Annals of Mathematical Sciences and Applications
Volume x, Number x, 1, 2019

Fast spectral solver for Poisson equation in an annular domain*

T.-S. Lin†, C.-Y. He, and W.-F. Hu

A simple and efficient spectral method is formulated to solve Poisson equation in an annular domain. The solver relies on the Fourier expansion, where the differential equations for the Fourier coefficients are solved using an ultraspherical spectral method. For a domain with N grid points in the polar direction and M grid points in the radial direction, the solver only requires $O(NM \log_2 N)$ arithmetic operations.

AMS 2000 subject classifications: Primary 65N35.
Keywords and phrases: fast Fourier transform, ultraspherical spectral method, fast Poisson solver, annular domain.

1. Introduction

The aim of the present manuscript is to develop a simple and efficient spectral method to solve Poisson equation in an annular domain. This problem had been well-studied and fast solvers had been constructed using Fourier and 2nd-order finite difference method [5, 7], Fourier and compact 4th-order finite difference method [6], spectral-Galerkin algorithms [11] and Fourier-Chebyshev algorithm [12]. All these approaches took advantages of the Fourier transform to transform Poisson equation, a partial differential equation (PDE), to a set of ordinary differential equations (ODEs) for each of the Fourier modes. The ODEs are then approximated by sparse or structured linear systems that can be solved efficiently. In the present manuscript, we follow the same idea to transform the PDE into a set of ODEs. The ODEs are then solved using ultraspherical spectral method [10]. The advantage of the present approach is that it converges to the solution super-algebraically, i.e., it is faster than any algebraic convergence, and only requires solving an almost banded, well-conditioned linear system.

*This work was supported by the Ministry of Science and Technology of Taiwan under grants MOST-107-2115-M-009-008-MY2 (TSL and CYH) and MOST-107-2115-M-005-004-MY2 (WFH).

†Corresponding author.

The rest of the manuscript is organized as follows. In Section 2, we provide a detailed presentation for the Fourier-ultraspherical method to solve Poisson equation in an annular domain. In Section 3 several numerical tests are conducted. Finally, the concluding remarks are given in Section 4.

2. Fast Fourier-ultraspherical spectral solver for Poisson equation in an annular domain

We consider Poisson equation in a two-dimensional annular domain that, written in polar coordinate system, has the form

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} = f(r,\theta) \quad 0 \le R_i < r < R_o, \quad 0 \le \theta < 2\pi, \tag{1}$$

where R_i and R_o are given fixed numbers that define either an annular domain ($R_i > 0$) or a disk ($R_i = 0$). The boundary conditions, either Dirichlet, Neumann or Robin type, shall be specified to ensure the well-posedness of the problem. In the following, we will assume that the Dirichlet boundary conditions are imposed:

$$u(R_i,\theta) = g(\theta), \quad u(R_o,\theta) = h(\theta), \tag{2}$$

for some functions g and h. However, we note that, the boundary conditions of the proposed method are not limited to Dirichlet type: Neumann or Robin boundary conditions can be imposed easily as well.

We approximate the solution u by the truncated Fourier series as

$$u(r,\theta) = \sum_{n=-N/2+1}^{N/2} \hat{u}_n(r) e^{in\theta}, \tag{3}$$

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

$$\hat{u}_n(r) = \frac{1}{N}\sum_{j=0}^{N-1} u(r,\theta_j) e^{-in\theta_j}, \tag{4}$$

and $\theta_j = 2j\pi/N$ where N is the number of grid points in the θ-direction. It can be found easily that the n-th Fourier mode $\hat{u}_n(r)$ satisfies the ordinary differential equation

$$\tfrac{d^2\hat{u}_n}{dr^2} + \tfrac{1}{r}\tfrac{d\hat{u}_n}{dr} - \tfrac{n^2}{r^2}\hat{u}_n = \hat{f}_n, \quad R_i < r < R_o, \tag{5}$$

$$\hat{u}_n(R_i) = \hat{g}_n, \quad \hat{u}_n(R_o) = \hat{h}_n, \tag{6}$$

where $\hat{f}_n(r)$, $\hat{g}_n$ and $\hat{h}_n$ are the Fourier coefficients of f, g and h, respectively, defined in the same manner as $\hat{u}_n$. In the following we shall explain how to solve equations (5, 6) based on the ultraspherical spectral method.

