A fast spectral/difference method without pole conditions for Poisson-type equations in cylindrical and spherical geometries

Ming-Chih Lai
Department of Mathematics, Chung Cheng University,
Minghsiung, Chiayi 621, TAIWAN
mclai@math.ccu.edu.tw

Wen-Wei Lin
Department of Mathematics, National Tsing Hua University,
Hsinchu 300, TAIWAN
wwlin@math.nthu.edu.tw

Weichung Wang
Department of Mathematics Education,
National Tainan Teachers College, Tainan 700, TAIWAN
wwang@ipx.ntntc.edu.tw

Abstract

A simple and efficient FFT-based fast direct solver for Poisson-type equations on 3D cylindrical and spherical geometries is presented. The solver relies on the truncated Fourier series expansion, where the differential equations of Fourier coefficients are solved using second-order finite difference discretizations without pole conditions. Three different boundary conditions (Dirichlet, Neumann and Robin conditions) can be handled without substantial differences.

Keywords: Fast Poisson solver; cylindrical coordinates; spherical coordinates; symmetry constraint.

1 Introduction

Many physical problems involve solving the Poisson equation in three-dimensional cylindrical or spherical domains. For example, the projection method for the numerical solution of the incompressible Navier-Stokes equations requires solving the pressure Poisson equation. It is convenient to rewrite the equation in cylindrical or spherical coordinates. The first problem that must be dealt with is the coordinate singularities caused by the transformation. These singularities occur at the polar axis of those domains. It is important to note that the singularities occur because of the representation of the governing equation in those coordinates; the solution itself is regular if the right-hand side function and the boundary values are smooth enough.

Most of the finite difference, finite volume and spectral methods in the literature (see the references in [5]) must 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 [4], the first author of this paper applied the second-order centred difference method to the Poisson equation on a disk using a grid shifted a half mesh width in the radial direction, to avoid placing a mesh point at the origin. The resulting linear system was solved using the cyclic reduction algorithm [2] directly, and no pole approximation was required in this setting.

Recently, the first author and Wang combined the spectral and finite difference methods to develop FFT-based fast direct solvers for the Poisson equation on 2D polar and spherical geometries [5]. The method first uses the truncated Fourier series expansion to derive a set of singular ODEs for the Fourier coefficients, then solves those singular equations by second- and fourth-order finite difference discretizations. Using a uniform grid shifted a half mesh width from the origin/poles, and incorporating the derived symmetry constraint of the Fourier coefficients, the coordinate singularities can be handled easily 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 easy. Both second- and fourth-order schemes only require $O(MN \log_2 N)$ arithmetic operations for $M \times N$ mesh points. The authors also collected detailed references for other numerical approaches such as finite difference, finite volume and spectral/pseudospectral methods in [5]. It was shown there that the method is simple, fast, and able to handle different boundary conditions.

In this article, we extend the previous second-order schemes on two-dimensional cases [5] to three-dimensional domains. Using the truncated Fourier series expansion, the original 3D PDE now becomes a set of 2D PDEs (instead of ODEs in the 2D case) of the Fourier coefficients. We then solve those PDEs by second-order finite difference discretizations. Of course, our scheme retains three major features: there are no pole conditions, a fast direct solver can be applied, and the scheme is able to handle different boundary conditions.

2 Fast Poisson solver in cylindrical coordinates

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

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

For the sake of simplicity, we restrict the boundary conditions to the Dirichlet type 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 $\frac{\partial u}{\partial r} + \alpha u(1,z,\theta) = u_S(z,\theta), \alpha > 0$. The reason for considering these boundary conditions is that, in the computation of Rayleigh-Bénard convection, the heat transfer at the sidewall boundary can range from Dirichlet (perfect conducting) or Neumann (perfect insulating) to more realistic Robin-type boundary conditions.

The main issue in solving Eq. (1) is how to treat the coordinate singularity along the polar axis at the centre $r = 0$. Most Poisson solvers for Eq. (1), including finite difference and spectral methods (see the references in [3, 5]), involve imposing additional pole conditions to approximate the solution in the vicinity of the origin accurately. The accuracy of those methods depends greatly on the choice of pole conditions. Recently, Chen et al. [3] presented a direct spectral collocation method for Eq. (1) with Dirichlet boundary conditions on all boundaries without any pole conditions. Note that here we are not trying to compare or compete with other spectral methods such as [3] or the references therein, but are introducing a simple treatment for the coordinate singularity so that fast direct methods can be applied. Some comparisons between our approach and others for the 2D case can be found in [5].

