

# 國立交通大學

應用數學系  
碩士論文

三維 Poisson 方程在圓柱及球座標下  
之形式四階緊緻差分法

A formally fourth-order compact scheme for  
Poisson equation in cylindrical and  
spherical coordinates

研究生：曾瑞閔

指導老師：賴明治 教授

中華民國九十五年一月

三維 Poisson 方程在圓柱及球座標下  
之形式四階緊緻差分法

A formally fourth-order compact scheme for Poisson  
equation in cylindrical and spherical coordinates

研究生：曾瑞閔

Student : Jui-Ming Tsung

指導教授：賴明治

Advisor : Ming-Chih Lai



Submitted to Department of Applied Mathematics

College of Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master

in

Applied Mathematics

January 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年一月

# 三維 Poisson 方程在圓柱及球座標下 之形式四階緊緻差分法

學生：曾瑞閔

指導老師：賴明治 教授

國立交通大學應用數學系(研究所)碩士班

## 摘 要

在這篇論文裡，將會介紹三維 Poisson 方程在圓柱及球座標下簡單且有效率的四階緊緻解法。這個解法是由截斷 (truncated) 傅利葉級數展開式所產生，且得到一組傅利葉係數所形成的偏微分方程組，運用緊緻差分技巧，我們可以得到四階精確且不需奇異點條件的結果。接著利用兩種有效的迭代法(GMRES, BI-CGSTAB)來解離散後，傅利葉係數所形成非對稱的線性系統並配合不同的 preconditioner。

# **A formally fourth-order compact scheme for Poisson equation in cylindrical and spherical coordinates**

**Student : Jui-Ming Tseng**

**Advisors : Dr. Ming-Chih Lai**

**Department ( Institute ) of Applied Mathematics  
National Chiao Tung University**

## **ABSTRACT**

A simple and efficient compact fourth-order Poisson solver in cylindrical and spherical coordinates is presented. The solver relies on the truncated Fourier series expansion, where the differential equations of Fourier coefficients have been solved by fourth-order finite difference discretizations without pole conditions. And two kinds of efficient iterative method, GMRES and Bi-CGSTAB, with different preconditioners are applied to solve the resulted nonsymmetrical systems of Fourier coefficients.

## 誌謝

研究所求學過程中，承蒙指導教授 賴明治老師於學業與論文上的教誨，使學生在學術研究、待人處事與生涯規劃上受益良多，並讓本論文得以順利完成，學生在此謹致最誠摯的謝意。

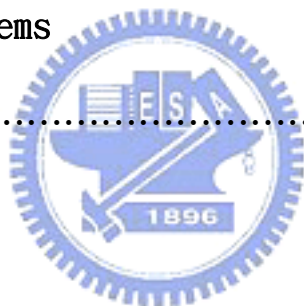
口試期間，承蒙吳金典教授、黃聰明教授及張書銘教授費心審閱並提供許多寶貴之意見，使本論文之完稿得以更加齊備，永誌於心。

由衷感謝父母親多年來無悔的養育與教導之恩，以及家人的照顧與關心，使我無後顧之憂的順利完成碩士學位。此外，在研究所求學過程中十分感謝研究室成員--曾昱豪學長、同學和學弟妹於課業上的協助與日常生活上的關懷與鼓勵，在此一併獻上誠摯感謝之意。

最後，謹將本論文之研究成果獻給我敬愛的父母親、家人以及所有關心我的人，共同分享這份喜悅與榮耀。

# 目 錄

中文提要	.....	i
英文提要	.....	ii
誌謝	.....	iii
目錄	.....	iv
1、	<b>Introduction</b>	1
2、	<b>Fast Poisson solver in cylindrical coordinates</b>	2
2.1	Fourier mode equations	2
2.2	Fourth-order finite difference discretization	3
3、	<b>Fast Poisson solver in spherical coordinates</b>	12
3.1	Fourier mode equations	13
3.2	Fourth-order finite difference discretization	14
4、	<b>Generalized Minimal Residual (GMRES)</b>	21
5、	<b>Bi-Conjugate Gradient Stabilized (Bi-CGSATB)</b>	26
6、	<b>The compact fourth-order scheme on polar geometry with Neumann problems</b>	33
7、	<b>Conclusions</b>	39
參考文獻	.....	39



# 1 Introduction

In many physical problems, one often needs to solve the Poisson equation on a non-Cartesian domain, such as polar or cylindrical or spherical domains. It is convenient to rewrite the equation in those coordinates. The first problem that must be dealt with is the coordinate singularities caused by the transformation. The singularities occur at the polar axis of those domains. It is important to note that the occurrence of those singularities is due to the representation of the governing equation in those coordinates.

Most of finite difference, finite volume and spectral methods in the literature (see Lai & Wang, 2002) need to either approximate the value of the solution or impose appropriate pole conditions for the solution at the singularities. This pole approximation provides a numerical boundary value for the finite difference scheme.

In Lai (2002), the author developed FFT-based fast direct solvers for Poisson equation on 2D polar geometry. The author uses the truncated Fourier series expansion to derive a set of singular ODEs for the Fourier coefficients, and then solves those singular equations by the compact fourth-order finite difference discretizations. By shifting a half mesh width from the origin, and incorporating with the derived symmetry constraint of Fourier coefficients, we can easily handle coordinate singularities without pole conditions. By manipulating the radial mesh width, three different boundary problems for polar geometry (Dirichlet, Neumann and Robin conditions) can be solved equally easily.

In this paper, we extend the previous fourth-order schemes on two dimensional cases (Lai, 2002) to the three-dimensional domains. Using the truncated Fourier series expansion, the original three-dimensional PDE now becomes a set of two-dimensional PDEs of the Fourier coefficients. Then we solve those PDEs by fourth-order finite difference discretizations.

In the following, we present two kinds of iterative method, GMRES and Bi-CGSTAB, to solve the nonsymmetric systems of two-dimensional PDEs of Fourier coefficients. Then some preconditioners can be used. In particular, a preconditioner arising from those singular equations have been solved by the second-order finite difference discretizations (see Lai *et al*, 2002) and shown to be the most efficient one.

## 2 Fast Poisson solver in cylindrical coordinates

The Poisson equation on a cylinder  $\Omega = \{0 < r \leq 1, 0 \leq \theta < 2\pi, 0 \leq z \leq 1\}$  can be conveniently written in cylindrical coordinates as

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} + \frac{\partial^2 u}{\partial z^2} = f(r, z, \theta) \quad (2.1)$$

For the sake of simplicity, we restrict the Dirichlet boundary conditions on the top and bottom boundaries,  $u(r, 1, \theta) = u_T(r, \theta)$ ,  $u(r, 0, \theta) = u_B(r, \theta)$ , but consider three different type of sidewall boundary conditions: Dirichlet  $u(1, z, \theta) = u_S(z, \theta)$ ; Neumann  $\frac{\partial u}{\partial r}(1, z, \theta) = u_S(z, \theta)$ ; or Robin condition  $\frac{\partial u}{\partial r} + \alpha u(1, z, \theta) = u_S(z, \theta)$ ,  $\alpha > 0$ .

The main issue for solving Eq.(2.1) is how to treat the coordinate singularity along the polar axis at the center  $r = 0$ . Most of Poisson solvers for Eq.(2.1) including finite difference and spectral methods (see Chen *et al.* (2000), Lai & Wang (2002)), involve imposing additional pole conditions to approximate accurately the solution in the vicinity of the origin. The accuracy of those methods depends greatly on the choice of pole conditions.

In the following, we develop a new class of fast direct solver for Eq.(2.1). Our approach relies on the truncated Fourier series expansion, where the differential equations of Fourier coefficients are solved by the fourth-order finite difference discretizations without pole condition.

### 2.1 Fourier mode equations

Since the solution  $u$  is periodic in  $\theta$ , we can approximate it by the truncated Fourier series as

$$u(r, z, \theta) = \sum_{n=-N/2}^{N/2-1} \hat{u}_n(r, z) e^{in\theta}, \quad (2.2)$$

where  $\hat{u}_n(r, z)$  is the complex Fourier coefficient given by

$$\hat{u}_n(r, z) = \frac{1}{N} \sum_{k=0}^{N-1} u(r, z, \theta_k) e^{-in\theta_k}, \quad (2.3)$$

and  $\theta_k = 2k\pi/N$ , and  $N$  is the number of grid points along a circle.



Substituting the expansions of (2.2) into Eq.(2.1), and equating the Fourier coefficients, we derive  $\hat{u}_n(r, z)$  satisfying the PDE

$$\frac{\partial^2 \hat{u}_n}{\partial r^2} + \frac{1}{r} \frac{\partial \hat{u}_n}{\partial r} + \frac{\partial^2 \hat{u}_n}{\partial z^2} - \frac{n^2}{r^2} \hat{u}_n = \hat{f}_n(r, z), 0 < r \leq 1, 0 \leq z \leq 1, \quad (2.4)$$

where the  $n$ th Fourier coefficient of the right-hand side function  $\hat{f}_n(r, z)$  is defined similarly as (2.3). The Fourier coefficients of the boundary values  $\hat{u}_S^n(z)$ ,  $\hat{u}_T^n(r)$ ,  $\hat{u}_B^n(r)$  are also defined in a similar fashion as to (2.3). So the remaining problem is to solve Eq.(2.4) with the top and bottom boundary conditions  $\hat{u}_n(r, 0) = \hat{u}_B^n(r)$ ,  $\hat{u}_n(r, 1) = \hat{u}_T^n(r)$ , and with one of the three sidewall boundary conditions  $\hat{u}_n(1, z) = \hat{u}_S^n(z)$ ,  $\frac{\partial \hat{u}_n}{\partial r}(1, z) = \hat{u}_S^n(z)$ , or  $\frac{\partial \hat{u}_n}{\partial r} + \alpha \hat{u}_n(1, z) = \hat{u}_S^n(z)$ .

## 2.2 Fourth-order finite difference discretization

We choose a grid in  $(r, z)$  plane to avoid the polar singularity by

$$r_i = (i - 1/2) \Delta r, \quad z_j = j \Delta z, \quad (2.5)$$

for  $1 \leq i \leq L + 1; 0 \leq j \leq M + 1$ , with  $\Delta r = 2/(2L + 1)$  and  $\Delta z = 1/(M + 1)$ . Let the discrete values be denoted by  $U(r_i, z_j) \approx \hat{u}_n(r_i, z_j)$ ,  $F(r_i, z_j) \approx \hat{f}_n(r_i, z_j)$ .

