Journal of Electrical and Electronic Engineering
2021; 9(6): 186-193
http://www.sciencepublishinggroup.com/j/jeee
doi: 10.11648/j.jeee.20210906.12
ISSN: 2329-1613 (Print); ISSN: 2329-1605 (Online)

Fast Parallel FDFD Algorithm for Solving Electromagnetic Scattering Problems

Jianming Wu[1], Xinbo He[2, *], Bing Wei[2], Xianglin Li[2]

[1]The 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang, China
[2]School of Physics and Optoelectronic Engineering, Xidian University, Xi'an, China

Email address:

wujianming2105@sina.com (Jianming Wu), hexinbo@xidian.edu.cn (Xinbo He), lxl_ahcz@163.com (Xianglin Li), bwei@xidian.edu.cn (Bing Wei)

[*]Corresponding author

To cite this article:

Jianming Wu, Xinbo He, Bing Wei, Xianglin Li. Fast Parallel FDFD Algorithm for Solving Electromagnetic Scattering Problems. *Journal of Electrical and Electronic Engineering*. Vol. 9, No. 6, 2021, pp. 186-193. doi: 10.11648/j.jeee.20210906.12

Received: November 25, 2021; **Accepted**: December 8, 2021; **Published**: December 24, 2021

Abstract: The finite difference frequency domain (FDFD) method is very suitable for working out narrowband problems and resonance problems. However, the FDFD method needs to solve a large complex sparse matrix equation. With the increase of computing scale, the dimension of matrix will increase rapidly, which is difficult to simulate. For improving the computational efficiency of solving the large complex sparse matrix equation and extend the application scope of the FDFD method, a fast parallel FDFD method on the basis of message passing interface (MPI) shared memory technology is proposed in this paper, which is used to solve the electromagnetic scattering problems of electrically large targets. Based on the conjugate gradient iterative algorithm, the large complex sparse matrix is reasonably distributed to each process according to the unequal row allocation scheme, so as to guarantee the load balancing of each process. In addition, the intermediate vectors utilized in total processes are stored in the shared memory of MPI, which reduces the communication time and the consumption of computer memory. The proposed parallel FDFD method is employed to solve the bistatic RCS of the PEC sphere, composite Von warhead and an automobile, compared with the serial FDFD method, the parallel FDFD method greatly improves the computational efficiency when the memory is not increased much.

Keywords: FDFD, Complex Sparse Matrix, Conjugate Gradient Iteration, MPI, Shared Memory

1. Introduction

The finite difference time domain (FDTD) method is very popular in computational electromagnetics. It solves Maxwell's equations in the time domain and is very suitable for solving broadband electromagnetic problems [1]. However, when dealing with the electromagnetic problem with resonant structure, the FDTD method requires lots of time iterations to guarantee the accuracy of the simulation. Besides, the time step used in the FDTD method is constricted by the CFL condition [2], it is difficult for FDTD method to solving long pulse electromagnetic problems. For example, the duration of the E3 (late-time) waveform for the high altitude electromagnetic pulse (HEMP) is about 400s, while the grid size of the target is meter-scale. Because of the restriction of the stability criterion, the time step of the FDTD method should be nanosecond, then hundreds of millions of time steps need to be iterated to complete the simulation [3].

The FDFD method applies the differential Maxwell equation to describe electromagnetic field relationships. The central difference is employed to discretize the spatial partial derivative, and the time harmonic factor is used to replace the time partial derivative. Afterwards, the electromagnetic scattering properties of targets at a certain frequency can be acquired by solving the sparse matrix equation [4]. The FDFD method is not limited by the stability condition, so it has a great advantage in dealing with electromagnetic problems such as resonance and long pulse. In recent years, FDFD method has also been widely used in periodic structures [4], microstrip structures and waveguides [10, 13], long-path propagation [12], the band gap of the photonic crystal [5], bioelectromagnetic uncertainty analysis [15], etc. Researchers are also studying the absorbing boundary condition [11], fast algorithm of the FDFD method [14, 16].

However, the FDFD method needs to solve a large complex sparse matrix equation. With the increase of computing scale, the dimension of matrix will increase rapidly, which is difficult to simulate. For improving the computational efficiency of solving the large complex sparse matrix equation and extend the application scope of the FDFD method, the parallel FDFD method is proposed in this paper. Based on the conjugate gradient method, a shared memory technology of the message passing interface (MPI) is adopted in the FDFD method, and the large sparse matrix is allocated reasonably to each process. Then, the load balancing is ensured and the solution efficiency of the large complex sparse matrix equation is improved.

The rest of this paper is structured as follows. The FDFD algorithm is described in Section II. The parallel FDFD scheme is explained in Section III. Its accuracy and efficiency are proved in details in Section IV. Section V closes the paper with some conclusions.

