

A Fast High Order Algorithm for 3D Helmholtz Equation with Dirichlet Boundary

Sheng An^{*}, Gendai Gu, Meiling Zhao

School of Mathematics and Physics, North China Electric Power University, Baoding, China

Email address:

anshengmath@163.com (Sheng An), ggd999@126.com (Gendai Gu), meilingzhao@sina.com (Meiling Zhao)

^{*}Corresponding author

To cite this article:

Sheng An, Gendai Gu, Meiling Zhao. A Fast High Order Algorithm for 3D Helmholtz Equation with Dirichlet Boundary. *Applied and Computational Mathematics*. Vol. 7, No. 4, 2018, pp. 180-187. doi: 10.11648/j.acm.20180704.11

Received: July 26, 2018; **Accepted:** August 13, 2018; **Published:** September 11, 2018

Abstract: Helmholtz equation is widely applied in the scientific and engineering problem. For the solution of the three-dimensional Helmholtz equation, the computational efficiency of the algorithm is especially important. In this paper, in order to solve the contradiction between accuracy and efficiency, a fast high order finite difference method is proposed for solving the three-dimensional Helmholtz equation. First, a traditional fourth order method is constructed. Then, fast Fourier transformation are introduced to generate a block-tridiagonal structure which can easily divide the original problem into small and independent subsystems. For large 3D problems, the computation of traditional discrete Fourier transformation is less efficient, and the memory requirements increase rapidly. To fix this problem, the transformed coefficient matrix is constructed as a sparse structure. In light of the sparsity, the algorithm presented in this paper requires less memory space and computational cost. This sparse structure also leads to independent solving procedure of any plane in the domain. Therefore, parallel method can be used to solve the Helmholtz equation with large grid number. In the end, three numerical experiments are presented to verify the effectiveness of the fast fourth-order algorithm, and the acceleration effect to use the parallel method has been demonstrated.

Keywords: Helmholtz Equation, Fourier Transformation, Parallel Implementation

1. Introduction

In this paper, we consider the following Helmholtz equation

$$\Delta u(x, y, z) + k^2 u(x, y, z) = f(x, y, z), \text{ in } \Omega, \# \quad (1)$$

with Dirichlet boundary condition

$$u(x, y, z) = u_0(x, y, z), \text{ on } \partial\Omega, \# \quad (2)$$

where k is the wave number. Helmholtz equation is widely applied to wave propagation and acoustic scattering problem, viscous and heat transfer problems and electromagnetic scattering problems.

In recent years, there has been tremendous interest in developing algorithms for solving the Helmholtz equation. The precise numerical solution of Helmholtz equation is of great importance. Most algorithms were initially developed for the two-dimensional case, such as finite element method,

finite difference method and other iterative methods [1-4]. However, the accuracy of the finite element method reduces rapidly with the increase of the wave number. In addition, finite difference method is not competitive for solving the Helmholtz equation with large wave number if no fine mesh is provided. Therefore, some high order finite difference methods were proposed for solving the two-dimensional Helmholtz equation in [5-10]. This method is prevalent since they can offer high accuracy solutions with less computational cost. There have been applications for solving the three-dimensional Helmholtz equations. Braverman et al. proposed a fast spectral Helmholtz solver which incorporates the application of the FFT with a preliminary subtraction technique in [11]. Lu presented a fourth-order compact difference scheme based on the Laplace operator in [12]. A noniterative solver based on the domain decomposition was developed in [13] for solving the three-dimensional

Helmholtz equation. Sutmann combined different high order finite difference schemes and proposed a more efficient algorithm in [14, 15]. Furthermore, Gordon et al. extend the sixth-order finite difference method to high and variable wave number k with CARP-CG algorithm in [16, 17]. Due to the heavy computational cost the efficiency of the algorithm for solving three-dimensional Helmholtz equation is especially important. The Fourier method has many advantages for solving the Helmholtz equation. Together with the symmetry of the high-order finite difference operator, the Dirichlet problems can be solved easily with sine transformation.

In this paper, our aim is to discretize the Helmholtz equation with fourth-order finite difference method and build a fast algorithm using the discrete Fourier-sine transformation. On the other hand, the traditional Fourier transformation in three dimensions needs considerable memory requirement especially for large wave number equations. In views of this problem, twice Fourier transformation are applied for the discrete Helmholtz equation.

The paper is organized as follows. In Section 2, a fourth-order finite difference method was proposed to discretize the three-dimensional Helmholtz equation. In Sections 3 and 4, a fast high algorithm is developed for solving the three-dimensional Helmholtz equation. Section 5 describes the parallel implementation for the three-dimensional Helmholtz equation. Moreover, three numerical examples prove the validity and feasibility of the proposed method. The paper is concluded in Section 6.