Our goal is to derive a fourth-order finite difference approximation to Eq.(2.4). Obviously, the first and second derivatives,  $U_r$ ,  $U_{rr}$  and  $U_{zz}$ , must be approximated to fourth-order accurately. First, let us write down two difference formulas for the first and second derivatives with the truncation errors  $O(\Delta r^4)$  and  $O(\Delta z^4)$ :

$$U_r = \delta_{0(r)} U_{ij} - \frac{\Delta r^2}{6} U_{rrr} + O(\Delta r^4), \quad (2.6)$$

$$U_{rr} = \delta_{(r)}^2 U_{ij} - \frac{\Delta r^2}{12} U_{rrrr} + O(\Delta r^4), \quad (2.7)$$

$$U_{zz} = \delta_{(z)}^2 U_{ij} - \frac{\Delta z^2}{12} U_{zzzz} + O(\Delta z^4). \quad (2.8)$$

Here  $\delta_{0(r)}U_{ij}$ ,  $\delta_{(r)}^2U_{ij}$  and  $\delta_{(z)}^2U_{ij}$  are the centered difference operators for the first and second derivatives, defined as

$$\delta_{0(r)}U_{ij} = \frac{U_{i+1,j} - U_{i-1,j}}{2\Delta r}, \delta_{(r)}^2U_{ij} = \frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{\Delta r^2},$$

$$\delta_{(z)}^2U_{ij} = \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{\Delta z^2}, \quad (2.9)$$

where  $U_{i,j}$  are the discrete values defined at the grid points  $r_i$  and  $z_j$ .

In order to have fourth-order approximations for  $U_r$ ,  $U_{rr}$  and  $U_{zz}$ , we need to approximate the higher order derivatives  $U_{rrr}$ ,  $U_{rrrr}$  and  $U_{zzzz}$  in Eqs.(2.6), (2.7) and (2.8) to be second-order accurate. To accomplish this, we differentiate Eq.(2.4) once and twice for radial and axial directions, respectively, to obtain the higher order derivatives of U :

$$U_{rrr} = F_r - \frac{U_{rr}}{r} + \frac{1+n^2}{r^2}U_r - \frac{2n^2}{r^3}U - U_{zzr}, \quad (2.10)$$

$$U_{rrrr} = F_{rr} - \frac{F_r}{r} + \frac{3+n^2}{r^2}U_{rr} - \frac{3+5n^2}{r^3}U_r + \frac{8n^2}{r^4}U + \frac{U_{zzr}}{r} - U_{zzrr}, \quad (2.11)$$

$$U_{zzz} = F_z - U_{rrz} - \frac{U_{rz}}{r} + \frac{n^2}{r^2}U_z, \quad (2.12)$$

$$U_{zzzz} = F_{zz} - U_{rrzz} - \frac{U_{rzz}}{r} + \frac{n^2}{r^2}U_{zz}. \quad (2.13)$$

In Eqs.(2.10), (2.11), (2.12), (2.13), those differential operators in right-hand side can be approximated further by the centered difference formulas to achieve second-order accuracy. Substituting those approximations into Eqs.(2.6), (2.7) and (2.8) then applying to Eq.(2.4), we obtain the finite difference scheme as follows. For  $1 \leq i \leq L, 1 \leq j \leq M$ , we need to solve

$$\begin{aligned} & \delta_{(r)}^2U_{i,j} - \frac{\Delta r^2}{12}[\delta_{(r)}^2F_{i,j} - \frac{1}{r_i}\delta_{0(r)}F_{i,j} + \frac{3+n^2}{r_i^2}\delta_{(r)}^2U_{i,j} - \frac{3+5n^2}{r_i^3}\delta_{0(r)}U_{i,j} \\ & + \frac{8n^2}{r_i^4}U_{i,j} + \frac{1}{r_i}\delta_{0(r)}\delta_{(z)}^2U_{i,j} - \delta_{(r)}^2\delta_{(z)}^2U_{i,j}] + \frac{1}{r_i}\delta_{0(r)}U_{i,j} - \frac{\Delta r^2}{6r_i}[\delta_{0(r)}F_{i,j} \\ & - \frac{1}{r_i}\delta_{(r)}^2U_{i,j} + \frac{1+n^2}{r_i^2}\delta_{0(r)}U_{i,j} - \frac{2n^2}{r_i^3}U_{i,j} - \delta_{0(r)}\delta_{(z)}^2U_{i,j}] - \frac{n^2}{r_i^2}U_{i,j} + \delta_{(z)}^2U_{i,j} \end{aligned}$$

$$-\frac{\Delta z^2}{12}[\delta_{(z)}^2 F_{i,j} - \delta_{(z)}^2 \delta_{(r)}^2 U_{i,j} - \frac{1}{r_i} \delta_{(z)}^2 \delta_{0(r)} U_{i,j} + \frac{n^2}{r_i^2} \delta_{(z)}^2 U_{i,j}] = F_{i,j}. \quad (2.14)$$

In order to close the linear system, the numerical boundary values  $U_{0,j}$  and  $U_{L+1,j}$  in the  $r$  direction should be supplied. Choosing of  $r_i$  as described in (2.5), we have  $r_{L+1} = 1$ ; thus, the numerical boundary value  $U_{L+1,j}$  can either be given by the Dirichlet boundary value  $\hat{u}_S^n(z_j)$  or be determined by imposing the condition on the boundary (Neumann and Robin). The numerical boundary value  $U_{0,j}$  can be obtained by the symmetry constraint of Fourier coefficients, which is derived as follows. The transformation between Cartesian and cylindrical coordinates can be written as  $x = r \cos \theta$ ,  $y = r \sin \theta$ ,  $z = z$ . When we replace  $r$  with  $-r$ , and  $\theta$  with  $\theta + \pi$ , the Cartesian coordinates of a point remain the same. Therefore, any scalar function  $u(r, \theta, z)$  satisfies  $u(-r, \theta, z) = u(r, \theta + \pi, z)$ . Using this equality, we have

$$\begin{aligned} u(-r, \theta, z) &= \sum_{n=-\infty}^{\infty} \hat{u}_n(-r, z) e^{in\theta} = \sum_{n=-\infty}^{\infty} \hat{u}_n(r, z) e^{in(\theta+\pi)} \\ &= \sum_{n=-\infty}^{\infty} \hat{u}_n(r, z) e^{in\theta} e^{in\pi} = \sum_{n=-\infty}^{\infty} (-1)^n \hat{u}_n(r, z) e^{in\theta}. \end{aligned} \quad (2.15)$$

Thus, when the domain of a function is extended to a negative value of  $r$ , the  $n$ th Fourier coefficient of this function satisfies  $\hat{u}_n(-r, z) = (-1)^n \hat{u}_n(r, z)$ . Using the above condition, we have

$$\begin{aligned} U_{0,j} &= U(r_0, z_j) = U\left(-\frac{\Delta r}{2}, z_j\right) = (-1)^n U\left(\frac{\Delta r}{2}, z_j\right) \\ &= (-1)^n U(r_1, z_j) = (-1)^n U_{1,j}. \end{aligned} \quad (2.16)$$

Therefore, the numerical boundary value  $U_{0,j}$  has been supplied. And the numerical boundary values in the  $z$  direction can be easily obtained by the given Dirichlet boundary values  $U_{i,0} = \hat{u}_B^n(r_i)$  and  $U_{i,M+1} = \hat{u}_T^n(r_i)$ .

Let us order the unknowns  $U_{ij}$  by first grouping the same  $i$  so that the solution vector  $v$  is defined by

$$v = \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_L \end{bmatrix}, \quad U_i = \begin{bmatrix} U_{i1} \\ U_{i2} \\ \vdots \\ U_{iM} \end{bmatrix}.$$

Solving the discrete equations (2.14) results in a large sparse linear system  $Av = b$ , where the coefficient matrix  $A$  and the right-hand side vector  $b$  are defined as follows. The matrix  $A$  is a  $L \times L$  block tridiagonal matrix

$$A = \begin{bmatrix} T_1 & I_1 & & & & & \\ H_2 & T_2 & I_2 & & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & & & \\ & & & & H_{L-1} & T_{L-1} & I_{L-1} \\ & & & & & H_L & T_L \end{bmatrix},$$

where  $T_i, H_i$  and  $I_i$ ,  $1 \leq i \leq L$  are the tridiagonal matrices given by

$$H_i = \begin{bmatrix} H1_i & H2_i & & & & & \\ H2_i & H1_i & H2_i & & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & H2_i & H1_i & H2_i \\ & & & & & H2_i & H1_i \end{bmatrix},$$

$$H1_i = \frac{5a}{6} - \frac{b}{6} - (1+n^2)c_i - (1+3n^2)d_i + 2e_i - 10f_i,$$

$$H2_i = \frac{a}{12} + \frac{b}{12} - e_i - f_i,$$

$$I_i = \begin{bmatrix} I1_i & I2_i & & & & & \\ I2_i & I1_i & I2_i & & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & & & \\ & & & & I2_i & I2_i & I2_i \\ & & & & & I2_i & I1_i \end{bmatrix},$$

$$I1_i = \frac{5a}{6} - \frac{b}{6} - (1+n^2)c_i + (1+3n^2)d_i - 2e_i + 10f_i,$$

$$I2_i = \frac{a}{12} + \frac{b}{12} + e_i + f_i,$$



For the Neumann ( $\alpha = 0$ ) or Robin boundary cases, we use the same mesh points but with different radial mesh width  $\Delta r = 1/L$ . With this choice of radial mesh width, the discrete values of  $U$  are defined midway between sidewall boundary so that the first derivative can be centered on the grid points. That is, at  $r = 1$ ,

$$\frac{\partial U}{\partial r} + \alpha U \approx \frac{U_{L+1,j} - U_{L,j}}{\Delta r} + \alpha \frac{U_{L+1,j} + U_{L,j}}{2} = \hat{u}_S^n(z_j). \quad (2.17)$$

The numerical boundary value  $U_{L+1,j}$  can be approximated by

$$U_{L+1,j} = \frac{(1 - \alpha\Delta r/2)U_{L,j} + \hat{u}_S^n(z_j)\Delta r}{1 + \alpha\Delta r/2}. \quad (2.18)$$

Therefore, we only need to modify  $T_L$  in the matrix  $A$  and  $b_{L,j}^T$  in the vector  $b$  by

$$\begin{aligned} T_L &= T_L + \gamma I_L, \\ b_{L,j}^T &= b_{L,j}^T - \zeta(I1_L \hat{u}_S^n(Z_j) - I2_L(\hat{u}_S^n(Z_{j-1}) + \hat{u}_S^n(Z_{j+1}))) - b1^T, \end{aligned} \quad (2.19)$$

with  $b1 = \gamma I2_L(\hat{u}_B^n(r_L), 0, \dots, 0, \hat{u}_T^n(r_L))$ ,  $\gamma = \frac{1 - \alpha\frac{\Delta r}{2}}{1 + \alpha\frac{\Delta r}{2}}$  and  $\zeta = \frac{\Delta r}{1 + \alpha\frac{\Delta r}{2}}$ .