2. Review of the FDFD Method

The 3D Maxwell equation in frequency domain is

$$\begin{cases}\nabla\times\boldsymbol{E}=j\omega\mu_r\mu_0\boldsymbol{H}\\ \nabla\times\boldsymbol{H}=-j\omega\varepsilon_r\varepsilon_0\boldsymbol{E}\end{cases}\tag{1}$$

where E, H represents electric field and magnetic field in the calculation region, respectively. ω is the angular frequency. μ_r, μ_0, ε_r, ε_0 represents relative permeability, permeability of vacuum, relative permittivity and permittivity of vacuum, respectively.

The Yee's cells are adopted to discretize the FDFD computation domain, and electric field sampling points and magnetic field sampling points are staggered with each other. Each electric field is surrounded by four magnetic fields, and each magnetic field is also surrounded by four electric fields.

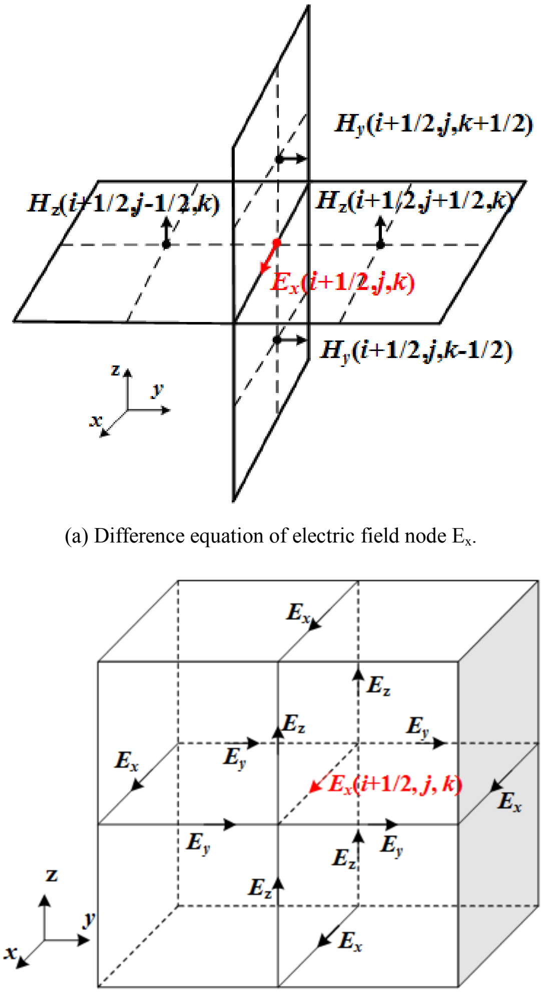
In the Cartesian coordinate system, the discretization equation related to the x direction's electric field $E_x(i+1/2,j,k)$ in (1) is

$$\begin{aligned}&j\omega\varepsilon_0\varepsilon_r(i+1/2,j,k)E_x(i+1/2,j,k)\\&=\frac{H_z(i+1/2,j+1/2,k)-H_z(i+1/2,j-1/2,k)}{\Delta y}\\&-\frac{H_y(i+1/2,j,k+1/2)-H_y(i+1/2,j,k-1/2)}{\Delta z}\end{aligned}\tag{2}$$

Similarly, we can get the other components of the electric field and magnetic field by discretizing (1) using central difference method. Equation (2) contains both electric field and magnetic field, for purpose of reducing the computational complexity, we substitute the difference equations for H_y and H_z into (2). Then, we can get an iterative equation containing only the electric field. For example, the FDFD iterative equation for electric field node $E_x(i+1/2,j,k)$ is as follows:

$$\begin{aligned}&cx0\cdot E_x(i+1/2,j,k)\\&+cx1\cdot E_x(i+1/2,j-1,k)+cx2\cdot E_x(i+1/2,j+1,k)\\&+cx3\cdot E_x(i+1/2,j,k-1)+cx4\cdot E_x(i+1/2,j,k+1)\\&+cy1\cdot E_y(i,j+1/2,k)+cy2\cdot E_y(i+1,j+1/2,k)\\&+cy3\cdot E_y(i,j-1/2,k)+cy4\cdot E_y(i+1,j-1/2,k)\\&+cz1\cdot E_z(i,j,k+1/2)+cz2\cdot E_z(i+1,j,k+1/2)\\&+cz3\cdot E_z(i,j,k-1/2)+cz4\cdot E_z(i+1,j,k-1/2)=0\end{aligned}\tag{3}$$

As can be seen from (3), the equation of each electric field involves its own node and its twelve adjacent nodes. And the twelve nodes are four E_x nodes, four E_y nodes and four E_z nodes respectively, as shown in Figure 1.



(a) Difference equation of electric field node E_x.