In the following, we combine the spectral and finite difference methods to develop a new class of fast direct solvers for Eq. (1). Our approach relies on the truncated Fourier series expansion, where the differential equations of Fourier coefficients are solved using second-order finite difference discretizations without pole conditions.

2.1 Fourier mode equations

Because the solution u is periodic in θ, we can approximate it by the truncated Fourier series as

$$u(r,z,\theta) = \sum_{n=-N/2}^{N/2-1} u_n(r,z)\, e^{in\theta}, \tag{2}$$

where $u_n(r,z)$ is the complex Fourier coefficient given by

$$u_n(r,z) = \frac{1}{N}\sum_{k=0}^{N-1} u(r,z,\theta_k)\, e^{-in\theta_k}, \tag{3}$$

and $\theta_k = 2k\pi/N$, and N is the number of grid points along a circle. The above transformation between the physical space and Fourier space can be efficiently performed using the fast Fourier transform (FFT) with $O(N\log_2 N)$ arithmetic operations.

Substituting the expansions of (3) into Eq. (1), and equating the Fourier coefficients, we derive $u_n(r,z)$ satisfying the PDE

$$\frac{\partial^2 u_n}{\partial r^2} + \frac{1}{r}\frac{\partial u_n}{\partial r} + \frac{\partial^2 u_n}{\partial z^2} - \frac{n^2}{r^2}\, u_n = f_n(r,z), \quad 0 < r \le 1, \quad 0 \le z \le 1, \tag{4}$$

where the nth Fourier coefficient of the right-hand side function $f_n(r,z)$ is defined similarly to (3). The Fourier coefficients of the boundary values $u_S^n(z)$, $u_T^n(r)$, $u_B^n(r)$ are also defined in a similar fashion as in (3). Thus, the remaining problem is to solve Eq. (4) with the top and bottom boundary conditions $u_n(r,0) = u_B^n(r)$, $u_n(r,1) = u_T^n(r)$, and with one of the three sidewall boundary conditions $u_n(1,z) = u_S^n(z)$, $\frac{\partial u_n}{\partial r}(1,z) = u_S^n(z)$, or $\frac{\partial u_n}{\partial r} + \alpha u_n(1,z) = u_S^n(z)$.

2.2 Second-order finite difference discretization

We choose a grid in the (r, z) plane by

$$r_i = (i - 1/2)\,\Delta r, \qquad z_j = j\,\Delta z, \tag{5}$$

for $1 \leq i \leq L + 1; 0 \leq j \leq M + 1$, with $\Delta r = 2/(2L + 1)$ and $\Delta z = 1/(M + 1)$. By the choice of these mesh points and widths, we have $r_{L+1} = 1$, $z_0 = 0$ and $z_{M+1} = 1$. Let the discrete values be denoted by $u_{ij} \approx u_n(r_i, z_j)$, $f_{ij} \approx f_n(r_i, z_j)$. At the interior points $(r_i, z_j), 1 \leq i \leq L, 1 \leq j \leq M$, using the centred difference approximations for the derivatives to discretize Eq. (4), we have

$$\frac{u_{i+1,j} - 2\,u_{ij} + u_{i-1,j}}{\Delta r^2} + \frac{1}{r_i}\frac{u_{i+1,j} - u_{i-1,j}}{2\,\Delta r} + \frac{u_{i,j+1} - 2\,u_{ij} + u_{i,j-1}}{\Delta z^2} - \frac{n^2}{r_i^2}\,u_{ij} = f_{ij}. \tag{6}$$

The numerical boundary values in the z direction can easily be obtained by the given Dirichlet boundary values $u_{i0} = u_B^n(r_i)$ and $u_{i,M+1} = u_T^n(r_i)$. When $i = 1$ for Eq. (6), we immediately observe that the coefficient of u_{0j} is zero because $r_1 = \Delta r/2$. Thus, the above discretization does not require any approximation for u_{0j} and no pole conditions are necessary. The other numerical boundary condition $u_{L+1,j}$ can either be given (Dirichlet) or be determined by imposing the condition on the boundary (Neumann and Robin).