Table 2.1 shows the maximum errors of the method for three different solutions of Poisson equation in a cylinder with Dirichlet boundary condition. In all our tests, we use  $L$  mesh points in the radial and axial directions, and  $2L$  points in the azimuthal direction. The rate of convergence is computed by the formula  $\log_2(\frac{E_{L/2}}{E_L})$ , where  $E_L$  is the maximum error. One can see that the errors of the solutions show third-order convergence for all solutions. The loss of one order of accuracy seems to come from the discretization near the origin. This can be seen from the following truncation error analysis. In the Fourier mode equation Eq.(2.4), the  $U'$  ( $= \frac{\partial \hat{u}_m}{\partial r}$ ) term is divided by  $r$ . So the second-order approximation of  $U'''$  in (2.6) is divided by an  $O(\Delta r)$  term near the origin, which makes the approximation of  $U'''/r$  first-order accurate. This has the consequence that the overall truncation error of the  $U'/r$  term in the vicinity of the origin is  $O(\Delta r^3)$  and thus so is the Fourier mode equation (2.4). However, this loss of accuracy does not appear when solving the problem on a hollow cylinder. Let us explain why that is the case next. The present scheme can be easily applied to solve the Poisson equation on a hollow cylinder  $\{a \leq r \leq b\}$ , where  $a > 0$ . As the cylinder case, we need

to solve Eq.(2.4) and three different boundary conditions at  $r = b$  with an additional boundary condition imposed at  $r = a$ . Instead of setting a grid as in (2.5), we choose a regular grid,

$$r_i = a + i\Delta r, \quad i = 0, 1, 2, \dots, L, L + 1, \quad (2.21)$$

with the mesh width  $\Delta r = (b - a)/(L + 1)$ . Now the second-order approximation of  $U'''$  in (2.6) is divided by an  $O(a + \Delta r)$  term instead of an  $O(\Delta r)$  term, so the truncation error of the  $U'/r$  term is still  $O(\Delta r^2)$ . Therefore, the overall truncation error of Eq.(2.4) is  $O(\Delta r^4)$ .

The fourth and fifth columns of Table 2.1 show the errors and the rate of convergence for the solutions on a hollow cylinder  $\{0.5 \leq r \leq 1\}$ . We can see that the fourth-order convergence can be achieved for all examples. In the following, the Table 2.2 and 2.3 show the maximum errors of the method for three different solutions of Poisson equation in a cylinder and hollow cylinder with remaining boundary conditions (Neumann and Robin).



**Table 2.1**  
**The Maximum Errors of Different Solutions to the**  
**Poisson Equation with Dirichlet Boundary**

L	$0 < r \leq 1$		$0.5 < r \leq 1$	
	$\ u\ _\infty$	Rate	$\ u\ _\infty$	Rate
$u(r, z, \theta) = e^{r \cos \theta + r \sin \theta + z}$				
8	7.8137E-05		1.5465E-07	
16	9.8506E-06	2.99	1.0920E-08	3.82
32	1.2566E-06	2.97	7.3409E-10	3.90
64	1.5941E-07	2.98	4.7580E-11	3.95
$u(r, z, \theta) = r^3(\cos \theta + \sin \theta)z(1 - z)$				
8	9.1438E-04		8.8994E-07	
16	1.0755E-04	3.09	6.4128E-08	3.80
32	1.3008E-05	3.05	4.3173E-09	3.89
64	1.5966E-06	3.03	2.8035E-10	3.95
$u(r, z, \theta) = \cos(\pi(r^2 \cos^2 \theta + r \sin \theta)) \sin(\pi z^2)$				
8	7.4000E-03		7.5000E-03	
16	3.3150E-04	4.48	1.7101E-05	8.78
32	4.0782E-05	3.02	1.2221E-06	3.81
64	5.0424E-06	3.02	8.1011E-08	3.92



**Table 2.2**  
**The Maximum Errors of Different Solutions to the**  
**Poisson Equation with Neumann Boundary**

L	$0 < r \leq 1$		$0.5 < r \leq 1$	
	$\ u\ _\infty$	Rate	$\ u\ _\infty$	Rate
$u(r, z, \theta) = e^{r \cos \theta + r \sin \theta + z}$				
8	4.2863E-04		9.3911E-05	
16	1.0817E-04	1.99	2.4471E-05	1.94
32	2.6993E-05	2.00	6.2247E-06	1.96
64	6.7313E-06	2.00	1.5682E-06	1.99
$u(r, z, \theta) = r^3(\cos \theta + \sin \theta)z(1 - z)$				
8	1.4000E-03		2.9138E-04	
16	3.5261E-04	1.99	7.7016E-05	1.92
32	8.8215E-05	2.00	1.9828E-05	1.96
64	2.2041E-05	2.00	5.0323E-06	1.99
$u(r, z, \theta) = \cos(\pi(r^2 \cos^2 \theta + r \sin \theta)) \sin(\pi z^2)$				
8	2.4900E-02		2.0200E-02	
16	4.1000E-03	2.60	1.1000E-03	4.20
32	1.1000E-03	1.90	2.8932E-04	1.93
64	3.0123E-04	1.87	7.5139E-05	1.95

**Table 2.3**  
**The Maximum Errors of Different Solutions to the**  
**Poisson Equation with Robin Boundary ( $\alpha = 1$ )**

L	$0 < r \leq 1$		$0.5 < r \leq 1$	
	$\ u\ _\infty$	Rate	$\ u\ _\infty$	Rate
$u(r, z, \theta) = e^{r \cos \theta + r \sin \theta + z}$				
8	1.0000E-03		2.3024E-04	
16	2.5937E-04	1.95	5.9649E-05	1.95
32	6.4653E-05	2.00	1.5131E-05	1.98
64	1.6112E-05	2.00	3.8114E-06	1.99
$u(r, z, \theta) = r^3(\cos \theta + \sin \theta)z(1 - z)$				
8	4.2000E-03		8.7050E-04	
16	1.0000E-03	2.07	2.3109E-04	1.91
32	2.6050E-04	1.94	5.9561E-05	1.96
64	6.5035E-05	2.00	1.5121E-05	1.98
$u(r, z, \theta) = \cos(\pi(r^2 \cos^2 \theta + r \sin \theta)) \sin(\pi z^2)$				
8	2.3100E-02		1.9100E-02	
16	5.3000E-03	2.12	1.3000E-03	3.88
32	1.6000E-03	1.73	3.9232E-04	1.73
64	4.1390E-04	1.95	1.0176E-04	1.95

### 3 Fast Poisson solver in spherical coordinates

The Poisson equation in a spherical shell  $\Omega = \{R_0 \leq r \leq 1, 0 \leq \phi \leq \pi, 0 \leq \theta \leq 2\pi\}$  can be written in spherical coordinates as

$$\frac{\partial^2 u}{\partial r^2} + \frac{2}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \phi^2} + \frac{\cot \phi}{r^2} \frac{\partial u}{\partial \phi} + \frac{1}{r^2 \sin^2 \phi} \frac{\partial^2 u}{\partial \theta^2} = f(r, \phi, \theta). \quad (3.1)$$

The boundary condition should be imposed on the inner ( $r = R_0 > 0$ ) and outer ( $r = 1$ ) surfaces of the sphere. Here, for convenience of exposition, we assume the Dirichlet boundary on the inner surface  $u(R_0, \phi, \theta) = u_I(\phi, \theta)$ . Three different boundary conditions can be considered on the outer surface: Dirichlet  $u(1, \phi, \theta) = u_S(\phi, \theta)$ ; Neumann  $\frac{\partial u}{\partial r}(1, \phi, \theta) = u_S(\phi, \theta)$ ; or Robin condition  $\frac{\partial u}{\partial r} + \alpha u(1, \phi, \theta) = u_S(\phi, \theta)$ ,  $\alpha > 0$ . However, the method to be

described can be easily adapted to different boundary conditions on the inner surface.

As in the cylindrical case of the previous section, the main difficulty for solving Eq.(3.1) is to treat the coordinate singularities along the polar axis where north ( $\phi = 0$ ) and south ( $\phi = \pi$ ) poles are located. Again, most of numerical approaches including finite difference and spectral methods involve imposing additional pole conditions to capture the behavior of the solution in the vicinity of the poles. In the following, we will present a numerical method to solve Eq.(3.1) which uses the symmetry constraint of Fourier coefficient to handle the coordinate singularities without pole condition.

### 3.1 *Fourier mode equations*

As in the cylindrical coordinate case, we approximate  $u$  by the truncated Fourier series as

$$u(r, \phi, \theta) = \sum_{n=-N/2}^{N/2-1} \hat{u}_n(r, \phi) e^{in\theta}, \quad (3.2)$$

where  $\hat{u}_n(r, \phi)$  is the complex Fourier coefficient given by

$$\hat{u}_n(r, \phi) = \frac{1}{N} \sum_{k=0}^{N-1} u(r, \phi, \theta_k) e^{-in\theta_k}, \quad (3.3)$$

and  $\theta_k = 2k\pi/N$  and  $N$  is the number of grid points along a latitude circle. The expansion for the function  $f$  can be written in the similar fashion. Substituting those expansions into Eq.(3.1), and equating the Fourier coefficients,  $\hat{u}_n(r, \phi)$  then satisfies the PDE

$$\frac{\partial^2 \hat{u}_n}{\partial r^2} + \frac{2}{r} \frac{\partial \hat{u}_n}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \hat{u}_n}{\partial \phi^2} + \frac{\cot \phi}{r^2} \frac{\partial \hat{u}_n}{\partial \phi} - \frac{n^2}{r^2 \sin^2 \phi} \hat{u}_n = \hat{f}_n(r, \phi), \quad (3.4)$$

with  $\hat{u}_n(R_0, \phi) = \hat{u}_I^n(\phi)$  and one of the three boundary conditions: Dirichlet  $\hat{u}_n(1, \phi) = \hat{u}_S^n(\phi)$ ; Neumann  $\frac{\partial \hat{u}_n}{\partial r}(1, \phi) = \hat{u}_S^n(\phi)$ ; or Robin condition  $\frac{\partial \hat{u}_n}{\partial r} + \alpha \hat{u}_n(1, \phi) = \hat{u}_S^n(\phi)$ . Here,  $\hat{u}_I^n(\phi)$  and  $\hat{u}_S^n(\phi)$  are the  $n$ th Fourier coefficient of  $u_I(\phi, \theta)$  and  $u_S(\phi, \theta)$ , respectively.