(b) Basic equation of electric field node E_x.

Figure 1. *Peripheral nodes involved in different situations of electric field nodes.*

where the coefficient are as follows, which are related to the grid size and the medium parameters.

$$\begin{cases} cx0=\omega^2\mu_0\varepsilon_0\Delta y\Delta z\varepsilon_r\left(i+1/2,j,k\right)-\left[\mu_r^{-1}\left(i+1/2,j-1/2,k\right)+{\mu^{-1}}_r\left(i+1/2,j+1/2,k\right)\right]\cdot\Delta z/\Delta y \\ \quad -\left[\mu_r^{-1}\left(i+1/2,j,k-1/2\right)+{\mu^{-1}}_r\left(i+1/2,j,k+1/2\right)\right]\cdot\Delta z/\Delta y \\ cx1=\mu_r^{-1}\left(i+1/2,j-1/2,k\right)\Delta z/\Delta y \\ cx2=\mu_r^{-1}\left(i+1/2,j+1/2,k\right)\Delta z/\Delta y \\ cx3=\mu_r^{-1}\left(i+1/2,j,k-1/2\right)\Delta y/\Delta z \\ cx4=\mu_r^{-1}\left(i+1/2,j,k+1/2\right)\Delta y/\Delta z \\ cy1=-cy2=\mu_r^{-1}\left(i+1/2,j+1/2,k\right)\Delta z/\Delta x \\ cy3=-cy4=-\mu_r^{-1}\left(i+1/2,j-1/2,k\right)\Delta z/\Delta x \\ cz1=-cz2=\mu_r^{-1}\left(i+1/2,j,k+1/2\right)\Delta y/\Delta x \\ cz3=-cz4=-\mu_r^{-1}\left(i+1/2,j,k-1/2\right)\Delta z/\Delta x \end{cases} \tag{4}$$

Similarly, the FDFD iterative equations for the other electric field components can be obtained. By combining the FDFD equations corresponding to all electric field nodes, the solution of the field in the FDFD calculation domain can be simplified to the solution of the matrix equation $\boldsymbol{A}\cdot x=b$, where $\boldsymbol{A}$ is the coefficient matrix composed of the coefficients in (4), *x* is the vector formed by the electric field nodes to be solved in the computational domain, and b is the excitation source vector.

3. Fast Solution of Large Complex Sparse Matrix Equations

3.1. Conjugate Gradient Iteration

The FDFD method can be transformed into the form of the matrix equation $\boldsymbol{A}\cdot x=b$, where the coefficient matrix $\boldsymbol{A}$ is a complex sparse matrix. To reduce the consumption of computing resources, the format of Coordinate (COO) [6] is used to store the complex sparse matrix. This format stores only the row, column and value of each non-zero element for the sparse matrix, which is very efficient. Because $\boldsymbol{A}$ is not a special matrix such as triangular matrix and banded matrix, etc. the existing efficient iterative solvers are not suitable. Therefore, in this paper, the conjugate gradient method (CGM) of complex matrix is used to solve the matrix equation [7, 8], and the specific steps are as follows [9]:

(i) Initialization, set initial value r_0 and convergence accuracy *err*:

$$\begin{cases} r_0=\boldsymbol{A}x_0-b \\ p_1=-\boldsymbol{A}^+r_0 \end{cases} \tag{5}$$

where $\boldsymbol{A}^+$ is the conjugate transpose of matrix $\boldsymbol{A}$.

(ii) Iteration (n = 1, 2, 3, 4…)

$$\begin{cases} \alpha_n &= \left\|\boldsymbol{A}^+r_{n-1}\right\|^2/\left\|\boldsymbol{A}p_n\right\|^2 \\ x_n &= x_{n-1}+\alpha_n p_n \\ r_n &= \boldsymbol{A}x_n-b=r_{n-1}+\alpha_n\boldsymbol{A}p_n \\ \beta_n &= \left\|\boldsymbol{A}^+r_n\right\|^2/\left\|\boldsymbol{A}^+r_{n-1}\right\|^2 \\ p_{n+1} &= -\boldsymbol{A}^+r_n+\beta_n p_n \end{cases} \tag{6}$$

(iii) Make a judgment after each iteration. Once $\left\|\boldsymbol{A}x-b\right\|/\left\|b\right\| \le err$, output x_n and finish the iteration. Otherwise, continue with step (ii).