2. The Fourth-Order Finite Difference Method

We consider a fourth-order finite difference method of (1), where Ω is the model domain and $\partial\Omega$ is the boundary of the domain. $f(x, y, z)$ is an given continuous function. In this paper, a cubic domain $\Omega = [0, a] \times [0, b] \times [0, c]$ is considered as the solution field. Firstly, the uniform partition is define as $\{x_i, y_j, z_l\}_{i,j,l=0}^{M+1, N+1, L+1}$ in Ω . Without loss of generality, we consider the case of $h = \Delta x = \Delta y = \Delta z$ since it can be extended to the general situation. For any point (x_i, y_j, z_l) in Ω the standard second order central difference operator can be written as

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

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

The fourth-order finite difference form can be obtained in the interior of Ω .

$$\begin{aligned}& \left(1 + \frac{k^2 h^2}{12}\right) (\delta_x^2 + \delta_y^2 + \delta_z^2) u_{i,j,l} + \frac{h^2}{6} (\delta_x^2 \delta_y^2 + \delta_x^2 \delta_z^2 + \delta_y^2 \delta_z^2) u_{i,j,l} + k^2 u_{i,j,l} \\ &= f + \frac{h^2}{12} (\delta_x^2 \delta_y^2 + \delta_x^2 \delta_z^2 + \delta_y^2 \delta_z^2) f_{i,j,l} + O(h^4).\end{aligned}\quad (3)$$

Figure 1 depicts the contributions to the 19-points stencil in an given axes, where $p_1 = 1 + \frac{k^2 h^2}{12}$, $p_2 = \frac{h^2}{6}$.

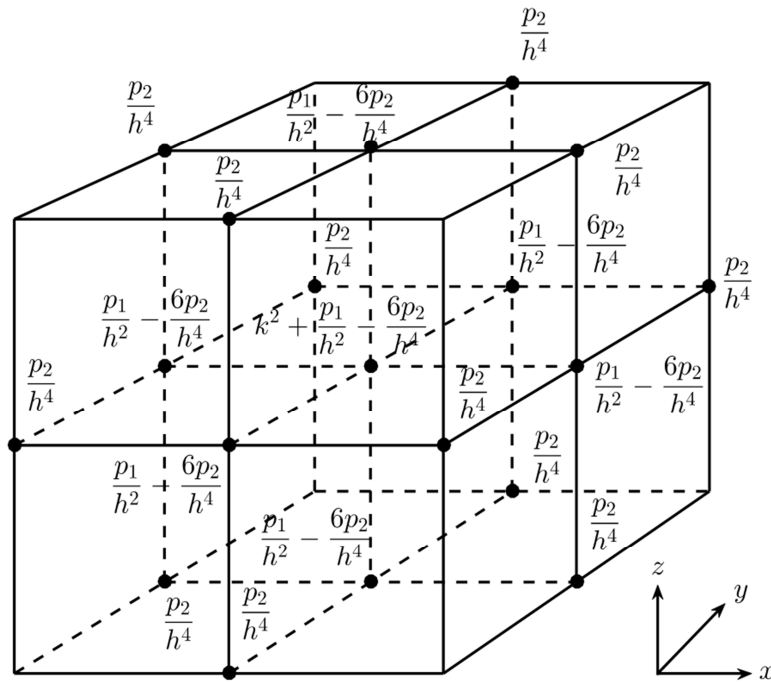


Figure 1. The 19 points stencil of fourth-order finite difference method for 3D Helmholtz equation.

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

$$\begin{aligned} & \left(1 + \frac{k^2 h^2}{12}\right) (A_M \otimes I_N \otimes I_L + I_M \otimes A_N \otimes I_L + I_M \otimes I_N \otimes A_L) U \\ & + \frac{h^2}{6} (A_M \otimes A_N \otimes I_L + I_M \otimes A_N \otimes A_L + A_M \otimes I_N \otimes A_L) U + k^2 U + U_B \# \\ & = F + \frac{h^2}{12} (A_M \otimes I_N \otimes I_L + I_M \otimes A_N \otimes I_L + I_M \otimes I_N \otimes A_L) F + F_B, \end{aligned} \quad (4)$$

where

$$A_M = \frac{1}{h^2} \text{tridiag}(1, -2, 1), \quad A_N = \frac{1}{h^2} \text{tridiag}(1, -2, 1), \quad A_L = \frac{1}{h^2} \text{tridiag}(1, -2, 1),$$

$$U = (u_{1,1,1}, \dots, u_{1,1,L}, u_{1,2,1}, \dots, u_{1,2,L}, \dots, u_{1,N,L}, \dots, u_{M,N,L})^T, \\ F = (f_{1,1,1}, \dots, f_{1,1,L}, f_{1,2,1}, \dots, f_{1,2,L}, \dots, f_{1,N,L}, \dots, f_{M,N,L})^T,$$

and the symbol \otimes represents the Kronecker product. I_M, I_N, I_L and I_{MNL} 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 .

3. The Explanation of the Boundary Parts

The boundary part U_B consists 18 parts which are related to six surfaces and twelve edges of the domain, they are $S_{B_{top}}, S_{B_{bottom}}, S_{B_{left}}, S_{B_{right}}, S_{B_{front}}, S_{B_{back}}, E_{B_{t1}}, E_{B_{t2}}, E_{B_{t3}}, E_{B_{t4}}, E_{B_{c1}}, E_{B_{c2}}, E_{B_{c3}}, E_{B_{c4}}, E_{B_{b1}}, E_{B_{b2}}, E_{B_{b3}}, E_{B_{b4}}$. In this section, only the detailed discussion of $S_{B_{left}}$ and $E_{B_{t1}}$ will be given, the other items can be deduced in similar way.