### 3.2 Fourth-order finite difference discretization

We consider the Dirichlet boundary on the outer surface first and will discuss the other cases later. Let us choose a grid in  $(r, \phi)$  plane by

$$r_i = R_0 + i \Delta r, \quad \phi_j = (j - 1/2) \Delta \phi, \quad (3.5)$$

for  $0 \leq i \leq L+1, 0 \leq j \leq M+1$  with  $\Delta r = (1 - R_0)/(L+1)$  and  $\Delta \phi = \pi/M$ . By the choice of those mesh points, we avoid placing points directly at north ( $\phi = 0$ ) and south ( $\phi = \pi$ ) poles. Again, let the discrete values be denoted by  $U(r_i, \phi_j) \approx \hat{u}_n(r_i, \phi_j)$ , and  $F(r_i, \phi_j) \approx \hat{f}_n(r_i, \phi_j)$ .

Our goal is to derive a fourth-order finite difference approximation to Eq.(3.4). As in the cylindrical coordinate case, we obtain the finite difference scheme as follows. For  $1 \leq i \leq L, 1 \leq j \leq M$ , we need to solve

$$\begin{aligned} & \delta_{(r)}^2 U_{i,j} - \frac{\Delta r^2}{12} [\delta_{(r)}^2 F_{i,j} - \frac{2}{r_i} \delta_{0(r)} F_{i,j} + (\frac{8 + n^2 \csc^2 \phi_j}{r_i^2}) \delta_{(r)}^2 U_{i,j} + \frac{10n^2 \csc^2 \phi_j}{r_i^4} U_{i,j} \\ & + (\frac{-8 - 6n^2 \csc^2 \phi_j}{r_i^3}) \delta_{0(r)} U_{i,j} - \frac{10}{r_i^4} \delta_{(\phi)}^2 U_{i,j} - \frac{10 \cot \phi_j}{r_i^4} \delta_{0(\phi)} U_{i,j} + \frac{6 \cot \phi_j}{r_i^3} \delta_{0(r)} \delta_{0(\phi)} U_{i,j} \\ & + \frac{6}{r_i^3} \delta_{0(r)} \delta_{(\phi)}^2 U_{i,j} - \frac{\cot \phi_j}{r_i^2} \delta_{(r)}^2 \delta_{0(\phi)} U_{i,j} - \frac{1}{r_i^2} \delta_{(r)}^2 \delta_{(\phi)}^2 U_{i,j}] + \frac{2}{r_i} \{ \delta_{0(r)} U_{i,j} - \frac{\Delta r^2}{6} [\delta_{0(r)} F_{i,j} \\ & - \frac{2}{r_i} \delta_{(r)}^2 U_{i,j} + (\frac{2 + n^2 \csc^2 \phi_j}{r_i^2}) \delta_{0(r)} U_{i,j} + \frac{2 \cot \phi_j}{r_i^3} \delta_{0(\phi)} U_{i,j} + \frac{2}{r_i^3} \delta_{(\phi)}^2 U_{i,j} - \frac{\cot \phi_j}{r_i^2} \delta_{0(r)} \delta_{0(\phi)} U_{i,j} \\ & - \frac{1}{r_i^2} \delta_{0(r)} \delta_{(\phi)}^2 U_{i,j} - \frac{2n^2 \csc^2 \phi_j}{r_i^3} U_{i,j}] \} + \frac{1}{r_i^2} \{ \delta_{(\phi)}^2 U_{i,j} - \frac{\Delta \phi^2}{12} [r_i^2 \delta_{(\phi)}^2 F_{i,j} - r_i^2 \cot \phi_j \delta_{0(\phi)} F_{i,j} \\ & + (-3 - 5n^2) \csc^2 \phi_j \cot \phi_j \delta_{0(\phi)} U_{i,j} + 2n^2 \csc^2 \phi_j (4 \csc^2 \phi_j - 3) U_{i,j} + 2r_i \cot \phi_j \delta_{0(\phi)} \delta_{0(r)} U_{i,j} \\ & + ((3 + n^2) \csc^2 \phi_j - 1) \delta_{(\phi)}^2 U_{i,j} + r_i^2 \cot \phi_j \delta_{0(\phi)} \delta_{(r)}^2 U_{i,j} - 2r_i \delta_{(\phi)}^2 \delta_{0(r)} U_{i,j} - r_i^2 \delta_{(\phi)}^2 \delta_{(r)}^2 U_{i,j}] \} \end{aligned}$$

$$\begin{aligned}
& + \frac{\cot \phi_j}{r_i^2} \left\{ \delta_{0(\phi)} U_{i,j} - \frac{\Delta \phi^2}{6} [-r_i^2 \delta_{0(\phi)} \delta_{(r)}^2 U_{i,j} + (1+n^2) \csc^2 \phi_j \delta_{0(\phi)} U_{i,j} - \cot \phi_j \delta_{(\phi)}^2 U_{i,j} \right. \\
& \quad \left. - 2n^2 \csc^2 \phi_j \cot \phi_j U_{i,j} + r_i^2 \delta_{0(\phi)} F_{i,j}] \right\} - \frac{n^2 \csc^2 \phi_j}{r_i^2} U_{i,j} = F_{i,j}. \quad (3.6)
\end{aligned}$$

When  $j = 1$  for Eq.(3.6), the numerical boundary value  $U_{i,0}$  can be given by  $U_{i,0} = (-1)^n U_{i,1}$ . This is because the Fourier coefficient satisfies the symmetry constraint  $\hat{u}_n(r_i, -\Delta\phi/2) = (-1)^n \hat{u}_n(r_i, \Delta\phi/2)$  (Lai & Wang, 2002). Similarly, another numerical boundary value  $U_{i,M+1}$  can also be obtained by  $U_{i,M+1} = (-1)^n U_{i,M}$  for the same reason. So the numerical boundary values in the  $\phi$  direction are provided and no pole condition is needed in our finite difference setting. The numerical boundary values in the radial direction  $U_{0,j}, U_{L+1,j}$  are given by the boundary values  $\hat{u}_I^n(\phi_j), \hat{u}_S^n(\phi_j)$ .

Let us order the unknowns  $U_{ij}$  by first grouping the same  $i$  so that the solution vector  $v$  is defined by

$$v = \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_L \end{bmatrix}, \quad U_i = \begin{bmatrix} U_{i1} \\ U_{i2} \\ \vdots \\ U_{iM} \end{bmatrix}.$$

The remaining problem is to solve a large sparse linear system  $Av = b$ , where the coefficient matrix  $A$  and the right-hand side vector  $b$  are defined as follows. The matrix  $A$  is a  $L \times L$  block tridiagonal matrix

$$A = \begin{bmatrix} T_1 & I_1 & & & & & \\ H_2 & T_2 & I_2 & & & & \\ & \cdot & \cdot & \cdot & & & \\ & & & & H_{L-1} & T_{L-1} & I_{L-1} \\ & & & & & H_L & T_L \end{bmatrix},$$

where  $T_i, H_i$  and  $I_i$ ,  $1 \leq i \leq L$  are the tridiagonal matrices given by

$$T_i = \begin{bmatrix} T1_{i,1} + (-1)^n T3_{i,1} & T2_{i,1} & & & & & \\ & T3_{i,2} & T1_{i,2} & T2_{i,2} & & & \\ & & & \cdot & \cdot & \cdot & \\ & & & & \cdot & \cdot & \\ & & & & & \cdot & \\ & & & & & & T3_{i,M-1} & T1_{i,M-1} & T2_{i,M-1} \\ & & & & & & & T3_{i,M} & T1_{i,M} + (-1)^n T2_{i,M} \end{bmatrix},$$



with

$$\begin{aligned}
P1 &= \frac{1}{12\Delta r^2}, P2_i = \frac{1}{12r_i\Delta r}, P3_{i,j} = \frac{n^2 \csc^2 \phi_j}{12r_i^2}, P4_{i,j} = \frac{\Delta r n^2 \csc^2 \phi_j}{12r_i^3}, \\
P5_i &= \frac{\Delta r}{12r_i^3\Delta\phi^2}, P6_i = \frac{1}{6r_i^2\Delta\phi^2}, P7_{i,j} = \frac{\Delta r^2 n^2 \csc^2 \phi_j}{6r_i^4}, \\
P8_{i,j} &= \frac{\Delta\phi^2 n^2 \csc^4 \phi_j}{3r_i^2}, P9_i = \frac{\Delta r^2}{6r_i^4\Delta\phi^2}, P10_{i,j} = \frac{\csc^2 \phi_j}{6r_i^2}, \\
P11_{i,j} &= \frac{\Delta r^2 \cot \phi_j}{12r_i^4\Delta\phi}, P12_{i,j} = \frac{5 \cot \phi_j}{12r_i^2\Delta\phi}, P13_{i,j} = \frac{(1+3n^2)(\Delta\phi \cot \phi_j \csc^2 \phi_j)}{24r_i^2}, \\
P14_j &= \frac{\Delta\phi \cot \phi_j}{12\Delta r^2}, P15_{i,j} = \frac{\Delta r \cot \phi_j}{24r_i^3\Delta\phi}, P16_{i,j} = \frac{\Delta\phi \cot \phi_j}{24r_i\Delta r}, 1 \leq j \leq M.
\end{aligned}$$

Incorporating with the boundary value and the function  $F$ , the right-hand side vector  $b$  can be written as

$$b = \begin{bmatrix} b_{1,j}^T - H3_{1,j} \hat{u}_I^n(\phi_{j-1}) - H1_{1,j} \hat{u}_I^n(\phi_j) - H2_{1,j} \hat{u}_I^n(\phi_{j+1}) \\ \vdots \\ b_{i,j}^T \\ \vdots \\ b_{L,j}^T - I3_{M,j} \hat{u}_s^n(\phi_{j-1}) - I1_{M,j} \hat{u}_s^n(\phi_j) - I2_{M,j} \hat{u}_s^n(\phi_{j+1}) \end{bmatrix},$$

where

$$\begin{aligned}
b_{i,j} &= \frac{2}{3}F_{i,j} + \left(\frac{1}{12} + P17_i\right)F_{i+1,j} + \left(\frac{1}{12} - P17_i\right)F_{i-1,j} \\
&\quad + \left(\frac{1}{12} + \frac{\Delta r^2 P14_j}{2}\right)F_{i,j+1} + \left(\frac{1}{12} - \frac{\Delta r^2 P14_j}{2}\right)F_{i,j-1},
\end{aligned}$$

with  $P17_i = \frac{\Delta r}{12r_i}$ ,  $1 \leq i \leq L$  and  $1 \leq j \leq M$ .