3.2. Parallel Algorithm of Conjugate Gradient Iteration Based on MPI

It can be seen from the above section that the conjugate gradient algorithm can greatly improve the computational efficiency of matrix equations by transforming the solution of matrix equations into simple sparse matrix vector multiplication, vector inner product and vector addition and subtraction. However, when the electrical size of the calculated target increases, the scale of the coefficient matrix also increases. On the one hand, it will lead to a sharp increase in the amount of computation and an increase in the computational resources consumed by each iteration. On the other hand, it makes the convergence of iteration worse, that is, it needs more iterations to complete the computation. In order to improve efficiency, parallel computing based on MPI is a better solution. However, due to the sparsity of the coefficient matrix, it is not feasible to use the parallel scheme of dense matrix partition by block. This is because the distribution of nonzero elements in the coefficient matrix is not uniform, and partition by block will lead to unbalanced load, which affects the parallel efficiency. In addition, because the number of nonzero elements in each row of the coefficient matrix is not equal, the common equal row allocation scheme will also lead to load imbalance. According to the characteristics of coefficient matrix, a scheme of unequal row allocation according to the number of elements is proposed in this paper. The coefficient matrix is evenly distributed to each process to ensure load balancing.

The idea of parallelization in this paper is:

(i) Matrix $\boldsymbol{A}$ is a large complex sparse matrix, whose most elements are zero, and only nonzero elements affect the calculation results. Thus, the total number of nonzero elements N_{tot} is counted first.

(ii) To guarantee the load balancing of each process, it is necessary to ensure that the number of nonzero elements allocated to each process is basically the same.

Therefore, the number of nonzero elements that should be allocated to each process is about $Aver = N_{tot} / n$, where n stands for the number of processes.

(iii) Following the principle of allocation by row, start to allocate nonzero elements to n processes. Because the number of nonzero elements in each row of the coefficient matrix is not equal, on the premise of ensuring that the number of nonzero elements contained in each process is greater than or equal to A_{ver}, the number of matrix rows allocated to each process may be inconsistent, but it must be ensured that the nonzero elements in each process are all nonzero elements of a row or several rows in the coefficient matrix.

The specific allocation steps are given below.

a) Traverse all nonzero elements N_{tot} and count the number of nonzero elements in each row.

b) An auxiliary array is allocated to store the start position and end position of nonzero elements in each row.

c) Start assigning rows to each process until the number of nonzero elements in the process is greater than or equal to A_{ver}.

d) Count the remaining rows of the matrix and assign them to a subprocess other than the main process.

e) Count the start and end positions of the matrix rows allocated by each process.

(iv) The above processes need to be allocated only once before the iteration, but the intermediate vectors r_n, p_n and x_n used by each process need to be assigned to each process during the iteration. Because the intermediate vectors are allocated during the iteration, the communication time is long. In addition, intermediate vectors need to be fully allocated to each process, so the computer memory consumption will increase with the increase of the number of processes. Therefore, the shared memory technology of MPI is introduced in this paper to optimize the allocation of intermediate vectors. In our work, the intermediate vectors needed by each process are stored in the shared memory, this shared memory technology eliminates the requirement for the main process to send intermediate vectors to each subprocess, which not only reduces the memory occupation, but also reduces the communication time between processes. Figure 2 shows the storage location of intermediate vectors in memory.