For a point (x_1, y_j, z_l) in the domain, points on the left surface of the domain $(x_0, y_j, z_l), (x_0, y_j, z_{l+1}), (x_0, y_j, z_{l-1}), (x_0, y_{j-1}, z_l), (x_0, y_{j+1}, z_l)$ will generate the boundary part $S_{B_{left}}$ as shown in Figure 2.

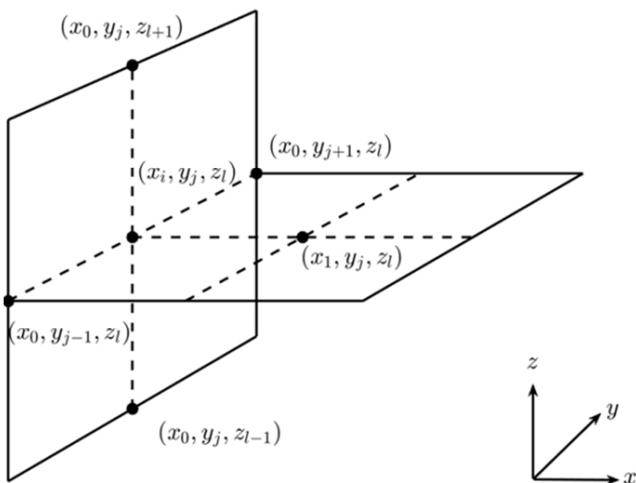


Figure 2. The corresponding points of $S_{B_{left}}$.

$S_{B_{left}}$ can be written as follows

$$S_{B_{left}} = (p_2 a_{M1} \otimes A_N \otimes I_L + p_2 a_{M1} \otimes I_N \otimes A_L) U_{0,:} \# + (p_1 a_{M1} \otimes I_N \otimes I_L) U_{0,:} \quad (5)$$

where $a_{M1} = \frac{1}{h^2} (1, 0, \dots, 0)_M^T$.

Analogously, $E_{B_{t1}}$ is related to three points $(x_{i+1}, y_j, z_l), (x_{i-1}, y_j, z_l), (x_i, y_j, z_{l+1})$. Therefore, we have

$$E_{B_{t1}} = (p_2 I_M \otimes a_{N1} \otimes a_{L2}) U_{:,0,L+1} \# \quad (6)$$

where $a_{N1} = \frac{1}{h^2} (1, 0, \dots, 0)_N^T, a_{L2} = \frac{1}{h^2} (0, 0, \dots, 1)_L^T$.

Moreover, the boundary part BF includes six parts. F_B can be written as follows

$$F_B = F_{B_{top}} + F_{B_{bottom}} + F_{B_{left}} + F_{B_{right}} + F_{B_{front}} + F_{B_{back}} \# \quad (7)$$

where

$$F_{B_{top}} = p_1 (I_M \otimes I_N \otimes a_{L2}) F_{:,L+1},$$

$$F_{B_{bottom}} = p_1 (I_M \otimes I_N \otimes a_{L1}) F_{:,0},$$

$$F_{B_{left}} = p_1 (a_{M1} \otimes I_N \otimes I_L) F_{0,:},$$

$$F_{B_{right}} = p_1 (a_{M2} \otimes I_N \otimes I_L) F_{M+1,:},$$

$$F_{B_{front}} = p_1 (I_M \otimes a_{N1} \otimes I_L) F_{:,0},$$

$$F_{B_{back}} = p_1 (I_M \otimes a_{N2} \otimes I_L) F_{:,0},$$

$$a_{L1} = \frac{1}{h^2} (1, 0, \dots, 0)_L^T, \quad a_{M2} = \frac{1}{h^2} (0, 0, \dots, 1)_M^T,$$

$$a_{N2} = \frac{1}{h^2} (0, 0, \dots, 1)_N^T.$$

4. The Fast Algorithm for 3D Helmholtz Equation

For the tridiagonal Toeplitz matrix A_M and A_N , we have

$$S_M A_M S_M = \Lambda_1 = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_M),$$

$$S_N A_N S_N = \Lambda_2 = \text{diag}(\mu_1, \mu_2, \dots, \mu_N),$$

where

$$(S_M)_{i,j} = \sqrt{\frac{2}{M+1}} \left(\sin \frac{ij\pi}{M+1} \right), \quad \# \quad (8)$$

$$\lambda_i = -\frac{4(M+1)^2}{a} \sin^2 \frac{i\pi}{2(M+1)}, \quad 1 \leq i, j \leq M, \#$$