Let us order the unknowns u_{ij} by first grouping by i values so that the solution vector v is defined by

$$v = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_L \end{bmatrix}, \qquad u_i = \begin{bmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{iM} \end{bmatrix}.$$

Solving the discrete equations (6) 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

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

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

$$T_i = \begin{bmatrix} -2-4\lambda_i^2 n^2 - 2\beta & \beta & & & \\ \beta & -2-4\lambda_i^2 n^2 - 2\beta & \beta & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \cdot \\ & & \beta & -2-4\lambda_i^2 n^2 - 2\beta & \beta \\ & & & \beta & -2-4\lambda_i^2 n^2 - 2\beta \end{bmatrix},$$

with $\beta = \Delta r^2/\Delta z^2, \lambda_i = 1/(2i-1)$. Incorporating the boundary values and the function f, the right-hand side vector b can be written as

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_L - (1+\lambda_L) u_S^n(z) \end{bmatrix}, \qquad b_i = \begin{bmatrix} \Delta r^2\, f_{i1} - \beta\, u_B^n(r_i) \\ \Delta r^2\, f_{i2} \\ \vdots \\ \Delta r^2\, f_{i,M-1} \\ \Delta r^2\, f_{iM} - \beta\, u_T^n(r_i) \end{bmatrix}.$$

Note that the above form is for the Dirichlet sidewall boundary. We shall discuss the other cases below.

The above linear equations $Av = b$ can easily be solved by the Fourier method [1]. Each tridiagonal matrix T_i can be diagonalized using the same eigenvectors as are used in applying the Fast Sine Transform. Moreover, the eigenvalues of T_i are known quantities. Thus, after transforming the right-hand side vector ($O(LM \log_2 M)$ operations) and reordering the equations, this leaves M decoupled $L \times L$ tridiagonal linear systems to be solved ($O(ML)$

operations). Transforming the solution vector back requires another Fast Sine Transform ($O(LM\log_2 M)$ operations). Because there are N Fourier mode equations to be solved, we will require $O(LMN\log_2 M)$ operations. If we add the FFT transforms at the beginning and the end, then the total number of operations becomes $O(LMN\log_2(MN))$. It is also worth mentioning here that the present approach is highly parallelizable because an FFT can be performed on each concentric grid circle and the resulting decoupled tridiagonal linear systems can be solved independently. The present scheme can also be extended easily to the Helmholtz equation, as only the diagonal part of the matrix A must be modified.

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 boundaries so that the first derivative can be centred on the grid points. That is, at $r = 1$,

$$\frac{\partial u}{\partial r} + \alpha u \approx \frac{u_{L+1,j} - u_{Lj}}{\Delta r} + \alpha\,\frac{u_{L+1,j} + u_{Lj}}{2} = u_S^n(z_j). \tag{7}$$

The numerical boundary values $u_{L+1,j}$ can therefore be approximated by

$$u_{L+1,j} = \frac{(1 - \alpha\Delta r/2)\,u_{Lj} + u_S^n(z_j)\,\Delta r}{1 + \alpha\Delta r/2}. \tag{8}$$

Therefore, it is only necessary to modify T_L in the matrix A and b_L in the vector b by

$$T_L = T_L + (1 + \lambda_L)\,\frac{1 - \alpha\Delta r/2}{1 + \alpha\Delta r/2}\,I \tag{9}$$

$$b_L = b_L - \frac{1 + \lambda_L}{1 + \alpha\Delta r/2}\,u_S^n(z)\,\Delta r. \tag{10}$$

We close this section by performing three numerical tests and comparing the present approach with the well-known software in the public domain called Fishpack [Adams, Swarztrauber, and Sweet (1980)]. Unlike Cartesian coordinates, the package does not have a direct solver for the cylindrical Poisson equation (1). Instead, the package provides a second-order difference solver for the Fourier mode equation (4), which uses the cyclic reduction method [2] to solve the resulting linear systems. The subroutine is called "HWSCYL.f" and calculates the correct approximation at coordinate singularity ($r = 0$), but our approach avoids doing that. In our implementation of the Fishpack solver, we must put pole conditions at $r = 0$. Here, we simply leave the zeroth-mode coefficient $u_0(0, z)$ undetermined, but let the other Fourier coefficients $u_n(0,z) = 0, n \neq 0$ to fulfil the requirements of calling HWSCYL.f. One should notice that these conditions are simply the requirements for the solution to be single-valued at the axis $r = 0$.

