

加權餘量法(MWR)中最關鍵的就是 trial functions 與 test functions 的選取。trial functions 常用的選擇方法有有限元素法、有限體積法、有限差分法和頻譜法等等。本論文採取的是頻譜法，因此 trial functions 是一個 global smooth functions。而頻譜法常用來選擇 test functions 的方法有 Galerkin method、Tau method 和 Collocation(配點法)。本論文採取的是配點法。

頻譜法 trial functions 的選取特別重視下列三個條件：

1. 近似值 $v^N(x, t) = \sum_{k=0}^N a_k(t)\phi_k(x)$ 必須快速的收斂。
2. 若 $\frac{\partial}{\partial x}(\sum_{k=0}^N a_k(t)\phi_k(x)) = \sum_{k=0}^N b_k(t)\phi_k(x)$ ，則當 $a_k(t)$ 的值被給定時， $b_k(t)$ 的值必須容易被決定。
3. 當 $a_k(t)$ 改變時， $v(x_i)$ 的和在某些 x_i 的集合，必須相對快速的改變。

故在本論文中，對於有界(bounded)且週期格點(periodic grid)問題，採用三角多項式(trigonometric polynomials)中的 Fourier 級數；而對於有界(bounded)且非週期性格點(nonperiodic grid)問題，則採用正交多項式(orthogonal polynomials)中的 Chebyshev 級數。

當 test functions $\chi_k(x)$ 與 trial functions $\phi_k(x)$ 選取相同函數的時後，若 $\phi_k(x)$ 滿足邊界條件時，此方法稱為 Galerkin method；若 $\phi_k(x)$ 不

滿足邊界條件時，此方法稱為 Tau method ；若 test functions $\chi_k(x)$ 是採用 δ -function，即 $\chi_j(z) = \delta(z - z_j)$ ，其中 z_j 是 collocation points，此法即為配點法。配點法一般不直接解出係數 $a_k(t)$ ，而是解出在配點上的近似值。由於配點法的發展已經十分完備，且在處理邊界問題有很好的效果，故在本論文中採用 collocation method。

2.2 Fourier級數

週期格點不一定是物理上的，也許是座標空間的，如極座標或球面座標。本論文是採用Fourier級數，應用在週期為 2π 的極座標上。

所謂三角多項式是

$$a_0 + \sum_{n=1}^N a_n \cos nx + b_n \sin nx,$$

形式的函數。其中 N 是一個正整數， x 是實數，並假設 a_n, b_n 也都是實數。很明顯的，三角多項式是一個以 2π 為週期的函數。由Euler identity

$$e^{i\theta} = \cos \theta + i \sin \theta, \quad i = \sqrt{-1}.$$

可改寫

$$\cos nx = \frac{e^{inx} + e^{-inx}}{2}, \quad \sin nx = \frac{e^{inx} - e^{-inx}}{2}.$$

則三角多項式可寫成

$$\sum_{n=-N}^N c_n e^{inx}.$$

此時 x 是實數， c_n 是複數。

若 $u(x)$ 是一個以 2π 為週期的函數，則其 Fourier 級數展開可表示成

$$u(x) = \sum_{n=-\infty}^{\infty} u_n e^{inx},$$

則其 Fourier transform 和 inverse Fourier transform 公式可以表示成

$$\begin{aligned}\widehat{u}(k) &= \int_{-\infty}^{\infty} e^{-ikx} u(x) dx, \quad k \in \mathbb{R}. \\ u(x) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikx} \widehat{u}(k) dk, \quad x \in \mathbb{R}.\end{aligned}$$

如果只考慮 $x \in h\mathbb{Z}$ ，因為空間的定義域是離散的，所以 k 的定義域也不在是在 \mathbb{R} 整個空間上。因此，一個適當的有界定義域長度為 $\frac{2\pi}{h}$ ，且合適的範圍選擇是 $[-\frac{\pi}{h}, \frac{\pi}{h}]$ 。若函數 v 的定義域為 $h\mathbb{Z}$ ，且在 x_j 點的函數值為 v_j ，則其 semidiscrete Fourier transform 和 inverse semidiscrete Fourier transform 公式可以表示成

$$\begin{aligned}\widehat{v}(k) &= h \sum_{j=-\infty}^{\infty} e^{-ikx_j} v_j, \quad k \in [-\frac{\pi}{h}, \frac{\pi}{h}]. \\ v_j &= \frac{1}{2\pi} \int_{-\frac{\pi}{h}}^{\frac{\pi}{h}} e^{ikx_j} \widehat{v}(k) dk, \quad j \in \mathbb{Z}.\end{aligned}$$

令函數 $u(x) \in L^2$ 是一個以 2π 為週期的函數， $x \in [0, 2\pi]$ 。在本論文中週期格點的格點數都取偶數。因此，將 $[0, 2\pi]$ 等分成 N 等分， N 是偶數，且每一段的長 $h = \frac{2\pi}{N}$ 。因為 k 的分部範圍在 $[-\frac{\pi}{h}, \frac{\pi}{h}]$ ，且 $\frac{\pi}{h} = \frac{N}{2}$ ，所以 discrete Fourier transform 和 inverse discrete Fourier transform 公式可以表示成

$$\begin{aligned}\widehat{u}_k &= h \sum_{j=1}^N e^{-ikx_j} u_j, \quad k = -\frac{N}{2} + 1, \dots, \frac{N}{2}. \\ u_j &= \frac{1}{2\pi} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} e^{ikx_j} \widehat{u}_k, \quad j = 1, \dots, N.\end{aligned}$$

取週期的 delta functions 為

$$\delta_j = \begin{cases} 1, & j = 0(\text{mod}N) \\ 0, & j \neq 0(\text{mod}N) \end{cases}$$

則由 discrete Fourier transform 可以得到 $\widehat{\delta}_k = h$ for each k 。因為是週期格點，令 $p(x_j) = v_j$, $x \in [0, 2\pi]$

$$\begin{aligned} p(x) &= \frac{h}{2\pi} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}} e^{ikx} = \frac{h}{2\pi} \left(\frac{1}{2} \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} e^{ikx} + \frac{1}{2} \sum_{k=-\frac{N}{2}+1}^{\frac{N}{2}} e^{ikx} \right) \\ &= \frac{h}{2\pi} \cos\left(\frac{x}{2}\right) \sum_{k=-\frac{N}{2}+\frac{1}{2}}^{\frac{N}{2}-\frac{1}{2}} e^{ikx} \\ &= \frac{h}{2\pi} \cos\left(\frac{x}{2}\right) \frac{e^{i(-\frac{N}{2}+\frac{1}{2})x} - e^{i(\frac{N}{2}+\frac{1}{2})x}}{1 - e^{ix}} \\ &= \frac{h}{2\pi} \cos\left(\frac{x}{2}\right) \frac{e^{-i(\frac{N}{2})x} - e^{i(\frac{N}{2})x}}{e^{-i(\frac{x}{2})} - e^{i(\frac{x}{2})}} \\ &= \frac{h}{2\pi} \cos\left(\frac{x}{2}\right) \frac{\sin\left(\frac{Nx}{2}\right)}{\sin\left(\frac{x}{2}\right)} \end{aligned}$$

由此可知對 delta functions 的插值(interpolant)。若週期的 sinc function $S_N(x) = \frac{\sin(\frac{\pi x}{h})}{(\frac{2\pi}{h})\tan(\frac{x}{2})}$ ，則對任意其他的函數的差值可表示如下：

$$p(x) = \sum_{m=1}^N S_N(x - x_m).$$

因為

$$S'_N(x_j) = \begin{cases} 0, & j = 0(\text{mod}N) \\ \frac{1}{2}(-1)^j \cot\left(\frac{jh}{2}\right), & j \neq 0(\text{mod}N) \end{cases}$$

所以，一階 Fourier collocation derivative matrix D_N 如下：

$$(D_N)_{ij} = \begin{cases} 0, & j = 0(\text{mod}N) \\ \frac{1}{2}(-1)^j \cot(\frac{jh}{2}), & j \neq 0(\text{mod}N) \end{cases}$$

又因為

$$S_N''(x_j) = \begin{cases} -\frac{\pi^2}{3h^2} - \frac{1}{6}, & j = 0(\text{mod}N) \\ -\frac{(-1)^j}{2\sin^2(\frac{jh}{2})}, & j \neq 0(\text{mod}N) \end{cases}$$

所以，二階 Fourier collocation derivative matrix D_N 如下：

$$(D_N^2)_{ij} = \begin{cases} -\frac{\pi^2}{3h^2} - \frac{1}{6}, & j = 0(\text{mod}N) \\ -\frac{(-1)^j}{2\sin^2(\frac{jh}{2})}, & j \neq 0(\text{mod}N) \end{cases}$$

Fourier spectral differentiation 的準確度可由下面的定理得知[1]：

定理 1 *Accuracy of Fourier spectral differentiation*

Let $u \in L^2(\mathbb{R})$ have a ν th derivative ($\nu \geq 1$) of bounded variation, and let w be the ν th spectral derivative of u on the grid $h\mathbb{Z}$. The following estimates hold uniformly for all $x \in h\mathbb{Z}$.

1. If u has $p-1$ continuous derivatives in $L^2(\mathbb{R})$ for some $p \geq \nu+1$ and a p th derivative of bounded variation, then

$$|w_j - u^\nu(x_j)| = O(h^{p-\nu}) \quad \text{as } h \rightarrow \infty.$$

2. If u has infinitely many continuous derivatives in $L^2(\mathbb{R})$, then

$$|w_j - u^\nu(x_j)| = O(h^m) \quad \text{as } h \rightarrow \infty$$

for every $m \geq 0$.

3. If there exist $a, c > 0$ such that u can be extended to an analytic function in the complex strip $|Imz| < a$ with $\|u(\cdot + iy)\| \leq c$ uniformly for all $y \in (-a, a)$, then

$$|w_j - u^{(\nu)}(x_j)| = O(e^{-\frac{\pi(a-\epsilon)}{h}}) \quad \text{as } h \rightarrow \infty$$

for every $\epsilon > 0$.

4. If u can be extended to an entire function and there exists $a > 0$ such that for $z \in \mathbb{C}$, $|u(z)| = o(e^{a|z|})$ as $|z| \rightarrow \infty$, then, provided $h \leq \frac{\pi}{a}$, $w_j = u^{(\nu)}(x_j)$.

2.3 Chebyshev 級數

k 次 Chebyshev 多項式定義為

$$T_k(x) = \cos(k \arccos x), \quad x \in [-1, 1].$$

$$\theta = \arccos x, \quad \theta \in [0, \pi].$$

在經過一獨立變數的轉換後，是一餘弦函數。由於此一特性，使得 Chebyshev 多項式可用於解非週期性的邊界值問題，而具有類似餘弦級數之 spectral accuracy。

Chebyshev 多項式的特性如下：

$$|T_k(x)| \leq 1, \quad -1 \leq x \leq 1.$$

$$T_k(\pm 1) = (\pm 1)^k.$$

根據三角恆等式 $\cos(k+1)\theta + \cos(k-1)\theta = 2 \cos \theta \cos k\theta$ 可得 Chebyshev 多項式的三項遞迴關係式：

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x).$$

$$\begin{aligned}
T_0(x) &= 1, \\
T_1(x) &= x, \\
T_2(x) &= 2x^2 - 1, \\
T_3(x) &= 4x^3 - 3x, \\
T_4(x) &= 8x^4 - 8x^2 + 1, \\
&\vdots
\end{aligned}$$

而 Chebyshev 多項式的正交關係為：

$$\begin{aligned}
\langle T_i(x), T_j(x) \rangle &= \int_{-1}^1 T_i(x)T_j(x)w(x)dx \\
&= \begin{cases} 0, & i \neq j \\ \frac{\pi}{2}, & i = j \neq 0 \\ \pi, & i = j = 0 \end{cases}
\end{aligned}$$

其中權重係數 $w(x) = \frac{1}{\sqrt{1-x^2}}$ 為 Chebyshev weight，函數必須要乘上此一特殊權重係數，才會得到正交關係。Chebyshev 轉換公式如下：

$$u = \sum_{k=0}^{\infty} \hat{u}_k T_k(x),$$

由正交關係可知：

$$\begin{aligned}
\hat{u}_k &= \frac{2}{c_k \pi} \int_{-1}^1 \frac{u(x)T_k(x)}{\sqrt{1-x^2}} \\
c_k &= \begin{cases} 2, & k = 0 \\ 1, & k > 0 \end{cases}
\end{aligned}$$

Chebyshev 離散轉換公式如下：

$$u = \sum_{k=0}^N \tilde{u}_k T_k(x),$$

由離散正交關係可知：

$$\tilde{u}_k = \frac{2}{\pi \bar{c}_k} \sum_{j=0}^N u(x_j) T_k(x_j) w_j$$

$$\bar{c}_k = \begin{cases} 2, & k = 0, N \\ 1, & k = 1, \dots, N-1 \end{cases}$$

所以，

$$u = \sum_{k=0}^N L_{N,k}(x) u(x_k)$$

$$L_{N,k}(x) = \prod_{\substack{l=0 \\ l \neq k}}^N \frac{x - x_l}{x_k - x_l}$$

其中， $L_{N,k}(x)$ 是 Lagrange 多項式，滿足 $L_{N,k}(x_l) = \delta_{kl}$ 。令 $L_{N,k}(x) = \psi_k(x) = \prod_{\substack{l=0 \\ l \neq k}}^N \frac{x - x_l}{x_k - x_l}$ 。

常用的 collocation points 為 Chebyshev-Gauss point，就是 Chebyshev 多項式 $T_{N+1}(x) = 0$ 的根：

$$x_j = \cos \frac{(2j+1)\pi}{2N+2}, \quad j = 0, 1, \dots, N.$$

但不含端點。為了使 collocation points 包含邊界點其一或其二，可作如下的改良：

$$\begin{aligned} q(x) &= T_{N+1}(x) - T_N(x) \\ &= \cos(N+1)\theta - \cos N\theta \\ &= -2 \sin \frac{2N+1}{2} \theta \sin \frac{\theta}{2} \end{aligned}$$

，則當

$$\frac{2N+1}{2} \theta = j\pi, \quad j = 0, \pm 1, \pm 2, \dots$$

時， $q(x) = 0$ 。為了滿足 $0 \leq \theta \leq \pi$ ，所以

$$\theta_j = \frac{2j\pi}{2N+1}, \quad j = 0, 1, \dots, N.$$

則 Chebyshev-Gauss-Radau points：

$$x_j = \cos \frac{2j\pi}{2N+1}, \quad j = 0, 1, \dots, N.$$

含一端點 $x = 1$ 。同理，

$$\begin{aligned} q(x) &= T'_N(x)(1-x^2) \\ &= \frac{N \sin N\theta}{\sin \theta} \end{aligned}$$

則當

$$N\theta_j = j\pi, \quad j = 1, 2, \dots, N-1,$$

或

$$\theta_j = \frac{j\pi}{N}, \quad j = 1, 2, \dots, N-1$$

時， $q(x) = 0$ 。為了包含兩端點 $x = \pm 1$ ，所以

$$\theta_j = \frac{j\pi}{N}, \quad j = 0, 1, 2, \dots, N.$$

則 Chebyshev-Gauss-Lobatto points：

$$x_j = \cos \frac{j\pi}{N}, \quad j = 0, 1, 2, \dots, N.$$

含兩端點 $x = \pm 1$ 。通常，常微分方程式邊界值問題都採用 Chebyshev-Gauss-Lobatto points。

已知 Chebyshev 多項式 $T_N(x) = \cos N\theta$, $\theta = \arccos x$, 則

$$\begin{aligned} \frac{d}{dx}T_N(x) &= \frac{d\theta}{dx} \frac{d}{d\theta} \cos N\theta \\ &= \frac{N \sin N\theta}{\sin \theta}, \end{aligned}$$

而

$$\begin{aligned} \frac{d^2}{dx^2}T_N(x) &= \frac{d\theta}{dx} \frac{d}{d\theta} \left(\frac{N \sin N\theta}{\sin \theta} \right) \\ &= -\frac{N^2 \cos N\theta \sin \theta - N \sin N\theta \cos \theta}{\sin^3 \theta}. \end{aligned}$$

現在觀察 $T'_N(x)$ 與 $T''_N(x)$ 在 collocation points $x_j = \cos \frac{j\pi}{N}$ 上的值：

$$\begin{aligned} T'_N(x_j) &= \begin{cases} 0, & j \neq 0, N \\ N^2, & j = 0 \\ (-1)^{N+1}N^2, & j = N \end{cases} \\ &= \begin{cases} 0, & j \neq 0, N \\ (\pm 1)^{N+1}N^2, & j = 0, N \end{cases} \end{aligned}$$

而

$$T''_N(x_j) = \begin{cases} \frac{(-1)^{j+1}N^2}{1-x_j^2}, & j \neq 0, N \\ (\pm 1)^N N^2 \left(\frac{N^2-1}{3} \right), & j = 0, N \end{cases}$$

同理，亦可推得 $T'''_N(x)$ 在 collocation points $x_j = \cos \frac{j\pi}{N}$ 上的值：

$$T'''_N(x_j) = \begin{cases} \frac{(-1)^{j+1}3N^2x_j}{(1-x_j^2)^2}, & j \neq 0, N \\ (\pm 1)^{N+1}N^2 \left(\frac{N^2-1}{3} \right) \left(\frac{N^2-4}{5} \right), & j = 0, N \end{cases}$$

Chebyshev collocation 導數可以矩陣形式

$$\mathcal{D}_N u(x_l) = \sum_{j=0}^N (D_N)_{lj} u(x_j), \quad l = 0, 1, \dots, N.$$

表之。其中 $\mathcal{D}_N u = (I_N u)'$ 。而 $(D_N)_{lj}$ 可藉由微分 Lagrange 多項式 ψ_j 得到，意即 $(D_N)_{lj} = \psi_j'(x_l)$ 。當使用 Chebyshev-Gauss-Lobatto 時，

$$\begin{aligned} \psi_k(x) &= \prod_{\substack{l=0 \\ l \neq k}}^N \frac{x - x_l}{x_k - x_l} \\ &= \frac{(-1)^{k+1} (1 - x^2) T_N'(x)}{\bar{c}_k N^2 (x - x_k)} \end{aligned}$$

則

$$(x - x_j) \psi_j(x) = \frac{(-1)^{j+1}}{\bar{c}_j N^2} (1 - x^2) T_N'(x).$$

等號兩邊微分，可得

$$\psi_j(x) + (x - x_j) \psi_j'(x) = \frac{(-1)^{j+1}}{\bar{c}_j N^2} [-2x T_N'(x) + (1 - x^2) T_N''(x)].$$