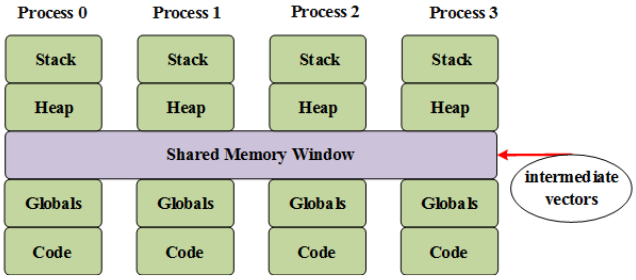


Figure 2. *The intermediate vector is stored in the shared memory area.*

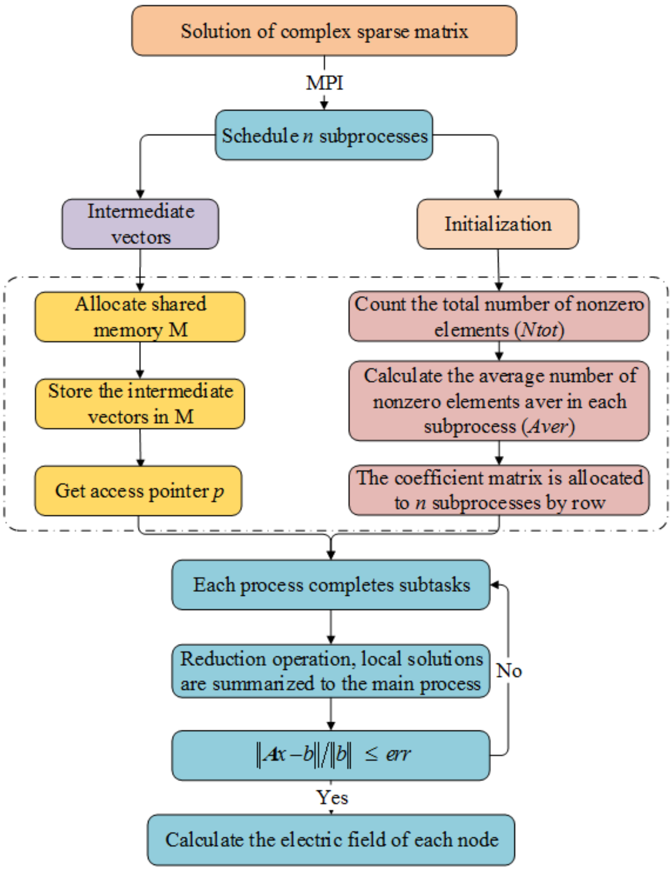

Solution of complex sparse matrix
MPI
Schedule n subprocesses
Intermediate vectors
Initialization
Allocate shared memory M
Store the intermediate vectors in M
Get access pointer p
Count the total number of nonzero elements ($Ntot$)
Calculate the average number of nonzero elements aver in each subprocess ($Aver$)
The coefficient matrix is allocated to n subprocesses by row
Each process completes subtasks
Reduction operation, local solutions are summarized to the main process
No
$\|Ax-b\|/\|b\| \le err$
Yes
Calculate the electric field of each node

Figure 3. *MPI based conjugate gradient iterative parallel algorithm.*

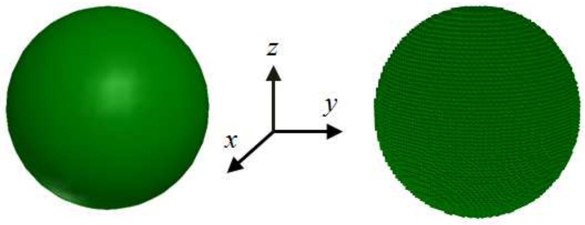
(v) After the assignment is completed, each process obtains the corresponding local solution according to the process of complex conjugate gradient method, and the final solution x_n can be obtained by gathering the local solution to the main process.

The flow chart of the parallel algorithm of conjugate gradient iterative based MPI is shown in Figure 3.

4. Numerical Results

4.1. Bistatic RCS of PEC Sphere

The geometric diagram and grid discretization of the PEC sphere are shown in Figure 4, with a radius $r = 1.0\text{m}$, the frequency of the plane wave is $f = 0.3\text{GHz}$ and its wavelength is $\lambda = c/f = 1.0\text{m}$. The plane wave is incident along the z-direction, and the electric field is in the xoy plane, i.e. the pitch angle is $\theta = 0^\circ$, the azimuth angle is $\varphi = 0^\circ$, and the polarization angle is $\alpha = 0^\circ$. The discrete grid size is taken as $\Delta x = \Delta y = \Delta z = \lambda / 20 = 0.05\text{m}$. In all directions, the target boundary is extrapolated to the truncated boundary by 10 grids. Finally, the whole calculation area is ($-30\Delta x : 30\Delta x, -30\Delta y : 30\Delta y, -30\Delta z : 30\Delta z$). The dimension of the coefficient matrix is 703452, and there are 494844716304 elements in total, including 8928744 non-zero elements, accounting for 0.0018% of the total matrix elements. The calculation results of FDFD serial algorithm, parallel scheme and commercial software are shown in Figure 5. It is obvious that the calculation results of FDFD parallel scheme are in good agreement with FDFD serial algorithm and commercial software HFSS, which proves the correctness and feasibility of FDFD algorithm and parallel scheme. Table 1 shows the calculation time and computer memory occupied by FDFD parallel algorithm in solving PEC sphere when the number of processes is different. From Table 1, it is clear that the parallel efficiency of FDFD first increases and then decreases as the increase of the number of processes. This is because the calculation scale of this example is small, and the communication time between processes increases with the number of processes increases, which reduces the calculation efficiency. In addition, because the number of processes will also affect the number of conjugate gradient iterations, the parallel efficiency in Table 1 may be greater than 100%. In the case of same convergence accuracy, the number of iterations of serial algorithm and parallel algorithm is different.



(a) The geometric diagram. (b) The grid discretization.

