Fast Poisson Solver in a Three-dimensional Ellipsoid

Ming-Chih Lai

ABSTRACT. We present a simple and efficient fast direct solver for 3D Poisson problem in elliptical coordinates. The method relies on representing the solution as a truncated Fourier series, then solving the partial differential equation of Fourier coefficients by second-order finite difference discretizations. Using a grid by shifting half mesh away from the poles and incorporating the derived numerical boundary values, the difficulty of coordinate singularities can be elevated easily. Besides, the resulting linear equations can be solved very efficiently by some existing numerical algorithms.

1. Introduction

Many physical problems involve solving the Poisson equation in three-dimensional domains. For example, the projection method for the numerical solution of the incompressible Navier-Stokes equations requires solving the pressure Poisson equation. Very often, the problem we are interested in is no longer in a rectangular domain. For instance, many applications in the areas of meteorology, geophysics, and astrophysics involve solving a Poisson equation on spherical geometry. Therefore, developing a simple and efficient Poisson solver on geometries other than Cartesian becomes an important issue in the scientific computing community.

In [**3**], the author and his collaborators have developed a class of FFT-based fast direct solvers for Poisson equation in three-dimensional cylindrical and spherical domains. The methods have three major features, namely, the coordinate singularities can be treated easily, the resulting linear equations can be solved efficiently by existing fast algorithms, and the different boundary conditions can be handled without substantial differences. Besides, the method is easy to implement.

In this paper, we extend the underlying method presented in [**3**] to the solution of Poisson equation in a three-dimensional elliptical region. Instead of solving the equation in an irregular Cartesian domain, we write the equation in elliptical coordinates. Our method relies on representing the solution as a truncated Fourier series, then solving the partial differential equations of Fourier coefficients by second-order

1991 *Mathematics Subject Classification.* 65N06.

Key words and phrases. Fast Poisson solver; elliptical coordinates; prolate spheroidal coordinates; symmetry condition.

The author was supported in part by National Science Council of Taiwan under research grant NSC-91-2115-M-009-016.

finite difference discretizations. Using a grid by shifting half mesh away from the poles and incorporating the derived numerical boundary values, the main difficulty of coordinate singularities can be elevated easily. Although the present approach used in elliptical domain is similar to the one used in the spherical domain, there are still some significant differences between them. We shall discuss them in appropriate places.

2. Fast Poisson solver inside a spheroid

A spheroid or an ellipsoid of revolution is the surface of revolution produced by rotating an ellipse in three-dimensional space about one of its axes (the axis of revolution). When the ellipse is rotated about its major axis, the ellipsoid is called *prolate*, when rotated about its minor axis, it is *oblate*. In this section, we will develop the numerical solution for Poisson equation with Dirichlet boundary inside a three-dimensional prolate spheroidal region Ω. However, the present method can be applied to the oblate case without any difficulty. Let us assume that the ellipse rotating about the z axis (the axis of revolution) has the length $2a\cosh b$, and the minor axis $2a\sinh b$. Then the prolate spheroidal domain Ω can be described by

$$\Omega = \left\{ (x,y,z) \mid \frac{x^2}{(a\,\sinh b)^2} + \frac{y^2}{(a\,\sinh b)^2} + \frac{z^2}{(a\,\cosh b)^2} \le 1 \right\}. \tag{2.1}$$

It is more convenient to use the prolate spheroidal coordinates (ρ, ϕ, θ) to represent the domain Ω and to write the governing equation in those new coordinates. The Cartesian-Prolate spheroidal coordinates transformation are

$$x = a\,\sinh\rho\,\sin\phi\,\cos\theta, \quad y = a\,\sinh\rho\,\sin\phi\,\sin\theta, \quad z = a\,\cosh\rho\,\cos\phi, \tag{2.2}$$

so the domain Ω can be described as $\{(\rho,\phi,\theta) | 0 < \rho \le b, 0 \le \phi \le \pi, 0 \le \theta < 2\pi\}$. The prolate spheroidal surface $\rho = b$ is the boundary of Ω.