S_M and S_N are discrete Fourier-sine transformation matrices. S_N and $\mu_t, t = 1, 2, \dots, N$ can be defined similarly as (8). Multiplying $S_M \otimes S_N \otimes I_L$ on both side of (4), the following formula can be obtained

$$\begin{aligned} & \left(1 + \frac{k^2 h^2}{12}\right) (\Lambda_1 \otimes I_N \otimes I_L + I_M \otimes \Lambda_2 \otimes I_L + I_M \otimes I_N \otimes A_L) \bar{U} \\ & + \frac{h^2}{6} (\Lambda_1 \otimes \Lambda_2 \otimes I_L + I_M \otimes \Lambda_2 \otimes A_L + \Lambda_1 \otimes I_N \otimes A_L) \bar{U} + k^2 \bar{U} + \bar{U}_B \# \\ & = \bar{F} + \frac{h^2}{12} (\Lambda_1 \otimes I_N \otimes I_L + I_M \otimes \Lambda_2 \otimes I_L + I_M \otimes I_N \otimes A_L) \bar{F} + \bar{F}_B, \end{aligned} \quad (9)$$

where $\bar{U} = (S_M \otimes S_N \otimes I_L)U, \bar{F} = (S_M \otimes S_N \otimes I_L)F, \bar{U}_B = (S_M \otimes S_N \otimes I_L)U_B, \bar{F}_B = (S_M \otimes S_N \otimes I_L)F_B$.

Multiplying $S_M \otimes S_N \otimes I_L$ on each part of \bar{U}_B , there follows

$$\begin{aligned} \bar{S}_{B_{top}} &= [p_2(\Lambda_1 \otimes I_N \otimes a_{L2}) + p_2(I_M \otimes \Lambda_2 \otimes a_{L2}) + p_1(I_M \otimes I_N \otimes a_{L2})] \bar{U}_{:,L+1}, \\ \bar{S}_{B_{bottom}} &= [p_2(\Lambda_1 \otimes I_N \otimes a_{L1}) + p_2(I_M \otimes \Lambda_2 \otimes a_{L1}) + p_1(I_M \otimes I_N \otimes a_{L1})] \bar{U}_{:,0}, \\ \bar{S}_{B_{front}} &= [p_2(\Lambda_1 \otimes S_N \otimes I_L) + p_2(I_M \otimes S_N \otimes A_L) + p_1(I_M \otimes S_N \otimes I_L)] \bar{U}_{:,0,:}, \\ \bar{S}_{B_{back}} &= [p_2(\Lambda_1 \otimes S_N \otimes I_L) + p_2(I_M \otimes S_N \otimes A_L) + p_1(I_M \otimes S_N \otimes I_L)] \bar{U}_{:,N+1,:}, \\ \bar{S}_{B_{left}} &= p_2(S_M \otimes \Lambda_2 \otimes I_L) \bar{U}_{0,:} + p_2(S_M \otimes I_N \otimes I_L) \bar{U}_{0,:} + p_1(S_M \otimes I_N \otimes I_L) \bar{U}_{0,:}, \\ \bar{S}_{B_{right}} &= p_2(S_M \otimes \Lambda_2 \otimes I_L) \bar{U}_{M,:} + p_2(S_M \otimes I_N \otimes I_L) \bar{U}_{M,:} + p_1(S_M \otimes I_N \otimes I_L) \bar{U}_{M,:}, \\ \bar{E}_{B_{l1}} &= p_2(S_M \otimes S_N \otimes I_L) \bar{U}_{0,0,:}, \bar{E}_{B_{l2}} = p_2(S_M \otimes S_N \otimes I_L) \bar{U}_{M+1,0,:}, \\ \bar{E}_{B_{l3}} &= p_2(S_M \otimes S_N \otimes I_L) \bar{U}_{M+1,N+1,:}, \bar{E}_{B_{l4}} = p_2(S_M \otimes S_N \otimes I_L) \bar{U}_{0,N+1,:}, \\ \bar{E}_{B_{t1}} &= p_2(I_M \otimes S_N \otimes a_{L2}) \bar{U}_{:,0,L+1}, \bar{E}_{B_{t2}} = p_2(I_M \otimes S_N \otimes a_{L1}) \bar{U}_{M+1,0,L+1}, \\ \bar{E}_{B_{t3}} &= p_2(S_M \otimes I_N \otimes a_{L2}) \bar{U}_{:,N+1,L+1}, \bar{E}_{B_{t3}} = p_2(S_M \otimes I_N \otimes a_{L2}) \bar{U}_{:,0,L+1}, \\ \bar{E}_{B_{c1}} &= p_2(S_M \otimes I_N \otimes a_{L1}) \bar{U}_{:,0,0}, \bar{E}_{B_{c2}} = p_2(S_M \otimes I_N \otimes a_{L1}) \bar{U}_{M+1,0}, \\ \bar{E}_{B_{c3}} &= p_2(I_M \otimes S_N \otimes a_{L1}) \bar{U}_{:,0N+1,0}, \bar{E}_{B_{c4}} = p_2(I_M \otimes S_N \otimes a_{L1}) \bar{U}_{0,0}, \end{aligned}$$

where

$$\begin{aligned} \bar{U}_{0,:} &= (a_{M1} \otimes S_N \otimes I_L)U_{0,:}, \bar{U}_{0,:} = (a_{M1} \otimes S_N \otimes A_L)U_{0,:}, \\ \bar{U}_{M,:} &= (a_{M2} \otimes S_N \otimes I_L)U_{M,:}, \bar{U}_{M,:} = (a_{M2} \otimes S_N \otimes A_L)U_{M,:}. \end{aligned}$$