For brevity of notations, in the following we drop the hat and subscript and denote the Fourier coefficients $U(r) = \hat{u}_n(r)$, $F(r) = \hat{f}_n(r)$, $G = \hat{g}_n$ and $H = \hat{h}_n$.

2.1. Ultraspherical spectral method

The ultraspherical (or Gegenbauer) polynomials, $\{C_i^{(\lambda)}(x),\, i = 1, \cdots\}$, are a family of orthogonal polynomials on the interval $[-1, 1]$ with respect to the weight function $(1-x^2)^{\lambda-1/2}$. Two special cases are $\lambda = 0$, the Chebyshev polynomials of the first kind (also denoted by $\{T_i(x)\}$), and $\lambda = 1$, the Chebyshev polynomials of the second kind [1]. More properties of the ultraspherical polynomials can be found at Ref. [9].

In order to take advantages of the ultraspherical polynomials, we introduce a new variable x that is defined on $[-1, 1]$ and make the following transformation between variables r and x:

$$x = \alpha r - \beta, \quad \alpha = \frac{2}{R_o - R_i}, \quad \beta = \frac{R_o + R_i}{R_o - R_i}.$$

The Fourier mode equations (5, 6) can then be rewritten in terms of the new variable as

$$(x+\beta)^2 \tfrac{d^2U}{dx^2} + (x+\beta)\tfrac{dU}{dx} - n^2\, U = \left(\tfrac{x+\beta}{\alpha}\right)^2 F, \quad -1 < x < 1, \tag{7}$$

$$U(-1) = G, \quad U(1) = H. \tag{8}$$

Next, we express the function $U(x)$ in terms of Chebyshev basis functions

$$U(x) = \sum_{k=0}^{\infty} c_k T_k(x), \tag{9}$$

and seek to solve its coefficients c_k's.

Following the approaches of Olver et. al. [10], we rewrite equations (7, 8) in the coefficient space. To do so, we need to take into account the differentiation, conversion and multiplication operators in the coefficient space. Specifically, the first and second order differentiation matrices are defined as

$$D_1 = \begin{pmatrix} 0 & 1 & & & \\ & & 2 & & \\ & & & 3 & \\ & & & & \ddots \end{pmatrix}, \quad D_2 = \begin{pmatrix} 0 & 0 & 4 & & & & \\ & & & 6 & & & \\ & & & & 8 & & \\ & & & & & 10 & \\ & & & & & & \ddots \end{pmatrix},$$

respectively. We note that differentiation is realized by matrix-vector multiplication where by performing $D_1\mathbf{c}$ we obtain the coefficients of $\frac{dU}{dx}$ in terms of the $C^{(1)}$ basis functions, where $\mathbf{c} = [c_0, c_1, \cdots]^T$. Similarly, by performing $D_2\mathbf{c}$ we obtain the coefficients of $\frac{d^2U}{dx^2}$ in terms of the $C^{(2)}$ basis functions.

In practice we need to truncate the infinite Chebyshev series expansions into a finite one, and the matrix representing the linear operators are of finite size correspondingly. Here we formulate the problem firstly in infinite expansions and infinite matrices. The truncation of terms and matrices for practical usage are attained by introducing the projection matrix later in this section.

