

An Effective Parallel FDTD Algorithm For Modeling 3D Frequency-Dependent Electromagnetic Applications

Omar Ramadan, Muhammed Salamah and Ahmad Salh

Computer Engineering Department
Eastern Mediterranean University
GaziMagusa, Mersin 10, Turkey

omar.ramadan@emu.edu.tr; muhammed.salamah@emu.edu.tr; asdwifi@googlemail.com

Abstract—Full-wave parallel finite difference time domain (FDTD) algorithm is presented for modeling open region dispersive electromagnetic applications. The algorithm is based on spatial partitioning of the problem geometry into adjacent non-overlapping sub-domains using two-dimensional topology. The inter-processor communication among the neighboring processors is carried out by using the message passing interface (MPI) library. The performance of the proposed parallel system, which is composed of 16 PCs interconnected through 100Mbps Ethernet, was illustrated for a point source radiating in three dimensional Lorentz dispersive domain and it has been found that the proposed algorithm not only speed up computations but also increases the maximum solvable problem size.

Keywords—Parallel programming; message passing interface (MPI); finite difference time domain (FDTD); anisotropic perfectly matched layer (APML); dispersive media.

I. INTRODUCTION

In the last decade, the finite difference time domain (FDTD) method [1] has been widely used for solving many electromagnetic problems [2]. This is due to its simplicity and direct applicability to Maxwell's curl equations. Nevertheless, when the FDTD method is used for modeling open region problems, efficient absorbing boundary conditions (ABCs) are needed to truncate the computational domains. The perfectly matched layer (PML) [3]-[6] has been shown to be one of the most effective FDTD ABCs. This ABCs surrounds the FDTD computational domain with a lossy layer that absorbs outgoing waves with minimal reflections.

To model large problems using the FDTD method, intensive computational time and memory storage are needed. Hence, parallelizing the FDTD method has been shown to be one of the latest challenges in the FDTD research. In last few years, different parallel PML-FDTD algorithms, based on the message passing interface (MPI) library [7], have been successfully introduced [8]-[14]. Nevertheless, these algorithms are suitable only for non-dispersive electromagnetic applications. In [15], dispersive parallel scalar wave equation FDTD algorithm has been presented. This approach, however, is valid only for source free applications only.

In this paper, full-wave parallel PML-FDTD algorithm is presented for modeling open region dispersive electromagnetic problems. The algorithm is based on spatial partitioning of the problem geometry into adjacent non-overlapping sub-domains using two-dimensional (2-D) topology and the inter-processor communication among the neighboring processors is carried out by using the MPI library. The performance of the proposed parallel system, which is composed of 16 PCs interconnected through 100Mbps ethernet, was illustrated for a point source radiating in three dimensional Lorentz dispersive domain and it has been observed that the parallel algorithm not only speed up computations but also increases the maximum solvable problem size. The paper is organized as follows. In Section II, the basic formulations of the FDTD and the APML ABCs approaches are presented. In Section II, the proposed parallel strategy is described. Numerical example to show the validity of the proposed parallel algorithm is included in Section III. Finally, summary and conclusions are included in Section IV.

II. THEORY

A. Basic Formulations

Considering an isotropic, homogeneous and dispersive computational domain, the frequency domain Maxwell's curl equations can be written as

$$j\omega\epsilon_0\epsilon_r(\omega)\mathbf{E}(\mathbf{r},\omega) = \nabla \times \mathbf{H}(\mathbf{r},\omega) \quad (1)$$

$$j\omega\mu_0\mathbf{H}(\mathbf{r}, \omega) = -\nabla \times \mathbf{E}(\mathbf{r}, \omega) \quad (2)$$

where \mathbf{E} and \mathbf{H} are, respectively, the electric and the magnetic field vectors, and $\varepsilon_r(\omega)$ is the relative permittivity of the domain which can be written as

$$\varepsilon_r(\omega) = \frac{\sum_{m=0}^M a_m(j\omega)^m}{\sum_{m=0}^M b_m(j\omega)^m} \quad (3)$$

where a_m and b_m , ($m = 0, 1, \dots, M$), are the coefficients of the rational polynomials and M is the maximum order of the dispersive domain. To discretize (1) and (2), consider, as an example, the E_z -field component of (1), i.e.,

$$j\omega\varepsilon_0\varepsilon_r(\omega)E_z = \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \quad (4)$$

Equation (4) can be written as

$$j\omega\varepsilon_0 D_z = \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \quad (5)$$

where D_z is related to E_z through the constitutive relation:

$$D_z = \varepsilon_r(\omega)E_z = \frac{\sum_{m=0}^M a_m(j\omega)^m}{\sum_{m=0}^M b_m(j\omega)^m} E_z \quad (6)$$