可分為以下四個部分討論：

1. 當 $l \neq j$ 時，

$$\begin{aligned} \psi_j(x_l) + (x_l - x_j) \psi_j'(x_l) &= \frac{(-1)^{j+1}}{\bar{c}_j N^2} [-2x_l T_N'(x_l) + (1 - x_l^2) T_N''(x_l)] \\ &= \frac{(-1)^{j+1}}{\bar{c}_j N^2} \begin{cases} -2(\pm 1)^N N^2 \\ (-1)^{l+1} N^2 \end{cases} \\ &= \frac{(-1)^{j+1}}{\bar{c}_j N^2} [\bar{c}_l (-1)^{l+1} N^2] \\ &= \frac{\bar{c}_l (-1)^{l+j}}{\bar{c}_j} \end{aligned}$$

因此， $\psi_j'(x_l) = \frac{\bar{c}_l (-1)^{l+j}}{\bar{c}_j (x_l - x_j)}$ 。

2. 當 $l = j = 0$ 時，意即 $x_j = x_l = 1$ ，則

$$\begin{aligned}\psi_j(x) &= \frac{(-1)^{j+1}(1-x^2)T'_N(x)}{\bar{c}_j N^2(x-x_1)} \\ &= \frac{(1+x)T'_N(x)}{2N^2}\end{aligned}$$

那麼， $\psi'_j(x) = \frac{1}{2N^2} [T'_N(x) + (1+x)T''_N(x)]$ 。因此，

$$\begin{aligned}\psi'_0(x_0) &= \psi'_0(x_1) \\ &= \frac{1}{2N^2} [T'_N(1) + (1+1)T''_N(1)] \\ &= \frac{1}{2N^2} \left[N^2 + \frac{2N^2(N^2-1)}{3} \right] \\ &= \frac{2N^2+1}{6}\end{aligned}$$

3. 當 $l = j = N$ 時，意即 $x_j = x_l = -1$ ，則

$$\begin{aligned}\psi_N(x) &= \frac{(-1)^{N+1}(1-x)T'_N(x)}{2N^2} \\ &= \frac{(-1)^{N+1}}{2N^2} [(1-x)T'_N(x)].\end{aligned}$$

那麼， $\psi'_N(x) = \frac{(-1)^{N+1}}{2N^2} [-T'_N(x) + (1-x)T''_N(x)]$ 。因此，

$$\begin{aligned}\psi'_N(-1) &= \frac{(-1)^{N+1}}{2N^2} [-T'_N(-1) + 2T''_N(-1)] \\ &= \frac{(-1)^{N+1}}{2N^2} \left[(-1)^N N^2 + (-1)^N \frac{2N^2(N^2-1)}{3} \right] \\ &= -\frac{1}{2} \left[1 + \frac{2N^2-2}{3} \right] \\ &= -\frac{2N^2+1}{6} \\ &= -\psi'_0(1)\end{aligned}$$

4. 當 $l = j \neq 0, N$ 時，則

$$2\psi'_j(x) + (x-x_j)\psi''_j(x)$$

$$= \frac{(-1)^{j+1}}{\bar{c}_j N^2} [-2xT'_N(x) - 4xT''_N(x) + (1-x^2)T'''_N(x)].$$

那麼， $x = x_l = x_j$ 代入上式得到：

$$\begin{aligned} 2\psi'_j(x_j) &= \frac{(-1)^{j+1}}{\bar{c}_j N^2} [-2T'_N(x_j) - 4x_j T''_N(x_j) + (1-x_j^2)T'''_N(x_j)] \\ &= \frac{(-1)^{j+1}}{\bar{c}_j N^2} \left[\frac{-4x_j(-1)^{j+1}N^2}{1-x_j^2} + \frac{3x_j(-1)^{j+1}N^2}{1-x_j^2} \right] \\ &= \frac{-x_j}{1-x_j^2} \end{aligned}$$

所以， $\psi'_j(x_j) = \frac{-x_j}{2(1-x_j^2)}$ 。

所以，一階 Chebyshev collocation derivative matrix D_N 如下：

$$(D_N)_{ij} = \begin{cases} \frac{\bar{c}_i(-1)^{i+j}}{\bar{c}_j(z_i-z_j)}, & i \neq j, \\ \frac{-z_j}{2(1-z_j^2)}, & 1 \leq i = j \leq N-1, \\ \frac{2N^2+1}{6}, & i = j = 0, \\ -\frac{2N^2+1}{6}, & i = j = N, \end{cases}$$

其中 z_j ($j = 0, 1, \dots, N$) 是 Gauss-Labotto collocation points，而 $\bar{c}_k = 1$ ，除了 $\bar{c}_0 = \bar{c}_N = 2$ 。二階 Chebyshev collocation derivative matrix D_N^2 可藉由平方一階 collocation derivative matrix D_N 得到。

Chebyshev spectral differentiation 的準確度可由下面的定理得知[1]：

定理 2 Accuracy of Chebyshev spectral differentiation

Suppose u is analytic on and inside the ellipse with foci ± 1 on which the Chebyshev potential takes the value ϕ_u , that is, the ellipse whose semimajor and semiminor axis lengths sum to $K = e^{\phi_u + \log 2}$. Let w be the ν th Chebyshev spectral derivative of u ($\nu \geq 1$). Then

$$|w_j - u^{(\nu)}(x_j)| = O(e^{-N(\phi_u + \log 2)}) = O(K^{-N}) \text{ as } N \rightarrow \infty.$$

2.4 邊界處理

頻譜配點法處理邊界條件有兩類基本的方法：

1. 重新建構原有的微分矩陣，使它滿足邊界條件。
2. 增加額外的方程式，滿足邊界條件。

在本論文中是採取第一類方法來處理不同的邊界條件。

若是一維的題目(圖1)，邊界只出現在兩端，微分矩陣的第一列與最後一列就是用來計算邊界的差值。因此要改變矩陣的第一列與最後一列，滿足已知的邊界條件。

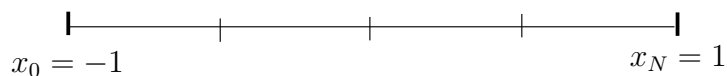


圖 1: 一維Chebyshev格點示意圖

若是二維的題目，且定義域是矩形，則邊界出現在矩形的四個邊(圖2)。若 x 方向與 y 方向的自由度均為 N ，則要改變矩陣的前 N 列與末 N 列，滿足已知邊界條件的南北兩邊界；改變矩陣的第 M_1 列，滿足已知邊界條件的西邊邊界；改變矩陣的第 M_2 列，滿足已知邊界條件的東邊邊界。其中

$$M_1 = 1 + k * N,$$

$$M_2 = N + k * N,$$

$$\text{for } 1 \leq k \leq N - 2$$

更多的說明見2.5節。

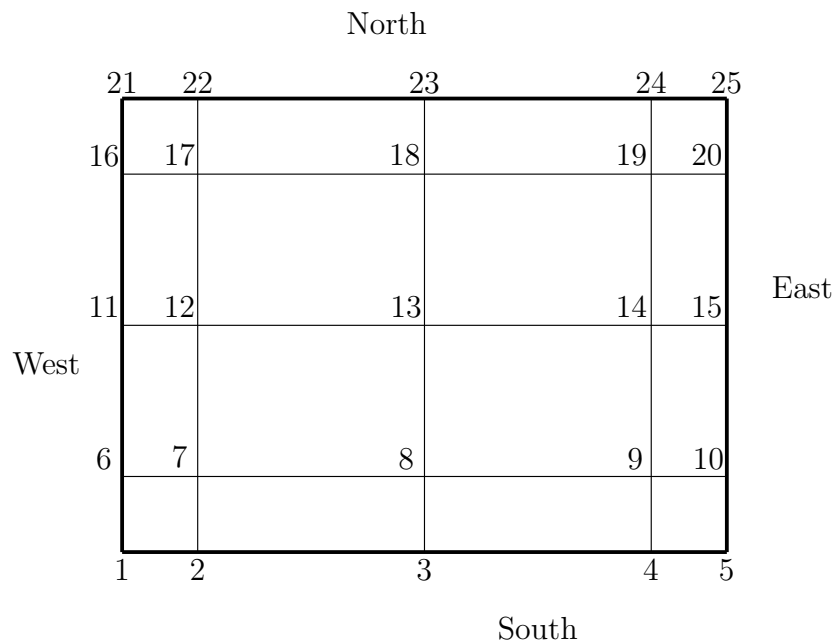


圖 2: 二維Chebyshev格點矩形定義域示意圖

若是二維的題目，且定義域是圓形，則邊界出現在最外側圓形的圓周上的點。(圖3) 若 θ 方向上的自由度為 M ，則要改變矩陣的前 M 列，滿足已知的邊界條件。

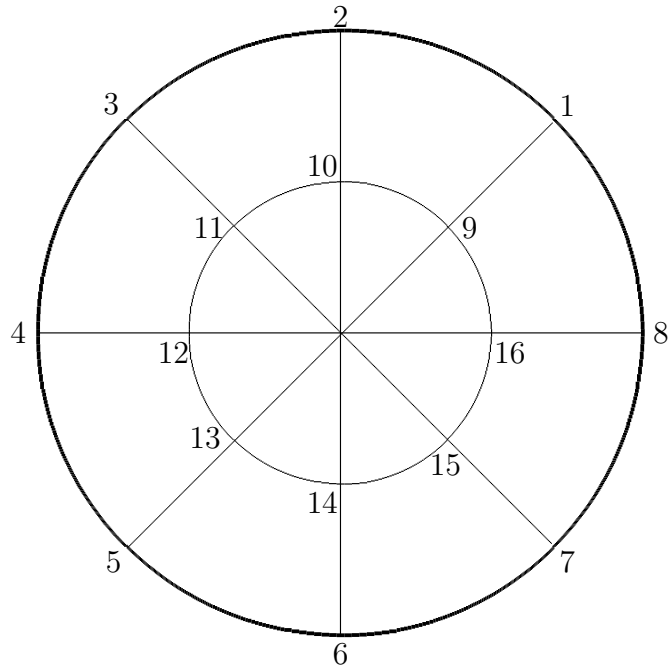


圖 3: 二維Chebyshev格點圓形定義域示意圖

若是三維的題目，且定義域是立方體，則邊界出現在立方體的六個面上的點(圖4)。若 x 方向與 y 方向與 z 方向的自由度均為 N ，則要改變矩陣的前 $N * N$ 列與末 $N * N$ 列，滿足已知邊界條件的上下兩邊界；改變矩陣的第 M_1 列，滿足已知邊界條件的南邊邊界；改變矩陣的第 M_2 列，滿足已知邊界條件的北邊邊界；

改變矩陣的第 M_3 列，滿足已知邊界條件的西邊邊界；改變矩陣的第 M_4 列，滿足已知邊界條件的東邊邊界。其中

$$M_1 = N * N * k + i,$$

$$M_2 = N * N * k + N * (N - 2) + i, \quad \text{for } 1 \leq i \leq N,$$

$$M_3 = N * N * k + 1 + j * N, \quad 1 \leq j \leq N - 2,$$

$$M_4 = N * N * k + N + j * N, \quad 1 \leq k \leq N - 2.$$

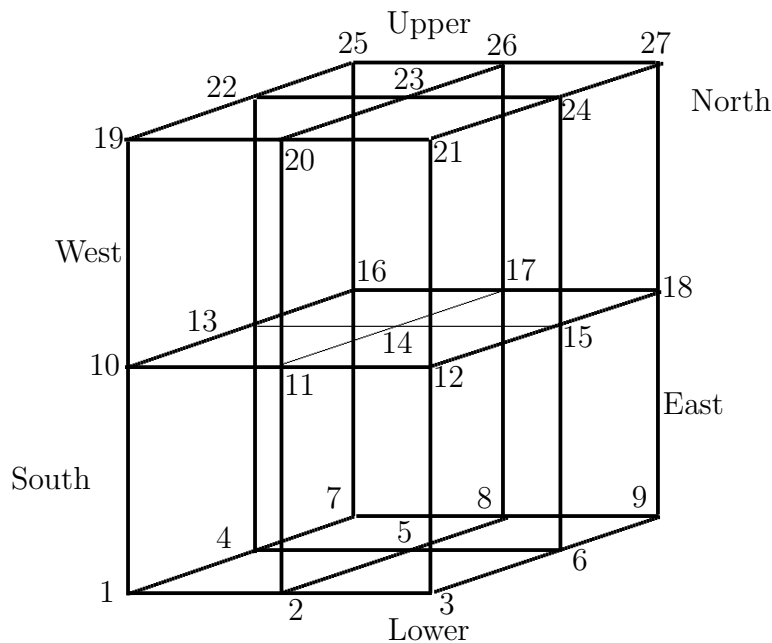


圖 4: 三維Chebyshev格點立方體定義域示意圖

若是三維的題目，且定義域是圓柱，則邊界出現在最外側圓形的圓周上的點與上下兩面的點。(圖5) 若 r 方向的自由度為 N ， θ 方向的自由度為 M ， z 方向的自由度為 H ，則要改變矩陣的前 $N * M$ 列與末 $N * M$ 列，滿足邊界條件的上下兩邊界；改變矩陣的第 M_1 列，滿足最外側圓周上的邊界條件。

$$M_1 = M * N * k + j, \quad \text{for } 1 \leq k \leq H - 2, 1 \leq j \leq M.$$

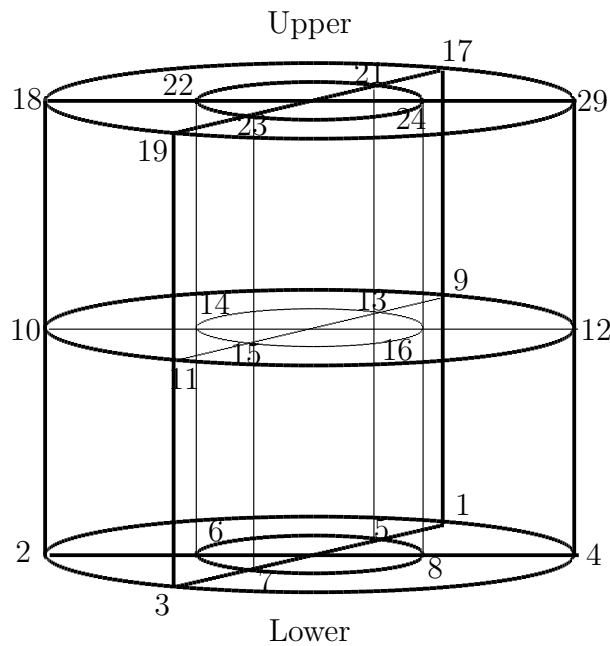


圖 5: 三維Chebyshev格點圓柱定義域示意圖

2.5 相關數值方法

以上的問題，都是 tensor product grid (見圖6)。

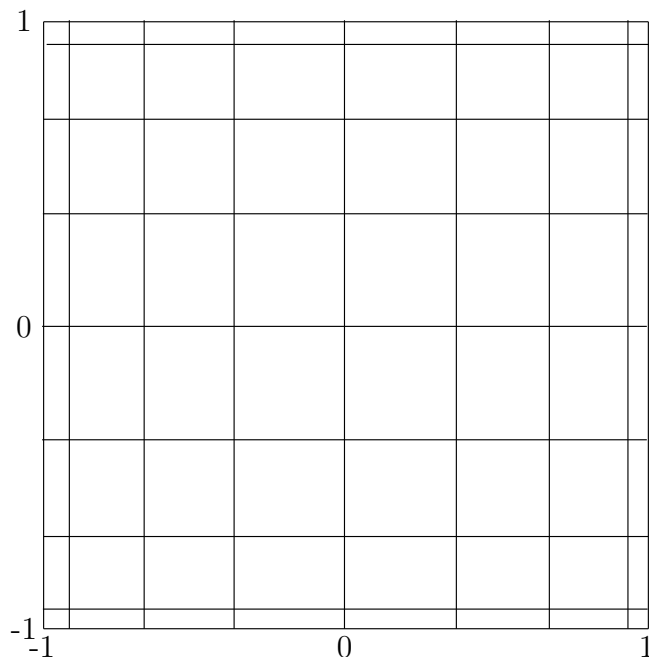


圖 6: tensor product 格點

要利用線性代數方法解決 tensor product grid，可利用 Kronecker products。兩矩陣A、B間的 Kronecker products 定義成 $A \otimes B$ 。假如A、B矩陣的維度分別是 $p \times q$ 與 $r \times s$ ，則 $A \otimes B$ 矩陣的維度是 $pr \times qs$ ，且有 $p \times q$ 個區塊。例如：

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \otimes \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \left(\begin{array}{cc|cc} a & b & 2a & 2b \\ c & d & 2c & 2d \\ \hline 3a & 3b & 4a & 4b \\ 3c & 3d & 4c & 4d \end{array} \right).$$

若以“lexicographic”順序(見圖7)來數二維的 tensor product grid，則離散

的 Laplacian 是兩個 Kronecker product 的和：

$$L = I_1 \otimes D_N + D_M \otimes I_2.$$

其中， N 是 x 方向的自由度， D_N 是對 x 的一階偏微分矩陣； M 是 y 方向的自由度， D_M 是對 y 的一階偏微分矩陣； I_1 是 $(M+1) \times (M+1)$ 的單位矩陣， I_2 是 $(N+1) \times (N+1)$ 的單位矩陣。