For the Neumann ( $\alpha = 0$ ) or Robin boundary cases, we use the same grid described in (3.5) but with different radial mesh width  $\Delta r = 2(1 - R_0)/(2L + 1)$ . With this choice of radial mesh width, the discrete values of  $U$  are defined midway between boundary so that the first derivative can be centered on the mesh points. That is, at  $r = 1$ ,

$$\frac{\partial U}{\partial r} + \alpha U \approx \frac{U_{L+1,j} - U_{L,j}}{\Delta r} + \alpha \frac{U_{L+1,j} + U_{L,j}}{2} = \hat{u}_s^n(\phi_j). \quad (3.7)$$

So the numerical boundary value  $U_{L+1,j}$  can be approximated by

$$U_{L+1,j} = \frac{(1 - \alpha\Delta r/2)U_{L,j} + \hat{u}_s^n(\phi_j)\Delta r}{1 + \alpha\Delta r/2}. \quad (3.8)$$

Therefore, we only need to modify  $T_L$  in the matrix  $A$  and the vector  $b_{L,j}^T$  by

$$\begin{aligned} T_L &= T_L + \gamma I_L, \\ b_{L,j}^T &= b_{L,j}^T - \zeta(I3_{M,j}\hat{u}_s^n(\phi_{j-1}) - I1_{M,j}\hat{u}_s^n(\phi_j) - I2_{M,j}\hat{u}_s^n(\phi_{j+1})), \end{aligned} \quad (3.9)$$

with  $\zeta = \frac{\Delta r}{1 + \alpha\frac{\Delta r}{2}}$ , and  $\gamma = \frac{1 - \alpha\frac{\Delta r}{2}}{1 + \alpha\frac{\Delta r}{2}}$ .

Table 3.1 shows the maximum errors of the method for three different solutions of Poisson equation in a spherical shell with Dirichlet boundary condition. In all our tests, we use  $L$  mesh points in the radial and colatitude directions, and  $2L$  points in the longitude direction. The inner radius is chosen by  $R_0 = 0.5$ .

One can see that the errors of the solutions show third-order convergence for all solutions. The loss of one order of accuracy seems to come from the discretization near the north ( $\phi = 0$ ) pole. This can be seen from the following truncation error analysis. In the Fourier mode equation Eq.(3.4), the  $U' (= \frac{\partial \hat{u}_n}{\partial \phi})$  term is divided by  $\sin \phi$ . So the fourth-order approximation of  $U'$  in (3.4) is divided by an  $O(\Delta\phi)$  term near the north ( $\phi = 0$ ) pole. This has the consequence that the overall truncation error of the  $\frac{\cot \phi}{r^2}U'$  term in the vicinity of the north ( $\phi = 0$ ) pole is  $O(\Delta\phi^3)$  and thus so is the Fourier mode equation (3.4). In the following, the Table 3.2 and 3.3 show the maximum errors of the method for three different solutions of Poisson equation in a spherical shell with remaining boundary conditions (Neumann and Robin).



**Table 3.1**  
**The Maximum Errors of Different Solutions to the**  
**Poisson Equation with Dirichlet Boundary**

$0.5 < r \leq 1$		
L	$\ u\ _\infty$	Rate
$u(r, \phi, \theta) = e^{r \sin \phi \cos \theta + r \sin \phi \sin \theta + r \cos \phi}$		
8	3.7000E-03	
16	5.4095E-04	2.77
32	7.4825E-05	2.85
64	1.0067E-05	2.89
$u(r, \phi, \theta) = r^3(\cos \theta + \sin \theta) \sin \phi(1 - r \cos \phi)$		
8	5.2000E-03	
16	9.5143E-04	2.45
32	1.5249E-04	2.64
64	2.2549E-05	2.76
$u(r, \phi, \theta) = \cos(\pi(r^2 \cos^2 \theta \sin^2 \phi + r \sin \theta \sin \phi)) \sin(\pi r^2 \cos^2 \phi)$		
8	5.3700E-02	
16	7.1000E-03	2.92
32	8.3568E-04	3.09
64	9.7240E-05	3.10

**Table 3.2**  
**The Maximum Errors of Different Solutions to the**  
**Poisson Equation with Neumann Boundary**

$0.5 < r \leq 1$		
L	$\ u\ _\infty$	Rate
$u(r, \phi, \theta) = e^{r \sin \phi \cos \theta + r \sin \phi \sin \theta + r \cos \phi}$		
8	6.8000E-03	
16	8.5296E-04	3.00
32	1.0804E-04	2.98
64	1.4317E-05	2.92
$u(r, \phi, \theta) = r^3(\cos \theta + \sin \theta) \sin \phi(1 - r \cos \phi)$		
8	1.0500E-02	
16	1.6000E-03	2.71
32	2.2105E-04	2.86
64	2.9431E-05	2.91
$u(r, \phi, \theta) = \cos(\pi(r^2 \cos^2 \theta \sin^2 \phi + r \sin \theta \sin \phi)) \sin(\pi r^2 \cos^2 \phi)$		
8	1.3090E-01	
16	1.6500E-02	2.99
32	2.2000E-03	2.91
64	3.8002E-04	2.53

**Table 3.3**  
**The Maximum Errors of Different Solutions to the**  
**Poisson Equation with Robin Boundary ( $\alpha = 1$ )**

$0.5 < r \leq 1$		
L	$\ u\ _\infty$	Rate
$u(r, \phi, \theta) = e^{r \sin \phi \cos \theta + r \sin \phi \sin \theta + r \cos \phi}$		
8	6.0000E-03	
16	8.0381E-04	2.90
32	1.0805E-04	2.90
64	1.5225E-05	2.83
$u(r, \phi, \theta) = r^3(\cos \theta + \sin \theta) \sin \phi(1 - r \cos \phi)$		
8	9.6000E-03	
16	1.5000E-03	2.68
32	2.1788E-04	2.78
64	3.3247E-05	2.71
$u(r, \phi, \theta) = \cos(\pi(r^2 \cos^2 \theta \sin^2 \phi + r \sin \theta \sin \phi)) \sin(\pi r^2 \cos^2 \phi)$		
8	1.0880E-01	
16	1.2900E-02	3.08
32	1.6000E-03	3.01
64	2.4752E-04	2.69

## 4 Generalized Minimal Residual (GMRES)

In this section, we present an iterative method for solving nonsymmetric linear systems of the Fourier mode equation (2.4). The Generalized Minimal Residual (GMRES) method is an extension of MINRES (which is only applicable to symmetric systems) to nonsymmetric linear systems (see Saad and Schultz). The stopping criterion of the convergence is based on the relative residual which the tolerance ranges from  $10^{-12}$  –  $10^{-16}$  depending on the different Fourier modes.

In the Conjugate Gradient method, the residuals form an orthogonal basis for the space  $\text{span}\{r^{(0)}, Ar^{(0)}, A^2r^{(0)}, \dots\}$ . In GMRES, this orthonormal basis is formed explicitly:

$\omega^{(i)} = Av^{(i)}$   
for  $k = 1, \dots, i$   
 $\omega^{(i)} = \omega^{(i)} - (\omega^{(i)}, v^{(k)})v^{(k)}$   
end  
 $v^{(i+1)} = \omega^{(i)} / \|\omega^{(i)}\|$

The reader may recognize this as a modified Gram-Schmidt orthogonalization. The GMRES iterates are constructed as

$$x^{(i)} = x^{(0)} + y_1v^{(1)} + \dots + y_iv^{(i)}, \quad (4.1)$$

where the coefficients  $y_k$  have been chosen to minimize the residual norm  $\|b - Ax^{(i)}\|$ .

Then we perform a preconditioner which can be applied to the GMRES method to solve the compact fourth-order scheme. We take the preconditioner  $M$  arising from the Eq.(2.4) has been solved by the second-order finite difference discretization with Dirichlet boundary condition(see Lai *et al.* (2002) ). The matrix  $M$  is a  $L \times L$  block tridiagonal matrix

$$M = \begin{bmatrix} T_1 & (1 + \lambda_1)I & & & & & & & & & \\ (1 - \lambda_2)I & T_2 & (1 + \lambda_2)I & & & & & & & & \\ & & & \ddots & & & & & & & \\ & & & & \ddots & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & & & & & \\ & & & & & & (1 - \lambda_{L-1})I & T_{L-1} & (1 + \lambda_{L-1})I & & \\ & & & & & & (1 - \lambda_L)I & T_L & & & \end{bmatrix},$$

where  $T_i, 1 \leq i \leq L$  is a tridiagonal matrix given by

$$T_i = \begin{bmatrix} \alpha_i & \beta & & & & & & & & & \\ \beta & \alpha_i & \beta & & & & & & & & \\ & & & \ddots & & & & & & & \\ & & & & \ddots & & & & & & \\ & & & & & & \beta & \alpha_i & \beta & & \\ & & & & & & & \beta & \alpha_i & & \end{bmatrix},$$

with  $\alpha_i = -2 - 4\lambda_i^2n^2 - 2\beta, \beta = \Delta r^2/\Delta z^2, \lambda_i = 1/(2i - 1)$ .

Table 4.1 shows the number of iterations needed for solving the solution of Example 1 of Poisson equation in a cylinder with the Fourier mode number  $n = 1$  and Dirichlet boundary condition by GMRES method with different

preconditioners. The tolerance for the relative residual is chosen as  $10^{-13}$ . Those preconditioners include block Jacobi (BJ), incomplete LU factorization (LUINC), and the fast Poisson solver (FPS) described as before. Indeed, the fast Poisson solver preconditioner turns out to be the most efficient one since it has the least number of iterations, and the iterations are kept to be a constant when we increase the grid points.