Table 1 shows the maximum errors of both methods for three different solutions of the Poisson equation in a cylinder with different sidewall boundary conditions. In all our tests, we used L mesh points in the radial and axial directions, and $2L$ points in the azimuthal direction. The rate of convergence was computed by the formula $\log_2(\frac{E_{L/2}}{E_L})$, where E_L is the maximum error. One can easily check from this table that second-order convergence is achieved for both methods on all three different solutions. Generally, the errors obtained by the present scheme are a little smaller than the ones obtained by the Fishpack solver, although the difference is not significant. The current subroutine used by Fishpack cannot handle the Robin boundary problem, so we list only the errors obtained by our present scheme for the case of $\alpha = 1$. Again, second-order accuracy can be seen for this problem.

As we mentioned before, the subroutine HWSCYL.f uses the cyclic reduction algorithm to solve the block tridiagonal systems. Thus, the computational costs for each solution of Eq. (4) is $O(LM \log_2 M)$ operations. Because there are N Fourier mode equations to be solved, the operation count is $O(NLM \log_2 M)$. If we take the FFT steps (2)-(3) into account, then the total number of operations becomes $O(LMN \log_2(MN))$. Therefore, the Fishpack solver has asymptotically the same costs as our method.

3 Fast Poisson solver in spherical coordinates

The solution of elliptic equations in spherical geometries has many applications in the areas of meteorology, geophysics, and astrophysics. 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

$$\frac{\partial^2 u}{\partial r^2} + \frac{2}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \phi^2} + \frac{\cot\phi}{r^2}\frac{\partial u}{\partial \phi} + \frac{1}{r^2 \sin^2\phi}\frac{\partial^2 u}{\partial \theta^2} = f(r,\phi,\theta). \tag{11}$$

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 $\frac{\partial u}{\partial r} + \alpha u(1,\phi,\theta) = u_S(\phi,\theta), \alpha > 0$. However, the method to be described can be adapted easily to different boundary conditions on the inner surface.

As in the cylindrical case of the previous section, the main difficulty in solving Eq. (11) is to treat the coordinate singularities along the polar axis where the north ($\phi = 0$) and south ($\phi = \pi$) poles are located. Again, most numerical approaches, including finite difference and spectral methods, involve imposing additional pole conditions to capture the behaviour of the solution in the vicinity of the poles. Readers who are interested in those approaches in 2D geometries (the surface of the sphere) can find treatments in the references of [5]. In a recent paper, Shen [6] used a spectral-Galerkin method to solve Eq. (11) on 2D and 3D domains where some necessary pole conditions must also be satisfied. In the following, we present a numerical method to solve Eq. (11), which uses the symmetry constraint of the Fourier coefficients to handle the coordinate singularities without pole conditions.

3.1 Fourier mode equations

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

$$u(r,\phi,\theta) = \sum_{n=-N/2}^{N/2-1} u_n(r,\phi)\, e^{in\theta}, \tag{12}$$

where $u_n(r,\phi)$ is the complex Fourier coefficient given by

$$u_n(r,\phi) = \frac{1}{N}\sum_{k=0}^{N-1} u(r,\phi,\theta_k)\, e^{-in\theta_k}, \tag{13}$$

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 a similar fashion. Substituting those expansions into Eq. (11), and equating the Fourier coefficients, $u_n(r,\phi)$ then satisfies the PDE

$$\frac{\partial^2 u_n}{\partial r^2} + \frac{2}{r}\frac{\partial u_n}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u_n}{\partial \phi^2} + \frac{\cot\phi}{r^2}\frac{\partial u_n}{\partial \phi} - \frac{n^2}{r^2\sin^2\phi}u_n = f_n(r,\phi), \tag{14}$$

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

3.2 Second-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 the (r, ϕ) plane by

$$r_i = R_0 + i\,\Delta r, \qquad \phi_j = (j - 1/2)\,\Delta\phi, \tag{15}$$

for $0 \le i \le L+1, 0 \le j \le M+1$ with $\Delta r = (1 - R_0)/(L+1)$ and $\Delta\phi = \pi/M$. By choosing these mesh points, we avoid placing points directly at the north ($\phi = 0$) and south ($\phi = \pi$) poles. Again, let the discrete values be denoted by $u_{ij} \approx u_n(r_i, \phi_j)$, and $f_{ij} \approx f_n(r_i, \phi_j)$. Using centred difference approximations to discretize Eq. (14) at the internal mesh points $(r_i, \phi_j), 1 \le i \le L, 1 \le j \le M$, we have

