A Fast-high Order Algorithm for Three-dimensional Poisson Equations

Wen-jie HE and Na ZHU*
School of Mathematics and Physics, North China Electric Power University Department, Baoding, China
*Corresponding author

Keywords: Three-dimensional Poisson equation, Fast Fourier transform, High-order algorithm.

Abstract. The three-dimensional Poisson equations widely exist in many physical or engineering problems. We proposed a fourth-order fast algorithm for solving the three-dimensional Poisson equation. By the fast Fourier transform, block tridiagonal structure can be generated, and the original problem can be easily decomposed into small independent systems. Fourier operators accelerate the process of solving numerical solutions and greatly reduce the computation time. The accuracy and efficiency of the method are verified by several numerical experiments.

Introduction
In the paper, we consider the following Poisson equation

\[ \Delta u(x, y, z) = -f(x, y, z), \text{ in } \Omega \]  \hspace{1cm} (1)

with Dirichlet boundary condition

\[ u(x, y, z) = u_0(x, y, z), \text{ on } \partial \Omega \]  \hspace{1cm} (2)

where \( \Omega \) is a three-dimensional continuous convex domain and \( \partial \Omega \) is its boundary. The source function \( f(x, y, z) \), the boundary condition \( u_0(x, y, z) \) and the exact solution \( u(x, y, z) \) are sufficiently smooth and have the necessary continuous partial derivatives up to certain orders.

The three-dimensional Poisson equation is a kind of partial differential equation with wide application range, and its numerical calculation plays a very important role in the numerical simulation of fluid mechanics, heat and mass transfer and so on. In recent years, there has been great interest in algorithms to solve the three-dimensional Poisson equation.

In modern numerical methods, there are many methods to solve Poisson equation, such as Ritz-Galerkin method [1], finite difference method [2], finite volume method and so on [3, 4]. Finite difference method is the earliest and perfect method to solve elliptic problems such as Poisson equation. However, most algorithms are developed for two-dimensional problems.

The numerical solution of the three-dimensional Poisson equation is very important for many applications in the computational physics and theoretical chemistry [5]. However, the great challenge has been encountered in the numerical approximation of three-dimensional problems, because it takes much memory and time to obtain sufficiently accuracy resolution. For two-dimensional Poisson equation, Dou [6] derived the seven-point difference scheme for three-dimensional Poisson equation, and solved the first-side value problem numerically. Later, we obtained a 19-point fourth order compact difference scheme for three-dimension Poisson equation. Zhai [8] eliminated the intermediate derivative terms by introducing the symmetric sum method from Taylor’s hand, and thus easily obtained the compacted difference schemes of the fourth and sixth order for the numerical solution of Poisson equation [7-10]. The algorithm in [11-13] for solving three-dimensional Poisson equation does not have obvious parallelism, and it is based on the idea of overall iteration, so it takes a long time when it is faced with a large amount of computation. Ge [14] modified and parallel optimized the algorithm for solving three-dimensional Poisson equation at the algorithm level, and transformed the entire solution problem into multiple independent problems for solving, which greatly improved the stability and parallel performance.
Fast Fourier transform is a powerful technique for solving three-dimensional Poisson equation. This paper presents a fast algorithm for solving three-dimensional Poisson equations. The large linear system is decomposed into small independent systems by fast Fourier transform, which greatly reduces computation time.

The rest of the paper is organized as follows. In Section 2, a fourth-order finite difference method for the Poisson equation is constructed. Sections 3 proposed a fast algorithm for solving the three-dimensional Poisson equation. Two numerical experiments verify the efficiency of the fast fourth-order algorithm in Section 4. The paper is concluded in Section 5.

The Fourth-Order Finite Difference Method

We consider a fourth-order finite difference method of Equation (1), where $\Omega$ is a continuous convex domain in three-dimensional space and $\partial \Omega$ is the boundary of the domain. The source function $f(x, y, z)$ is a given continuous function. And $f(x, y, z)$ is assumed to be sufficiently smooth and has the necessary continuous partial derivatives up to certain orders. The boundary condition $u_0(x, y, z)$ is suitable. In this paper, a cubic domain $\Omega = [0, a] \times [0, b] \times [0, c]$ is considered as the solution field. For the primal partition, we discrete $\Omega$ with uniform mesh sizes $h_x = \frac{a}{M+1}, h_y = \frac{b}{N+1}, h_z = \frac{c}{L+1}$ in the $x, y$ and $z$ coordinate directions respectively. Here $M, N$ and $L$ are the number of components in the $x, y$ and $z$ coordinate directions. The uniform partition is defined as $\{(i, j, l) : i = 0, 1, 2, \ldots, M+1; j = 0, 1, 2, \ldots, N+1; l = 0, 1, 2, \ldots, L+1 \}$ in $\Omega$. Without loss of generality, we consider the case of $h = h_x = h_y = h_z$ since it can be extended second order central difference operator can be written as