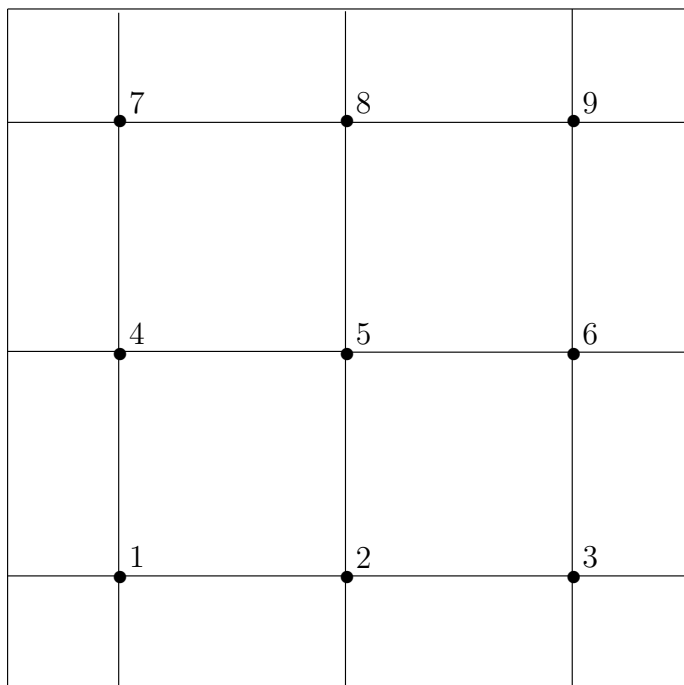


圖 7: lexicographic 順序

若是以另一種順序來數二維的 tensor product grid (即先數 y 方向，再數 x 方向)，則離散的 Laplacian 是兩個 Kronecker product 的和：

$$L = I_1 \otimes D_M + D_N \otimes I_2.$$

其中， N 是 x 方向的自由度， D_N 是對 x 的一階偏微分矩陣； M 是 y 方向的自由度， D_M 是對 y 的一階偏微分矩陣； I_1 是 $(N+1) \times (N+1)$ 的單位矩陣， I_2 是 $(M+1) \times (M+1)$ 的單位矩陣。三維以上的情形以此類推。

2.6 座標轉換

頻譜配點法可應用在許多定義域上。常用到的就是利用極座標(polar coordinates)轉換，將頻譜配點法應用在單位圓盤和圓柱上。離散化時，使用週期的 Fourier 格點在角度 θ 上，使用非週期的 Chebyshev 格點在半徑 r 上。其中， $\theta \in [0, 2\pi]$ ， $r \in [-1, 1]$ 。

由於 (r, θ) 轉換到 (x, y) 是2-to-1，為了避免處理 (x, y) 在原點位置的問題，在 r 方向上的自由度 N 都取奇數。經過極座標轉換，原來的2維偏微分方程式 $u_{xx} + u_{yy}$ 可以轉換成 $u_{rr} + r^{-1}u_r + r^{-2}u_{\theta\theta}$ 。在 (r, θ) 空間中有 $(N_r + 1) \times (N_\theta)$ 個格點落在 (r, θ) 平面上，因此在這些格點離散的 Laplacian 矩陣大小為 $((N_r + 1) \times (N_\theta)) \times ((N_r + 1) \times (N_\theta))$ 。因為 $u(r, \theta) = u(-r, (\theta + \pi)(\text{mod}2\pi))$ ，所以可以拋棄原矩陣第三和第四部份(見圖8)。

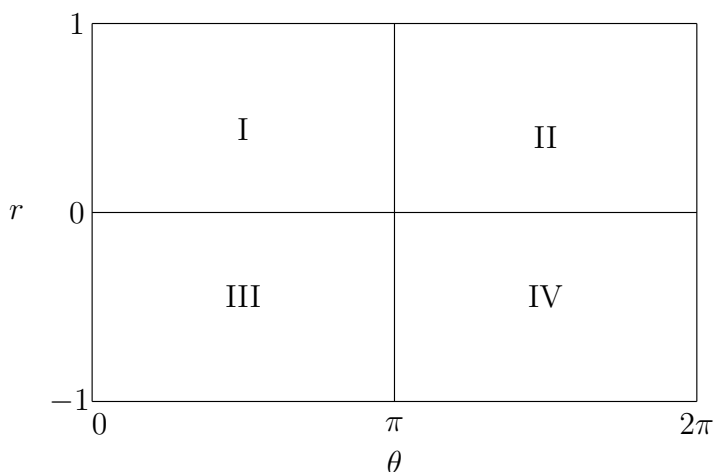


圖 8: 極座標所對應之矩陣示意圖

$$D_r = \begin{array}{c} r > 0 & r < 0 \\ \left(\begin{array}{c|c} D_1 & D_2 \\ \hline D_3 & D_4 \end{array} \right) & \begin{array}{l} r > 0 \quad \leftarrow \text{added together} \\ r < 0 \quad \leftarrow \text{discarded} \end{array} \end{array}$$

$$D_r^2 = \begin{array}{c} r > 0 & r < 0 \\ \left(\begin{array}{c|c} D_1^2 & D_2^2 \\ \hline D_3^2 & D_4^2 \end{array} \right) & \begin{array}{l} r > 0 \quad \leftarrow \text{added together} \\ r < 0 \quad \leftarrow \text{discarded} \end{array} \end{array}$$

因為對 r 方向作一次和二次偏微分矩陣 \tilde{D}_r^2 和 \tilde{D}_r 的大小，都是 $(N_r + 1) \times (N_r + 1)$ ；且對 θ 方向作二次偏微分矩陣 D_θ^2 的大小，是 $(N_\theta) \times (N_\theta)$ 。所以，Laplacian 離散化後的矩陣 L 在極座標的表示為

$$L = (D_1^2 + RD_1) \otimes \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix} + (D_2^2 + RD_2) \otimes \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix} + R^2 \otimes D_\theta^2.$$

其中 I 是一個大小為 $(\frac{N_\theta}{2}) \times (\frac{N_\theta}{2})$ 的單位矩陣； R 是一個對角線矩陣，對角線元素為 r_j^{-1} ， $0 \leq j \leq \frac{N_r-1}{2}$ ； \otimes 代表Kronecker products。三維以上的情形以此類推。

3 頻譜配點法之應用(一) 邊界值問題

本章將測試頻譜配點法應用於一維、二維、三維的Poisson方程式。用不同的測試函數，在不同定義域與不同邊界條件下，先算出正確解的答案，再和近似解比較，算出誤差。測試中採取直接解法的LU分解法，與BCR(Bi-Conjugate Residual method)[11]迭代方法。

3.1 1D Poisson equation

一維的Poisson方程式： $u_{xx} = f$, $-1 \leq x \leq 1$.

設定基本參數：

N : x 方向的自由度

$iter$: 迭代次數總和

res_max : 最大 *residual norm*

err_max : 最大誤差

cpu_time : 測試所花的全部時間

由表1至表9可知，無論何種邊界條件，頻譜配點法都只用到少數的自由度，即可快速且準確的逼近到正確答案。

3.1.1 Dirichlet Boundary Condition

使用LU分解法與BCR迭代法[11]解一維的Poisson方程式。

邊界值：

- $u(-1)$.
- $u(1)$.

case	RHS function of PDE	exact solution
1	0.0	1.0
2	1.0	x
3	2.0	x^2
4	$6.0 * x$	x^3
5	$exp(x)$	$exp(x)$
6	$-\pi^2 * sin(x)$	$sin(\pi * x)$
7	$-\pi^2 * cos(x)$	$cos(\pi * x)$

表 1: 1D Poisson with Dirichlet B.C.

eq	N	method	res_max	err_max	iter	cpu_t
1	5	LU	1.90e-15	4.44e-16	1	0.00
1	6	LU	3.18e-15	7.77e-16	1	0.00
2	5	LU	1.48e-15	2.78e-16	1	0.00
2	6	LU	6.82e-16	5.55e-16	1	0.00
3	5	LU	3.63e-16	3.75e-16	1	0.00
3	6	LU	4.44e-16	5.55e-16	1	0.00
4	5	LU	8.16e-16	2.84e-16	1	0.00
4	6	LU	8.22e-16	4.44e-16	1	0.00
5	12	LU	4.04e-14	1.64e-14	1	0.00
5	13	LU	5.08e-14	5.00e-15	1	0.00
6	17	LU	7.60e-15	4.62e-14	1	0.00
6	18	LU	1.03e-14	1.77e-14	1	0.00
7	17	LU	6.20e-14	2.26e-13	1	0.00
7	18	LU	9.08e-14	4.77e-15	1	0.00

表 2: 1D Poisson with Dirichlet B.C. and LU Method

eq	N	method	res_max	err_max	iter	cpu_t
5	12	BCR	3.57e-13	1.78e-14	14	0.00
5	13	BCR	9.70e-14	6.66e-15	14	0.00
6	17	BCR	5.97e-14	5.05e-14	13	0.00
6	18	BCR	6.88e-14	1.77e-14	14	0.00
7	17	BCR	1.10e-12	2.26e-13	15	0.00
7	18	BCR	2.13e-12	9.66e-15	16	0.00

表 3: 1D Poisson with Dirichlet B.C. and BCR Method

3.1.2 Robin Boundary Condition

使用LU分解法與BCR迭代法[11]解一維的Poisson方程式。

邊界值：

- $u(-1)$.
- $u_x(1)$.

case	RHS function of PDE	exact solution
1	0.0	1.0
2	1.0	x
3	2.0	x^2
4	$6.0 * x$	x^3
5	$exp(x)$	$exp(x)$
6	$-\pi^2 * sin(x)$	$sin(\pi * x)$
7	$-\pi^2 * cos(x)$	$cos(\pi * x)$

表 4: 1D Poisson with Robin B.C.

eq	N	method	res_max	err_max	iter	cpu_t
1	5	LU	2.01e-15	5.55e-16	1	0.00
1	6	LU	1.57e-15	8.88e-16	1	0.00
2	5	LU	1.23e-15	3.11e-15	1	0.00
2	6	LU	1.04e-15	3.44e-15	1	0.00
3	7	LU	1.29e-15	3.11e-15	1	0.00
3	8	LU	1.58e-15	9.10e-15	1	0.00
4	7	LU	3.63e-15	2.89e-15	1	0.00
4	8	LU	1.55e-15	7.77e-15	1	0.00
5	13	LU	4.09e-14	3.91e-14	1	0.00
5	14	LU	3.35e-14	6.13e-14	1	0.00
6	17	LU	9.06e-15	1.24e-11	1	0.00
6	18	LU	1.14e-14	5.79e-12	1	0.00
7	17	LU	1.22e-13	6.54e-11	1	0.00
7	18	LU	1.08e-13	1.08e-12	1	0.00

表 5: 1D Poisson with Robin B.C. and LU Method

eq	N	method	res_max	err_max	iter	cpu_t
5	13	BCR	2.47e-11	2.71e-12	19	0.00
5	14	BCR	9.33e-13	6.55e-13	16	0.00
6	17	BCR	6.88e-14	1.23e-11	16	0.00
6	18	BCR	5.37e-14	5.80e-12	17	0.00
7	17	BCR	2.42e-13	6.54e-11	17	0.00
7	18	BCR	6.03e-13	1.06e-12	18	0.00

表 6: 1D Poisson with Robin B.C. and BCR Method

3.1.3 Neumann Boundary Condition

使用LU分解法與BCR迭代法[11]解一維的Poisson方程式。

邊界值：

- $u_x(-1)$.
- $u_x(1)$.

case	RHS function of PDE	exact solution
1	0.0	1.0
2	1.0	x
3	2.0	x^2
4	$6.0 * x$	x^3
5	$exp(x)$	$exp(x)$
6	$-\pi^2 * sin(x)$	$sin(\pi * x)$
7	$-\pi^2 * cos(x)$	$cos(\pi * x)$

表 7: 1D Poisson with Neumann B.C.

eq	N	method	res_max	err_max	iter	cpu_t
1	4	LU	0.00e+00	0.00e+00	1	0.00
1	5	LU	0.00e+00	0.00e+00	1	0.00
2	4	LU	1.05e-15	3.72e-15	1	0.00
2	5	LU	6.02e-15	9.77e-15	1	0.00
3	6	LU	5.63e-16	1.33e-15	1	0.00
3	7	LU	7.69e-16	5.52e-16	1	0.00
4	6	LU	2.16e-15	7.73e-15	1	0.00
4	7	LU	2.63e-14	1.60e-14	1	0.00
5	13	LU	1.66e-13	1.40e-13	1	0.00
5	14	LU	1.20e-13	6.83e-15	1	0.00
6	17	LU	6.88e-14	1.23e-11	1	0.00
6	18	LU	1.12e-13	5.61e-12	1	0.00
7	17	LU	5.29e-10	3.67e-10	1	0.00
7	18	LU	8.01e-12	1.43e-11	1	0.00

表 8: 1D Poisson with Neumann B.C. and LU Method

eq	N	method	res_max	err_max	iter	cpu_t
5	13	BCR	1.35e-12	3.49e-13	21	0.00
5	14	BCR	1.09e-13	2.58e-14	16	0.00
6	17	BCR	4.94e-14	1.24e-11	15	0.00
6	18	BCR	3.16e-14	5.79e-12	16	0.00
7	17	BCR	9.64e-10	1.21e-11	174	0.00
7	18	BCR	7.67e-12	4.04e-14	32	0.00

表 9: 1D Poisson with Neumann B.C. and BCR Method

3.2 2D Poisson equation in a square

二維的Poisson方程式： $u_{xx} + u_{yy} = f$, $-1 \leq x \leq 1$, $-1 \leq y \leq 1$.

設定基本參數：

N : x 方向與 y 方向的自由度

$iter$: 迭代次數總和

res_max : 最大 *residual norm*

err_max : 最大誤差

cpu_time : 測試所花的全部時間

由表10至表18可知，無論何種邊界條件，頻譜配點法都只用到少數的自由度，即可快速且準確的逼近到正確答案。

3.2.1 Dirichlet Boundary Condition

使用LU分解法與BCR迭代法[11]解二維的Poisson方程式。

邊界值：

- $u(-1, y), \quad -1 < y < 1.$
- $u(1, y), \quad -1 < y < 1.$
- $u(x, -1), \quad -1 \leq x \leq 1.$
- $u(x, 1), \quad -1 \leq x \leq 1.$

case	RHS function of PDE	exact solution
1	0.0	1.0
2	0.0	x
3	0.0	y
4	0.0	$x + y$
5	0.0	$x * y + x + y$
6	4.0	$x^2 + y^2 + x * y$
7	$6.0 * x + 6.0 * y$	$x^3 + y^3 + x * y$
8	$2.0 * exp(x + y)$	$exp(x + y)$
9	$-2.0 * \pi^2 * sin(\pi * x) * sin(\pi * y)$	$sin(\pi * x) * sin(\pi * y)$
10	$-2.0 * \pi^2 * cos(\pi * x) * cos(\pi * y)$	$cos(\pi * x) * cos(\pi * y)$

表 10: 2D Poisson in a square with Dirichlet B.C.

eq	N	method	res_max	err_max	iter	cpu_t
1	7	LU	8.53e-15	1.22e-15	1	0.00
1	8	LU	1.22e-14	1.22e-15	1	0.00
2	7	LU	6.28e-15	8.88e-16	1	0.00
2	8	LU	9.36e-15	7.77e-16	1	0.00
3	7	LU	6.97e-15	9.99e-16	1	0.00
3	8	LU	9.32e-15	8.88e-16	1	0.00
4	7	LU	6.28e-15	2.00e-15	1	0.00
4	8	LU	8.50e-15	1.78e-15	1	0.00
5	7	LU	7.87e-15	1.33e-15	1	0.01
5	8	LU	1.59e-14	2.22e-15	1	0.00
6	7	LU	8.83e-15	2.66e-15	1	0.00
6	8	LU	1.47e-14	2.22e-15	1	0.00
7	7	LU	7.33e-15	1.11e-15	1	0.00
7	8	LU	1.30e-14	1.78e-15	1	0.00
8	12	LU	1.33e-13	5.60e-14	1	0.01
8	13	LU	1.60e-13	8.88e-15	1	0.02
9	15	LU	1.31e-14	9.99e-12	1	0.03
9	16	LU	1.20e-14	2.68e-12	1	0.05
10	17	LU	1.50e-13	2.88e-13	1	0.09
10	18	LU	1.72e-13	6.44e-15	1	0.13

表 11: 2D Poisson in a square with Dirichlet B.C. and LU Method

eq	N	method	res_max	err_max	iter	cpu_t
8	12	BCR	7.61e-13	5.77e-14	27	0.04
8	13	BCR	8.48e-13	1.27e-14	29	0.06
9	15	BCR	3.35e-13	9.99e-12	26	0.11
9	16	BCR	6.46e-14	2.68e-12	26	0.14
10	17	BCR	6.82e-13	2.88e-13	30	0.20
10	18	BCR	7.83e-13	6.66e-15	31	0.26

表 12: 2D Poisson in a square with Dirichlet B.C. and BCR Method

3.2.2 Robin Boundary Condition

使用LU分解法與BCR迭代法[10]解二維的Poisson方程式。

邊界值：

- $u(-1, y), \quad -1 < y < 1.$
- $u_y(1, y), \quad -1 \leq y < 1.$
- $u(x, -1), \quad -1 \leq x < 1.$
- $u_x(x, 1), \quad -1 \leq x \leq 1.$

case	RHS function of PDE	exact solution
1	0.0	1.0
2	0.0	x
3	0.0	y
4	1.0	$\frac{(x-1.0)^2}{2.0}$
5	$-2.0 * x + 3.0$	$\frac{-(x-1.0)^3}{3.0} + \frac{(x-1.0)^2}{2.0}$
6	$12.0 * x^2 - 4.0 + 12.0 * y^2 - 4.0$	$(x + 1.0)^2 * (1.0 - x)^2$ $+(y + 1.0)^2 * (1.0 - y)^2$
7	$2.0 * \pi^2 *$ $(\cos(2.0 * \pi * x) * \sin(\pi * y)^2$ $+ \cos(2.0 * \pi * y) * \sin(\pi * x)^2)$	$\sin(\pi * x)^2 * \sin(\pi * y)^2$
8	$-0.5 * \pi^2 * \sin(\frac{\pi * x}{2.0}) * \sin(\frac{\pi * y}{2.0})$	$\sin(\frac{\pi * x}{2.0}) * \sin(\frac{\pi * y}{2.0})$

表 13: 2D Poisson in a square with Robin B.C.

eq	N	method	res_max	err_max	iter	cpu_t
1	5	LU	3.28e-15	2.33e-15	1	0.00
1	6	LU	3.65e-15	6.22e-15	1	0.00
2	5	LU	2.22e-15	8.88e-16	1	0.00
2	6	LU	3.21e-15	5.66e-15	1	0.00
3	5	LU	1.85e-15	2.00e-15	1	0.00
3	6	LU	3.90e-15	5.22e-15	1	0.00
4	5	LU	2.17e-15	1.72e-15	1	0.00
4	6	LU	4.86e-15	1.33e-15	1	0.00
5	5	LU	3.59e-15	3.86e-15	1	0.00
5	6	LU	1.19e-14	3.30e-15	1	0.00
6	5	LU	1.76e-15	9.72e-16	1	0.00
6	6	LU	2.97e-15	6.74e-15	1	0.00
7	17	LU	9.42e-15	3.26e-06	1	0.00
7	18	LU	8.95e-15	2.21e-07	1	0.00
8	17	LU	1.91e-13	2.92e-14	1	0.00
8	18	LU	1.72e-13	6.81e-14	1	0.00