Multiplying $(S_M \otimes S_N \otimes I_N)$ on Equation (6), there holds

$$\begin{aligned} \bar{F}_{B_{top}} &= p_1(I_M \otimes I_N \otimes a_{L2}) \bar{F}_{:,L+1}, \bar{F}_{B_{bottom}} = p_1(I_M \otimes I_N \otimes a_{L1}) \bar{F}_{:,0}, \\ \bar{F}_{B_{left}} &= p_1(a_{M1} \otimes I_N \otimes I_L) \bar{F}_{0,:}, \bar{F}_{B_{right}} = p_1(a_{M2} \otimes I_N \otimes I_L) \bar{F}_{M+1,:}, \\ \bar{F}_{B_{front}} &= p_1(I_M \otimes a_{N1} \otimes I_L) \bar{F}_{:,0}, \bar{F}_{B_{back}} = p_1(I_M \otimes a_{N2} \otimes I_L) \bar{F}_{:,0}, \end{aligned}$$

Equation (9) is transformed into block-tridiagonal system, therefore, we can transform the original problem into the following equations

$$\begin{aligned} & \left[\left(1 + \frac{k^2 h^2}{12}\right) (\lambda_i I_L + \mu_j I_L + A_L) + \frac{h^2}{6} (\lambda_i \mu_j I_L + \mu_j A_L + \lambda_i A_L) + k^2 I_L \right] \bar{U}_{i,j,:} + \bar{U}_{B_{i,j,:}} \# \\ & = \bar{F}_{i,j,:} + \bar{F}_{B_{i,j,:}}, i = 1, 2, \dots, M; j = 1, 2, \dots, N. \end{aligned} \quad (10)$$

5. Numerical Experiments

5.1. Preparation for the Numerical Experiments

In this section, the algorithm is applied for three numerical experiments. And the performance of the algorithm is described using MATLAB on an eight processor computer.

One error measurement is used to weigh the difference between $u_{i,j,k}^*$ and $u_{i,j,k}$, where $u_{i,j,k}^*$ denotes the real solution of the three-dimensional Helmholtz equation.

Define the error measurement as follows

$$error = \left(\frac{H^3}{MNL} \sum_{i,j,l=1}^{M,N,L} |u_{i,j,k}^* - u_{i,j,k}|^2 \right)^{\frac{1}{2}}$$

For different kinds of meshes, the convergence order can be written as

$$order = \log_2 \frac{error(n)}{error(n+1)}.$$

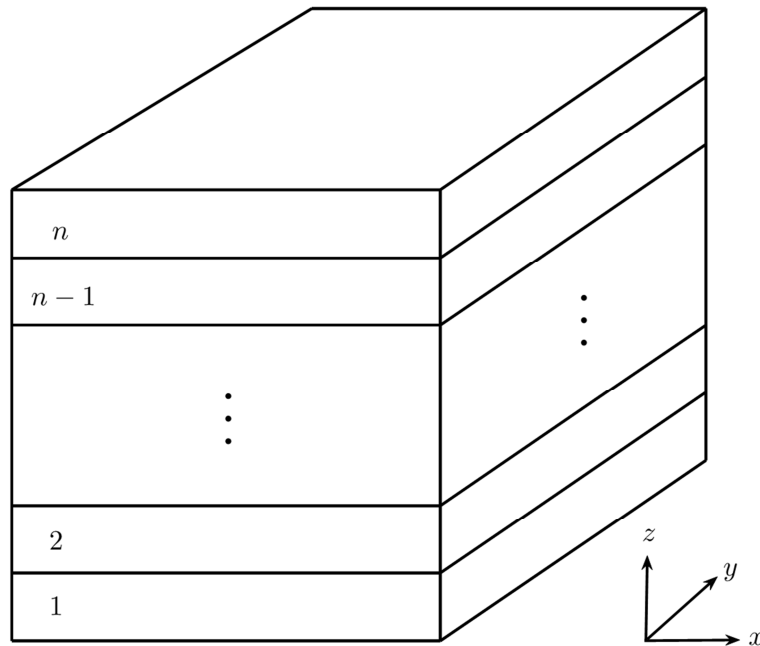


Figure 3. Partition of the three-dimensional mesh among n processors.

Example 1. Consider the problem

$$u(x, y, z) = \frac{\sin(\pi y) \sin(\pi z)}{\sin(\sqrt{2}\pi)} \cdot \left[2 \sinh(\sqrt{2}\pi x) + \sinh(\sqrt{2}\pi(1-x)) \right], \quad \# (11)$$

with Dirichlet boundary conditions, where $\Omega = [0,1] \times [0,1] \times [0,1]$ is a unit cubic.

$$\begin{cases} u(x, y, z) = \sin(\pi y) \sin(\pi z), & x = 0, \\ u(x, y, z) = 2 \sin(\pi y) \sin(\pi z), & x = 1, \\ 0, & y, z \in \{0,1\}, \end{cases}$$

5.2. Parallel Implementation

After deriving \bar{U} by (10), the solution U of the three-dimensional Helmholtz equation is calculated by $U = (S_M \otimes S_N \otimes I_L) \bar{U}$. Due to the special form of the coefficient matrix, the computational procedure is independent for each $z_l, l = 0, 1, \dots, L$. Therefore, the fast algorithm proposed in Section 3 is well adapted for the parallel computation. The implementation under n processors is shown in Figure 3.