Figure 4. *The model of the PEC sphere model..*

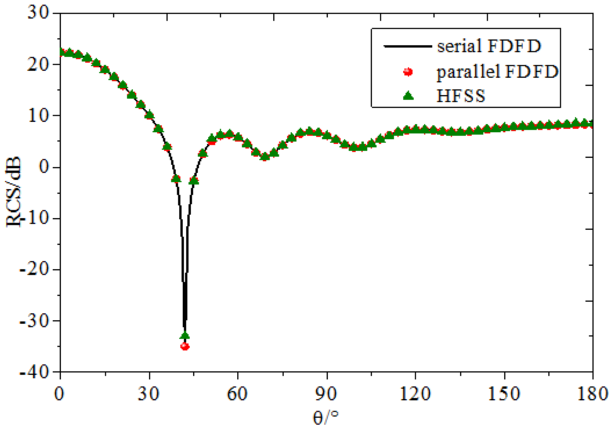
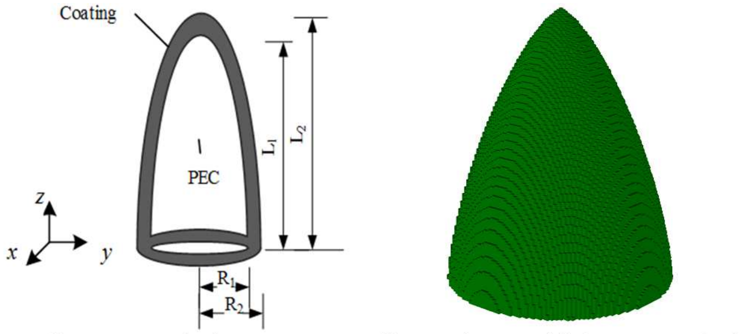


Figure 5. *The bistatic RCS of PEC sphere.*



(a) Composite warhead geometry (b) FDFD discrete model of composite warhead.

Figure 6. *Von warhead model.*

Table 1. *Performance comparison of PEC sphere in different number of processes.*

Number of processes	**1 (serial)**	**2**	**4**	**8**	**12**	**16**	**24**
Memory/MB	182	316	341	391	430	491	590
Time/s	1181	600	265	123	101	90	81
Speed up	1	1.96	4.45	9.6	11.69	13.1	14.6
Parallel efficiency /%	-	98	111	120	97	82	61

4.2. Bistatic RCS of Composite Von Warhead

The composite Von warhead is composed of inner and outer layers, the inner layer is PEC structure, and the outer layer is coated with absorbing materials to reduce its RCS. The geometric diagram and grid discretization of the composite warhead are shown in Figure 6. Its dimensions are $R_1 = 0.1$m, $R_2 = 0.15$m, $L_1 = 0.4$m, $L_1 = 0.4$m, $L_2 = 0.6$m. The frequency of the incident wave is f = 3GHz, and the plane wave is incident along the *z*-direction, that is, the incident angle are $\theta = 0°$, $\varphi = 0°$, and $\alpha = 0°$. The medium parameters of the outer coating are $\varepsilon_r = (2.0 - j0.5)$, $\mu_r = (1.24 - j0.2)$. The grid size is $\Delta x = \Delta y = \Delta z = \lambda/20$ = 5mm, the serial FDFD and parallel FDFD methods are adopted to calculate the bistatic RCS of the warhead, and the calculation results are shown in Figure 7. It can be seen that the calculation results of the two methods agree with well. Table 2 shows the results of time consumption and memory occupation of parallel methods with different process numbers. In this example, the parallel efficiency can reach 93% when the number of processes is $n = 24$.

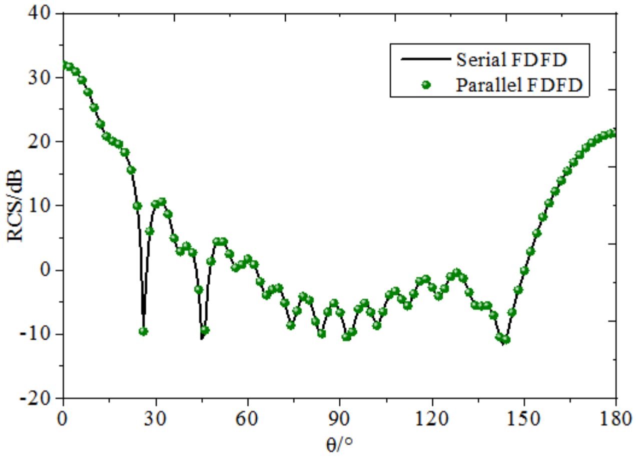


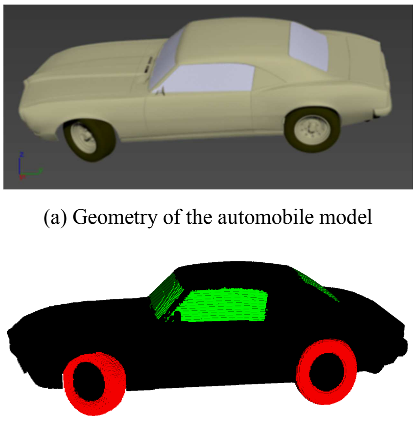
Figure 7. *Bistatic RCS of Von warhead.*

Table 2. *Performance comparison of Von warhead model in different number of processes.*