表 14: 2D Poisson in a square with Robin B.C. and LU Method

eq	N	method	res_max	err_max	iter	cpu_t
7	20	BCR	3.29e-13	5.42e-09	75	0.91
7	21	BCR	1.38e-13	2.52e-09	80	1.23
8	20	BCR	1.84e-12	4.45e-14	91	1.12
8	21	BCR	2.83e-12	3.66e-14	105	1.53

表 15: 2D Poisson in a square with Robin B.C. and BCR Method

3.2.3 Neumann Boundary Condition

使用LU分解法與BCR迭代法[10]解二維的Poisson方程式。

邊界值：

- $u_y(-1, y), \quad -1 < y < 1.$
- $u_y(1, y), \quad -1 < y < 1.$
- $u_x(x, -1), \quad -1 \leq x \leq 1.$
- $u_x(x, 1), \quad -1 \leq x \leq 1.$

case	RHS function of PDE	exact solution
1	0.0	1.0
2	0.0	x
3	0.0	y
4	$12.0 * x^2 - 4.0$	$(x - 1.0)^2 + (x + 1.0)^2$
5	$12.0 * y^2 - 4.0$	$(y - 1.0)^2 + (y + 1.0)^2$
6	$2.0 * \pi^2 * (\cos(2.0 * \pi * x) * \sin^2(\pi * y) + \cos(2.0 * \pi * y) * \sin^2(\pi * x))$	$\sin^2(\pi * x) * \sin^2(\pi * y)$
7	$-2.0 * \pi^2 * \cos(\pi * x) * \cos(\pi * y)$	$\cos(\pi * x) * \cos(\pi * y)$

表 16: 2D Poisson in a square with Neumann B.C.

eq	N	method	res_max	err_max	iter	cpu_t
1	5	LU	0.00e+00	0.00e+00	1	0.00
1	6	LU	0.00e+00	0.00e+00	1	0.00
2	5	LU	4.27e-15	3.36e-15	1	0.00
2	6	LU	4.88e-15	8.55e-15	1	0.00
3	5	LU	4.32e-15	3.14e-15	1	0.00
3	6	LU	5.14e-15	9.12e-15	1	0.00
4	5	LU	1.57e-15	1.90e-15	1	0.00
4	6	LU	2.26e-15	9.99e-16	1	0.00
5	5	LU	6.86e-16	9.71e-16	1	0.00
5	6	LU	1.84e-15	9.44e-16	1	0.00
6	21	LU	1.30e-08	1.97e-09	1	0.38
6	22	LU	3.89e-10	1.67e-10	1	0.53
7	21	LU	2.73e-13	4.90e-14	1	0.40
7	22	LU	3.29e-13	9.57e-14	1	0.53

表 17: 2D Poisson in a square with Neumann B.C. and LU Method

eq	N	method	res_max	err_max	iter	cpu_t
6	21	BCR	5.56e-08	4.90e-09	500	7.30
6	22	BCR	8.04e-08	4.01e-09	500	8.54
7	21	BCR	3.57e-11	2.81e-12	500	7.17
7	22	BCR	3.45e-12	1.28e-13	165	2.83

表 18: 2D Poisson in a square with Neumann B.C. and BCR Method

3.3 2D Poisson equation in a circle

二維的Poisson方程式： $u_{rr} + r^{-1}u_r + r^{-2}u_{\theta\theta} = f$,

$$0 < r \leq 1, \quad 0 < \theta \leq 2\pi.$$

$$u = u(r, \theta)$$

設定基本參數：

N : r 方向之自由度，取奇數

M : θ 方向之自由度，取偶數

$iter$: 迭代次數總和

res_max : 最大 *residual norm*

err_max : 最大誤差

cpu_time : 測試所花之全部時間

由表19至表27可知，無論何種邊界條件，頻譜配點法都只用到少數之自由度，即可快速且準確之逼近到正確答案。

3.3.1 Dirichlet Boundary Condition

使用LU分解法與BCR迭代法[11]解二維的Poisson方程式。

邊界值：

$$\cdot u(1, \theta), \quad 0 < \theta \leq 2\pi.$$

case	RHS function of PDE	exact solution
1	0.0	1.0
2	0.0	$r * \cos(t)$
3	0.0	$r * \sin(t)$
4	4.0	$r^2 * \cos(t)^2 + r^2 * \sin(t)^2$
5	$2.0 * \exp(r * \cos(t) + r * \sin(t))$	$\exp(r * \cos(t) + r * \sin(t))$
6	$4.0 * r^2 - 2.0$	$\frac{r^4}{4.0} - \frac{r^2}{2.0}$
7	$4.0 - \frac{2.0}{r}$	$r^2 - 2.0 * r$

表 19: 2D Poisson in a circle with Dirichlet B.C.

eq	N	M	method	res_max	err_max	iter	cpu_t
1	3	8	LU	1.42e-15	1.22e-15	1	0.00
1	3	10	LU	3.05e-15	2.00e-15	1	0.00
1	5	8	LU	4.18e-15	1.78e-15	1	0.00
1	5	10	LU	5.27e-15	2.89e-15	1	0.00
2	3	8	LU	8.42e-16	4.44e-16	1	0.00
2	3	10	LU	7.57e-16	4.44e-16	1	0.00
2	5	8	LU	1.89e-15	6.11e-16	1	0.00
2	5	10	LU	2.14e-15	5.55e-16	1	0.00
3	3	8	LU	6.53e-16	2.78e-16	1	0.00
3	3	10	LU	8.89e-16	3.89e-16	1	0.00
3	5	8	LU	1.71e-15	4.44e-16	1	0.00
3	5	10	LU	1.87e-15	2.78e-16	1	0.00
4	3	8	LU	6.08e-16	3.05e-16	1	0.00
4	3	10	LU	7.82e-16	5.83e-16	1	0.00
4	5	8	LU	2.32e-15	6.66e-16	1	0.00
4	5	10	LU	1.88e-15	4.44e-16	1	0.00
5	3	8	LU	1.11e-16	4.69e-02	1	0.00
5	3	10	LU	2.29e-16	4.69e-02	1	0.00
5	5	8	LU	5.61e-16	1.94e-16	1	0.00
5	5	10	LU	6.42e-16	1.39e-16	1	0.00
6	11	22	LU	1.02e-13	1.02e-10	1	0.00
6	11	24	LU	1.22e-13	1.02e-10	1	0.00
6	13	22	LU	1.63e-13	4.66e-12	1	0.00
6	13	24	LU	1.55e-13	2.31e-13	1	0.01

表 20: 2D Poisson in a circle with Dirichlet B.C. and LU Method

eq	N	M	method	res_max	err_max	iter	cpu_t
6	11	18	BCR	5.62e-13	1.45e-09	34	0.02
6	11	20	BCR	3.02e-13	1.01e-10	34	0.02
6	11	22	BCR	3.68e-13	1.02e-10	36	0.03
6	13	18	BCR	7.24e-13	1.61e-09	33	0.03
6	13	20	BCR	8.75e-13	8.40e-11	35	0.04
6	13	22	BCR	5.76e-13	4.66e-12	36	0.04

表 21: 2D Poisson in a circle with Dirichlet B.C. and BCR Method

3.3.2 Robin Boundary Condition

使用LU分解法與BCR迭代法[11]解二維的Poisson方程式。

邊界值：

- $u_r(1, \theta), \quad 0 < \theta \leq \pi.$
- $u(1, \theta), \quad \pi < \theta \leq 2\pi.$

case	RHS function of PDE	exact solution
1	0.0	1.0
2	0.0	$r * \cos(t)$
3	0.0	$r * \sin(t)$
4	4.0	$r^2 * \cos^2(t) + r^2 * \sin^2(t)$
5	$4.0 * r^2 - 2.0$	$\frac{r^4}{4} - \frac{r^2}{2}$
6	$2.0 * \exp(r * \cos(t) + r * \sin(t))$	$\exp(r * \cos(t) + r * \sin(t))$

表 22: 2D Poisson in a circle with Robin B.C.

eq	N	M	method	res_max	err_max	iter	cpu_t
1	3	8	LU	1.36e-15	4.77e-15	1	0.00
1	3	10	LU	2.37e-15	7.55e-15	1	0.00
1	5	8	LU	3.25e-15	1.08e-14	1	0.00
1	5	10	LU	5.42e-15	2.55e-15	1	0.00
2	3	8	LU	4.86e-16	1.33e-15	1	0.00
2	3	10	LU	1.09e-15	1.44e-15	1	0.00
2	5	8	LU	1.58e-15	1.78e-15	1	0.00
2	5	10	LU	2.12e-15	1.89e-15	1	0.00
3	3	8	LU	3.63e-16	2.22e-15	1	0.00
3	3	10	LU	8.10e-16	1.11e-15	1	0.00
3	5	8	LU	1.41e-15	3.77e-15	1	0.00
3	5	10	LU	1.45e-15	2.89e-15	1	0.00
4	3	8	LU	7.45e-16	1.33e-15	1	0.00
4	3	10	LU	7.90e-16	1.80e-15	1	0.00
4	5	8	LU	1.44e-15	6.88e-15	1	0.00
4	5	10	LU	1.94e-15	4.44e-15	1	0.00
5	3	8	LU	4.78e-16	9.69e-01	1	0.00
5	3	10	LU	7.45e-16	9.81e-01	1	0.00
5	5	8	LU	5.09e-16	2.16e-15	1	0.00
5	5	10	LU	6.13e-16	1.14e-15	1	0.00
6	11	22	LU	9.27e-14	1.38e-09	1	0.00
6	11	24	LU	1.08e-13	1.34e-09	1	0.00
6	13	22	LU	1.78e-13	2.42e-11	1	0.00
6	13	24	LU	1.66e-13	4.47e-12	1	0.00

表 23: 2D Poisson in a circle with Robin B.C. and LU Method

eq	N	M	method	res_max	err_max	iter	cpu_t
6	11	18	BCR	1.40e-11	6.24e-09	61	0.04
6	11	20	BCR	1.17e-12	1.36e-09	58	0.04
6	11	22	BCR	1.27e-12	1.38e-09	60	0.05
6	13	18	BCR	1.05e-12	5.34e-09	57	0.05
6	13	20	BCR	4.18e-12	3.09e-10	64	0.06
6	13	22	BCR	3.03e-11	2.39e-11	78	0.09

表 24: 2D Poisson in a circle with Robin B.C. and BCR Method

3.3.3 Neumann Boundary Condition

使用LU分解法與BCR迭代法[11]解二維的Poisson方程式。

邊界值：

$$\cdot u_r(1, \theta), \quad 0 < \theta \leq 2\pi.$$

case	RHS function of PDE	exact solution
1	0.0	1.0
2	0.0	$r * \cos(t)$
3	0.0	$r * \sin(t)$
4	4.0	$r^2 * \cos^2(t) + r^2 * \sin^2(t)$
5	$4.0 * r^2 - 2.0$	$\frac{r^4}{4} - \frac{r^2}{2}$
6	$2.0 * \exp(r * \cos(t) + r * \sin(t))$	$\exp(r * \cos(t) + r * \sin(t))$

表 25: 2D Poisson in a circle with Neumann B.C.

eq	N	M	method	res_max	err_max	iter	cpu_t
1	3	8	LU	0.00e+00	0.00e+00	1	0.00
1	3	10	LU	0.00e+00	0.00e+00	1	0.00
1	5	8	LU	0.00e+00	0.00e+00	1	0.00
1	5	10	LU	0.00e+00	0.00e+00	1	0.00
2	3	8	LU	8.14e-16	2.89e-15	1	0.00
2	3	10	LU	1.19e-15	3.11e-15	1	0.00
2	5	8	LU	1.72e-15	4.11e-15	1	0.00
2	5	10	LU	4.39e-15	2.73e-15	1	0.00
3	3	8	LU	5.11e-16	2.00e-15	1	0.00
3	3	10	LU	7.53e-16	1.55e-15	1	0.00
3	5	8	LU	1.70e-15	3.44e-15	1	0.00
3	5	10	LU	5.14e-15	3.61e-15	1	0.00
4	3	8	LU	5.44e-16	1.11e-16	1	0.00
4	3	10	LU	3.85e-16	1.66e-16	1	0.00
4	5	8	LU	2.04e-15	8.05e-16	1	0.00
4	5	10	LU	1.14e-14	1.05e-14	1	0.00
5	3	8	LU	3.53e-01	7.81e-01	1	0.00
5	3	10	LU	2.43e-01	9.84e-01	1	0.00
5	5	8	LU	5.17e-16	2.43e-16	1	0.00
5	5	10	LU	2.25e-15	2.76e-15	1	0.00
6	11	22	LU	5.61e-09	9.63e-10	1	0.00
6	11	24	LU	4.79e-10	9.27e-10	1	0.00
6	13	22	LU	2.50e-12	3.00e-11	1	0.00
6	13	24	LU	6.32e-12	5.45e-12	1	0.00

表 26: 2D Poisson in a circle with Neumann B.C. and LU Method

eq	N	M	method	res_max	err_max	iter	cpu_t
6	11	18	BCR	1.35e-06	4.76e-08	500	0.29
6	11	20	BCR	1.81e-05	1.46e-07	500	0.35
6	11	22	BCR	1.45e-07	6.41e-09	500	0.43
6	13	18	BCR	3.38e-11	6.48e-09	82	0.06
6	13	20	BCR	2.05e-09	4.21e-10	500	0.48
6	13	22	BCR	3.17e-08	3.25e-10	500	0.58

表 27: 2D Poisson in a circle with Neumann B.C. and BCR Method

3.4 3D Poisson equation in a cube

三維的Poisson方程式： $u_{xx} + u_{yy} + u_{zz} = f$,

$$-1 \leq x \leq 1, \quad -1 \leq y \leq 1, \quad -1 \leq z \leq 1.$$

設定基本參數：

N : x 方向與 y 方向與 z 方向的自由度

$iter$: 迭代次數總和

res_max : 最大 *residual norm*

err_max : 最大誤差

cpu_time : 測試所花的全部時間

由表28至表36可知，無論何種邊界條件，頻譜配點法都只用到少數的自由度，即可快速且準確的逼近到正確答案。

3.4.1 Dirichlet Boundary Condition

使用LU分解法與BCR迭代法[11]解三維的Poisson方程式。

邊界值：

- $u(-1, y, z), \quad -1 < y < 1, \quad -1 \leq z \leq 1.$
- $u(1, y, z), \quad -1 < y < 1, \quad -1 \leq z \leq 1.$
- $u(x, -1, z), \quad -1 \leq x \leq 1, \quad -1 \leq z \leq 1.$
- $u(x, 1, z), \quad -1 \leq x \leq 1, \quad -1 \leq z \leq 1.$
- $u(x, y, -1), \quad -1 < x < 1, \quad -1 < y < 1.$
- $u(x, y, 1), \quad -1 < x < 1, \quad -1 < y < 1.$

case	RHS function of PDE	exact solution
1	0.0	1.0
2	0.0	x
3	0.0	y
4	0.0	z
5	0.0	$x + y + z$
6	0.0	$x * y * z + x + y + z$
7	$6.0 * x + 6.0 * y + 6.0 * z$	$x^3 + y^3 + z^3 + x * y * z$
8	$3.0 * \exp(x + y + z)$	$\exp(x + y + z)$
9	$3.0 * \pi^2 * \sin(\pi * x) * \sin(\pi * y) * \sin(\pi * z)$	$\sin(\pi * x) * \sin(\pi * y) * \sin(\pi * z)$
10	$3.0 * \pi^2 * \cos(\pi * x) * \cos(\pi * y) * \cos(\pi * z)$	$\cos(\pi * x) * \cos(\pi * y) * \cos(\pi * z)$

表 28: 3D Poisson in a cube with Dirichlet B.C.

eq	N	method	res_max	err_max	iter	cpu_t
1	6	LU	6.55e-15	1.11e-15	1	0.10
1	7	LU	1.15e-14	1.55e-15	1	0.47
2	7	LU	7.17e-15	1.22e-15	1	0.46
2	8	LU	1.18e-14	8.88e-16	1	1.36
3	7	LU	7.21e-15	1.22e-15	1	0.47
3	8	LU	1.45e-14	9.99e-16	1	1.40
4	7	LU	7.84e-15	1.11e-15	1	0.52
4	8	LU	1.45e-14	8.88e-16	1	1.53
5	6	LU	5.42e-15	1.78e-15	1	0.13
5	7	LU	1.21e-14	3.55e-15	1	0.56
6	6	LU	7.10e-15	2.22e-15	1	0.10
6	7	LU	1.64e-14	4.00e-15	1	0.46
7	6	LU	8.20e-15	1.78e-15	1	0.10
7	7	LU	8.92e-15	3.55e-15	1	0.46
8	12	LU	3.55e-13	1.63e-13	1	45.80
8	13	LU	3.84e-13	2.13e-14	1	87.80
9	12	LU	7.25e-15	3.23e-08	1	39.89
9	13	LU	7.65e-15	1.51e-09	1	86.39
10	12	LU	4.49e-14	2.51e-08	1	40.21
10	13	LU	6.15e-14	4.70e-09	1	94.58

表 29: 3D Poisson in a cube with Dirichlet B.C. and LU Method

eq	N	method	res_max	err_max	iter	cpu_t
8	12	BCR	1.37e-12	1.67e-13	27	8.17
8	13	BCR	1.87e-12	2.13e-14	28	13.14
9	12	BCR	3.05e-14	3.23e-08	25	7.61
9	13	BCR	5.34e-14	1.51e-09	25	11.79
10	12	BCR	1.74e-13	2.51e-08	27	8.16
10	13	BCR	2.70e-13	4.70e-09	26	12.36

