An FFT Based Fast Poisson Solver on Spherical Shells

Yin-Liang Huang\(^1\), Jian-Guo Liu\(^2\) and Wei-Cheng Wang\(^3\)

\(^1\) Department of Applied Mathematics, National University of Tainan, Tainan 70005, Taiwan.
\(^2\) Department of Physics and Department of Mathematics, Duke University, Durham, NC 27708, USA.
\(^3\) Department of Mathematics, National Tsing Hua University, Hsinchu 30013, Taiwan.

Received 6 May 2009; Accepted (in revised version) 8 June 2009
Available online 17 September 2010

To the memory of David Gottlieb

Abstract. We present a fast Poisson solver on spherical shells. With a special change of variable, the radial part of the Laplacian transforms to a constant coefficient differential operator. As a result, the Fast Fourier Transform can be applied to solve the Poisson equation with \(O(N^3 \log N)\) operations. Numerical examples have confirmed the accuracy and robustness of the new scheme.

AMS subject classifications: 35Q86, 65N06, 65N15, 65N22, 65T50

Key words: Poisson equation, spherical coordinate, FFT, spectral-finite difference method, fast diagonalization, high order accuracy, error estimate, trapezoidal rule, Euler-Maclaurin formula, Bernoulli numbers.

1 Introduction

The purpose of this paper is to propose a simple fast solver for the Poisson equation in a spherical shell

\[
\begin{cases}
\frac{\partial}{\partial \rho} \left( \rho^2 \frac{\partial u}{\partial \rho} \right) + \frac{\partial}{\partial \theta} \left( \sin \theta \frac{\partial u}{\partial \theta} \right) + \frac{\partial^2 u}{\rho^2 \sin^2 \theta} = f, & \text{in } \Omega, \\
u \big|_{\rho = \rho_{\min}} = u^L(\theta, \phi), \\
u \big|_{\rho = \rho_{\max}} = u^R(\theta, \phi),
\end{cases}
\]

(1.1)
where
\[ \Omega = \left\{ \rho_{\text{min}} < \rho < \rho_{\text{max}}, \ 0 \leq \theta \leq \pi, \ 0 \leq \phi \leq 2\pi \right\}. \]

The Poisson equation in the spherical shell geometry is important in many geophysical and solar-physical applications [5, 14, 15].

Eq. (1.1) can be put in a more symmetric form
\[
\begin{align*}
\frac{\partial}{\partial \rho} \left( \rho^2 \partial_{\rho} \sin^2 \theta \ u \right) + \left( \sin \theta \partial_{\theta} \right)^2 u + \partial_{\phi}^2 u &= \rho^2 (\sin^2 \theta) f, \quad \text{in} \ \Omega, \\
u|_{\rho=\rho_{\text{min}}} &= u^L(\theta,\phi), \\
u|_{\rho=\rho_{\text{max}}} &= u^R(\theta,\phi).
\end{align*}
\]

In this symmetric form (1.2), one can apply Fast Fourier Transform to both the \( \theta \) and \( \phi \) derivatives (see Section 2 for details) to obtain optimal efficiency. The major obstacle for developing an overall fast solver is the radial derivatives which constitute a variable coefficient differential operator. The most popular approaches include Poisson solvers based on FFT in two directions or spherical harmonic functions which requires a Fast Legendre transform [1, 4, 6, 7, 9, 12, 13, 16]. There are also other approaches using different sets of grids such as the Cubed Sphere grid [11] and the Yin-Yang grid [17].

In this paper, we propose a simple alternative, which provides a more accessible fast solver to (1.2) via FFT in all three variables. We propose the following simultaneous change of dependent and independent variables
\[
\begin{align*}
s &= \frac{\ln \rho - \ln \rho_{\text{min}}}{\ln \rho_{\text{max}} - \ln \rho_{\text{min}}}, \quad (1.3a) \\
v &= \sqrt{\rho} \ u. \quad (1.3b)
\end{align*}
\]

It is easy to see that, under the transformation (1.3), the Poisson equation (1.1) now takes the form
\[
\sin^2 \theta \left( \alpha \partial_s^2 - \frac{1}{4} \right) v + \left( \sin \theta \partial_{\theta} \right)^2 v + \partial_{\phi}^2 v = g \equiv \rho^2 \sin^2 \theta f, \quad (1.4)
\]
where
\[ \alpha = (\ln \rho_{\text{max}} - \ln \rho_{\text{min}})^{-2}, \quad (1.5) \]
with boundary data
\[
\begin{align*}
v|_{s=0} = v^L(\theta,\phi) &= \sqrt{\rho_{\text{min}}} \ u^L(\theta,\phi), \\
v|_{s=1} = v^R(\theta,\phi) &= \sqrt{\rho_{\text{max}}} \ u^R(\theta,\phi). \quad (1.6a)
\end{align*}
\]

The significance of the transformation (1.3) is that the radial part now becomes a constant coefficient differential operator. As a consequence, the discretized operator for \( (\alpha \partial_s^2 - 1/4) \) can be fast-diagonalized via FFT, resulting in an fast solver with total \( O(N^3 \log N) \)