Using the prolate spheroidal coordinates, the Poisson equation in Ω can be written as [**2**]

$$\frac{1}{h^2}\left(\frac{\partial^2 u}{\partial \rho^2} + \coth\rho\,\frac{\partial u}{\partial \rho} + \frac{\partial^2 u}{\partial \phi^2} + \cot\phi\,\frac{\partial u}{\partial \phi}\right) + \frac{1}{a^2\,\sinh^2\rho\,\sin^2\phi}\,\frac{\partial^2 u}{\partial \theta^2} = f, \tag{2.3}$$

where the factor $h^2 = a^2(\sinh^2\rho + \sin^2\phi)$. Multiplying h^2 on both sides of Eq. (2.3) and imposing the boundary condition, we have the Dirichlet problem

$$\frac{\partial^2 u}{\partial \rho^2} + \coth\rho\,\frac{\partial u}{\partial \rho} + \frac{\partial^2 u}{\partial \phi^2} + \cot\phi\,\frac{\partial u}{\partial \phi} + \left(\frac{1}{\sinh^2\rho} + \frac{1}{\sin^2\phi}\right)\frac{\partial^2 u}{\partial \theta^2} = h^2\,f, \tag{2.4}$$

$$u(b,\phi,\theta) = g(\phi,\theta). \tag{2.5}$$

Once we write the equation in prolate spheroidal coordinates, it becomes to solve the equation in (ρ, ϕ, θ) rectangular domain. Let us discuss what kind of boundary conditions can be imposed on this rectangular domain. First of all, the periodic boundary condition is imposed in the θ direction. The boundary value at $\rho = b$ is given by the Dirichlet data. However, there is lack of boundary conditions at $\rho = 0$ and $\phi = 0, \pi$ which are the corresponding coordinates singularities of Eq. (2.4). This can be seen easily from the equation (2.4) which is singular at $\rho = 0$ and $\phi = 0, \pi$. So the main difficulty for solving Eqs. (2.4)-(2.5) is how to handle those coordinate singularities. In the following, we adapt the similar approach used for spherical domain [**3**] to develop a fast direct solver for Eqs. (2.4)-(2.5).

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

$$u(\rho,\phi,\theta) = \sum_{n=-N/2}^{N/2-1} \widehat{u_n}(\rho,\phi)\, e^{in\theta}, \tag{2.6}$$

where $\widehat{u_n}(\rho,\phi)$ is the complex Fourier coefficient given by

$$\widehat{u_n}(\rho,\phi) = \frac{1}{N}\sum_{k=0}^{N-1} u(\rho,\phi,\theta_k)\, e^{-in\theta_k}, \tag{2.7}$$

with $\theta_k = 2k\pi/N$ and N is the number of grid points along a circle. Substituting those expansions into Eq. (2.4), and equating the Fourier coefficients, $\widehat{u_n}(\rho,\phi)$ then satisfies the partial differential equation

$$\frac{\partial^2\widehat{u_n}}{\partial\rho^2} + \coth\rho\,\frac{\partial\widehat{u_n}}{\partial\rho} + \frac{\partial^2\widehat{u_n}}{\partial\phi^2} + \cot\phi\,\frac{\partial\widehat{u_n}}{\partial\phi} - n^2\left(\frac{1}{\sinh^2\rho} + \frac{1}{\sin^2\phi}\right)\widehat{u_n} = \widehat{f_n}(\rho,\phi) \tag{2.8}$$

$$\widehat{u_n}(b,\phi) = \widehat{g_n}(\phi), \tag{2.9}$$

where $\widehat{f_n}(\rho,\phi)$ and $\widehat{g_n}(\phi)$ are the corresponding Fourier coefficients of the functions $h^2 f$ and g, respectively, and are defined in the same manner as Eqs. (2.6)-(2.7). Next, we will solve Eq. (2.8) by a second-order finite difference discretization.