**Table 4.1**  
**The performance comparison for using different preconditioners**

$0 < r \leq 1$				
$u(r, z, \theta) = e^{r \cos \theta + r \sin \theta + z}$				
L	Outer/Inner Iteration	Relative residual	Tolerance	Relative Error
GMRES				
8	1/38	4.2482E-14	1.0E-13	7.8137E-05
16	1/74	8.7889E-14	1.0E-13	9.8506E-06
32	1/145	8.3080E-14	1.0E-13	1.2566E-06
64	2/1	9.6686E-14	1.0E-13	1.5941E-07
BJ				
8	1/25	2.5087E-14	1.0E-13	7.8137E-05
16	1/48	5.4973E-14	1.0E-13	9.8507E-06
32	1/90	6.7025E-14	1.0E-13	1.2565E-06
64	1/164	8.5684E-14	1.0E-13	1.5909E-07
LUINC				
8	1/10	1.2748E-14	1.0E-13	7.8137E-05
16	1/16	2.1491E-14	1.0E-13	9.8505E-06
32	1/27	3.7427E-14	1.0E-13	1.2566E-06
64	1/47	7.4488E-14	1.0E-13	1.5913E-07
(2-order)FPS				
8	1/15	1.5611E-14	1.0E-13	7.8137E-05
16	1/15	5.0064E-14	1.0E-13	9.8506E-06
32	1/15	2.7255E-14	1.0E-13	1.2566E-06
64	2/1	2.4130E-14	1.0E-13	1.5941E-07

In the following, Table 4.2 and 4.3 show the number of iterations needed for solving the solution of remaining examples with the Fourier mode number  $n = 1$  by GMRES method with different preconditioners. The tolerance for the relative residual is chosen as  $10^{-13}$ . Similarly, the fast Poisson solver preconditioner turns out to be the most efficient one since it has the least number of iterations, and the iterations are kept to be a constant when we increase the grid points.

**Table 4.2**  
**The performance comparison for using different preconditioners**

$0 < r \leq 1$				
$u(r, z, \theta) = r^3(\cos \theta + \sin \theta)z(1 - z)$				
L	Outer/Inner Iteration	Relative residual	Tolerance	Relative Error
GMRES				
8	1/28	1.0713E-14	1.0E-13	9.1438E-04
16	1/58	7.3844E-14	1.0E-13	1.0755E-04
32	1/115	8.7476E-14	1.0E-13	1.3008E-05
64	2/1	9.5299E-14	1.0E-13	1.5966E-06
8	1/19	5.6023E-14	1.0E-13	9.1438E-04
16	1/36	7.6266E-14	1.0E-13	1.0755E-04
32	1/67	7.0556E-14	1.0E-13	1.3008E-05
64	1/125	6.3349E-14	1.0E-13	1.5966E-06
LUINC				
8	1/10	2.1958E-14	1.0E-13	9.1438E-04
16	1/15	6.7705E-14	1.0E-13	1.0755E-04
32	1/26	5.5145E-14	1.0E-13	1.3008E-05
64	1/45	6.0695E-14	1.0E-13	1.5966E-06
(2-order)FPS				
8	1/13	1.4886E-14	1.0E-13	9.1438E-04
16	1/12	7.5673E-14	1.0E-13	1.0755E-04
32	1/12	2.3112E-14	1.0E-13	1.3008E-05
64	1/11	1.9006E-14	1.0E-13	1.5966E-06

**Table 4.3**  
**The performance comparison for using different preconditioners**

$0 < r \leq 1$				
$u(r, z, \theta) = r^2 \sin 2\theta \sin(\pi z)$				
L	Outer/Inner Iteration	Relative residual	Tolerance	Relative Error
GMRES				
8	1/35	6.2341E-14	1.0E-13	3.0938E-05
16	1/69	7.0950E-14	1.0E-13	2.0650E-06
32	1/136	8.1227E-14	1.0E-13	1.3481E-07
64	1/267	9.4758E-14	1.0E-13	8.6033E-09
BJ				
8	1/22	6.8349E-14	1.0E-13	3.0938E-05
16	1/43	6.1935E-14	1.0E-13	2.0650E-06
32	1/79	7.4148E-14	1.0E-13	1.3481E-07
64	1/143	8.8186E-14	1.0E-13	8.6030E-09
LUINC				
8	1/10	2.0703E-14	1.0E-13	3.0938E-05
16	1/15	6.1446E-14	1.0E-13	2.0650E-06
32	1/26	7.2626E-14	1.0E-13	1.3481E-07
64	1/44	7.6683E-14	1.0E-13	8.6124E-09
(2-order)FPS				
8	1/13	1.5440E-14	1.0E-13	3.0938E-05
16	1/13	3.2524E-14	1.0E-13	2.0650E-06
32	1/13	1.7154E-14	1.0E-13	1.3481E-07
64	1/13	2.3399E-14	1.0E-13	8.6031E-09

## 5 Bi-Conjugate Gradient Stabilized

In this section, we present the other efficient iterative method for solving non-symmetric linear systems. The stopping criterion of the convergence is based on the relative residual which the tolerance ranges from  $10^{-9} - 10^{-13}$  depending on the different Fourier modes. First, we present the Conjugate Gradient Squared (CG-S) algorithm that was developed by Sonneveld (1984), mainly to avoid using the transpose of  $A$  in the Bi-Conjugate Gradient (Bi-CG) method and to gain faster convergence for roughly the same computational cost. In the Bi-CG algorithm, the approximations are constructed in such a way that the residual vector  $r^{(j)}$  is orthogonal with respect to another vectors  $\tilde{r}^{(0)}, \tilde{r}^{(1)}, \dots, \tilde{r}^{(j-1)}$ , and, vice versa,  $\tilde{r}^{(j)}$  is orthogonal with respect to  $r^{(0)}, r^{(1)}, \dots, r^{(j-1)}$ .

Sonneveld observed that, in the case of convergence, both the rows  $\{r^{(j)}\}$  and  $\{\tilde{r}^{(j)}\}$  converge to zero, but that only the convergence of the  $\{r^{(j)}\}$  is used. He proposes the following modification to Bi-CG by which all the convergence effort is focused in the  $r^{(j)}$  vectors. For the Bi-CG vectors it is well known that they can be written as  $r^{(j)} = P_j(A)r^{(0)}$  and  $\tilde{r}^{(j)} = P_j(A^T)\tilde{r}^{(0)}$ , and because of the bi-orthogonality relation we have that

$$\begin{aligned} (r^{(j)}, \tilde{r}^{(i)}) &= (P_j(A)r^{(0)}, P_i(A^T)\tilde{r}^{(0)}) \\ &= (P_i(A)P_j(A)r^{(0)}, \tilde{r}^{(0)}) = 0 \text{ for } i < j. \end{aligned} \quad (5.1)$$

The iteration parameters for Bi-CG are computed from innerproducts like the above. Sonneveld observed that we can also construct the vectors  $\hat{r}^{(j)} = P_j^2(A)r^{(0)}$ , using only the latter form of the innerproduct for recovering the Bi-CG parameters. By doing so, it can be avoided that the vectors  $\tilde{r}^{(j)}$  have to be formed, nor is here any multiplication with the matrix  $A^T$ . The resulting algorithm can be represented by the following scheme  $M$  is a preconditioning matrix.



*PRECONDITIONED CG-S ALGORITHM* —————

Compute  $r^{(0)} = b - Ax^{(0)}$  for some initial guess  $x^{(0)}$

$\tilde{r}^{(0)}$  is an arbitrary vector, such that  $(r^{(0)}, \tilde{r}^{(0)}) \neq 0$ , e.g.,  $r^{(0)} = \tilde{r}^{(0)}$ ;

for  $i = 1, 2, \dots$

$$\rho_{i-1} = (\tilde{r}^{(0)}, r^{(i-1)})$$

if  $\rho_{i-1} = 0$  method fails

if  $i = 1$

$$u^{(1)} = r^{(0)}$$

$$p^{(1)} = u^{(1)}$$

else

$$\beta_{i-1} = \rho_{i-1} / \rho_{i-2}$$

$$u^{(i)} = r^{(i-1)} + \beta_{i-1} q^{(i-1)}$$

$$p^{(i)} = u^{(i)} + \beta_{i-1} (q^{(i-1)} + \beta_{i-1} p^{(i-1)})$$

end if

Solve  $\hat{p}$  from  $M\hat{p} = p^{(i)}$

$$\hat{v} = A\hat{p}$$

$$\alpha_i = \rho_{i-1} / (\tilde{r}^{(0)}, \hat{v})$$

$$q^{(i)} = u^{(i)} - \alpha_i \hat{v}$$

Solve  $\hat{u}$  from  $M\hat{u} = u^{(i)} + q^{(i)}$

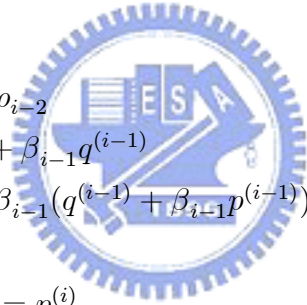
$$x^{(i)} = x^{(i-1)} + \alpha_i \hat{u}$$

$$\hat{q} = A\hat{u}$$

$$r^{(i)} = r^{(i-1)} - \alpha_i \hat{q}$$

check convergence; continue if necessary

end



The CG-S algorithm is based on squaring the residual polynomial, and, in case of irregular convergence, this may lead to a great quantity increasing of rounding errors, or possibly even overflow. The Bi-Conjugate Gradient Stabilized (Bi-CGSTAB) algorithm is a variation of CG-S which was developed to treat this difficulty (see Van der Vorst). Instead of seeking a method which expresses a residual vector of the form  $\hat{r}^{(j)}$ , Bi-CGSTAB produces iterates whose residual are of the form

$$\hat{r}^{(j)} = Q_j(A)P_j(A)r^{(0)}, \quad (5.2)$$

in which, as before,  $P_j(t)$  is the residual polynomial associate with the Bi-CG algorithm and  $Q_j(t)$  is a new polynomial which is defined recursively at each step with the goal of "stabilizing" or "smoothing" the convergence behavior of the original algorithm.

The resulting algorithm can be represented by the following scheme.  $M$  is a preconditioning matrix.