The conversion matrices to transfer between different basis functions are defined as the following: The operator S_0 that converts coefficients in $C^{(0)}$ to $C^{(1)}$ and S_1 that converts coefficients in $C^{(1)}$ to $C^{(2)}$ are defined as

$$S_0 = \begin{pmatrix} 1 & 0 & -\frac{1}{2} & & \\ & \frac{1}{2} & 0 & -\frac{1}{2} & \\ & & \frac{1}{2} & 0 & \ddots \\ & & & \frac{1}{2} & \ddots \\ & & & & \ddots \end{pmatrix}, \quad S_1 = \begin{pmatrix} 1 & 0 & -\frac{1}{3} & & \\ & \frac{1}{2} & 0 & -\frac{1}{4} & \\ & & \frac{1}{3} & 0 & \ddots \\ & & & \frac{1}{4} & \ddots \\ & & & & \ddots \end{pmatrix},$$

respectively.

Notice that equation (7) is an ODE with non-constant coefficients, so we need to take into account the multiplication between functions in the coefficient space. The multiplication matrix for $(x+\beta)^2$, denoted by M_2, should be written under the $C^{(2)}$ basis. It was found that M_2 is a pentadiagonal matrix that has entries, for $j \geq 1$,

$$\begin{aligned} M_2(j,j) &= \beta^2 + \frac{1}{6} + \frac{(j-1)(j+3)}{3j(j+2)}, \quad M_2(j,j+1) = \frac{j+3}{j+2}\,\beta, \\ M_2(j+1,j) &= \frac{j}{j+1}\,\beta, \quad M_2(j,j+2) = \frac{(j+4)}{4(j+2)}, \quad M_2(j+2,j) = \frac{j}{4(j+2)}. \end{aligned}$$

The multiplication matrix for $(x+\beta)$, denoted by M_1, should be written under the $C^{(1)}$ basis. It was found that

$$M_1 = \begin{pmatrix} \beta & 0.5 & & & \\ 0.5 & \beta & 0.5 & & \\ & 0.5 & \beta & 0.5 & \\ & & 0.5 & \beta & \ddots \\ & & & \ddots & \ddots \end{pmatrix},$$

that is a tri-diagonal matrix.

Finally, we are able to rewrite equation (7) as

$$L\mathbf{c} = S_1 S_0 \mathbf{F}, \tag{10}$$

where $L := M_2 D_2 + S_1 M_1 D_1 - n^2\, S_1 S_0$, and $\mathbf{F}$ is a vector that represents the Chebyshev coefficients of the function $((x+\beta)/\alpha)^2 F(x)$.

2.2. Boundary conditions

Boundary conditions are forced directly that can be written down as a linear system

$$B\mathbf{c} = \mathbf{d}, \tag{11}$$

where

$$B = \begin{pmatrix} T_0(-1) & T_1(-1) & T_2(-1)\dots \\ T_0(1) & T_1(1) & T_2(1)\dots \end{pmatrix},$$

and $\mathbf{d} = [G, H]^T$ are the imposed Dirichlet boundary conditions.

Remark 1. *We note that different kinds of conditions can be imposed easily in a similar manner. For example, consider a Neumann boundary condition at $r = R_o$, i.e.,*

$$\frac{\partial}{\partial r} u(R_o, \theta) = h(\theta). \tag{12}$$

The equation in the coefficient space is then written as

$$\begin{pmatrix} T_0'(1) & T_1'(1) & T_2'(1)\dots \end{pmatrix} \mathbf{c} = H.$$

2.3. Full linear system

The full linear system presenting equations (7, 8) is obtained by putting together the discrete equation of the ODE, (10), and the equation for the boundary conditions, (11). We seek to represent the solution in terms of a $(M-1)$th-degree Chebyshev polynomial. By introducing the projection matrix $P_n = (I_n, \mathbf{0})$, where I_n is a n-by-n identity matrix and $\mathbf{0}$ is a n-by-∞ zero matrix, the full system is given by

$$\begin{pmatrix} BP_M^T \\ \\ P_{M-2}LP_M^T \\ \\ \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{M-1} \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \\ P_{M-2}S_1S_0\mathbf{F} \\ \\ \end{pmatrix}, \tag{13}$$

where the superscript T denotes matrix transpose. This system is a sparse one that consists of two dense rows representing the boundary conditions and a penta-diagonal matrix. It can be solved efficiently by using the Woodbury formula [3]:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}. \tag{14}$$

In this way, the overall cost of the proposed fast Poisson solver is as follows. Solving one Fourier mode equation takes $O(M)$ operations and the cost of solving all the Fourier mode equations is $O(NM)$. Besides, the Fourier transform requires $O(N \log_2 N)$ operations in each azimuthal direction and is $O(NM \log_2 N)$ in the whole domain. In total, the solver requires $O(NM \log_2 N)$ operations.