表 30: 3D Poisson in a cube with Dirichlet B.C. and BCR Method



3.4.2 Robin Boundary Condition

使用LU分解法與BCR迭代法[11]解三維的Poisson方程式。

邊界值：

- $u(-1, y, z), \quad -1 < y < 1, \quad -1 \leq z \leq 1.$
- $u_y(1, y, z), \quad -1 \leq y < 1, \quad -1 \leq z \leq 1.$
- $u(x, -1, z), \quad -1 \leq x < 1, \quad -1 \leq z \leq 1.$
- $u_x(x, 1, z), \quad -1 \leq x \leq 1, \quad -1 \leq z \leq 1.$
- $u(x, y, -1), \quad -1 < x < 1, \quad -1 < y < 1.$
- $u_z(x, y, 1), \quad -1 < x < 1, \quad -1 < y < 1.$

case	RHS function of PDE	exact solution
1	0.0	1.0
2	0.0	x
3	0.0	y
4	0.0	z
5	$-2.0 * x + 3.0$	$-\frac{(x-1.0)^3}{3.0} + \frac{(x-1.0)^2}{2.0}$
6	$12.0 * x^2 + 12.0 * y^2 + 12.0 * z^2 - 12.0$	$(x + 1.0)^2 * (1.0 - x)^2$ $+(y + 1.0)^2 * (1.0 - y)^2$ $+(z + 1.0)^2 * (1.0 - z)^2$
7	$2.0 * \pi^2 *$ $(\cos(2.0 * \pi * x) * \sin^2(\pi * y) * \sin^2(\pi * z)$ $+ \cos(2.0 * \pi * y) * \sin^2(\pi * x) * \sin^2(\pi * z)$ $+ \cos(2.0 * \pi * z) * \sin^2(\pi * x) * \sin^2(\pi * y))$	$\sin^2(\pi * x)$ $* \sin^2(\pi * y)$ $* \sin^2(\pi * z)$
8	$-\frac{3}{4} * \pi^2 * \sin(\frac{\pi * x}{2.0}) * \sin(\frac{\pi * y}{2.0}) * \sin(\frac{\pi * z}{2.0})$	$\sin(\frac{\pi * x}{2.0}) * \sin(\frac{\pi * y}{2.0}) * \sin(\frac{\pi * z}{2.0})$

表 31: 3D Poisson in a cube with Robin B.C.

eq	N	method	res_max	err_max	iter	cpu_t
1	5	LU	3.88e-15	1.33e-15	1	0.02
1	6	LU	5.95e-15	6.22e-15	1	0.11
2	5	LU	2.60e-15	3.11e-15	1	0.02
2	6	LU	4.15e-15	4.00e-15	1	0.11
3	5	LU	2.51e-15	1.11e-15	1	0.02
3	6	LU	4.27e-15	4.55e-15	1	0.10
4	5	LU	2.45e-15	2.89e-15	1	0.01
4	6	LU	5.00e-15	4.22e-15	1	0.10
5	5	LU	6.68e-15	4.62e-15	1	0.02
5	6	LU	1.44e-14	5.77e-15	1	0.12
6	6	LU	7.85e-15	1.09e-14	1	0.10
6	7	LU	1.17e-14	3.00e-15	1	0.48
7	15	LU	5.84e-15	4.91e-05	1	748.74
7	16	LU	6.57e-15	4.34e-06	1	2201.47
8	12	LU	5.86e-14	5.08e-10	1	40.24
8	13	LU	7.24e-14	3.78e-12	1	95.43

表 32: 3D Poisson in a cube with Robin B.C. and LU Method

eq	N	method	res_max	err_max	iter	cpu_t
7	15	BCR	2.95e-10	4.91e-05	80	83.55
7	16	BCR	1.22e-09	4.34e-06	80	126.58
8	12	BCR	6.29e-12	5.08e-10	80	23.41
8	13	BCR	8.26e-10	6.15e-10	80	36.52

表 33: 3D Poisson in a cube with Robin B.C. and BCR Method

3.4.3 Neumann Boundary Condition

使用LU分解法與BCR迭代法[11]解三維的Poisson方程式。

邊界值：

- $u_y(-1, y, z), \quad -1 < y < 1, \quad -1 \leq z \leq 1.$
- $u_y(1, y, z), \quad -1 < y < 1, \quad -1 \leq z \leq 1.$
- $u_x(x, -1, z), \quad -1 \leq x \leq 1, \quad -1 \leq z \leq 1.$
- $u_x(x, 1, z), \quad -1 \leq x \leq 1, \quad -1 \leq z \leq 1.$
- $u_z(x, y, -1), \quad -1 < x < 1, \quad -1 < y < 1.$
- $u_z(x, y, 1), \quad -1 < x < 1, \quad -1 < y < 1.$

case	RHS function of PDE	exact solution
1	0.0	1.0
2	0.0	x
3	0.0	y
4	0.0	z
5	$12.0 * x^2 - 4.0$	$(x - 1.0)^2 * (x + 1.0)^2$
6	$12.0 * y^2 - 4.0$	$(y - 1.0)^2 * (y + 1.0)^2$
7	$12.0 * z^2 - 4.0$	$(z - 1.0)^2 * (z + 1.0)^2$
8	$2.0 * \pi^2 *$ $(\cos(2.0 * \pi * x) * \sin^2(\pi * y) * \sin^2(\pi * z)$ $+ \cos(2.0 * \pi * y) * \sin^2(\pi * x) * \sin^2(\pi * z)$ $+ \cos(2.0 * \pi * z) * \sin^2(\pi * x) * \sin^2(\pi * y))$	$\sin^2(\pi * x)$ $* \sin^2(\pi * y)$ $* \sin^2(\pi * z)$
9	$-3.0 * \pi^2$ $* \cos(\pi * x) * \cos(\pi * y) * \cos(\pi * z)$	$\cos(\pi * x)$ $* \cos(\pi * y)$ $* \cos(\pi * z)$

表 34: 3D Poisson in a cube with Neumann B.C.

eq	N	method	res_max	err_max	iter	cpu_t
1	6	LU	0.00e+00	0.00e+00	1	0.11
1	7	LU	0.00e+00	0.00e+00	1	0.47
2	6	LU	6.37e-15	1.24e-14	1	0.10
2	7	LU	1.50e-14	9.05e-15	1	0.46
3	6	LU	6.64e-15	1.09e-14	1	0.10
3	7	LU	1.73e-14	8.94e-15	1	0.47
4	6	LU	5.89e-15	5.33e-15	1	0.11
4	7	LU	1.88e-14	1.18e-14	1	0.50
5	6	LU	2.63e-15	1.55e-15	1	0.11
5	7	LU	3.94e-15	2.00e-15	1	0.48
6	6	LU	2.59e-15	2.00e-15	1	0.10
6	7	LU	3.73e-15	1.83e-15	1	0.46
7	6	LU	2.73e-15	2.89e-15	1	0.11
7	7	LU	6.40e-15	2.78e-15	1	0.46
8	15	LU	1.78e-04	1.81e-04	1	931.16
8	16	LU	6.05e-05	8.53e-06	1	2494.20
9	11	LU	3.12e-14	1.99e-05	1	18.78
9	12	LU	4.67e-14	7.51e-07	1	44.11

表 35: 3D Poisson in a cube with Neumann B.C. and LU Method

eq	N	method	res_max	err_max	iter	cpu_t
8	15	BCR	2.08e-04	1.82e-05	80	80.80
8	16	BCR	2.03e-05	1.53e-06	80	114.24
9	11	BCR	9.52e-11	1.99e-05	80	14.38
9	12	BCR	1.13e-09	7.51e-07	80	23.29

表 36: 3D Poisson 3D in a cube with Neumann B.C. and BCR Method

3.5 3D Poisson equation in a cylinder

三維的Poisson方程式： $u_{rrr} + r^{-1}u_r + r^{-2}u_{\theta\theta} + u_{zz} = f$,

$$0 < r \leq 1, \quad 0 < \theta \leq 2\pi, \quad -1 \leq z \leq 1.$$

$$u = u(r, \theta, z)$$

設定基本參數：

N : r 方向의 自由度，取奇數

M : θ 方向의 自由度，取偶數

H : z 方向의 自由度

$iter$: 迭代次數總和

res_max : 最大 *residual norm*

err_max : 最大誤差

cpu_time : 測試所花의 全部時間

由表37至表51可知，無論何種邊界條件，頻譜配點法都只用到少數의 自由度，即可快速且準確的逼近到正確答案。

3.5.1 Dirichlet Boundary Condition

使用LU分解法與BCR迭代法[11]解三維的Poisson方程式。

邊界值：

- $u(1, \theta, z), \quad 0 < \theta \leq 2\pi, \quad -1 \leq z \leq 1.$
- $u(r, \theta, -1), \quad 0 < \theta \leq 2\pi, \quad 0 < r < 1.$
- $u(r, \theta, 1), \quad 0 < \theta \leq 2\pi, \quad 0 < r < 1.$

case	RHS function of PDE	exact solution
1	0.0	1.0
2	0.0	$r * \cos(t)$
3	0.0	$r * \sin(t)$
4	0.0	z
5	4.0	$r^2 * \cos^2(t) + r^2 * \sin^2(t)$
6	$4.0 * r^2 - 2.0$	$\frac{r^4}{4.0} - \frac{r^2}{2.0}$
7	$8.0 * r * \cos(t)$	$(r^2 * \cos^2(t) + r^2 * \sin^2(t)) * r * \cos(t)$
8	$8.0 * r * \sin(t)$	$(r^2 * \cos^2(t) + r^2 * \sin^2(t)) * r * \sin(t)$

表 37: 3D Poisson in a cylinder with Dirichlet B.C.

eq	N	M	H	method	res_max	err_max	iter	cpu_t
1	3	6	5	LU	2.37e-15	2.00e-15	1	0.00
1	3	8	5	LU	2.66e-15	1.22e-15	1	0.00
1	5	6	5	LU	3.85e-15	2.44e-15	1	0.00
1	5	8	5	LU	4.85e-15	2.11e-15	1	0.00
1	3	6	6	LU	2.13e-15	1.55e-15	1	0.00
1	3	8	6	LU	4.50e-15	1.11e-15	1	0.00
1	5	6	6	LU	6.80e-15	2.66e-15	1	0.00
1	5	8	6	LU	5.63e-15	2.44e-15	1	0.01
2	3	6	5	LU	6.44e-16	6.66e-16	1	0.00
2	3	8	5	LU	9.42e-16	3.89e-16	1	0.00
2	5	6	5	LU	1.89e-15	7.49e-16	1	0.00
2	5	8	5	LU	1.79e-15	6.66e-16	1	0.00
2	3	6	6	LU	1.04e-15	6.66e-16	1	0.00
2	3	8	6	LU	1.09e-15	3.89e-16	1	0.00
2	5	6	6	LU	1.83e-15	8.60e-16	1	0.00
2	5	8	6	LU	2.29e-15	6.66e-16	1	0.01
3	3	6	5	LU	7.67e-16	4.44e-16	1	0.00
3	3	8	5	LU	9.95e-16	2.78e-16	1	0.00
3	5	6	5	LU	1.70e-15	6.16e-16	1	0.00
3	5	8	5	LU	2.34e-15	6.66e-16	1	0.00
3	3	6	6	LU	1.13e-15	4.38e-16	1	0.00
3	3	8	6	LU	1.02e-15	2.22e-16	1	0.00
3	5	6	6	LU	2.00e-15	5.75e-16	1	0.00
3	5	8	6	LU	2.31e-15	5.55e-16	1	0.01

表 38: 3D Poisson in a cylinder with Dirichlet B.C. (1) and LU Method

eq	N	M	H	method	res_max	err_max	iter	cpu_t
4	3	6	5	LU	1.55e-15	8.88e-16	1	0.00
4	3	8	5	LU	2.53e-15	5.55e-16	1	0.00
4	5	6	5	LU	3.06e-15	1.33e-15	1	0.00
4	5	8	5	LU	3.29e-15	6.66e-16	1	0.00
4	3	6	6	LU	2.37e-15	7.77e-16	1	0.00
4	3	8	6	LU	2.51e-15	5.55e-16	1	0.00
4	5	6	6	LU	4.02e-15	9.99e-16	1	0.00
4	5	8	6	LU	3.86e-15	6.66e-16	1	0.01
5	3	6	5	LU	5.94e-16	5.83e-16	1	0.00
5	3	8	5	LU	8.85e-16	3.05e-16	1	0.00
5	5	6	5	LU	1.85e-15	4.44e-16	1	0.00
5	5	8	5	LU	2.52e-15	6.66e-16	1	0.00
5	3	6	6	LU	1.06e-15	4.72e-16	1	0.00
5	3	8	6	LU	1.05e-15	2.78e-16	1	0.00
5	5	6	6	LU	2.32e-15	4.44e-16	1	0.00
5	5	8	6	LU	3.38e-15	6.66e-16	1	0.01
6	3	6	5	LU	2.16e-16	3.52e-02	1	0.00
6	3	8	5	LU	3.41e-16	3.52e-02	1	0.00
6	5	6	5	LU	7.15e-16	1.39e-16	1	0.00
6	5	8	5	LU	8.55e-16	2.22e-16	1	0.00
6	3	6	6	LU	3.45e-16	3.77e-02	1	0.00
6	3	8	6	LU	4.10e-16	3.77e-02	1	0.00
6	5	6	6	LU	1.11e-15	1.39e-16	1	0.00
6	5	8	6	LU	1.09e-15	2.01e-16	1	0.01

表 39: 3D Poisson in a cylinder with Dirichlet B.C. (2) and LU Method

eq	N	M	H	method	res_max	err_max	iter	cpu.t
7	3	6	5	LU	3.84e-16	1.94e-16	1	0.00
7	3	8	5	LU	4.67e-16	1.11e-16	1	0.00
7	5	6	5	LU	1.98e-15	2.22e-16	1	0.00
7	5	8	5	LU	1.15e-15	4.44e-16	1	0.00
7	3	6	6	LU	3.92e-16	1.73e-16	1	0.00
7	3	8	6	LU	5.00e-16	1.11e-16	1	0.00
7	5	6	6	LU	1.98e-15	2.22e-16	1	0.00
7	5	8	6	LU	1.49e-15	3.89e-16	1	0.01
8	3	6	5	LU	2.89e-16	1.25e-16	1	0.00
8	3	8	5	LU	4.36e-16	8.33e-17	1	0.00
8	5	6	5	LU	1.05e-15	3.33e-16	1	0.00
8	5	8	5	LU	1.57e-15	4.44e-16	1	0.00
8	3	6	6	LU	3.83e-16	1.10e-16	1	0.00
8	3	8	6	LU	5.99e-16	6.94e-17	1	0.00
8	5	6	6	LU	1.07e-15	2.78e-16	1	0.00
8	5	8	6	LU	1.54e-15	3.33e-16	1	0.01

表 40: 3D Poisson in a cylinder with Dirichlet B.C. (3) and LU Method

eq	N	M	H	method	res_max	err_max	iter	cpu_t
6	3	6	5	BCR	3.11e-15	3.52e-02	19	0.01
6	3	8	5	BCR	1.75e-15	3.52e-02	22	0.01
6	5	6	5	BCR	5.68e-15	6.66e-16	24	0.01
6	5	8	5	BCR	3.59e-15	4.44e-16	26	0.03
6	3	6	6	BCR	8.01e-16	3.77e-02	22	0.01
6	3	8	6	BCR	3.25e-14	3.77e-02	22	0.02
6	5	6	6	BCR	1.72e-14	8.33e-16	24	0.02
6	5	8	6	BCR	6.00e-15	2.50e-16	26	0.04
7	3	6	5	BCR	1.03e-14	1.06e-15	19	0.01
7	3	8	5	BCR	1.82e-14	1.64e-15	22	0.01
7	5	6	5	BCR	2.17e-14	1.28e-15	23	0.01
7	5	8	5	BCR	4.23e-15	5.55e-16	27	0.03
7	3	6	6	BCR	1.67e-14	9.44e-16	20	0.01
7	3	8	6	BCR	4.11e-15	2.73e-16	22	0.02
7	5	6	6	BCR	8.80e-15	6.66e-16	25	0.02
7	5	8	6	BCR	8.13e-15	9.99e-16	27	0.04
8	3	6	5	BCR	5.73e-15	7.33e-16	19	0.01
8	3	8	5	BCR	1.26e-14	1.40e-15	21	0.01
8	5	6	5	BCR	1.25e-14	8.68e-16	24	0.01
8	5	8	5	BCR	1.41e-14	9.99e-16	26	0.03
8	3	6	6	BCR	6.01e-15	4.30e-16	21	0.01
8	3	8	6	BCR	5.35e-15	3.61e-16	22	0.01
8	5	6	6	BCR	5.58e-15	5.55e-16	25	0.02
8	5	8	6	BCR	1.07e-14	6.14e-16	26	0.04

表 41: 3D Poisson in a cylinder with Dirichlet B.C. and BCR Method

3.5.2 Robin Boundary Condition

使用LU分解法與BCR迭代法[11]解三維的Poisson方程式。

邊界值：

- $u_r(1, \theta, z), \quad 0 < \theta \leq \pi, \quad -1 \leq z \leq 1.$
- $u(1, \theta, z), \quad \pi < \theta \leq 2\pi, \quad -1 \leq z \leq 1.$
- $u_z(r, \theta, -1), \quad 0 < \theta \leq 2\pi, \quad 0 < r < 1.$
- $u_z(r, \theta, 1), \quad 0 < \theta \leq 2\pi, \quad 0 < r < 1.$

case	RHS function of PDE	exact solution
1	0.0	1.0
2	0.0	$r * \cos(t)$
3	0.0	$r * \sin(t)$
4	0.0	z
5	4.0	$r^2 * \cos^2(t) + r^2 * \sin^2(t)$
6	$4.0 * r^2 - 2.0$	$\frac{r^4}{4.0} - \frac{r^2}{2.0}$
7	$8.0 * r * \cos(t)$	$(r^2 * \cos^2(t) + r^2 * \sin^2(t)) * r * \cos(t)$
8	$8.0 * r * \sin(t)$	$(r^2 * \cos^2(t) + r^2 * \sin^2(t)) * r * \sin(t)$

表 42: 3D Poisson in a cylinder with Robin B.C.