*PRECONDITIONED Bi-CGSTAB ALGORITHM* —————

Compute  $r^{(0)} = b - Ax^{(0)}$  for some initial guess  $x^{(0)}$

$\tilde{r}^{(0)}$  is an arbitrary vector, such that  $(r^{(0)}, \tilde{r}^{(0)}) \neq 0$ , e.g.,  $r^{(0)} = \tilde{r}^{(0)}$ ;

for  $i = 1, 2, \dots$

$$\rho_{i-1} = (\tilde{r}^{(0)}, r^{(i-1)})$$

if  $\rho_{i-1} = 0$  method fails

if  $i = 1$

$$p^{(i)} = r^{(i-1)}$$

else

$$\beta_{i-1} = (\rho_{i-1}/\rho_{i-2})(\alpha_{i-1}/\omega_{i-1})$$

$$p^{(i)} = r^{(i-1)} + \beta_{i-1}(p^{(i-1)} - \omega_{i-1}v^{(i-1)})$$

end if

Solve  $\hat{p}$  from  $M\hat{p} = p^{(i)}$

$$v^{(i)} = A\hat{p}$$

$$\alpha_i = \rho_{i-1}/(\tilde{r}^{(0)}, v^{(i)})$$

$$s = r^{(i-1)} - \alpha_i v^{(i)}$$

check norm of  $s$ ; if small enough: set  $x^{(i)} = x^{(i-1)} + \alpha_i \hat{p}$  and  
 stop  
 Solve  $\hat{s}$  from  $M\hat{s} = s$   
 $t = A\hat{s}$   
 $\omega_i = (s, t)/(t, t)$   
 $x^{(i)} = x^{(i-1)} + \alpha_i \hat{p}^{(i)} + \omega_i \hat{s}$   
 $r^{(i)} = s - \omega_i t$   
 check convergence; continue if necessary  
 for continuation it is necessary that  $\omega_i \neq 0$   
 end

Table 5.1 shows the number of iterations needed for solving the solution of Example 1 of Poisson equation in a cylinder with the Fourier mode number  $n = 1$  and Dirichlet boundary condition by Bi-CGSTAB method with different preconditioners. The tolerance for the relative residual is chosen as  $10^{-10}$ . Those preconditioners have been described in earlier section. Indeed, the fast Poisson solver preconditioner turns out to be the most efficient one since it has the least number of iterations, and the iterations are kept to be a constant when we increase the grid points.

**Table 5.1**  
**The performance comparison for using different preconditioners**

$$0 < r \leq 1$$

$$u(r, z, \theta) = e^{r \cos \theta + r \sin \theta + z}$$

L	Iteration number	Relative residual	Tolerance	Relative Error
Bi-CGSTAB				
8	25	2.3263E-11	1.0E-10	7.8136E-05
16	52	8.3558E-11	1.0E-10	9.8507E-06
32	97	8.2703E-11	1.0E-10	1.2564E-06
64	182	9.9557E-11	1.0E-10	1.5835E-07
BJ				
8	16	3.0152E-11	1.0E-10	7.8136E-05
16	34	4.6514E-11	1.0E-10	9.8505E-06
32	65	7.5701E-11	1.0E-10	1.2564E-06
64	134	8.0676E-11	1.0E-10	1.5885E-07
LUINC				
8	6	1.8371E-11	1.0E-10	7.8136E-05
16	9	7.8271E-11	1.0E-10	9.8503E-06
32	19	3.6401E-11	1.0E-10	1.2575E-06
64	38	7.1583E-11	1.0E-10	1.6127E-07
(2-order)FPS				
8	7	4.7806E-11	1.0E-10	7.8137E-05
16	7	6.9638E-11	1.0E-10	9.8505E-06
32	7	6.5055E-11	1.0E-10	1.2566E-06
64	7	2.7534E-11	1.0E-10	1.5950E-07

In the following, Table 5.2 and 5.3 show the number of iterations needed for solving the solution of remaining examples with the Fourier mode number  $n = 1$  by Bi-CGSTAB method with different preconditioners. The tolerance for the relative residual is chosen as  $10^{-10}$ . Similarly, the fast Poisson solver preconditioner turns out to be the most efficient one since it has the least number of iterations, and the iterations are kept to be a constant when we increase the grid points.

**Table 5.2**  
**The performance comparison for using different preconditioners**

$0 < r \leq 1$				
$u(r, z, \theta) = r^3(\cos \theta + \sin \theta)z(1 - z)$				
L	Iteration number	Relative residual	Tolerance	Relative Error
Bi-CGSTAB				
8	17	7.1236E-11	1.0E-10	9.1438E-04
16	34	6.7664E-11	1.0E-10	1.0755E-04
32	67	6.9265E-11	1.0E-10	1.3008E-05
64	138	9.2923E-11	1.0E-10	1.5966E-06
8	12	6.2911E-11	1.0E-10	9.1438E-04
16	25	5.2211E-11	1.0E-10	1.0755E-04
32	54	8.9216E-11	1.0E-10	1.3008E-05
64	109	8.3574E-11	1.0E-10	1.5966E-06
LUINC				
8	6	1.0459E-11	1.0E-10	9.1438E-04
16	10	1.5875E-11	1.0E-10	1.0755E-04
32	18	7.2357E-11	1.0E-10	1.3008E-05
64	35	7.2622E-11	1.0E-10	1.5966E-06
(2-order)FPS				
8	6	2.2892E-11	1.0E-10	9.1438E-04
16	5	5.1414E-11	1.0E-10	1.0755E-04
32	5	5.5197E-11	1.0E-10	1.3008E-05
64	5	2.0837E-11	1.0E-10	1.5966E-06

**Table 5.3**  
**The performance comparison for using different preconditioners**

$0 < r \leq 1$				
$u(r, z, \theta) = r^2 \sin 2\theta \sin(\pi z)$				
L	Iteration number	Relative residual	Tolerance	Relative Error
Bi-CGSTAB				
8	21	5.5726E-11	1.0E-10	3.0938E-05
16	40	8.3996E-11	1.0E-10	2.0650E-06
32	77	8.2066E-11	1.0E-10	1.3480E-07
64	171	8.9168E-11	1.0E-10	8.5930E-09
BJ				
8	15	1.0410E-11	1.0E-10	3.0938E-05
16	32	9.1358E-11	1.0E-10	2.0650E-06
32	62	9.9962E-11	1.0E-10	1.3481E-07
64	114	9.5901E-11	1.0E-10	8.6033E-09
LUINC				
8	6	5.8224E-12	1.0E-10	3.0938E-05
16	10	2.1490E-11	1.0E-10	2.0650E-06
32	18	9.4662E-11	1.0E-10	1.3481E-07
64	35	6.6702E-11	1.0E-10	8.5840E-09
(2-order)FPS				
8	6	1.1841E-11	1.0E-10	3.0938E-05
16	6	7.6716E-11	1.0E-10	2.0650E-06
32	6	6.6982E-11	1.0E-10	1.3481E-07
64	6	3.4051E-11	1.0E-10	8.6042E-09

## 6 The compact fourth-order scheme on polar geometry with Neumann problems

In Lai (2002), the author present a simple and efficient compact fourth-order Poisson solver in polar coordinates with Dirichlet problem. In this section, we will present the compact fourth-order scheme on polar geometry with different Neumann problems. The first, we consider a Neumann problem for the Poisson equation on a unit disk:

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} = f(r, \theta), 0 < r < 1, 0 \leq \theta < 2\pi, \quad (6.1)$$

$$\frac{\partial u}{\partial r}(1, \theta) = g(\theta). \quad (6.2)$$

Since the solution  $u$  is periodic in  $\theta$ , we can approximate it by the truncated Fourier series as

$$u(r, \theta) = \sum_{n=-N/2}^{N/2-1} \hat{u}_n(r) e^{in\theta}, \quad (6.3)$$

where  $\hat{u}_n(r)$  is the complex Fourier coefficient given by

$$\hat{u}_n(r) = \frac{1}{N} \sum_{k=0}^{N-1} u(r, \theta_k) e^{-in\theta_k}, \quad (6.4)$$

and  $\theta_k = 2k\pi/N$ , and  $N$  is the number of grid points along a circle.

Substituting those expansions into Eq.(6.1), and equating the Fourier coefficients, we derive  $\hat{u}_n(r)$  satisfying the ODE

$$\frac{d^2 \hat{u}_n}{dr^2} + \frac{1}{r} \frac{d\hat{u}_n}{dr} - \frac{n^2}{r^2} \hat{u}_n = \hat{f}_n, 0 < r < 1, \quad (6.5)$$

$$\frac{d\hat{u}_n}{dr}(1) = \hat{g}_n, \quad (6.6)$$

where  $\hat{f}_n(r)$  and  $\hat{g}_n$  are defined in a manner similar to that of (6.3) and (6.4).

For the Neumann boundary, we choose a grid to avoid the polar singularity by

$$r_i = (i - 1/2) \Delta r, i = 0, 1, \dots, M + 2, \quad (6.7)$$

with  $\Delta r = 2/(2M+1)$ . Let the discrete values be denoted by  $U(r_i) \approx \hat{u}_n(r_i)$ ,  $F(r_i) \approx \hat{f}_n(r_i)$ .