2.2. Second-order finite difference discretization. We first choose a grid in the (ρ,ϕ) plane by

$$\rho_i = (i-1/2)\,\Delta\rho, \qquad \phi_j = (j-1/2)\,\Delta\phi, \tag{2.10}$$

for $0 \le i \le L+1, 0 \le j \le M+1$ with $\Delta\rho = 2b/(2L+1)$ and $\Delta\phi = \pi/M$. By choosing these mesh points, we avoid placing points directly at the points $\rho = 0$ and $\phi = 0,\pi$. Again, let the discrete values be denoted by $U_{ij} \approx \widehat{u_n}(\rho_i,\phi_j)$, and $F_{ij} \approx \widehat{f_n}(\rho_i,\phi_j)$. The unknown indexes run from $i = 1$ to $i = L$, and $j = 1$ to $j = M$. Using centered difference approximations to discretize Eq. (2.8) at the internal mesh points $(\rho_i,\phi_j), 1 \le i \le L, 1 \le j \le M$, we have

$$\begin{aligned} &\frac{U_{i+1,j} - 2\,U_{ij} + U_{i-1,j}}{(\Delta\rho)^2} + \coth\rho_i\,\frac{U_{i+1,j} - U_{i-1,j}}{2\,\Delta\rho} + \frac{U_{i,j+1} - 2\,U_{ij} + U_{i,j-1}}{(\Delta\phi)^2} \\ &\quad + \cot\phi_j\,\frac{U_{i,j+1} - U_{i,j-1}}{2\Delta\phi} - n^2\left(\frac{1}{\sinh^2\rho_i} + \frac{1}{\sin^2\phi_j}\right)U_{ij} = F_{ij}. \end{aligned} \tag{2.11}$$

Incorporating the numerical boundary values derived below, the above linear system of equations resulting from the finite difference discretization of the separable elliptic equation (2.8) can be solved by the generalized cyclic reduction algorithm efficiently (with $O(ML\log_2 L)$ operations) [**5**]. This linear solver is written as a subroutine called CBLKTR.f which can be obtained from the popular software package Fishpack [**1**].

2.3. Numerical boundary values. In order to close the linear systems of (2.11), the numerical boundary values $U_{0j}, U_{L+1,j}, U_{i0}$ and $U_{i,M+1}$ must be provided. Under the grid (2.10), we have $\rho_{L+1} = b$; thus, the outer numerical boundary value $U_{L+1,j}$ is simply given by the boundary value $\widehat{g_n}(\phi_j)$. The inner numerical boundary value $U_{0j} \approx \widehat{u_n}(-\Delta\rho/2,\phi_j)$ can be obtained by the symmetry condition of Fourier coefficients as follows.

In the Cartesian-Prolate spheroidal coordinates transformation (2.2), if we replace ρ by $-\rho$, and θ by $\theta+\pi$, the Cartesian coordinates of a point remains the same. That is, any scalar function $u(\rho,\phi,\theta)$ satisfies $u(-\rho,\phi,\theta)=u(\rho,\phi,\theta+\pi)$. Using this equality, we have

$$\begin{aligned} u(-\rho,\phi,\theta) &= \sum_{n=-\infty}^{\infty} \widehat{u_n}(-\rho,\phi)\, e^{in\theta} \\ = u(\rho,\phi,\theta+\pi) &= \sum_{n=-\infty}^{\infty} \widehat{u_n}(\rho,\phi)\, e^{in(\theta+\pi)} = \sum_{n=-\infty}^{\infty} (-1)^n\, \widehat{u_n}(\rho,\phi)\, e^{in\theta}. \end{aligned} \tag{2.12}$$

Thus, when the domain of a function is extended to a negative value of ρ, the nth Fourier coefficient of this function satisfies

$$\widehat{u_n}(-\rho,\phi) = (-1)^n\, \widehat{u_n}(\rho,\phi). \tag{2.13}$$

Therefore, the numerical boundary value U_{0j} can be approximated by

$$U_{0j} \approx \widehat{u_n}(\rho_0,\phi_j) = \widehat{u_n}(-\Delta\rho/2,\phi_j) = (-1)^n\, \widehat{u_n}(\Delta\rho/2,\phi_j) \approx (-1)^n\, U_{1j}. \tag{2.14}$$

It is worth mentioning that the shifting half mesh grid is not used in the radial direction for the spherical coordinates case [**3**]. Instead, we use the uniform grid $r_i = i\Delta r$ so that the coefficient of U_{0j} in the finite difference scheme becomes zero. Thus, no pole approximation for U_{0j} is needed. This is the major difference between solving the Poisson equation in spherical and prolate spheroidal domains.