The numerical results for solving \bar{U} and U are displayed in Table 1. Here $CPU_{\bar{U}}$ and CPU_U denotes the computational time (s) for solving \bar{U} and U respectively. After deriving \bar{U} , we need to multiply $S_M \otimes S_N \otimes I_L$ on the left side of \bar{U} to compute the numerical solution of the Helmholtz equation. As we can see from Table 1, $CPU_{\bar{U}}$ time is notably less than CPU_U time especially for large grid points. Moreover, the comparison of the computational time of three times Fourier transformation and twice Fourier transformation are given in Table 1. Here $S_M \otimes S_N \otimes S_L$ and $S_M \otimes S_N \otimes I_L$ represent two different transform operators.

Table 1. CPU time (s) and error for solving \bar{U} and U .

M	$S_M \otimes S_N \otimes S_L$		$S_M \otimes S_N \otimes I_L$	
	$CPU_{\bar{U}}$ time (s)	CPU_U time (s)	$CPU_{\bar{U}}$ time (s)	CPU_U time (s)
8	0.0381	0.0449	0.0499	0.2240
16	0.0634	0.8158	0.1947	0.2717
32	0.4535	28.6402	0.7144	0.4436
64	1.5009	1052.0414	3.2476	3.1560
128	9.0472	46725.2783	14.0811	60.283293

If the practical problem only needs some of the solutions of the Helmholtz equation, the algorithm can provide more precious numerical solutions on one of the surfaces. The

numerical solution on the face $z = \frac{1}{2}$ with $1024 \times 1024 \times 1024$ meshes are presented in Figure 4. Color depicts the value of numerical solution.

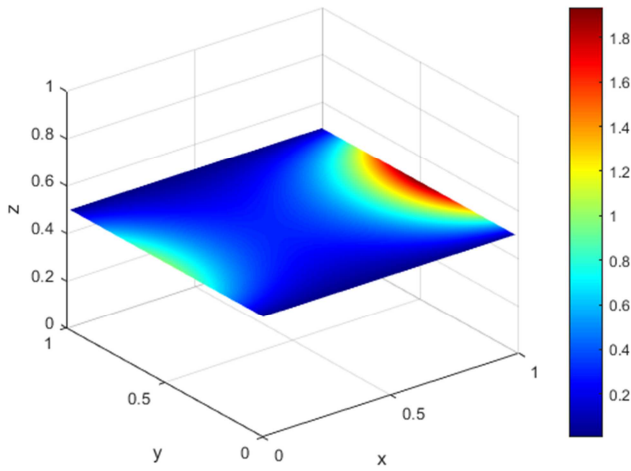


Figure 4. The numerical solution on the face $z = \frac{1}{2}$ with $M = 1024$.

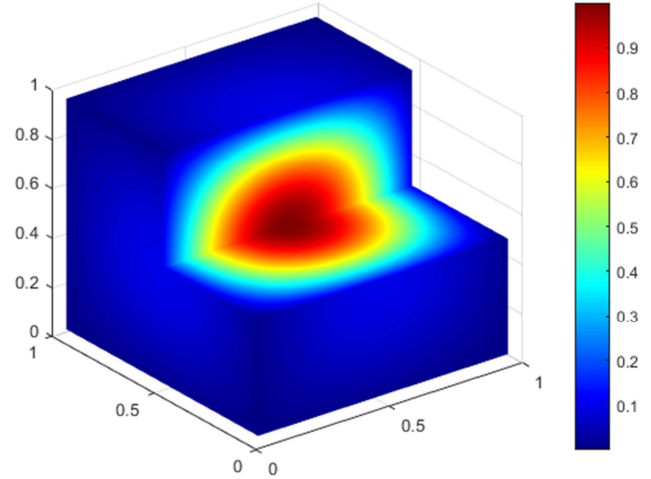


Figure 5. The numerical solution of Equation (12) with $M = 512$.

Example 2. Consider the problem

$$\Delta u + k^2 u = f, k = 5, \text{ in } \Omega,$$

with boundary conditions corresponding to the exact solution

$$u = \sin(\pi x) \sin(\pi y) \sin(\pi z). \quad (12)$$

Figure 5 illustrates the numerical solution of (12) with $512 \times 512 \times 512$ meshes.

The behavior of the parallel implementation was examined in Example 2. The results are displayed in Table 2. The results again verify the efficiency of the parallel processor. Moreover, we test the convergence order of the proposed fast high algorithm. The last column of Table 2 demonstrates the fourth-order convergence of the proposed algorithm. The line in Figure 6 is fitted by $\log(\text{error})$ and $\log(M)$ and its slope is 4.0545.