By the same method that described in second section, we obtain the finite difference scheme as follows. For  $1 \leq i \leq M+1$ , we need to solve

$$\begin{aligned} & \delta^2 U_i - \frac{\Delta r^2}{12} (\delta^2 F_i - \frac{1}{r_i} \delta_0 F_i + \frac{3+n^2}{r_i^2} \delta^2 U_i - \frac{3+5n^2}{r_i^3} \delta_0 U_i + \frac{8n^2}{r_i^4} U_i) \\ & + \frac{1}{r_i} \delta_0 U_i - \frac{\Delta r}{6r_i} (\delta_0 F_i - \frac{1}{r_i} \delta^2 U_i + \frac{1+n^2}{r_i^2} \delta_0 U_i - \frac{2n^2}{r_i^3} U_i) - \frac{n^2}{r_i^2} U_i = F_i. \end{aligned} \quad (6.8)$$

The inner numerical boundary value  $U_0 = (-1)^n U_1$  can be obtained by the symmetry constraint of Fourier coefficients as before. And the outer numerical boundary value  $U_{M+2}$  is derived as follows. That is, at  $r = 1$ ,

$$\begin{aligned} \delta_0 U &= U' + \frac{U'''}{6} \Delta r^2 + O(\Delta r^4) \\ &= U' + \frac{\Delta r^2}{6} (F' - \frac{U''}{r} + \frac{1+n^2}{r^2} U' - \frac{2n^2}{r^3} U) + O(\Delta r^4). \\ &\approx \frac{U_{M+2} - U_M}{2\Delta r} = \hat{g}_n + \frac{\Delta r^2}{6} (\frac{3F_{M+1} - 4F_M + F_{M-1}}{2\Delta r} - \frac{U_{M+2} - 2U_{M+1} + U_M}{\Delta r^2} \\ &\quad + (1+n^2)\hat{g}_n - 2n^2 U_{M+1}). \end{aligned} \quad (6.9)$$

The outer numerical boundary value  $U_{M+2}$  can be approximated by

$$\begin{aligned} U_{M+2} &= (\frac{6\Delta r}{3+\Delta r}) [(\frac{3-\Delta r}{6\Delta r}) U_M + (\frac{6+\Delta r^2(1+n^2)}{6}) \hat{g}_n + \frac{\Delta r}{4} F_{M+1} \\ &\quad - \frac{\Delta r}{3} F_M + \frac{\Delta r}{12} F_{M-1} + (\frac{1-n^2\Delta r^2}{3}) U_{M+1}]. \end{aligned} \quad (6.10)$$

Table 6.1 shows the maximum errors of this method for three different solutions of Poisson equation in unit disk with Neumann boundary condition at  $r = 1$ .



**Table 6.1**  
**The Maximum Errors of Different Solutions to**  
**the Poisson Equation with Neumann Boundary**  
 $\frac{\partial u}{\partial r}(1, \theta) = g(\theta)$  using  $N = 64$ .

$0 < r \leq 1$		
M	$\ u\ _\infty$	Rate
$u(x, y) = e^{x+y}$		
16	4.2008E-05	
32	5.3993E-06	2.960
64	6.8285E-07	2.983
128	8.5733E-08	2.994
$u(x, y) = 3e^{x+y}(x - x^2)(y - y^2) + 5$		
16	2.4161E-04	
32	1.7289E-05	3.805
64	1.1573E-06	3.901
128	7.4875E-08	3.950
$u(x, y) = \frac{e^x + e^y}{1 + xy}$		
16	7.2000E-03	
32	5.7732E-04	3.641
64	4.1560E-05	3.796
128	2.8026E-06	3.890

The second, we consider a Neumann problem for the Poisson equation on an annulus:

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} = f(r, \theta), 0.5 < r < 1, 0 \leq \theta < 2\pi, \quad (6.11)$$

$$u(0.5, \theta) = k(\theta), \quad \frac{\partial u}{\partial r}(1, \theta) = g(\theta). \quad (6.12)$$

We choose a regular grid,

$$r_i = 0.5 + i\Delta r, i = 0, 1, \dots, M + 2, \quad (6.13)$$

with  $\Delta r = 1/(2M + 2)$ .

The finite difference scheme can be obtained as Eq.(6.8). Similarly, the outer numerical boundary value  $U_{M+2}$  can be approximated by (6.10). Table

6.2 shows the maximum errors of this method for three different solutions of Poisson equation in an annulus with Neumann boundary condition at  $r = 1$ .

**Table 6.2**  
**The Maximum Errors of Different Solutions to**  
**the Poisson Equation with Neumann Boundary**  
 $u(0.5, \theta) = k(\theta), \frac{\partial u}{\partial r}(1, \theta) = g(\theta)$  using  $N = 64$ .

M	$0.5 \leq r \leq 1$	
	$\ u\ _\infty$	Rate
$u(x, y) = e^{x+y}$		
16	5.3054E-07	
32	3.7859E-08	3.809
64	2.5324E-09	3.902
128	1.6362E-10	3.952
$u(x, y) = 3e^{x+y}(x - x^2)(y - y^2) + 5$		
16	1.3822E-05	
32	1.0083E-06	3.777
64	6.8229E-08	3.885
128	4.4393E-09	3.942
$u(x, y) = \frac{e^x + e^y}{1 + xy}$		
16	6.6515E-04	
32	5.1868E-05	3.681
64	3.6497E-06	3.829
128	2.4233E-07	3.913

The last one, we consider a Neumann problem for the Poisson equation on an annulus:

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} = f(r, \theta), 0.5 < r < 1, 0 \leq \theta < 2\pi, \quad (6.14)$$

$$\frac{\partial u}{\partial r}(0.5, \theta) = k(\theta), \quad \frac{\partial u}{\partial r}(1, \theta) = g(\theta). \quad (6.15)$$

We choose a regular grid,

$$r_i = 0.5 + (i - 1)\Delta r, i = 0, 1, \dots, M + 2, \quad (6.16)$$

with  $\Delta r = 1/(2M)$ . Then the outer numerical boundary value  $U_{M+2}$  can be approximated by (6.10). And the inner numerical boundary value  $U_0$  is derived as follows. That is, at  $r = 0.5$ ,

$$\begin{aligned}
\delta_0 U &= U' + \frac{U'''}{6} \Delta r^2 + O(\Delta r^4) \\
&= U' + \frac{\Delta r^2}{6} \left( F' - \frac{U''}{r} + \frac{1+n^2}{r^2} U' - \frac{2n^2}{r^3} U \right) + O(\Delta r^4). \\
&\approx \frac{U_2 - U_0}{2\Delta r} = \hat{k}_n + \frac{\Delta r^2}{6} \left( \frac{-3F_1 + 4F_2 - F_3}{2\Delta r} - \frac{U_2 - 2U_1 + U_0}{0.5\Delta r^2} + \frac{(1+n^2)\hat{k}_n}{(0.5)^2} \right. \\
&\quad \left. - \frac{2n^2}{(0.5)^3} U_1 \right). \tag{6.17}
\end{aligned}$$

The inner numerical boundary value  $U_0$  can be approximated by

$$\begin{aligned}
U_0 &= \left( \frac{6\Delta r}{3-2\Delta r} \right) \left[ \left( \frac{3+2\Delta r}{6\Delta r} \right) U_2 + \left( \frac{-3-2\Delta r^2(1+n^2)}{3} \right) \hat{k}_n + \frac{\Delta r}{4} F_1 \right. \\
&\quad \left. - \frac{\Delta r}{3} F_2 + \frac{\Delta r}{12} F_3 + \left( \frac{-2+8n^2\Delta r^2}{3} \right) U_1 \right]. \tag{6.18}
\end{aligned}$$

Table 6.3 shows the maximum errors of this method for three different solutions of Poisson equation in an annulus with Neumann boundary conditions at  $r = 1$  and  $0.5$ .

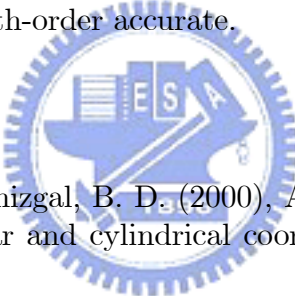
**Table 6.3**  
**The Maximum Errors of Different Solutions to**  
**the Poisson Equation with Neumann Boundary**  
 $\frac{\partial u}{\partial r}(0.5, \theta) = k(\theta), \frac{\partial u}{\partial r}(1, \theta) = g(\theta)$  using  $N = 64$ .

$0.5 \leq r \leq 1$		
M	$\ u\ _\infty$	Rate
$u(x, y) = e^{x+y}$		
16	6.7419E-07	
32	4.3156E-08	3.966
64	2.7284E-09	3.983
128	1.7218E-10	3.986
$u(x, y) = 3e^{x+y}(x - x^2)(y - y^2) + 5$		
16	2.5549E-05	
32	1.6625E-06	3.942
64	1.0601E-07	3.971
128	6.6921E-09	3.986
$u(x, y) = \frac{e^x + e^y}{1 + xy}$		
16	6.5994E-04	
32	4.6168E-05	3.837
64	3.0672E-06	3.912
128	1.9766E-07	3.956

## 7 Conclusions

In this paper, we present a simple and efficient compact fourth-order Poisson solver in cylindrical and spherical coordinates. The solver relies on the truncated Fourier series expansion, where the differential equations of Fourier coefficients have been solved by fourth-order finite difference discretizations without pole conditions. And two kinds of efficient iterative method, GMRES and Bi-CGSTAB, with different preconditioners are applied to solve the resulted nonsymmetric systems of Fourier coefficients. In particular, a preconditioner arising from those singular equations have been solved by the second-order finite difference discretizations and shown to be the most efficient one. Meanwhile, we can see that the numerical result confirms the third-order accuracy for the problem on a cylinder. The loss of one order of accuracy can be seen from the discretization near the origin. Therefore, the future research aspect is how to treat the coordinate singularity so that the scheme becomes fully fourth-order accurate.

## References

- 
- [1] Chen, H., Su, Y. & Shizgal, B. D. (2000), A direct spectral collocation Poisson solver in polar and cylindrical coordinates. *J. Comput. Phys.*, **160**, 456-469.
  - [2] Lai, M.-C., Lin, W.-W. & Wang, W.-C. (2002), A fast spectral/difference method without pole conditions for Poisson-type equations in cylindrical and spherical geometries. *IMA J. Numer. Anal.*, **22**, 537-548.
  - [3] Lai, M.-C. (2002), A simple compact fourth-order Poisson solver on polar geometry. *J. Comput. Phys.*, **182**, 337-345.
  - [4] Lai, M.-C. & Wang, W.-C. (2002), Fast direct solvers for Poisson equation on 2D polar and spherical geometries. *Numer. Meth. Partial Diff. Eq.*, **18**, 56-68.
  - [5] Quarteroni, A., Sacco, R. & Saleri, F. (2000), *Numerical Mathematics* (Springer).

- [6] Richard, B. & et al. (1994), *Templates for the solution of linear systems: Building blocks for iterative methods* (SIAM).
- [7] Saad, Y. & Schultz, M. (1986), GMRES : A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, **7**, 856-869.
- [8] Saad, Y. (1996), *Iterative Methods for Sparse Linear Systems* (PWS Publishing Company).
- [9] Sonneveld, P. (1989), CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, **10**, 36-52.
- [10] Strikwerda, J. C. (1989), *Finite Difference schemes and Partial Differential Equations* (Wadsworth & Brooks/Cole, Belmonts, CA).
- [11] Van Der Vorst, H. A. (1992), Bi-CGSTAB : A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, **13**, 631-644.