Remark 2. *For Poisson equation in a unit disk, $R_i = 0$, the boundary conditions required to be imposed at $r = 0$ is the so-called "pole conditions". We follow Shen [11] to impose the "essential pole conditions"*

$$\hat{u}_n(r = 0) = 0, \quad n \neq 0. \tag{15}$$

In such a case, for $n \neq 0$ the boundary conditions are of Dirichlet type at R_i so the same system of equations (13) is obtained. For $n = 0$, there is no condition at R_i so the equation for the boundary condition becomes

$$B = \begin{pmatrix} T_0(1) & T_1(1) & T_2(1) \ldots \end{pmatrix},$$

and $\boldsymbol{d} = H$. *The full system is then given by*

$$\begin{pmatrix} BP_M^T \\ \\ P_{M-1}LP_M^T \\ \\ \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{M-1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{d} \\ \\ P_{M-1}S_1S_0\boldsymbol{F} \\ \\ \end{pmatrix}. \tag{16}$$

3. Numerical examples

In this section, we perform numerical tests for the present Fourier-ultraspherical spectral method. The implementation in Matlab can be found at Ref. [8].

3.1. Annular domain

We solve Poisson equation in an annular domain with $R_i = 0.5$ and $R_o = 1$. Two cases are studied with exact solutions

1. $u(r,\theta) = \sin(10r\cos\theta)$,
2. $u(r,\theta) = \exp(r\cos\theta + r\sin\theta)$.

We fix the number of grid points in the θ-direction $N = 100$ and vary the ones in the r-direction. The results of the error in L_∞-norm ($\|u - v\|_\infty = \sup_{\mathbf{x}} |u(\mathbf{x}) - \mathbf{v}(\mathbf{x})|$) are shown in Table 1. One can see clearly the spectral convergence of the solutions.

Table 1: Poisson equation in an annular domain. The values on the table show the L_∞-error between the numerical solutions and the exact solutions.

M	10	14	18	20	100
$u(r,\theta) = \sin(10r\cos\theta)$	$1.30e-05$	$1.40e-09$	$4.75e-14$	$3.41e-15$	$3.62e-15$
$u(r,\theta) = \exp(r\cos\theta + r\sin\theta)$	$1.54e-13$	$1.78e-15$	$1.78e-15$	$1.33e-15$	$2.16e-15$

3.2. Disk domain

With $R_i = 0$ we solve Poisson equation in a unit disk with essential pole condition as discussed in Remark 2. Similar to annular domain we study two examples and fix the number of grid points in the θ-direction $N = 100$ and vary the ones in the r-direction. The results of the error in L_∞-norm are shown in Table 2. Again, we see clearly the spectral convergence of the solutions.

Table 2: Poisson equation in a unit disk. The values on the table show the L_∞-error between the numerical solutions and the exact solutions.

M	10	14	20	30	100
$u(r,\theta) = \sin(10r\cos\theta)$	$6.27e-02$	$1.94e-04$	$2.96e-09$	$8.60e-15$	$1.86e-13$
$u(r,\theta) = \exp(r\cos\theta + r\sin\theta)$	$1.67e-09$	$8.66e-15$	$4.78e-15$	$4.32e-15$	$1.94e-14$

Remark 3. *As pointed out by Olver et. al. [10], the condition number of the matrix of the linear system (13) grows proportional to M. We have verified that indeed for annular domains, $R_i > 0$, the condition number of the linear systems grows as $O(M)$. See, for example, numerical results in Table 3. However, for a disk domain, $R_i = 0$, it is found that the condition number of the system grows as $O(M^4)$. As a consequence the error may grow significantly when one uses too many grid points. For example, in Table 2 one can see that the solutions are accurate to machine precision at $M = 30$ but the error become larger at $M = 100$.*

Table 3: Condition number for the linear system of (13) for an annular domain $R_i = 0.5$ and $R_o = 1$.