eq	N	M	H	method	res_max	err_max	iter	cpu_t
1	3	6	5	LU	2.21e-15	7.77e-15	1	0.00
1	3	8	5	LU	2.26e-15	8.33e-15	1	0.00
1	5	6	5	LU	4.11e-15	3.77e-15	1	0.00
1	5	8	5	LU	4.09e-15	1.68e-14	1	0.00
1	3	6	6	LU	3.54e-15	7.77e-15	1	0.00
1	3	8	6	LU	2.98e-15	5.00e-15	1	0.00
1	5	6	6	LU	4.57e-15	4.44e-15	1	0.00
1	5	8	6	LU	5.65e-15	1.35e-14	1	0.01
2	3	6	5	LU	6.88e-16	2.44e-15	1	0.00
2	3	8	5	LU	9.83e-16	2.33e-15	1	0.00
2	5	6	5	LU	1.89e-15	2.00e-15	1	0.00
2	5	8	5	LU	1.65e-15	4.11e-15	1	0.01
2	3	6	6	LU	1.20e-15	2.78e-15	1	0.00
2	3	8	6	LU	1.14e-15	1.67e-15	1	0.00
2	5	6	6	LU	2.56e-15	2.05e-15	1	0.00
2	5	8	6	LU	1.77e-15	2.22e-15	1	0.01
3	3	6	5	LU	6.40e-16	9.99e-16	1	0.00
3	3	8	5	LU	6.13e-16	4.11e-15	1	0.00
3	5	6	5	LU	1.52e-15	2.66e-15	1	0.00
3	5	8	5	LU	2.12e-15	7.66e-15	1	0.00
3	3	6	6	LU	6.42e-16	1.22e-15	1	0.00
3	3	8	6	LU	1.27e-15	2.55e-15	1	0.00
3	5	6	6	LU	2.01e-15	1.67e-15	1	0.00
3	5	8	6	LU	2.31e-15	5.77e-15	1	0.01

表 43: 3D Poisson in a cylinder with Robin B.C. (1) and LU Method

eq	N	M	H	method	res_max	err_max	iter	cpu_t
4	3	6	5	LU	1.49e-15	2.78e-15	1	0.00
4	3	8	5	LU	1.95e-15	3.11e-15	1	0.00
4	5	6	5	LU	2.06e-15	2.00e-15	1	0.00
4	5	8	5	LU	2.48e-15	5.11e-15	1	0.00
4	3	6	6	LU	2.70e-15	3.77e-15	1	0.00
4	3	8	6	LU	2.36e-15	2.66e-15	1	0.00
4	5	6	6	LU	3.70e-15	3.33e-15	1	0.00
4	5	8	6	LU	4.36e-15	5.55e-15	1	0.01
5	3	6	5	LU	5.41e-16	1.78e-15	1	0.00
5	3	8	5	LU	6.11e-16	2.33e-15	1	0.00
5	5	6	5	LU	1.54e-15	3.77e-15	1	0.00
5	5	8	5	LU	2.23e-15	1.05e-14	1	0.00
5	3	6	6	LU	8.79e-16	1.78e-15	1	0.00
5	3	8	6	LU	1.10e-15	1.44e-15	1	0.00
5	5	6	6	LU	2.23e-15	2.89e-15	1	0.00
5	5	8	6	LU	2.61e-15	8.66e-15	1	0.01
6	3	6	5	LU	3.55e-16	6.00e-01	1	0.00
6	3	8	5	LU	8.49e-16	9.69e-01	1	0.00
6	5	6	5	LU	6.20e-16	1.11e-15	1	0.00
6	5	8	5	LU	6.56e-16	3.39e-15	1	0.00
6	3	6	6	LU	4.85e-16	6.00e-01	1	0.00
6	3	8	6	LU	1.26e-15	9.69e-01	1	0.00
6	5	6	6	LU	8.31e-16	6.94e-16	1	0.01
6	5	8	6	LU	9.90e-16	2.66e-15	1	0.01

表 44: 3D Poisson in a cylinder with Robin B.C. (2) and LU Method

eq	N	M	H	method	res_max	err_max	iter	cpu.t
7	3	6	5	LU	3.86e-16	8.88e-16	1	0.00
7	3	8	5	LU	4.13e-16	7.77e-16	1	0.00
7	5	6	5	LU	1.25e-15	9.99e-16	1	0.00
7	5	8	5	LU	1.31e-15	2.89e-15	1	0.01
7	3	6	6	LU	5.66e-16	8.88e-16	1	0.00
7	3	8	6	LU	3.77e-16	5.55e-16	1	0.00
7	5	6	6	LU	2.09e-15	9.44e-16	1	0.00
7	5	8	6	LU	1.32e-15	1.83e-15	1	0.01
8	3	6	5	LU	3.29e-16	3.33e-16	1	0.00
8	3	8	5	LU	4.24e-16	1.33e-15	1	0.00
8	5	6	5	LU	1.23e-15	2.11e-15	1	0.00
8	5	8	5	LU	1.33e-15	5.11e-15	1	0.00
8	3	6	6	LU	3.50e-16	2.22e-16	1	0.00
8	3	8	6	LU	3.90e-16	8.88e-16	1	0.00
8	5	6	6	LU	1.55e-15	1.55e-15	1	0.00
8	5	8	6	LU	1.53e-15	4.00e-15	1	0.01

表 45: 3D Poisson in a cylinder with Robin B.C. (3) and LU Method

eq	N	M	H	method	res_max	err_max	iter	cpu_t
6	3	6	5	BCR	3.37e-15	6.00e-01	37	0.01
6	3	8	5	BCR	2.09e-14	9.69e-01	44	0.02
6	5	6	5	BCR	4.84e-15	2.19e-15	45	0.03
6	5	8	5	BCR	6.32e-15	4.19e-15	51	0.05
6	3	6	6	BCR	1.47e-14	6.00e-01	42	0.02
6	3	8	6	BCR	9.99e-15	9.69e-01	46	0.03
6	5	6	6	BCR	2.11e-14	5.55e-15	47	0.04
6	5	8	6	BCR	8.74e-15	3.14e-15	57	0.09
7	3	6	5	BCR	1.44e-14	2.42e-14	37	0.01
7	3	8	5	BCR	9.80e-15	4.00e-15	40	0.02
7	5	6	5	BCR	1.48e-14	5.00e-15	43	0.03
7	5	8	5	BCR	2.54e-14	9.55e-15	50	0.06
7	3	6	6	BCR	2.89e-15	1.17e-15	43	0.02
7	3	8	6	BCR	1.04e-14	2.41e-15	46	0.03
7	5	6	6	BCR	1.60e-14	1.17e-15	48	0.04
7	5	8	6	BCR	2.07e-14	6.39e-15	54	0.08
8	3	6	5	BCR	1.56e-14	7.33e-15	36	0.01
8	3	8	5	BCR	1.01e-14	7.88e-15	40	0.02
8	5	6	5	BCR	2.13e-14	8.10e-15	43	0.03
8	5	8	5	BCR	2.12e-14	4.11e-15	49	0.06
8	3	6	6	BCR	2.41e-15	2.00e-15	44	0.02
8	3	8	6	BCR	1.32e-14	6.66e-15	46	0.03
8	5	6	6	BCR	2.57e-14	5.11e-15	47	0.04
8	5	8	6	BCR	1.01e-14	2.55e-15	55	0.08

表 46: 3D Poisson in a cylinder with Robin B.C. and BCR Method

3.5.3 Neumann Boundary Condition

使用LU分解法與BCR迭代法[11]解二維的Poisson方程式。

邊界值：

- $u_r(1, \theta), \quad 0 < \theta \leq 2\pi.$
- $u_z(r, \theta, -1), \quad 0 < \theta \leq 2\pi, \quad 0 < r < 1.$
- $u_z(r, \theta, 1), \quad 0 < \theta \leq 2\pi, \quad 0 < r < 1.$

case	RHS function of PDE	exact solution
1	0.0	1.0
2	0.0	$r * \cos(t)$
3	0.0	$r * \sin(t)$
4	0.0	z
5	4.0	$r^2 * \cos^2(t) + r^2 * \sin^2(t)$
6	$4.0 * r^2 - 2.0$	$\frac{r^4}{4.0} - \frac{r^2}{2.0}$
7	$8.0 * r * \cos(t)$	$(r^2 * \cos^2(t) + r^2 * \sin^2(t)) * r * \cos(t)$
8	$8.0 * r * \sin(t)$	$(r^2 * \cos^2(t) + r^2 * \sin^2(t)) * r * \sin(t)$

表 47: 3D Poisson in a cylinder with Neumann B.C.

eq	N	M	H	method	res_max	err_max	iter	cpu_t
1	3	6	5	LU	0.00e+00	0.00e+00	1	0.00
1	3	8	5	LU	0.00e+00	0.00e+00	1	0.00
1	5	6	5	LU	0.00e+00	0.00e+00	1	0.01
1	5	8	5	LU	0.00e+00	0.00e+00	1	0.00
1	3	6	6	LU	0.00e+00	0.00e+00	1	0.00
1	3	8	6	LU	0.00e+00	0.00e+00	1	0.00
1	5	6	6	LU	0.00e+00	0.00e+00	1	0.00
1	5	8	6	LU	0.00e+00	0.00e+00	1	0.01
2	3	6	5	LU	1.05e-15	1.25e-15	1	0.00
2	3	8	5	LU	1.13e-15	5.77e-15	1	0.00
2	5	6	5	LU	4.24e-15	7.22e-15	1	0.00
2	5	8	5	LU	1.73e-15	8.55e-15	1	0.00
2	3	6	6	LU	1.64e-15	1.65e-15	1	0.00
2	3	8	6	LU	1.18e-15	4.11e-15	1	0.00
2	5	6	6	LU	4.04e-14	4.52e-14	1	0.00
2	5	8	6	LU	2.49e-15	5.11e-15	1	0.01
3	3	6	5	LU	5.79e-16	2.11e-15	1	0.00
3	3	8	5	LU	8.83e-16	4.77e-15	1	0.00
3	5	6	5	LU	2.14e-15	4.33e-15	1	0.00
3	5	8	5	LU	2.04e-15	7.44e-15	1	0.01
3	3	6	6	LU	1.41e-15	2.44e-15	1	0.00
3	3	8	6	LU	1.24e-15	3.00e-15	1	0.00
3	5	6	6	LU	5.71e-15	7.68e-15	1	0.00
3	5	8	6	LU	2.03e-15	4.55e-15	1	0.01

表 48: 3D Poisson in a cylinder with Neumann B.C. (1) and LU Method

eq	N	M	H	method	res_max	err_max	iter	cpu_t
4	3	6	5	LU	1.77e-15	7.22e-15	1	0.00
4	3	8	5	LU	1.65e-15	6.88e-15	1	0.00
4	5	6	5	LU	2.80e-15	3.89e-15	1	0.00
4	5	8	5	LU	2.62e-15	1.27e-14	1	0.00
4	3	6	6	LU	1.86e-15	7.66e-15	1	0.00
4	3	8	6	LU	2.37e-15	5.77e-15	1	0.00
4	5	6	6	LU	1.35e-14	1.67e-14	1	0.00
4	5	8	6	LU	4.03e-15	7.99e-15	1	0.01
5	3	6	5	LU	3.79e-16	2.22e-16	1	0.00
5	3	8	5	LU	5.13e-16	1.11e-16	1	0.00
5	5	6	5	LU	4.04e-15	5.80e-15	1	0.00
5	5	8	5	LU	1.92e-15	1.47e-15	1	0.00
5	3	6	6	LU	4.54e-16	1.50e-16	1	0.00
5	3	8	6	LU	4.19e-16	1.11e-16	1	0.00
5	5	6	6	LU	3.24e-14	4.31e-14	1	0.00
5	5	8	6	LU	2.69e-15	9.71e-16	1	0.01
6	3	6	5	LU	2.51e-01	9.69e-01	1	0.00
6	3	8	5	LU	4.67e-01	1.50e+00	1	0.00
6	5	6	5	LU	1.02e-15	1.48e-15	1	0.00
6	5	8	5	LU	5.11e-16	5.00e-16	1	0.00
6	3	6	6	LU	3.64e-01	1.34e+00	1	0.00
6	3	8	6	LU	9.09e-01	7.50e-01	1	0.00
6	5	6	6	LU	7.06e-15	9.98e-15	1	0.00
6	5	8	6	LU	6.17e-16	3.05e-16	1	0.01

表 49: 3D Poisson in a cylinder with Neumann B.C. (2) and LU Method

eq	N	M	H	method	res_max	err_max	iter	cpu.t
7	3	6	5	LU	5.21e-16	6.11e-16	1	0.00
7	3	8	5	LU	2.88e-16	1.55e-15	1	0.00
7	5	6	5	LU	1.82e-15	3.17e-15	1	0.00
7	5	8	5	LU	2.27e-15	5.44e-15	1	0.01
7	3	6	6	LU	7.05e-16	8.88e-16	1	0.00
7	3	8	6	LU	5.32e-16	1.11e-15	1	0.00
7	5	6	6	LU	1.82e-14	2.08e-14	1	0.00
7	5	8	6	LU	1.92e-15	3.22e-15	1	0.01
8	3	6	5	LU	4.18e-16	1.11e-15	1	0.00
8	3	8	5	LU	4.76e-16	1.67e-15	1	0.00
8	5	6	5	LU	1.57e-15	2.66e-15	1	0.00
8	5	8	5	LU	1.48e-15	4.88e-15	1	0.00
8	3	6	6	LU	3.54e-16	1.33e-15	1	0.00
8	3	8	6	LU	5.84e-16	1.11e-15	1	0.00
8	5	6	6	LU	7.31e-15	8.35e-15	1	0.00
8	5	8	6	LU	1.48e-15	3.00e-15	1	0.01

表 50: 3D Poisson in a cylinder with Neumann B.C. (3) and LU

eq	N	M	H	method	res_max	err_max	iter	cpu_t
6	3	6	5	BCR	2.33e+03	2.80e+03	1000	0.27
6	3	8	5	BCR	2.95e+02	2.71e+02	1000	0.47
6	5	6	5	BCR	6.83e-15	5.29e-15	78	0.05
6	5	8	5	BCR	4.58e-15	1.32e-15	61	0.07
6	3	6	6	BCR	1.09e+03	9.30e+02	1000	0.35
6	3	8	6	BCR	1.16e+00	3.74e-01	1000	0.64
6	5	6	6	BCR	2.13e-14	1.53e-14	68	0.05
6	5	8	6	BCR	4.52e-13	1.14e-13	150	0.22
7	3	6	5	BCR	1.76e-14	8.49e-15	41	0.01
7	3	8	5	BCR	6.52e-15	2.89e-15	43	0.02
7	5	6	5	BCR	1.99e-14	5.65e-15	53	0.03
7	5	8	5	BCR	8.64e-15	8.55e-15	56	0.06
7	3	6	6	BCR	2.16e-14	9.10e-15	45	0.02
7	3	8	6	BCR	7.20e-15	2.09e-15	48	0.03
7	5	6	6	BCR	2.37e-14	7.22e-15	58	0.05
7	5	8	6	BCR	1.45e-14	7.77e-15	59	0.09
8	3	6	5	BCR	1.55e-14	1.27e-14	40	0.01
8	3	8	5	BCR	1.04e-14	1.34e-14	42	0.02
8	5	6	5	BCR	1.32e-14	4.76e-15	55	0.04
8	5	8	5	BCR	1.90e-14	1.03e-14	54	0.06
8	3	6	6	BCR	5.51e-15	2.07e-15	47	0.02
8	3	8	6	BCR	6.58e-15	2.78e-15	47	0.03
8	5	6	6	BCR	1.73e-14	4.11e-15	58	0.05
8	5	8	6	BCR	1.31e-14	6.00e-15	58	0.09

表 51: 3D Poisson in a cylinder with Neumann B.C. and BCR Method

4 頻譜配點法之應用(二) 線性波

上一章僅對空間微分方程式作討論，並未對有時間相依(time dependence)的例子作討論。本章開始將對時間相依的方程式作討論。

$$\text{一維的wave方程式： } u_{tt} - c^2 u_{xx} = 0, \quad -1 \leq x \leq 1$$

$$\left\{ \begin{array}{l} \text{Boundary conditions :} \quad u(-1, t) = 0, \quad u(1, t) = 0, \quad \forall t. \\ \text{Initial conditions :} \quad u(x, 0) = f(x), \quad u_t(x, 0) = 0, \quad \forall x. \\ \text{exact solution :} \quad u_{x,t} = \frac{F(x - c * t) + F(x + c * t)}{2}. \end{array} \right.$$

其中 $f(x)$ 為自訂的函數， $0 < x \leq 1$ 。而

$$F(z) = f(z), \quad \text{if } 0 < z \leq 1.$$

$$F(-z) = -F(z), \quad \text{if } -1 \leq z < 0.$$

$$F(z + 2) = F(z), \quad \forall z \in (R)$$

本次實驗中對函數 $f(x)$ 的設定如下：

$$\text{if } (|x - 0.5| < 0.1) \quad f(x) = \sin(\pi * 5 * (x - 0.4))$$

$$\text{else} \quad f(x) = 0.0$$

頻譜配點法與有限差分法的實驗結果可參見表52。頻譜配點法只需要極小的自由度，誤差就可以小於有限差分法。且其誤差累積的速度較慢。

time	finite difference	spectral collocation
	err_1_norm	err_1_norm
2.00e-06	1.258423e-04	5.44e-05
3.00e-06	1.256031e-04	5.44e-05
4.00e-06	2.733253e-04	1.09e-04
5.00e-06	2.736458e-04	1.09e-04
6.00e-06	5.823528e-04	1.63e-04
7.00e-06	7.751007e-04	1.63e-04
8.00e-06	1.459450e-03	2.17e-04
9.00e-06	2.323361e-03	2.17e-04
1.00e-05	4.242279e-03	2.72e-04
	N = 400	N = 13