$$\begin{aligned}\frac{u_{i+1,j} - 2\,u_{ij} + u_{i-1,j}}{(\Delta r)^2} + \frac{2}{r_i}\frac{u_{i+1,j} - u_{i-1,j}}{2\,\Delta r} + \frac{1}{r_i^2}\frac{u_{i,j+1} - 2\,u_{ij} + u_{i,j-1}}{(\Delta\phi)^2} \\ + \frac{\cot\phi_j}{r_i^2}\frac{u_{i,j+1} - u_{i,j-1}}{2\Delta\phi} - \frac{n^2}{r_i^2 \sin^2\phi_j}\,u_{ij} = f_{ij}.\end{aligned} \tag{16}$$

When $j = 1$ for Eq. (16), the numerical boundary value u_{i0} can be given by $u_{i0} = (-1)^n u_{i1}$. This is because the Fourier coefficient satisfies the symmetry constraint $u_n(r_i, -\Delta\phi/2) = (-1)^n u_n(r_i, \Delta\phi/2)$ [5]. Similarly, another numerical boundary value $u_{i,M+1}$ can also be obtained by $u_{i,M+1} = (-1)^n u_{iM}$ for the same reason. The numerical boundary values in the ϕ direction are therefore provided, and no pole conditions are required in our finite difference setting. The numerical boundary values in the radial direction $u_{0j}, u_{L+1,j}$ are given by the boundary values $u_I^n(\phi_j), u_S^n(\phi_j)$. It is also worth pointing out that, in the case of $R_0 = 0$ (the whole sphere case), the numerical boundary value u_{0j} seems to be redundant. This is because when $i = 1$, the coefficient of u_{0j} in Eq. (16) becomes zero because $r_1 = \Delta r$. Thus, the above discretization does not require any approximation for u_{0j}. This appears to be another advantage of using a grid like that in (15).

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

$$v = \begin{bmatrix} u_1 \\ \vdots \\ u_i \\ \vdots \\ u_L \end{bmatrix}, \qquad u_i = \begin{bmatrix} u_{i1} \\ \vdots \\ u_{ij} \\ \vdots \\ u_{iM} \end{bmatrix}.$$

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

$$A = \begin{bmatrix} T_1 & d_1 I & & & & \\ c_2 I & T_2 & d_2 I & & & \\ & \cdot & \cdot & \cdot & & \\ & & & c_{L-1} I & T_{L-1} & d_{L-1} I \\ & & & & c_L I & T_L \end{bmatrix},$$

where $c_i = i^2 - i, d_i = i^2 + i, 1 \le i \le L$, and T_i is a tridiagonal matrix

$$T_i = -(c_i + d_i)I + T.$$

Here, the matrix T is a tridiagonal matrix given by

$$T = \begin{bmatrix} -2\beta - a_1 + (-1)^n/12 & \beta + \lambda_1 & & & \\ & \cdot & \cdot & & \\ & \beta - \lambda_j & -2\beta - a_j & \beta + \lambda_j & \\ & & & \cdot & \cdot \\ & & & \beta - \lambda_M & -2\beta - a_M + (-1)^n/12 \end{bmatrix}.$$

where $\beta = 1/\Delta\phi^2$, $a_j = n^2/\sin^2\phi_j$, $\lambda_j = \cot\phi_j/(2\Delta\phi), 1 \le j \le M$. We modify the first and last elements in the diagonal part of the matrix T because we must incorporate the numerical boundary values obtained by the symmetry constraint with the difference scheme. Here, $\beta - \lambda_1$ can further be approximated by $1/12 + O(\Delta\phi^2)$ using the Taylor expansion for the function $\cot\phi$.

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

$$b = \begin{bmatrix} b_1 - d_1\, u_I^n(\phi) \\ \vdots \\ b_i \\ \vdots \\ b_L - d_L\, u_S^n(\phi) \end{bmatrix}, \qquad b_i = \begin{bmatrix} r_i^2\, f_{i1} \\ \vdots \\ r_i^2 f_{ij} \\ \vdots \\ r_i^2 f_{iM} \end{bmatrix}.$$

The above linear system $Av = b$ can be solved by the generalized cyclic reduction method [7] with $O(ML\log_2 L)$ operations. For the numerical results shown in Table 2, we solved the above linear system using the subroutine CBLKTR.f provided by Fishpack. Because there are N Fourier mode equations to be solved, the operation count is $O(NML\log_2 L)$. Again, we add the FFT at the beginning and the end, so the total operations will be $O(NML\log_2(LN))$. The above method can also be extended to the solution of the Helmholtz equation straightforwardly, as only the diagonal part of the matrix A must be modified.