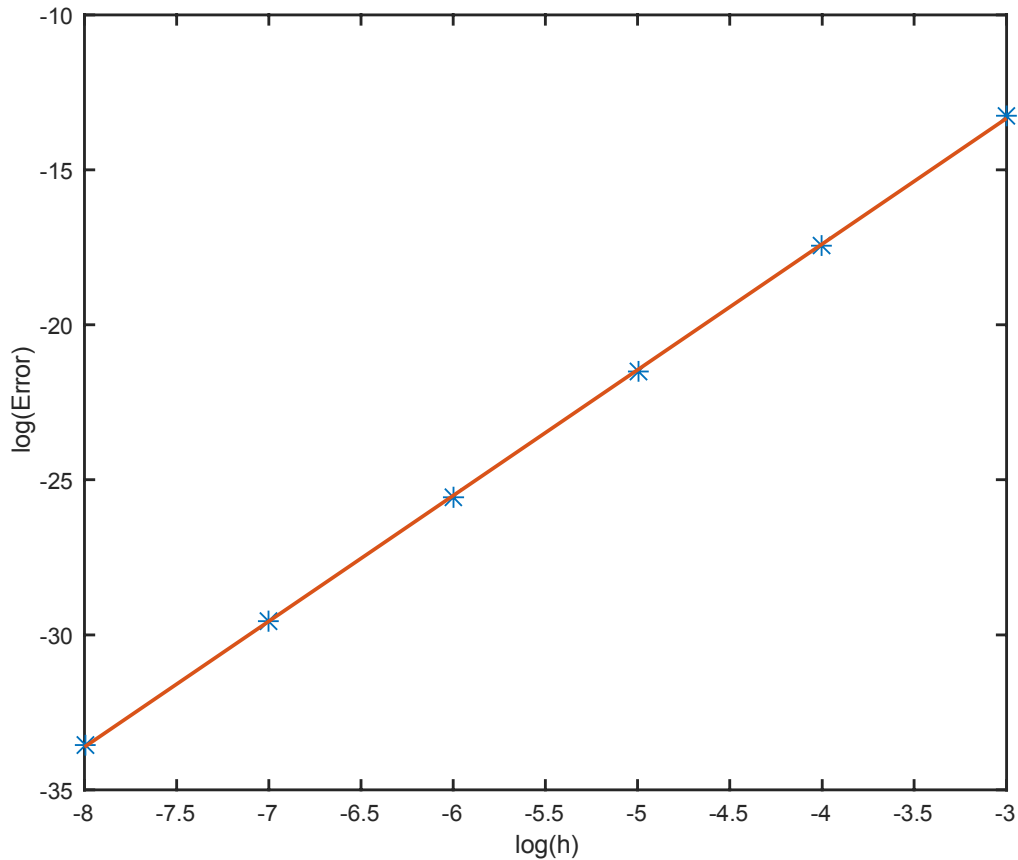


Figure 6. log-log picture of $\log(\text{error})$ and $\log(M)$ and the slope of the fitted line is 4.0545.

Table 2. CPU time (s) and error for solving \bar{U} and U .

M	CPU \bar{U} time (s)	CPU U time (s)		error	order
		1 core	8 cores		
8	0.0313	0.0162	0.2392	1.0145e-04	
16	0.1008	0.1902	0.3019	5.6505e-06	4.1662
32	0.6519	3.2303	0.4358	3.3522e-07	4.0752
64	2.8458	56.5750	3.1811	2.0439e-08	4.0357
128	12.4757	1020.6792	50.2228	1.2621e-09	4.0174
256	56.7871	12472.1685	909.4857	7.8732e-11	4.0027

Example 3. Consider the problem

$$\Delta u + k^2 u = 0, \text{ in } \Omega,$$

with boundary conditions corresponding to the exact solution

$$u(x, y, z) = \cos(2x) \cos(\sqrt{1+k^2}y) \frac{\sinh \sqrt{5}z}{\sinh \sqrt{5}\pi}. \quad (13)$$

The figures of the numerical solutions U with different wave number in Figure 7 and Figure 8. As shown in Figures 7-8, the solutions of the Helmholtz equation is highly oscillating for large wave number.

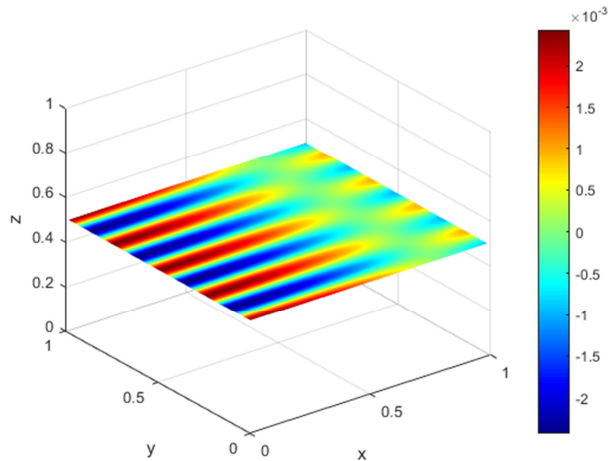


Figure 7. Numerical solutions on the face $z = \frac{1}{2}$ with $M = 512$ when $k = 8\pi$.