表 52: comparison of finite difference and spectral collocation

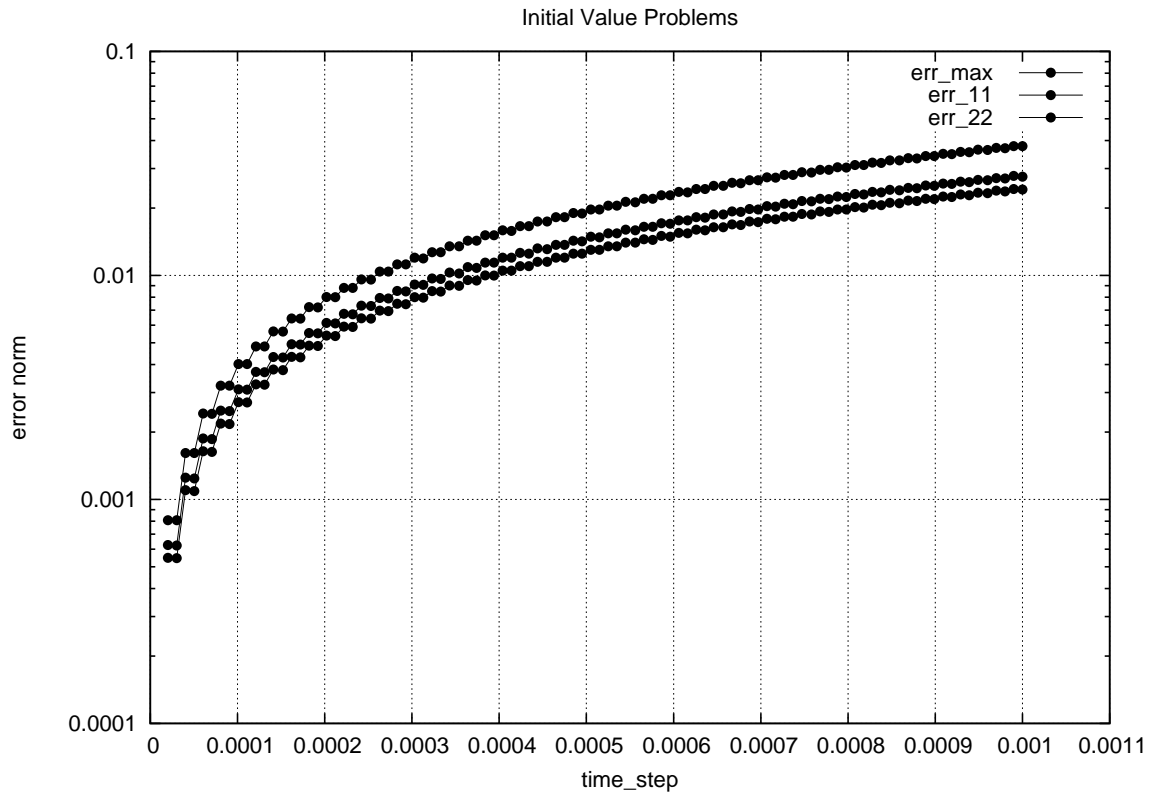


圖 9: 非線性波傳頻譜配點法誤差圖

此圖可以詳細看出頻譜配點法誤差的增加十分平緩。其中

err_max : 最大誤差

err_11 : 平均誤差

err_22 : 根號平均誤差

5 頻譜配點法之應用(三) 非線性波

本章將以頻譜配點法應用於有時間相依的一維、二維、三維Burgers方程式，在不同定義域所得到的結果。由於Burgers方程式有正確解，所以先算出正確解的答案，再和近似解比較，算出誤差。Burgers方程式是一個關於Convection-Diffusion兩者作用之下最簡單的非線性微分方程式。而在計算流體力學，有效而準確地處理此兩者作用的平衡性是一個重要而較困難的問題，且最重要的是其與Navier-Stokes方程式具有類似性(Analogy)，故研究者常用伯格方程式做為測試一個數值方法是否可以應用於Navier-Stokes方程式。測試中採取直接解法的LU分解法，與GMRes(Generalized minimal residual method)[11]迭代方法。

5.1 1D Burgers equation

使用LU分解法與GMRes迭代法[11]解一維的Burgers方程式。

$$\text{一維Burgers方程式：} u_t + u(u_x) - \frac{(u_{xx})}{Re} = 0, \quad -1 \leq x \leq 1.$$

$$\text{正確解：} u(x, t) = \frac{1.0}{1.0 + \exp\left(\frac{Re * (2.0 * x - t)}{4.0}\right)},$$

Re 為雷諾數。

設定基本參數：

N : x 方向的自由度

$iter_sum$: 線性迭代次數總和

nl_sum : 非線性迭代次數總和

res_max : 最大 *residual norm*

err_max : 最大誤差

cpu_time : 測試所花的全部時間

由表53至表58可知其residual很快就可以達到指定的要求，但誤差似乎比處理Poisson時大，應該是時間項沒有算的很準的結果。由於是一維的問題，所以cpu時間為0，但隨著題目複雜程度的增加，cpu時間也會逐漸增加。由表55與表57知道隨時間的減少，誤差也會隨著變小，且無論使用分解法或迭代法，都可以到達相同的誤差值。

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	LU	6.80e-17	1.52e-10	0.00
9.00e-04	9	LU	6.80e-17	1.38e-10	0.00
8.00e-04	9	LU	6.80e-17	1.24e-10	0.00
7.00e-04	9	LU	6.80e-17	1.10e-10	0.00
6.00e-04	9	LU	6.80e-17	9.50e-11	0.00
5.00e-04	9	LU	6.80e-17	8.00e-11	0.00
4.00e-04	9	LU	6.80e-17	6.46e-11	0.00
3.00e-04	9	LU	6.80e-17	4.90e-11	0.00
2.00e-04	9	LU	6.80e-17	3.30e-11	0.00
1.00e-04	9	LU	6.80e-17	1.67e-11	0.00
iter_ sum = 33, nl_ sum = 33.					

表 53: 1D Burgers with $Re = 1.0$ and LU Method

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	LU	4.99e-17	2.15e-05	0.00
9.00e-04	9	LU	4.99e-17	1.94e-05	0.00
8.00e-04	9	LU	4.99e-17	1.72e-05	0.00
7.00e-04	9	LU	4.99e-17	1.51e-05	0.00
6.00e-04	9	LU	4.99e-17	1.29e-05	0.00
5.00e-04	9	LU	4.99e-17	1.08e-05	0.00
4.00e-04	9	LU	4.99e-17	8.62e-06	0.00
3.00e-04	9	LU	4.99e-17	6.46e-06	0.00
2.00e-04	9	LU	4.99e-17	4.31e-06	0.00
1.00e-04	9	LU	4.99e-17	2.16e-06	0.00
iter_ sum = 40, nl_ sum = 40.					

表 54: 1D Burgers with Re =10.0 and LU Method

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	LU	3.51e-17	1.84e-03	0.00
1.00e-04	9	LU	7.02e-17	1.84e-04	0.00
1.00e-05	9	LU	1.67e-23	1.84e-05	0.00
1.00e-06	9	LU	8.60e-17	1.84e-06	0.00
1.00e-07	9	LU	4.97e-17	1.84e-07	0.00
1.00e-08	9	LU	6.08e-17	1.84e-08	0.00
1.00e-09	9	LU	6.08e-17	1.84e-09	0.00
1.00e-10	9	LU	7.85e-17	1.84e-10	0.00

表 55: 1D Burgers with $Re = 100$ and LU Method

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	GMRes	5.61e-20	1.52e-10	0.00
9.00e-04	9	GMRes	5.61e-20	1.38e-10	0.00
8.00e-04	9	GMRes	5.61e-20	1.24e-10	0.00
7.00e-04	9	GMRes	5.61e-20	1.10e-10	0.00
6.00e-04	9	GMRes	5.61e-20	9.50e-11	0.00
5.00e-04	9	GMRes	5.61e-20	8.00e-11	0.00
4.00e-04	9	GMRes	5.61e-20	6.46e-11	0.00
3.00e-04	9	GMRes	5.61e-20	4.90e-11	0.00
2.00e-04	9	GMRes	5.61e-20	3.30e-11	0.00
1.00e-04	9	GMRes	5.61e-20	1.67e-11	0.00
iter_ sum = 104, nl_ sum = 36.					

表 56: 1D Burgers with $Re = 1.0$ and GMRes Method

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	GMRes	3.51e-17	2.15e-05	0.00
9.00e-04	9	GMRes	3.51e-17	1.94e-05	0.00
8.00e-04	9	GMRes	3.51e-17	1.72e-05	0.00
7.00e-04	9	GMRes	3.51e-17	1.51e-05	0.00
6.00e-04	9	GMRes	3.51e-17	1.29e-05	0.00
5.00e-04	9	GMRes	3.51e-17	1.08e-05	0.00
4.00e-04	9	GMRes	3.51e-17	8.62e-06	0.00
3.00e-04	9	GMRes	3.51e-17	6.46e-06	0.00
2.00e-04	9	GMRes	3.51e-17	4.31e-06	0.00
1.00e-04	9	GMRes	3.51e-17	2.16e-06	0.00
iter_ sum = 100, nl_ sum = 40.					

表 57: 1D Burgers with $Re = 10.0$ and GMRes Method

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	GMRes	5.61e-20	1.84e-03	0.00
1.00e-04	9	GMRes	3.51e-17	1.84e-04	0.00
1.00e-05	9	GMRes	1.67e-23	1.84e-05	0.00
1.00e-06	9	GMRes	3.51e-17	1.84e-06	0.00
1.00e-07	9	GMRes	3.92e-25	1.84e-07	0.00
1.00e-08	9	GMRes	9.54e-20	1.84e-08	0.00
1.00e-09	9	GMRes	8.59e-21	1.84e-09	0.00
1.00e-10	9	GMRes	5.74e-24	1.84e-10	0.00

表 58: 1D Burgers with $Re = 100$ and GMRes Method

5.2 2D Burgers equation in a square

使用LU分解法與GMRes迭代法[11]解二維的Burgers方程式。

$$\text{二維Burgers方程式：} u_t + u(u_x + u_y) - \frac{(u_{xx} + u_{yy})}{Re} = 0,$$

$$-1 \leq x \leq 1, \quad -1 \leq y \leq 1.$$

$$\text{正確解：} u(x, y, t) = \frac{1.0}{1.0 + \exp\left(\frac{Re * (2.0 * x + 2.0 * y - 2.0 * t)}{4.0}\right)},$$

Re 為雷諾數。

設定基本參數：

N : x 方向與 y 方向的自由度

$iter_sum$: 線性迭代次數總和

nl_sum : 非線性迭代次數總和

res_max : 最大 *residual norm*

err_max : 最大誤差

cpu_time : 測試所花的全部時間

由表59至表64可知其residual很快就可以達到指定的要求，但誤差似乎比處理Poisson時大，應該是時間項沒有算的很準的結果。由表61與表64知道隨時間的減少，誤差也會隨著變小，且無論使用分解法或迭代法，都可以到達相同的誤差值。

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	LU	8.83e-13	1.00e-09	0.00
9.00e-04	9	LU	8.83e-13	9.02e-10	0.00
8.00e-04	9	LU	8.83e-13	8.01e-10	0.00
7.00e-04	9	LU	8.83e-13	7.00e-10	0.00
6.00e-04	9	LU	8.83e-13	5.99e-10	0.00
5.00e-04	9	LU	8.83e-13	4.99e-10	0.00
4.00e-04	9	LU	8.83e-13	4.01e-10	0.00
3.00e-04	9	LU	8.83e-13	3.05e-10	0.00
2.00e-04	9	LU	8.83e-13	2.08e-10	0.00
1.00e-04	9	LU	8.83e-13	1.06e-10	0.00
iter_ sum = 40, nl_ sum = 40.					

表 59: 2D Burgers in a square with $Re = 1.0$ and LU Method

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	LU	9.82e-13	4.45e-05	0.00
9.00e-04	9	LU	9.82e-13	4.01e-05	0.00
8.00e-04	9	LU	9.82e-13	3.57e-05	0.00
7.00e-04	9	LU	9.82e-13	3.12e-05	0.00
6.00e-04	9	LU	9.82e-13	2.68e-05	0.00
5.00e-04	9	LU	9.82e-13	2.23e-05	0.00
4.00e-04	9	LU	9.82e-13	1.79e-05	0.00
3.00e-04	9	LU	9.82e-13	1.34e-05	0.00
2.00e-04	9	LU	9.82e-13	8.96e-06	0.00
1.00e-04	9	LU	9.82e-13	4.48e-06	0.00
iter_ sum = 50, nl_ sum = 50.					

表 60: 2D Burgers in a square with $Re = 10.0$ and LU Method

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	LU	1.23e-12	1.05e-02	0.00
1.00e-04	9	LU	7.87e-12	1.05e-03	0.00
1.00e-05	9	LU	6.74e-11	1.05e-04	0.00
1.00e-06	9	LU	1.01e-09	1.05e-05	0.00
1.00e-07	9	LU	7.30e-09	1.05e-06	0.00
1.00e-08	9	LU	6.31e-08	1.05e-07	0.00
1.00e-09	9	LU	9.34e-07	1.05e-08	0.00
1.00e-10	9	LU	7.40e-06	1.05e-09	0.00

表 61: 2D Burgers in a square with $Re = 100$ and LU Method

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	GMRes	3.69e-13	1.00e-09	0.00
9.00e-04	9	GMRes	3.69e-13	9.02e-10	0.00
8.00e-04	9	GMRes	3.69e-13	8.01e-10	0.00
7.00e-04	9	GMRes	3.69e-13	7.00e-10	0.00
6.00e-04	9	GMRes	3.69e-13	5.99e-10	0.00
5.00e-04	9	GMRes	3.69e-13	4.99e-10	0.00
4.00e-04	9	GMRes	3.69e-13	4.01e-10	0.00
3.00e-04	9	GMRes	3.69e-13	3.05e-10	0.00
2.00e-04	9	GMRes	3.69e-13	2.08e-10	0.00
1.00e-04	9	GMRes	3.69e-13	1.06e-10	0.00
iter_ sum = 130, nl_ sum = 40.					

表 62: 2D Burgers in a square with $Re = 1.0$ and GMRes Method

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	GMRes	4.27e-13	4.45e-05	0.00
9.00e-04	9	GMRes	4.27e-13	4.01e-05	0.00
8.00e-04	9	GMRes	4.27e-13	3.57e-05	0.00
7.00e-04	9	GMRes	4.27e-13	3.12e-05	0.00
6.00e-04	9	GMRes	4.27e-13	2.68e-05	0.00
5.00e-04	9	GMRes	4.27e-13	2.23e-05	0.00
4.00e-04	9	GMRes	4.27e-13	1.79e-05	0.00
3.00e-04	9	GMRes	4.27e-13	1.34e-05	0.00
2.00e-04	9	GMRes	4.27e-13	8.96e-06	0.00
1.00e-04	9	GMRes	4.27e-13	4.48e-06	0.00
iter_ sum = 140, nl_ sum = 50.					

表 63: 2D Burgers in a square with $Re = 10.0$ and GMRes Method

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	GMRes	5.75e-13	1.05e-02	0.00
1.00e-04	9	GMRes	3.25e-12	1.05e-03	0.00
1.00e-05	9	GMRes	2.73e-11	1.05e-04	0.00
1.00e-06	9	GMRes	3.73e-10	1.05e-05	0.00
1.00e-07	9	GMRes	2.58e-09	1.05e-06	0.00
1.00e-08	9	GMRes	2.80e-08	1.05e-07	0.00
1.00e-09	9	GMRes	6.61e-07	1.05e-08	0.00
1.00e-10	9	GMRes	3.58e-06	1.05e-09	0.00

表 64: 2D Burgers in a square with $Re = 100$ and GMRes Method

5.3 2D Burgers equation in a circle

使用LU分解法與GMRes迭代法[11]解二維的Burgers方程式。

$$\text{二維Burgers方程式：} u_t + u(u_r) - \frac{1}{Re}(u_{rr} + r^{-1}u_r + r^{-2}u_{\theta\theta}) = 0,$$

$$0 < r \leq 1, \quad 0 < \theta \leq 2\pi.$$

$$\text{正確解：} u(r, \theta, t) = \frac{1.0}{1.0 + \exp\left(\frac{Re * (2.0 * x + 2.0 * y - 2.0 * t)}{4.0}\right)},$$

Re 為雷諾數。

設定基本參數：

N : r 方向의自由度，取奇數

M : θ 方向의自由度，取偶數

$iter_sum$: 線性迭代次數總和

nl_sum : 非線性迭代次數總和

res_max : 最大 *residual norm*

err_max : 最大誤差

cpu_time : 測試所花的全部時間

由表65至表70可知其residual很快就可以達到指定的要求，但誤差似乎比處理Poisson時大，應該是時間項沒有算的很準的結果。由表67與表70知道隨時間的減少，誤差也會隨著變小，且無論使用分解法或迭代法，都可以到達相同的誤差值。

time	N	M	method	res_ max	err_ max	cpu_ t
1.00e-03	9	8	LU	5.28e-13	2.52e-04	0.00
9.00e-04	9	8	LU	5.28e-13	2.27e-04	0.00
8.00e-04	9	8	LU	5.28e-13	2.01e-04	0.00
7.00e-04	9	8	LU	5.28e-13	1.76e-04	0.00
6.00e-04	9	8	LU	5.28e-13	1.51e-04	0.00
5.00e-04	9	8	LU	5.28e-13	1.26e-04	0.00
4.00e-04	9	8	LU	5.28e-13	1.01e-04	0.00
3.00e-04	9	8	LU	5.28e-13	7.54e-05	0.00
2.00e-04	9	8	LU	5.28e-13	5.02e-05	0.00
1.00e-04	9	8	LU	5.28e-13	2.51e-05	0.00
iter_ sum = 40, nl_ sum = 40.						

表 65: 2D Burgers in a circle with $Re = 1.0$ and LU Method

time	N	M	method	res_ max	err_ max	cpu_ t
1.00e-03	9	8	LU	8.29e-13	2.25e-03	0.00
9.00e-04	9	8	LU	8.29e-13	2.02e-03	0.00
8.00e-04	9	8	LU	8.29e-13	1.80e-03	0.00
7.00e-04	9	8	LU	8.29e-13	1.57e-03	0.00
6.00e-04	9	8	LU	8.29e-13	1.35e-03	0.00
5.00e-04	9	8	LU	8.29e-13	1.13e-03	0.00
4.00e-04	9	8	LU	8.29e-13	9.01e-04	0.00
3.00e-04	9	8	LU	8.29e-13	6.76e-04	0.00
2.00e-04	9	8	LU	8.29e-13	4.50e-04	0.00
1.00e-04	9	8	LU	8.29e-13	2.25e-04	0.00
iter_ sum = 40, nl_ sum = 40.						

表 66: 2D Burgers in a circle with $Re = 10.0$ and LU Method

time	N	M	method	res_ max	err_ max	cpu_ t
1.00e-03	19	14	LU	1.30e-12	1.48e-02	0.04
1.00e-04	19	14	LU	1.09e-11	1.48e-03	0.04
1.00e-05	19	14	LU	1.00e-10	1.48e-04	0.04
1.00e-06	19	14	LU	1.34e-09	1.48e-05	0.04
1.00e-07	19	14	LU	1.13e-08	1.48e-06	0.04
1.00e-08	19	14	LU	7.89e-08	1.48e-07	0.04
1.00e-09	19	14	LU	1.11e-06	1.48e-08	0.04
1.00e-10	19	14	LU	9.53e-06	1.48e-09	0.04

表 67: 2D Burgers in a circle with $Re = 100$ and LU Method

time	N	M	method	res_ max	err_ max	cpu_ t
1.00e-03	9	8	GMRes	3.67e-13	2.52e-04	0.00
9.00e-04	9	8	GMRes	3.67e-13	2.27e-04	0.00
8.00e-04	9	8	GMRes	3.67e-13	2.01e-04	0.00
7.00e-04	9	8	GMRes	3.67e-13	1.76e-04	0.00
6.00e-04	9	8	GMRes	3.67e-13	1.51e-04	0.00
5.00e-04	9	8	GMRes	3.67e-13	1.26e-04	0.00
4.00e-04	9	8	GMRes	3.67e-13	1.01e-04	0.00
3.00e-04	9	8	GMRes	3.67e-13	7.54e-05	0.00
2.00e-04	9	8	GMRes	3.67e-13	5.02e-05	0.00
1.00e-04	9	8	GMRes	3.67e-13	2.51e-05	0.00
iter_ sum = 129, nl_ sum = 40.						