For the Neumann ($\alpha = 0$) or Robin boundary cases, we use the grid described in (15) but with a 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 the boundaries so that the first derivative can be centred on the mesh points. That is, at $r = 1$,

$$\frac{\partial u}{\partial r} + \alpha u \approx \frac{u_{L+1,j} - u_{Lj}}{\Delta r} + \alpha\, \frac{u_{L+1,j} + u_{Lj}}{2} = u_S^n(\phi_j). \tag{17}$$

The numerical boundary values $u_{L+1,j}$ can therefore be approximated by

$$u_{L+1,j} = \frac{(1 - \alpha\Delta r/2)\, u_{Lj} + u_S^n(\phi_j)\,\Delta r}{1 + \alpha\Delta r/2}. \tag{18}$$

Therefore, only T_L requires modification in the matrix A and the vector b_L by

$$T_L = T_L + d_L\, \frac{1 - \alpha\Delta r/2}{1 + \alpha\Delta r/2}\, I, \tag{19}$$

$$b_L = b_L - \frac{d_L}{1 + \alpha\Delta r/2}\, u_S^n(\phi)\,\Delta r. \tag{20}$$

As in the cylindrical case, we performed three numerical tests and compared the errors with Fishpack. Again, the package does not have a direct solver for the Poisson equation (11).

Instead, the package provides a subroutine "HWSCSP.f" to solve the Fourier mode equation (14). This subroutine automatically calculates the correct approximation at the north ($\phi = 0$) and south ($\phi = \pi$) poles, while our approach finds a simple alternative to treat them. In the implementation of the Fishpack solver, we require some conditions at the poles. Here, we simply leave the zeroth-mode coefficient at the poles undetermined, but let the other Fourier coefficients $u_n(r, 0) = u_n(r, \pi) = 0, n \neq 0$. These conditions are set up to fulfil the requirements of calling HWSCSP.f. Their physical meaning is that they guarantee that the solution is single-valued at the poles.

Table 2 shows the maximum errors of both methods for three different solutions of the Poisson equation in spherical coordinates with different sidewall boundary conditions. In all our tests, we used L mesh points in the radial and co-latitude directions, and $2L$ points in the longitudinal direction. The inner radius is chosen as $R_0 = 0.5$. One can easily check from this table that second-order convergence is achieved for both methods. The errors obtained by our present approach are sometimes smaller and sometimes larger than the errors obtained by the Fishpack solver. Nevertheless, both methods have the same computational costs because the same block tridiagonal solver CBLKTR.f is called.

Acknowledgment

The first author was supported in part by National Science Council of Taiwan under research grant NSC-90-2115-M-194-018. The authors thank Prof. Wei-Cheng Wang for his helpful discussions. Finally, the authors thank the referees for their valuable comments to improve the article.

References

[1] B.L. Buzbee, G.H. Golub and C.W. Nielson, On direct methods for solving Poisson's equation, SIAM J. Numer. Anal., 7 (1970), 627–656.

[2] O. Buneman, A compact non-iterative Poisson solver, Rep. 294, Stanford University Institute for Plasma Research, Stanford, California, 1969.

[3] H. Chen, Y. Su and B.D. Shizgal, A direct spectral collocation Poisson solver in polar and cylindrical coordinates, J. Comput. Phys., 160 (2000), 453–469.

[4] M.-C. Lai, A note on finite difference discretizations for Poisson equation on a disk, Numer. Methods Partial Differential Eq., 17 (2001), 199–203.

[5] M.-C. Lai and W.-C. Wang, Fast direct solvers for Poisson equation on 2D polar and spherical geometries, Numer. Methods Partial Differential Eq., in press.

[6] J. Shen, Efficient spectral-Galerkin methods IV: Spherical geometries, SIAM. J. Sci. Comput., 20 (1999), 1438–1455.

[7] P.N. Swarztrauber, A direct method for the discrete solution of separable elliptic equations, SIAM. J. Numer. Anal., 11 (1974), 1136–1149.