$$
\delta_x^2 u_{i,j,l} = \frac{u_{i+1,j,l} - 2u_{i,j,l} + u_{i-1,j,l}}{h^2}, \\
\delta_y^2 u_{i,j,l} = \frac{u_{i,j+1,l} - 2u_{i,j,l} + u_{i,j-1,l}}{h^2}, \\
\delta_z^2 u_{i,j,l} = \frac{u_{i,j,l+1} - 2u_{i,j,l} + u_{i,j,l-1}}{h^2},
$$

where $u_{i,j,l}, i = 1, 2, \ldots, M, j = 1, 2, \ldots, N, l = 1, 2, \ldots, L$ refers to the fourth-order finite difference solution of the three-dimensional Poisson equation.

The fourth-order finite difference from can be obtained in the interior of $\Omega$.

$$
(\delta_x^2 + \delta_y^2 + \delta_z^2) u_{i,j,l} - \frac{h^2}{6} (\delta_x^2 \delta_y^2 + \delta_y^2 \delta_z^2 + \delta_z^2 \delta_x^2) u_{i,j,l} = -f + \frac{h^2}{12} (\delta_x^2 + \delta_y^2 + \delta_z^2) f_{i,j,l} + O(h^4) \tag{3}
$$

where $\delta_x^2, \delta_y^2$ and $\delta_z^2$ are standard second order central difference operator and $u_{i,j,l}$ is the fourth-order finite difference solution of Equation (1). Figure 1 depicts the contributions to the 19-points stencil in a given axis, where $p = \frac{h^2}{6}$. 

631
Moreover, Equation (3) can be written in the matrix form

\[
\begin{align*}
(A_M \otimes I_N \otimes I_L + I_M \otimes A_N \otimes I_L + I_M \otimes I_N \otimes A_L)U + U_B & = \frac{h^2}{6} (A_M \otimes A_N \otimes I_L + I_M \otimes A_N \otimes A_L + A_M \otimes I_N \otimes A_L)U = -F \\
+ \frac{h^2}{12} (A_M \otimes I_N \otimes I_L + I_M \otimes A_M \otimes I_L + I_M \otimes I_N \otimes A_L)F + F_B.
\end{align*}
\] (4)

where

\[
\begin{align*}
A_M &= \frac{1}{h^2} \text{tridiag}(1,-2,1), \\
A_N &= \frac{1}{h^2} \text{tridiag}(1,-2,1), \\
A_L &= \frac{1}{h^2} \text{tridiag}(1,-2,1), \\
U &= (u_{1,1,1}, \ldots, u_{1,1,L}, u_{1,2,1}, \ldots, u_{1,2,L}, \ldots, u_{1,N,L}, \ldots, u_{M,N,L})^T, \\
F &= (f_{1,1,1}, \ldots, f_{1,1,L}, f_{1,2,1}, \ldots, f_{1,2,L}, \ldots, f_{1,N,L}, \ldots, f_{M,N,L})^T,
\end{align*}
\]

and the symbol \( \otimes \) represents the Kronecker product. \( I_M, I_N \) and \( I_L \) are identity matrices, and the subscripts denote their dimension. \( A_M, A_N \) and \( A_L \) are \( M \times M, N \times N \) and \( L \times L \) tridiagonal matrices respectively. \( U_B \) and \( F_B \) contain boundary values subtracted from \( U \) and \( F \).