表 68: 2D Burgers in a circle with $Re = 1.0$ and GMRes Method

time	N	M	method	res_ max	err_ max	cpu_ t
1.00e-03	9	8	GMRes	4.00e-13	2.25e-03	0.00
9.00e-04	9	8	GMRes	4.00e-13	2.02e-03	0.00
8.00e-04	9	8	GMRes	4.00e-13	1.80e-03	0.00
7.00e-04	9	8	GMRes	4.00e-13	1.57e-03	0.00
6.00e-04	9	8	GMRes	4.00e-13	1.35e-03	0.00
5.00e-04	9	8	GMRes	4.00e-13	1.13e-03	0.00
4.00e-04	9	8	GMRes	4.00e-13	9.01e-04	0.00
3.00e-04	9	8	GMRes	4.00e-13	6.76e-04	0.00
2.00e-04	9	8	GMRes	4.00e-13	4.50e-04	0.00
1.00e-04	9	8	GMRes	4.00e-13	2.25e-04	0.00
iter_ sum = 110, nl_ sum = 40.						

表 69: 2D Burgers in a circle with $Re = 10.0$ and GMRes Method

time	N	M	method	res_ max	err_ max	cpu_ t
1.00e-03	19	14	GMRes	5.54e-13	1.48e-02	0.04
1.00e-04	19	14	GMRes	2.13e-12	1.48e-03	0.04
1.00e-05	19	14	GMRes	3.27e-11	1.48e-04	0.04
1.00e-06	19	14	GMRes	5.04e-10	1.48e-05	0.04
1.00e-07	19	14	GMRes	2.82e-09	1.48e-06	0.04
1.00e-08	19	14	GMRes	3.34e-08	1.48e-07	0.04
1.00e-09	19	14	GMRes	7.03e-07	1.48e-08	0.04
1.00e-10	19	14	GMRes	2.34e-06	1.48e-09	0.04

表 70: 2D Burgers in a circle with $Re = 100$ and GMRes Method

5.4 3D Burgers equation in a cube

使用LU分解法與GMRes迭代法[11]解三維的Burgers方程式。

$$\text{三維Burgers方程式：} u_t + u(u_x + u_y + u_z) - \frac{u_{xx} + u_{yy} + u_{zz}}{Re} = 0,$$

$$-1 \leq x \leq 1, \quad -1 \leq y \leq 1, \quad -1 \leq z \leq 1.$$

正確解：

$$u(x, y, z, t) = \frac{1.0}{1.0 + \exp\left(\frac{Re * (2.0 * x + 2.0 * y + 2.0 * z - 3.0 * t)}{4.0}\right)},$$

Re 為雷諾數。

設定基本參數：

N : x 方向與 y 方向與 z 方向的自由度

$iter_sum$: 線性迭代次數總和

nl_sum : 非線性迭代次數總和

res_max : 最大 *residual norm*

err_max : 最大誤差

cpu_time : 測試所花的全部時間

由表71至表76可知其residual很快就可以達到指定的要求，但誤差似乎比處理Poisson時大，應該是時間項沒有算的很準的結果。由表73與表76知道隨時間的減少，誤差也會隨著變小，且無論使用分解法或迭代法，都可以到達相同的誤差值。

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	LU	7.47e-13	5.35e-05	3.62
9.00e-04	9	LU	7.47e-13	4.82e-05	3.62
8.00e-04	9	LU	7.47e-13	4.28e-05	3.62
7.00e-04	9	LU	7.47e-13	3.74e-05	3.62
6.00e-04	9	LU	7.47e-13	3.24e-05	3.62
5.00e-04	9	LU	7.47e-13	2.72e-05	3.62
4.00e-04	9	LU	7.47e-13	2.20e-05	3.62
3.00e-04	9	LU	7.47e-13	1.67e-05	3.62
2.00e-04	9	LU	7.47e-13	1.12e-05	3.62
1.00e-04	9	LU	7.47e-13	5.66e-06	3.62
iter_ sum = 40, nl_ sum = 40.					

表 71: 3D Burgers in a cube with $Re = 1.0$ and LU Method

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	LU	1.11e-12	1.47e-03	3.81
9.00e-04	9	LU	1.11e-12	1.33e-03	3.81
8.00e-04	9	LU	1.11e-12	1.18e-03	3.81
7.00e-04	9	LU	1.11e-12	1.03e-03	3.81
6.00e-04	9	LU	1.11e-12	8.85e-04	3.81
5.00e-04	9	LU	1.11e-12	7.37e-04	3.81
4.00e-04	9	LU	1.11e-12	5.90e-04	3.81
3.00e-04	9	LU	1.11e-12	4.42e-04	3.81
2.00e-04	9	LU	1.11e-12	2.95e-04	3.81
1.00e-04	9	LU	1.11e-12	1.47e-04	3.81
iter_ sum = 40, nl_ sum = 40.					

表 72: 3D Burgers in a cube with $Re = 10.0$ and LU Method

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	LU	1.27e-12	1.78e-02	3.52
1.00e-04	9	LU	9.89e-12	1.78e-03	3.51
1.00e-05	9	LU	7.94e-11	1.78e-04	3.52
1.00e-06	9	LU	1.22e-09	1.78e-05	3.51
1.00e-07	9	LU	8.57e-09	1.78e-06	3.52
1.00e-08	9	LU	7.69e-08	1.78e-07	3.51
1.00e-09	9	LU	8.09e-07	1.78e-08	3.51
1.00e-10	9	LU	9.18e-06	1.78e-09	3.52

表 73: 3D Burgers in a cube with $Re = 100$ and LU Method

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	GMRes	3.15e-13	5.35e-05	0.15
9.00e-04	9	GMRes	3.15e-13	4.82e-05	0.15
8.00e-04	9	GMRes	3.15e-13	4.28e-05	0.15
7.00e-04	9	GMRes	3.15e-13	3.74e-05	0.15
6.00e-04	9	GMRes	3.15e-13	3.24e-05	0.15
5.00e-04	9	GMRes	3.15e-13	2.72e-05	0.15
4.00e-04	9	GMRes	3.15e-13	2.20e-05	0.15
3.00e-04	9	GMRes	3.15e-13	1.67e-05	0.15
2.00e-04	9	GMRes	3.15e-13	1.12e-05	0.15
1.00e-04	9	GMRes	3.15e-13	5.66e-06	0.15
iter_ sum = 130, nl_ sum = 40.					

表 74: 3D Burgers in a cube with $Re = 1.0$ and GMRes Method

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	GMRes	3.92e-13	1.47e-03	0.14
9.00e-04	9	GMRes	3.92e-13	1.33e-03	0.14
8.00e-04	9	GMRes	3.92e-13	1.18e-03	0.14
7.00e-04	9	GMRes	3.92e-13	1.03e-03	0.14
6.00e-04	9	GMRes	3.92e-13	8.85e-04	0.14
5.00e-04	9	GMRes	3.92e-13	7.37e-04	0.14
4.00e-04	9	GMRes	3.92e-13	5.90e-04	0.14
3.00e-04	9	GMRes	3.92e-13	4.42e-04	0.14
2.00e-04	9	GMRes	3.92e-13	2.95e-04	0.14
1.00e-04	9	GMRes	3.92e-13	1.47e-04	0.14
iter_ sum = 130, nl_ sum = 40.					

表 75: 3D Burgers in a cube with $Re = 10.0$ and GMRes Method

time	N	method	res_ max	err_ max	cpu_ t
1.00e-03	9	GMRes	4.30e-13	1.78e-02	0.12
1.00e-04	9	GMRes	3.14e-12	1.78e-03	0.12
1.00e-05	9	GMRes	2.91e-11	1.78e-04	0.12
1.00e-06	9	GMRes	4.33e-10	1.78e-05	0.12
1.00e-07	9	GMRes	2.64e-09	1.78e-06	0.12
1.00e-08	9	GMRes	2.72e-08	1.78e-07	0.11
1.00e-09	9	GMRes	4.14e-07	1.78e-08	0.12
1.00e-10	9	GMRes	2.51e-06	1.78e-09	0.11

表 76: 3D Burgers in a cube with $Re = 100$ and GMRes Method

5.5 3D Burgers equation in a cylinder

使用LU分解法與GMRes迭代法[11]解三維的Burgers方程式。

$$\text{三維Burgers方程式：} u_t + u(u_r + u_z) - \frac{1}{Re}(u_{rr} + r^{-1}u_r + r^{-2}u_{\theta\theta} + u_{zz}) = 0$$

$$0 < r \leq 1, \quad 0 < \theta \leq 2\pi, \quad -1 \leq z \leq 1.$$

正確解：

$$u(r, \theta, z, t) = \frac{1.0}{1.0 + \exp\left(\frac{Re * (2.0 * x + 2.0 * y + 2.0 * z - 3.0 * t)}{4.0}\right)}$$

Re 為雷諾數。

設定基本參數：

N : r 方向의 自由度，取奇數

M : θ 方向의 自由度，取偶數

H : z 方向의 自由度

$iter$: 迭代次數總和

res_max : 最大 *residual norm*

err_max : 最大誤差

cpu_time : 測試所花的全部時間

由表77至表82可知其residual很快就可以達到指定的要求，但誤差似乎比處理Poisson時大，應該是時間項沒有算的很準的結果。由表79與表82知道隨時間的減少，誤差也會隨著變小，且無論使用分解法或迭代法，都可以到達相同的誤差值。

time	N	M	H	method	res_ max	err_ max	cpu_ t
1.00e-03	9	8	9	LU	6.85e-13	2.54e-04	0.19
9.00e-04	9	8	9	LU	6.85e-13	2.28e-04	0.19
8.00e-04	9	8	9	LU	6.85e-13	2.03e-04	0.19
7.00e-04	9	8	9	LU	6.85e-13	1.77e-04	0.19
6.00e-04	9	8	9	LU	6.85e-13	1.52e-04	0.19
5.00e-04	9	8	9	LU	6.85e-13	1.27e-04	0.19
4.00e-04	9	8	9	LU	6.85e-13	1.01e-04	0.19
3.00e-04	9	8	9	LU	6.85e-13	7.59e-05	0.19
2.00e-04	9	8	9	LU	6.85e-13	5.06e-05	0.19
1.00e-04	9	8	9	LU	6.85e-13	2.53e-05	0.19
iter_ sum = 40, nl_ sum = 40.							

表 77: 3D Burgers in a cylinder with $Re = 1.0$ and LU Method

time	N	M	H	method	res_ max	err_ max	cpu_ t
1.00e-03	9	8	9	LU	9.54e-13	2.59e-03	0.19
9.00e-04	9	8	9	LU	9.54e-13	2.34e-03	0.19
8.00e-04	9	8	9	LU	9.54e-13	2.08e-03	0.19
7.00e-04	9	8	9	LU	9.54e-13	1.82e-03	0.19
6.00e-04	9	8	9	LU	9.54e-13	1.56e-03	0.19
5.00e-04	9	8	9	LU	9.54e-13	1.30e-03	0.19
4.00e-04	9	8	9	LU	9.54e-13	1.04e-03	0.19
3.00e-04	9	8	9	LU	9.54e-13	7.83e-04	0.19
2.00e-04	9	8	9	LU	9.54e-13	5.23e-04	0.19
1.00e-04	9	8	9	LU	9.54e-13	2.62e-04	0.19
iter_ sum = 50, nl_ sum = 50.							

表 78: 3D Burgers in a cylinder with $Re = 10.0$ and LU Method

time	N	M	H	method	res_ max	err_ max	cpu_ t
1.00e-03	19	14	3	LU	1.10e-12	2.05e-02	0.61
1.00e-04	19	14	3	LU	9.63e-12	2.07e-03	0.61
1.00e-05	19	14	3	LU	6.19e-11	2.07e-04	0.61
1.00e-06	19	14	3	LU	1.01e-09	2.07e-05	0.61
1.00e-07	19	14	3	LU	8.81e-09	2.07e-06	0.62
1.00e-08	19	14	3	LU	6.56e-08	2.07e-07	0.61
1.00e-09	19	14	3	LU	1.10e-06	2.07e-08	0.62
1.00e-10	19	14	3	LU	8.70e-06	2.07e-09	0.61

表 79: 3D Burgers in a cylinder with $Re = 100$ and LU Method

time	N	M	H	method	res_ max	err_ max	cpu_ t
1.00e-03	9	8	9	GMRes	3.50e-13	2.54e-04	0.02
9.00e-04	9	8	9	GMRes	3.50e-13	2.28e-04	0.02
8.00e-04	9	8	9	GMRes	3.50e-13	2.03e-04	0.02
7.00e-04	9	8	9	GMRes	3.50e-13	1.77e-04	0.02
6.00e-04	9	8	9	GMRes	3.50e-13	1.52e-04	0.02
5.00e-04	9	8	9	GMRes	3.50e-13	1.27e-04	0.02
4.00e-04	9	8	9	GMRes	3.50e-13	1.01e-04	0.02
3.00e-04	9	8	9	GMRes	3.50e-13	7.59e-05	0.02
2.00e-04	9	8	9	GMRes	3.50e-13	5.06e-05	0.02
1.00e-04	9	8	9	GMRes	3.50e-13	2.53e-05	0.02
iter_ sum = 150, nl_ sum = 40.							

表 80: 3D Burgers in a cylinder with $Re = 1.0$ and GMRes Method

time	N	M	H	method	res_ max	err_ max	cpu_ t
1.00e-03	9	8	9	GMRes	4.32e-13	2.59e-03	0.02
9.00e-04	9	8	9	GMRes	4.32e-13	2.34e-03	0.02
8.00e-04	9	8	9	GMRes	4.32e-13	2.08e-03	0.02
7.00e-04	9	8	9	GMRes	4.32e-13	1.82e-03	0.02
6.00e-04	9	8	9	GMRes	4.32e-13	1.56e-03	0.02
5.00e-04	9	8	9	GMRes	4.32e-13	1.30e-03	0.02
4.00e-04	9	8	9	GMRes	4.32e-13	1.04e-03	0.02
3.00e-04	9	8	9	GMRes	4.32e-13	7.83e-04	0.02
2.00e-04	9	8	9	GMRes	4.32e-13	5.23e-04	0.02
1.00e-04	9	8	9	GMRes	4.32e-13	2.62e-04	0.02
iter_ sum = 140, nl_ sum = 40.							

表 81: 3D Burgers in a cylinder with $Re = 10.0$ and GMRes Method

time	N	M	H	method	res_ max	err_ max	cpu_ t
1.00e-03	19	14	3	GMRes	3.99e-13	2.05e-02	0.04
1.00e-04	19	14	3	GMRes	3.53e-12	2.07e-03	0.04
1.00e-05	19	14	3	GMRes	2.87e-11	2.07e-04	0.04
1.00e-06	19	14	3	GMRes	3.52e-10	2.07e-05	0.04
1.00e-07	19	14	3	GMRes	2.95e-09	2.07e-06	0.04
1.00e-08	19	14	3	GMRes	2.08e-08	2.07e-07	0.04
1.00e-09	19	14	3	GMRes	4.11e-07	2.07e-08	0.04
1.00e-10	19	14	3	GMRes	2.58e-06	2.07e-09	0.04

表 82: 3D Burgers in a cylinder with $Re = 100$ and GMRes Method

6 結論

本論文是採用頻譜配點法來近似求解偏微分方程式。由前兩章可以知道，無論是初始值問題或邊界條件問題，頻譜配點法對空間微分項的處理可以得到非常好的解。頻譜配點法的好處就是，其誤差呈現指數形式的遞減，故具有以較小的自由度便可達到需要的精確度。且頻譜法是採用正交基底的函數展開的整體近似法，對空間微分項的處理可得非常好的解，故不會有相速度的誤差。

關於頻譜配點法的應用，有下列幾點說明：

1. 本論文討論的定義域僅限於 $[-1, 1]$ ，可以利用適當的座標轉換函數(mapping function)將之應用的其他的定義域。
2. 本論中處理時間項是採用FTCS Method，可利用Crank-Nicolson方法增加準確度。
3. 本論文中未利用到快速傅立葉轉換(Fast Fouier transform)，若能加入此法，應能加快計算速度。
4. 本論文中可知道頻譜配點法的優勢，未來可將之應用在更多實際的問題。

參考文獻

- [1] Lloyd N. Trefethen (2000) : *Spectral Methods in MATLAB*, SIAM.
- [2] John P. Boyd (2001) : *Chebyshev and Fourier Spectral Methods*, Springer-Verlag.
- [3] Bengt Fornberg (1998) : *A Practical Guide to Pseudospectral Methods*, Cambridge University Press, Cambridge, UK.
- [4] S.A. Orszag (1971) : *Accurate solution of the Orr-Sommerfeld equation*, Journal of Fluid Mechanics 50, 689-703.
- [5] D. Gottlieb and S.A. Orszag (1977) : *Numerical analysis of spectral methods*, Society for Industrial and Applied Mathematics, Philadelphia.
- [6] P. Moin and J. Kim (1980) : *On the numerical solution of time-dependent viscous incompressible fluid flows involving solid boundaries*, Journal of Computational Physics 35, 3.
- [7] A.T. Patera (1984) : *A spectral element method for fluid dynamics: laminar flow in a channel expansion*, Journal of Computational Physics 54, 468-488.
- [8] Y. Sadd and M. Schultz (1986) : *A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput, 7 pp.856-869.
- [9] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T. Zang (1988) : *Spectral Methods in Fluid Dynamics*, Springer-Verlag, New York.
- [10] 郭登堯(1996) : 解一維震波問題，碩士論文，逢甲大學應用數學研究所。
- [11] 郭銘斌(2003) : 線性系統之巢式迭代解法，碩士論文，輔仁大學數學研究所。