Sidewall cond.	L=8	L=16	L=32	L=64
$u(r,z,\theta)=e^{r\cos\theta+r\sin\theta+z}$				
Dirichlet (Present)	3.4254E-3	9.2502E-4	2.3943E-4	6.0933E-5
Dirichlet (Fishpack)	3.8827E-3	9.8408E-4	2.4473E-4	5.6267E-5
Neumann (Present)	4.4958E-3	1.1491E-3	2.9033E-4	7.2902E-5
Neumann (Fishpack)	1.1676E-2	2.9866E-3	7.5652E-4	1.9856E-4
Robin (Present)	3.9896E-3	1.2670E-3	3.5377E-4	9.3287E-5
Convergent rate		1.65	1.84	1.92
$u(r,z,\theta)=r^3(\cos\theta+\sin\theta)z(1-z)$				
Dirichlet (Present)	5.7639E-4	1.5301E-4	3.9536E-5	1.0038E-5
Dirichlet (Fishpack)	6.4235E-4	1.6280E-4	4.0731E-5	1.0167E-5
Neumann (Present)	6.6702E-4	1.6664E-4	4.1764E-5	1.0447E-5
Neumann (Fishpack)	1.3402E-3	3.3805E-4	8.4846E-5	2.1540E-5
Robin (Present)	7.2683E-4	2.1487E-4	5.8044E-5	1.5015E-5
Convergent rate		1.76	1.89	1.95
$u(r,z,\theta)=r^2\sin(5\theta)\sin(3.5z)$				
Dirichlet (Present)	1.1556E-3	3.3747E-4	9.0391E-5	2.3343E-5
Dirichlet (Fishpack)	1.4097E-3	3.8182E-4	9.5887E-5	2.3608E-5
Neumann (Present)	2.5564E-3	7.3981E-4	1.9828E-4	5.1200E-5
Neumann (Fishpack)	3.4777E-3	8.5330E-4	2.1140E-4	5.2074E-5
Robin (Present)	1.9134E-3	5.5023E-4	1.4814E-4	3.8333E-5
Convergent rate		1.80	1.89	1.95

Table 1: Error comparisons of the present scheme and Fishpack for the cylindrical case.

Outer cond.	L=8	L=16	L=32	L=64
$u(r,\phi,\theta) = e^{r\sin\phi\cos\theta + r\sin\phi\sin\theta + r\cos\phi}$				
Dirichlet (Present)	1.7511E-2	4.1757E-3	1.0503E-3	2.6190E-4
Dirichlet (Fishpack)	1.4589E-2	3.9633E-3	1.0096E-3	2.5731E-4
Neumann (Present)	3.6855E-2	8.4165E-3	2.1187E-3	5.2816E-4
Neumann (Fishpack)	4.0324E-2	1.0957E-2	2.7160E-3	6.8381E-4
Robin (Present)	3.3031E-2	7.4906E-3	1.9040E-3	4.7698E-4
Convergent rate		2.14	1.98	2.00
$u(r,\phi,\theta) = r^4(\cos\theta + \sin\theta)\cos\phi(1 - r\cos\phi)$				
Dirichlet (Present)	5.3901E-3	1.5394E-3	4.0541E-4	1.0365E-4
Dirichlet (Fishpack)	3.9091E-3	1.5637E-3	4.6397E-4	1.2371E-4
Neumann (Present)	1.5934E-2	4.2404E-3	1.0777E-3	2.7464E-4
Neumann (Fishpack)	7.2722E-3	2.2594E-3	6.2339E-4	1.6436E-4
Robin (Present)	9.3311E-3	2.5298E-3	6.5065E-4	1.6492E-4
Convergent rate		1.88	1.96	1.98
$u(r,\phi,\theta) = \sin(r\sin\phi\cos\theta)\sin(r\sin\phi\sin\theta)\sin(r\cos\phi)$				
Dirichlet (Present)	1.3726E-3	3.9972E-4	1.0226E-4	2.5476E-5
Dirichlet (Fishpack)	1.8520E-3	4.6624E-4	1.1826E-4	2.9856E-5
Neumann (Present)	3.7529E-3	1.0292E-3	2.5834E-4	6.4408E-5
Neumann (Fishpack)	5.0679E-3	1.2342E-3	3.1357E-4	7.8481E-5
Robin (Present)	2.9288E-3	8.1316E-4	2.0478E-4	5.0971E-5
Convergent rate		1.85	1.99	2.00

Table 2: Error comparisons of the present scheme and Fishpack for the spherical case.