Applying the similar arguments, we can derive

$$\widehat{u_n}(\rho,-\phi) = (-1)^n\, \widehat{u_n}(\rho,\phi), \qquad \widehat{u_n}(\rho,\pi+\phi) = (-1)^n\, \widehat{u_n}(\rho,\pi-\phi). \tag{2.15}$$

Thus, the numerical boundary values U_{i0} and $U_{i,M+1}$ can be obtained by

$$U_{i0} \approx \widehat{u_n}(\rho_i,\phi_0) = \widehat{u_n}(\rho_i,-\Delta\phi/2) = (-1)^n\, \widehat{u_n}(\rho_i,\Delta\phi/2) \approx (-1)^n\, U_{i1}, \tag{2.16}$$

$$\begin{aligned} U_{i,M+1} &\approx \widehat{u_n}(\rho_i,\phi_{M+1}) = \widehat{u_n}(\rho_i,\pi+\Delta\phi/2) \\ &= (-1)^n\, \widehat{u_n}(\rho_i,\pi-\Delta\phi/2) \approx (-1)^n\, U_{iM}. \end{aligned} \tag{2.17}$$

We close the section by summarizing the algorithm and the operation counts in the following three steps:

1. Compute the Fourier coefficients for the right-hand side function as in (2.7) by FFT, which requires $O(MLN\log_2 N)$ arithmetic operations.

2. Solve the block tridiagonal linear system (2.11) using the generalized cyclic reduction method for each Fourier mode. This requires $O(MLN\log_2 L)$ operations.

3. Convert the Fourier coefficients as in (2.6) by FFT to obtain the solution, which requires $O(MLN\log_2 N)$ operations. The overall operation count is thus $O(LMN\log_2(LN))$ for $L\times M\times N$ grid points.

3. Numerical results

In this section, we perform several numerical tests for the accuracy of our fast direct solver presented in Section 2. All numerical runs were carried out on a PC workstation with 512 RAM using double precision arithmetics. Our code was written in Fortran and can be available for public usage upon request.

Throughout this paper, the error is measured by the L_∞ relative error

$$E_\infty = \frac{\|u-U\|_\infty}{\|u\|_\infty}, \tag{3.1}$$

where u and U represent the exact and computed solutions, respectively. The order of accuracy (or rate of convergence) is calculated by the ratio of two consecutive errors. The following problems are simply described by the solution u in Cartesian coordinates so that the right hand side function f and the boundary conditions can easily be obtained from u. The domain parameters a and b are chosen by $a = 1$ and $b = 0.5$ for all test problems.

Example 1. The first example is taken as

$$u(x, y, z) = e^{x+y+z}. \tag{3.2}$$

Table 1 shows the relative errors in maximum norm and the rate of convergence of our second-order method. Indeed, the clean second-order accuracy has been seen.

(L,M,N)	error	rate
(16,32,16)	3.6927E-04	-
(32,64,32)	9.0408E-05	2.03
(64,128,64)	2.2379E-05	2.01

TABLE 1. The numerical results obtained by the second-order solver in a 3D spheroid for Example 1.

Example 2. The second solution is given by

$$u(x, y, z) = \frac{1}{x^2 + 2y^2 + 2z^2 + 1}. \tag{3.3}$$

Again, the second-order convergence has been achieved.

(L,M,N)	error	rate
(16,32,16)	1.7257E-03	-
(32,64,32)	4.4426E-04	1.96
(64,128,64)	1.1202E-04	1.99

TABLE 2. The numerical results obtained by the second-order solver in a 3D spheroid for Example 2.

References

[1] J Adams, P. Swarztrauber and R. Sweet, *Fishpack- A package of Fortran subprograms for the solution of separable elliptic partial differential equations*, 1980. Available in http://www.netlib.org/fishpack.
[2] A. Jeffrey, *Handbook of Mathematical Formulas and Integrals*, Second edition, Academic Press, (2000).
[3] M.-C. Lai, W.-W. Lin and W. Wang, *A fast spectral/difference method without pole conditions for Poisson-type equations in cylindrical and spherical geometries*, IMA J. Numer. Anal., vol 22, 537-548, (2002).
[4] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth & Brooks/Cole, (1989), pp.68.

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

Department of Applied Mathematics, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 300, TAIWAN

E-mail address: mclai@math.nctu.edu.tw