The boundary part \( U_B \) consists 18 parts which are related to six surfaces and twelve edges of the domain, they are

\[
\begin{align*}
S_{B\text{top}}, S_{B\text{bottom}}, S_{B\text{left}}, S_{B\text{right}}, S_{B\text{front}}, S_{B\text{back}}, E_{B_{c1}}, E_{B_{c2}}, E_{B_{c3}}, E_{B_{c4}}, E_{B_{b1}}, E_{B_{b2}}, E_{B_{b3}}, E_{B_{b4}}.
\end{align*}
\]

Moreover, the boundary part \( F_B \) includes six parts, and they are

\[
\begin{align*}
F_{B\text{top}}, F_{B\text{bottom}}, F_{B\text{left}}, F_{B\text{right}}, F_{B\text{front}}, F_{B\text{back}}.
\end{align*}
\]

### The Fast Algorithm for three-dimensional Poisson Equations

To accelerate the algorithm, we apply the Fourier-sine transformation. For the tridiagonal Toeplitz matrix \( A_M \) and \( A_N \), we have

\[
\begin{align*}
S_M A_M S_M &= A_1 = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_M), \\
S_N A_N S_N &= A_2 = \text{diag}(\mu_1, \mu_2, \ldots, \mu_M),
\end{align*}
\]

where

\[
(S_M)_{i,j} = \sqrt{\frac{2}{M+1}} \sin \frac{i \pi j}{M+1},
\] (5)
\[ \lambda_i = -\frac{4(M+1)^2}{a^2} \sin^2 \frac{i\pi}{2(M+1)}, \quad 1 \leq i, j \leq M \]  

(6)

\( S_M \) and \( S_N \) are discrete Fourier-sin transformation matrices. \( S_N \) and \( \mu_t, t = 1, 2, \cdots, N \) can be defined similarly as Equation (5) and Equation (6). Multiplying \( S_M \otimes S_N \otimes I_L \) on both side of Equation (4), the following formula can be obtained

\[
\begin{aligned}
& (A_1 \otimes I_N \otimes I_L + I_M \otimes A_2 \otimes I_L + I_M \otimes I_N \otimes A_L) \bar{U} \\
& - \frac{h^2}{6} (A_1 \otimes A_2 \otimes I_L + I_M \otimes A_2 \otimes A_L + A_1 \otimes I_N \otimes A_L) \bar{U} + \bar{U}_B = -\bar{F} \\
& + \frac{h^2}{12} (A_1 \otimes I_N \otimes I_L + I_M \otimes A_2 \otimes I_L + I_M \otimes I_N \otimes A_L) \bar{F} + \bar{F}_B,
\end{aligned}
\]

(7)

where

\[
\begin{align*}
\bar{U} &= (S_M \otimes S_N \otimes I_L) U, \\
\bar{F} &= (S_M \otimes S_N \otimes I_L) F, \\
\bar{U}_B &= (S_M \otimes S_N \otimes I_L) U, \\
\bar{F}_B &= (S_M \otimes S_N \otimes I_L) F.
\end{align*}
\]

Multiplying \( S_M \otimes S_N \otimes I_L \) on each part of \( \bar{U}_B \), we can obtain \( \bar{S}_{B_{top}}, \bar{S}_{B_{bottom}}, \bar{S}_{B_{left}}, \bar{S}_{B_{right}}, \bar{S}_{B_{front}}, \bar{S}_{B_{back}}, \bar{E}_{B_{t1}}, \bar{E}_{B_{t2}}, \bar{E}_{B_{t3}}, \bar{E}_{B_{t4}}, \bar{E}_{B_{c1}}, \bar{E}_{B_{c2}}, \bar{E}_{B_{c3}}, \bar{E}_{B_{c4}}, \bar{E}_{B_{b1}}, \bar{E}_{B_{b2}}, \bar{E}_{B_{b3}}, \bar{E}_{B_{b4}} \). 

Similarly, multiplying \( S_M \otimes S_N \otimes I_L \) each part of \( \bar{F}_B \), we can obtain \( \bar{F}_{B_{top}}, \bar{F}_{B_{bottom}}, \bar{F}_{B_{left}}, \bar{F}_{B_{right}}, \bar{F}_{B_{front}}, \bar{F}_{B_{back}} \).

Equation (6) is transformed into block-tridiagonal system. Therefore, we can transform the original problem into the following equations

\[
\begin{aligned}
& (\lambda_i I_L + \mu_j I_L + A_L) - \frac{h^2}{6} (\lambda_i \mu_j I_L + \mu_j A_L + \lambda_i A_L) \bar{U}_{i,j,:} \\
& + \bar{U}_{B_{i,j,:}} = -\bar{F}_{i,j,:} + \frac{h^2}{12} (\lambda_i I_L + \mu_j I_L + A_L) \bar{F}_{i,j,:} + \bar{F}_{B_{i,j,:}}
\end{aligned}
\]

(8)

where \( i = 1, 2, \cdots, M, j = 1, 2, \cdots, N \).