Number of processes	**1 (serial)**	**2**	**4**	**8**	**12**	**16**	**24**
Memory/MB	1233	2077	2104	2157	2204	2243	2347
Time/s	5867	2763	1428	600	376	330	263
Speed up	1.0	2.12	4.11	9.78	15.6	17.8	22.3
Parallel efficiency /%	-	106	103	122	125	111	93

4.3. Bistatic RCS of an Automobile

The geometric diagram and grid discretization of the automobile are shown in Figure 8. The length of automobile is $l = 4.2\text{m}$, the width is $w = 1.5\text{m}$ and the height is $h = 1.2\text{m}$. The incident frequency in the case of lightning is $f = 100\text{MHz}$ and the plane wave is incident from the automobile head direction, i.e. along the *y*-direction. The incident angle are $\theta = 0^\circ$, $\varphi = 0^\circ$ and the polarization angle is $\alpha = 0^\circ$ and $\alpha = 45^\circ$. Due to the complexity of automobile components, it is difficult to obtain the dielectric parameters of each component. In this example, the automobile components are mainly divided into three parts: automobile frame, tire and window glass. The material of automobile frame is regarded as PEC. The tire is mainly composed of rubber, and its corresponding dielectric parameters are $\mu_r = 1.0$ and $\varepsilon_r = 5.5$, The window glass is made of glass, and its dielectric parameters are $\mu_r = 1.0$ and $\varepsilon_r = 3.0$. In order to describe the fine structures of the automobile, the grid size is $\Delta x = \Delta y = \Delta z = \lambda / 100 = 0.03\text{m}$. The target boundary is extrapolated to the truncated boundary by 30 grids, and the whole calculation area is ($-100\Delta x : 100\Delta x$, $-55\Delta y : 55\Delta y$, $-50\Delta z : 50\Delta z$). The serial FDFD and parallel FDFD methods are employed to calculate the bistatic RCS of the automobile with polarization angle of $\alpha = 0^\circ$ and $\alpha = 45^\circ$. The simulation results of the two polarization cases are shown in Figure 9 and Figure 10, respectively. It can be seen that the consistency between the parallel FDFD and the serial FDFD algorithms.


(a) Geometry of the automobile model

(b) Grid discretization of the automobile model

Figure 8. *The automobile model.*

Table 3. *Performance comparison of the automobile model in different number of processes.*

Number of processes	**1 (serial)**	**2**	**4**	**8**	**12**	**16**	**24**
Memory/MB	3645	6150	6171	6230	6277	6329	6424
Time/s	20890	10351	4246	2311	1488	1190	927
Speed up	1.0	2.01	4.92	9.04	14.0	17.6	22.5
Parallel efficiency /%	-	100	123	113	117	110	94

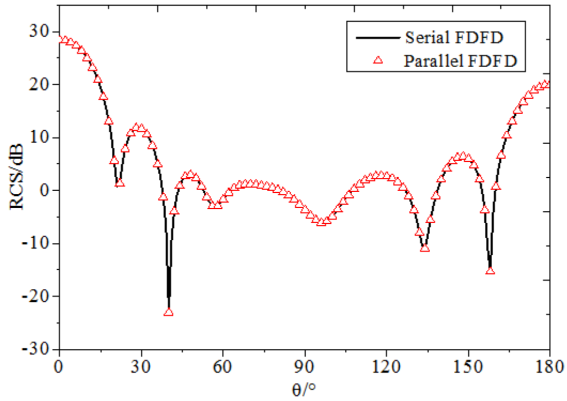


Figure 9. *Bistatic RCS of the automobile at polarization angle* $\alpha = 0^\circ$.

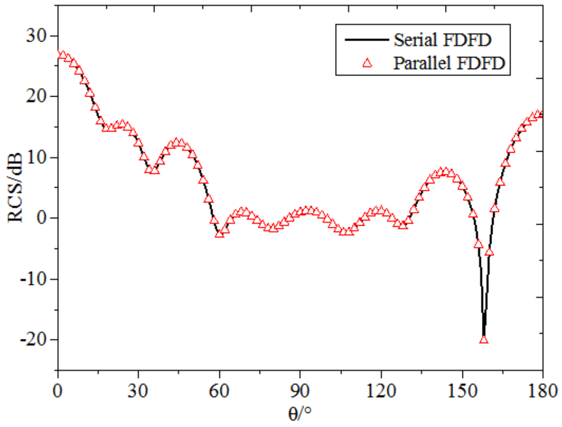


Figure 10. *Bistatic RCS of the automobile at polarization angle* $\alpha = 45^\circ$.

Table 3 shows the time and memory consumed by the parallel FDFD algorithm when the number of processes is different. It can be seen that the parallel efficiency can reach 94% when the number of processes is $n = 24$.