M	20	200	2000	20000
annular domain	$6.16 * 10^2$	$6.85 * 10^3$	$6.94 * 10^4$	$6.95 * 10^5$

Remark 4. *For the disk domain, $R_i = 0$, grid points are clustered at the pole that may cause numerical instabilities. To avoid such an ill-conditioning situation, fast Poisson solvers were formulated through the use of parity properties in a disk [12, 13], in a cylinder [2], or formulated through a low rank approximations [14].*

4. Conclusion

In the present manuscript a fast Poisson solver based on Fourier-ultraspherical spectral method in an annular domain is formulated. The algorithm is simple and easy to implement: all the required matrices are almost-banded and are given explicitly. The resultant matrix for the problem is almost-banded that can be inverted efficiently based on Woodbury formula. The overall cost of the algorithm is $O(NM\log_2 N)$ and the numerical results confirm the super-algebraical convergence to the solution. Finally, we note that the present elliptic solver can be extended straightforwardly to solve the diffusion equation in an annular domain, and had been implemented to study the chaotic swimming motion of phoretic particles [4].

References

[1] B. Fornberg, *A practical guide to pseudospectral method*, Cambridge University Press, 1996.

[2] D. Fortunato and A. Townsend, *Fast Poisson solvers for spectral methods*, https://arxiv.org/abs/1710.11259.

[3] G. H. Golub and C. F. Van Loan, *Matrix computation*, Johns Hopkins University Press, 1996.

[4] W.-F. Hu, T.-S. Lin, S. Rafai, and C. Misbah, *Chaotic swimming of phoretic self-propelled particles*, submitted, (2019).

[5] A. Iserles, *A first course in the numerical analysis of differential equations*, Cambridge University Press, 1996.

[6] M.-C. Lai, *A simple compact fourth-order Poisson solver on polar geometry*, J. Comput. Phys., 182 (2002), pp. 337–345.

[7] M.-C. Lai and W.-C. Wang, *Fast direct solvers for Poisson equation on 2D polar and spherical geometries*, Numer. Methods for Partial Diff. Eqs., 18 (2002), pp. 56–68.

[8] T.-S. Lin, https://github.com/teshenglin/fast_poisson_solvers.

[9] F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, *NIST Handbook of Mathematical Functions*, Cambridge University Press, 2010.

[10] S. Olver and A. Townsend, *A fast and well-conditioned spectral method*, SIAM Rev., 55 (2013), pp. 462–489.

[11] J. Shen, *Efficient spectral-galerkin methods III: Polar and cylindrical geometries*, SIAM J. Sci. Comput., 18 (1997), pp. 1583–1604.

[12] J. Shen, *A new fast Chebyshev-Fourier algorithm for Poisson-type equations in polar geometries*, Appl. Num. Math., 33 (2000), pp. 183–190.

[13] L. Y. Trefethen, *Spectral methods is Matlab*, SIAM, 2001.

[14] H. Wilber, A. Townsend, and G. B. Wright, *Computing with functions in spherical and polar geometries II. The disk*, SIAM J. Sci. Comput., 39 (2017), pp. C238–C262.

T.-S. Lin
Department of Applied Mathematics
National Chiao Tung University,
Hsinchu 300, Taiwan
E-mail address: tslin@math.nctu.edu.tw

C.-Y. He
Department of Applied Mathematics
National Chiao Tung University,
Hsinchu 300, Taiwan

W.-F. Hu
Department of Applied Mathematics
National Chung Hsing University,
Taichung 402, Taiwan

Received 20 August 2019