**Numerical Experiments**

In order to verify the accuracy and reliability of the method presented in this paper, we investigate the following two problems with exact solutions in the three-dimensional unit space \( \Omega = [0,1] \times [0,1] \times [0,1] \). Moreover, both experiments are implemented on MATLAB.

**Example 1.** Consider the following problem

\[ \Delta u(x, y, z) = -f(x, y, z), \]

with \( f(x, y, z) = 0 \) and the boundary conditions

\[
\begin{aligned}
& u(x, y, z) = \sin(\pi y) \sin(\pi z), \quad x = 0, \\
& u(x, y, z) = 2 \sin(\pi y) \sin(\pi z), \quad x = 1, \\
& u(x, y, z) = 0, \quad y, z = \{0,1\}.
\end{aligned}
\]

(9)

Here the exact solution is

\[ u(x, y, z) = \frac{\sin(\pi y) \sin(\pi z)}{\sinh(\pi \sqrt{2})} 2 \sinh(\pi \sqrt{2}x) + \sinh(\pi \sqrt{2}(1 - x)). \]

In order to express the result more intuitively, we line up the values of \( x \) direction and \( y \) direction as the rows of the matrix, and the values of \( z \) direction as the columns of the matrix, and express the result in three-dimensional space. The results are shown in Figure 2 and Figure 3.
Example 2. Consider the following problem

$$\Delta u(x, y, z) = -f(x, y, z),$$

$$f(x, y, z) = 3\pi^2 \sin(\pi x) \sin(\pi y) \sin(\pi z).$$

where the boundary conditions are

$$u(0, y, z) = u(1, y, z) = 0,$$

$$u(x, 0, z) = u(x, 1, z) = 0,$$

$$u(x, y, 0) = u(x, y, 1) = 0.$$

Here the exact solution is

$$u(x, y, z) = \sin(\pi x) \sin(\pi y) \sin(\pi z).$$

We give the figures of the numerical solutions $u(x, y, z)$. The numerical solution on the face $z = \frac{1}{2}$ with 32×32×32 meshes are presented in Figure 4. Color depicts the value of numerical solution. Figure 5 and Figure 6 illustrates the numerical solution of Equation (10) with 32×32×32 meshes and 256×256×256 meshes, respectively.
The results are displayed in Table 1. We test the convergent order of the proposed fast high algorithm. The last column of Table 1 demonstrates the fourth-order convergence of the proposed algorithm.
Table 1. CPU time (s), error and convergence order for solving example 2 with different operators.

<table>
<thead>
<tr>
<th>M</th>
<th>CPU_{10} time (s)</th>
<th>CPU_{10} time (s)</th>
<th>used memory (MB)</th>
<th>error</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0.0337</td>
<td>0.0087</td>
<td>0.0194</td>
<td>6.1655e-05</td>
<td>4.1488</td>
</tr>
<tr>
<td>16</td>
<td>0.0933</td>
<td>0.0860</td>
<td>0.0998</td>
<td>4.4059e-06</td>
<td>4.0714</td>
</tr>
<tr>
<td>32</td>
<td>0.3522</td>
<td>1.2344</td>
<td>0.6326</td>
<td>2.9595e-07</td>
<td>4.0348</td>
</tr>
<tr>
<td>64</td>
<td>1.5402</td>
<td>20.5315</td>
<td>4.4991</td>
<td>1.9203e-08</td>
<td>4.0172</td>
</tr>
<tr>
<td>128</td>
<td>6.3100</td>
<td>347.8190</td>
<td>33.9351</td>
<td>1.2234e-09</td>
<td>4.0132</td>
</tr>
<tr>
<td>256</td>
<td>29.8813</td>
<td>7186.8724</td>
<td>263.6197</td>
<td>7.6952e-11</td>
<td>4.0132</td>
</tr>
</tbody>
</table>

Conclusion

In this paper, we proposed a fourth-order fast algorithm for solving the three-dimensional Poisson equation. By fast Fourier transform, the large discrete linear system is decomposed into small independent systems, which greatly save computational memory and time. Several numerical experiments show the feasibility and efficiency of the method.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (No. 11401208), the Natural Science Foundation of Hebei Province (No. A2016502001) and the Fundamental Research Funds for the Central Universities (No. 2018MS129).

References