5. Conclusions

Aiming at the problem that the computational efficiency of FDFD method is low and difficult to solve electrically large targets, MPI is used to parallelize the FDFD algorithm, and the shared memory mechanism of MPI is introduced to store the intermediate vector, which reduces the computational resources and improves the computational efficiency of FDFD algorithm. Numerical results show that compared with the serial algorithm, the parallel algorithm occupies about two times as much memory as the serial algorithm, but its speedup ratio in 24 processes can reach 22.5, and the parallel efficiency can reach 94%. The parallel scheme greatly improves the computation efficiency and application scope of FDFD algorithm, and provides a certain reference value for further improving the computation performance of FDFD algorithm. In this paper, we only calculate the scattering problem of the target by FDFD method. In future work, the coupling effect of the HEMP E3 pulse and the rail, the propagation of low frequency electromagnetic wave in the ground-ionospheric waveguide will be studied.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61901324, 62001345); The China Postdoctoral Science foundation (2019M653548, 2019M663928XB); the Fundamental Research Funds for the Central Universities (XJS200501, XJS200507, JB200501, YJS2105).

References

[1] K. S. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. IEEE Transactions on Antennas and Propagation, vol. 14, no. 3, pp. 302-307, May 1966.

[2] T. Namiki. A new FDTD algorithm based on alternating-direction implicit method. IEEE Transactions on Microwave Theory and Techniques, vol. 47, no. 10, pp. 2003-2007, Oct. 1999.

[3] T. An, M. Wei, S. Y. Li, et al. Study on the comparation of field-wire coupling effect for long wire in UWB with HEMP. Journal of Microwaves, vol. 26, no. 4, pp. 14-18, Apr. 2010.

[4] F. Xu, Y. L. Zhang, W. Hong, et al. Finite-difference frequency-domain algorithm for modeling guided-wave properties of substrate integrated waveguide. IEEE Transactions on Microwave Theory and Techniques, vol. 51, no. 11, pp. 2221-2227, Nov. 2003.

[5] A. G. Hanif, T. Arima, T. Uno. Finite-difference frequency-domain algorithm for band-diagram calculation of 2-D photonic crystals composed of Debye-type dispersive materials. IEEE Antennas and Wireless Propagation Letters, vol. 11, pp. 41-44, 2012.

[6] N. Neuss. A new sparse-matrix storage method for adaptively solving large systems of reaction-diffusion-transport equations. Computing, vol. 68, no. 1, pp. 19-36, Sep. 2001.

[7] D. A. H. Jacobs. A Generalization of the conjugate-gradient method to solve complex systems. IMA Journal of Numerical Analysis, vol. 6, no. 4, pp. 447-452, 1986.

[8] V. Demir, E. Alkan, A. Z. Elsherbeni, et al. An algorithm for efficient solution of finite-difference frequency-domain (FDFD) methods. IEEE Antennas and Propagation Magazine, vol. 51, no. 6, pp. 143-150, 2010.

[9] X. L. Li, B. Wei, X. B. He, et al. Parallel FDFD Algorithm Based on MPI and Its Application. 2020 Cross Strait Radio Science & Wireless Technology Conference (CSRSWTC), pp. 1-3, 2020.

[10] J. N. Hwang. A compact 2-D FDFD method for modeling microstrip structures with nonuniform grids and perfectly matched layer. IEEE Transactions on Microwave Theory and Techniques, vol. 53, no. 2, pp. 653-659, Feb. 2005.

[11] C. M. Rappaport, M. Kilmer, E. Miller. Accuracy considerations in using the PML ABC with FDFD Helmholtz equation computation. International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, vol. 13, pp. 471-482, 2000.

[12] M. W. Chevalier, U. S. Inan. A technique for efficiently modeling long-path propagation for use in both FDFD and FDTD. IEEE Antennas and Wireless Propagation Letters, vol. 5, pp. 535-528, 2006.

[13] G. Zheng, B. Z. Wang. A hybrid MM-FDFD method for the analysis of waveguides with multiple discontinuities. IEEE Antennas and Wireless Propagation Letters, vol. 11, pp. 645-647, 2012.

[14] X. G. Xie, L. Wei, Ying L., et al. Using LU decomposition in FDFD for fast calculation of monostatic RCS, 2014 7th International Conference on Intelligent Computation Technology and Automation, pp. 887-889, 2014.

[15] K. Masumnia-Bisheh, K. Forooraghi, M. Ghaffari-Miab. Electromagnetic uncertainty analysis using stochastic FDFD method. IEEE Transactions on Antenna and Propagation, vol. 67, no. 5, pp. 3268-3277, May. 2019.

[16] X. Gu, X. L. Jin, J. X. Li, et al. Two-component compact 2-D FDFD method for waveguide structures with ARPACK. 2019 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting, pp. 187-188, 2019.