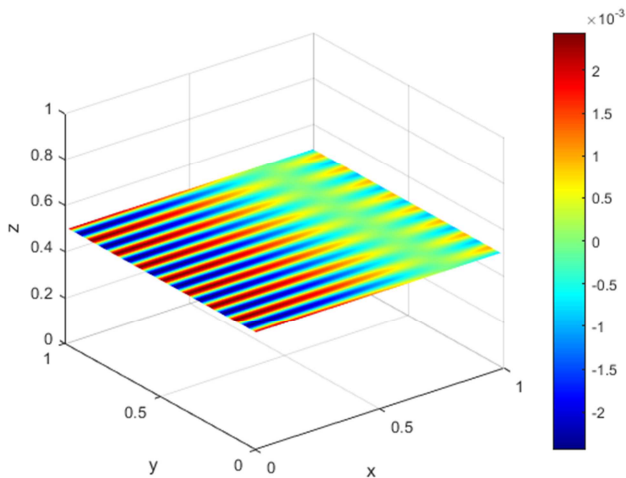


Figure 8. Numerical solutions on the face $z = \frac{1}{2}$ with $M = 1024$ when $k = 16\pi$.

6. Conclusion

In this paper, we proposed an fourth-order fast algorithm for solving the three-dimensional Helmholtz equation. By multiplying the Fourier transform operator, the large system becomes some small independent systems. Moreover, in view of the special format of the coefficient matrix, we utilize the parallel processors to solve the equation. This implementation is especially efficient for solving the numerical solutions on one of the surfaces. Three numerical experiments have demonstrated the validity of the fourth-order fast algorithm.

Acknowledgements

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

References

- [1] Jin J M, Liu J, Z. Lou, et al. A fully high-order finite-element simulation of scattering by deep cavities [J]. Antennas & Propagation IEEE Transactions on, 2003 51 (9): 2420-2429.
- [2] Li P J, and Wood A. A two-dimensional Helmholtz equation solution for the multiple cavity scattering problem [J]. Journal of Computational Physics, 2013 240 (1): 100-120.
- [3] Liu J, Jin J M. A special higher order finite-element method for scattering by deep cavities [J]. IEEE Transactions on Antennas & Propagation, 2000 48 (5): 694-703.
- [4] Wang Y X, Du K, Sun W W. A Second-Order Method for the Electromagnetic Scattering from a Large Cavity [J]. Numerical Mathematics: Theory, Methods and Applications, 2008 1 (4): 357-382.
- [5] Fu Y. Compact fourth-order finite difference schemes for Helmholtz equation with high wave numbers [J]. Journal of Computational Mathematics, 2008 26 (1): 98-111.
- [6] Hicks G J. Arbitrary source and receiver positioning in finite-difference schemes using Kaiser windowed sinc functions [J]. Geophysics, 2002 67 (1): 156-166.
- [7] Singer I, Turkel E. A perfectly matched layer for the Helmholtz equation in a semi-infinite strip [J]. Journal of Computational Physics, 2004 201 (1): 439-465.
- [8] Britt S, Tsynkov S, Turkel E. A Compact Fourth Order Scheme for the Helmholtz Equation in Polar Coordinates [J]. Journal of Scientific Computing, 2010, 45 (1): 26-47.

- [9] Nabavi M, Siddiqui M. H. K, Dargahi J. A new 9-point sixth-order accurate compact finite-difference method for the Helmholtz equation [J]. *Journal of Sound & Vibration*, 2007 307(3): 972-982.
- [10] Feng X F, Li Z L, Qiao Z H. High order compact finite difference schemes for the Helmholtz equation with discontinuous coefficients [J]. *Journal of Computational Mathematics*, 2011, 29 (3):324-340.
- [11] Singer I, Turkel E. Sixth-order accurate finite difference schemes for the Helmholtz equation [J]. *Journal of Computational Acoustics*, 2006 14 (3): 339-351.
- [12] Sutmann G. Compact finite difference schemes of sixth order for the Helmholtz equation [J]. *Journal of Computational & Applied Mathematics*, 2007 203 (1): 15-31.
- [13] Sutmann G. Compact finite difference schemes of sixth order for the Helmholtz equation [J]. *Journal of Computational & Applied Mathematics*, 2007 203 (1): 15-31.
- [14] Gordan D, Gordon R. Parallel solution of high frequency Helmholtz equations using high order finite difference schemes [J]. *Applied Mathematics & Computation*, 2012 218 (21): 10737-10754.
- [15] Helmholtz solver by domain decomposition and modified Fourier method [J]. *Siam Journal on Scientific Computing*, 2005 26 (5): 1504-1524.
- [16] Turkel E, Gordan D, Gordon R, et al.. Compact 2D and 3D sixth order schemes for the Helmholtz equation with variable wave number [J]. *Journal of Computational Physics*, 2013 232 (1): 72-287.
- [17] Lu Y Y. A fourth-order Magnus scheme for Helmholtz equation [J]. *Journal of Computational & Applied Mathematic*, 2005 173 (2): 247-258.