Using the inverse Fourier transform relation, $j\omega \Rightarrow \partial/\partial t$, and employing the FDTD time and space discretizations [1], (5) can be written in the discrete time domain as

$$D_{z,i,j,k+1/2}^{n+1} = D_{z,i,j,k+1/2}^n + \frac{\Delta t}{\Delta \varepsilon_0} \left[H_{y,i+1/2,j,k+1/2}^{n+1/2} - H_{y,i-1/2,j,k+1/2}^{n+1/2} - H_{x,i,j+1/2,k+1/2}^{n+1/2} + H_{x,i,j-1/2,k+1/2}^{n+1/2} \right] \quad (7)$$

where $\Delta = \Delta x = \Delta y$ is the space cell size. Equation (6) can be easily written in the discrete time domain easily by using the Bilinear transformation relation [16]

$$j\omega \Rightarrow \frac{z - z^{-1}}{\Delta t} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (8)$$

where Z^{-1} is the Z -transform variable which corresponds to a single delay element in the discrete time domain. To this end, (6) can be written in the Z -domain as

$$D_z(Z) = \frac{\sum_{m=0}^M c_m Z^{(1-m)}}{\sum_{m=0}^M d_m Z^{(1-m)}} E_z(Z) \quad (9)$$

where c_m and d_m , ($m = 0, 1, \dots, M$), are related to a_m and b_m and the time step Δt . Using the Z -transform relation

$$Z^{-m} G(Z) \rightarrow G^{n-m} \quad (10)$$

(9) can be written directly in the discrete time form as

$$E_{z,i,j,k+1/2}^{n+1} = \frac{d_0}{c_0} D_{z,i,j,k+1/2}^{n+1} + \Psi_{i,j,k+1/2}^n \quad (11)$$

where

$$\Psi_{i,j,k+1/2}^n = \frac{1}{c_0} \sum_{m=1}^M \left(d_m D_{z,i,j,k+1/2}^{n+(1-m)} - c_m E_{z,i,j,k+1/2}^{n+(1-m)} \right) \quad (12)$$

Similar equations can be obtained for the other field components.

B. Absorbing Boundary Conditions

Using the anisotropic PML (APML) formulations of [5], (1) and (2) can be written in the APML region at the domain boundaries as

$$j\omega\varepsilon_0\varepsilon_r(\omega)\bar{\varepsilon}(\mathbf{r}, \omega)\mathbf{E}(\mathbf{r}, \omega) = \nabla \times \mathbf{H}(\mathbf{r}, \omega) \quad (13)$$

$$j\omega\mu_0\bar{\mu}(\mathbf{r}, \omega)\mathbf{H}(\mathbf{r}, \omega) = -\nabla \times \mathbf{E}(\mathbf{r}, \omega) \quad (14)$$

where $\bar{\varepsilon}(\mathbf{r}, \omega)$ and $\bar{\mu}(\mathbf{r}, \omega)$ are, respectively, the APML permittivity and permeability diagonal tensors defined as [5]

$$\bar{\varepsilon}(\mathbf{r}, \omega) = \bar{\mu}(\mathbf{r}, \omega) = \begin{bmatrix} S_y S_z & & \\ & S_x & \\ & & S_y \\ & & & S_x S_y \\ & & & & S_z \end{bmatrix} \quad (15)$$

with S_η ($\eta = x, y, \text{ or } z$) are given by

$$S_\eta = 1 + \frac{\sigma_\eta}{j\omega\varepsilon_0} \quad (16)$$

where σ_η is the APML conductivity profile along the η -coordinate designed to absorb the outgoing waves with minimal reflections [3]. To discretize (13) and (14), consider, as an example, the E_z -field component of (13):

$$j\omega\varepsilon_0\varepsilon_r(\omega) \frac{\left(1 + \frac{\sigma_y}{j\omega\varepsilon_0}\right)\left(1 + \frac{\sigma_x}{j\omega\varepsilon_0}\right)}{\left(1 + \frac{\sigma_z}{j\omega\varepsilon_0}\right)} E_z = \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \quad (17)$$

equation (17) can be re-arranged as

$$j\omega\varepsilon_0 \left(1 + \frac{\sigma_x}{j\omega\varepsilon_0}\right) G_z = \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \quad (18)$$

where G_z is given by

$$G_z = \frac{\left(1 + \frac{\sigma_y}{j\omega\varepsilon_0}\right)}{\left(1 + \frac{\sigma_z}{j\omega\varepsilon_0}\right)} D_z \quad (19)$$

and D_z is related to E_z through (6). Using the inverse Fourier transform relation, $j\omega \Rightarrow \partial/\partial t$, (18) and (19) can be written in the time domain as

$$\frac{\partial G_z}{\partial t} + \frac{\sigma_x}{\varepsilon_0} G_z = \frac{1}{\varepsilon_0} \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) \quad (20)$$

and

$$\frac{\partial G_z}{\partial t} + \frac{\sigma_z}{\varepsilon_0} G_z = \frac{\partial D_z}{\partial t} + \frac{\sigma_y}{\varepsilon_0} D_z \quad (21)$$

Using the FDTD algorithm [2], (20) and (21) can be written in the discrete time domain as

$$G_{z,i,j,k+1/2}^{n+1} = \frac{\alpha_{x_i}^-}{\alpha_{x_i}^+} G_{z,i,j,k+1/2}^n + \frac{\Delta t}{\alpha_{x_i}^+ \Delta \varepsilon_0} \left[H_{y,i+1/2,j,k+1/2}^{n+1/2} - H_{y,i-1/2,j,k+1/2}^{n+1/2} - H_{x,i,j+1/2,k+1/2}^{n+1/2} + H_{x,i,j-1/2,k+1/2}^{n+1/2} \right] \quad (22)$$

$$D_{z,i,j,k+1/2}^{n+1} = \frac{\alpha_{z_j}^-}{\alpha_{z_j}^+} D_{z,i,j,k+1/2}^n + \frac{\alpha_z^{k+1/2}}{\alpha_{z_j}^+} \left[G_{z,i,j,k+1/2}^{n+1} - \frac{\alpha_{z_{k+1/2}}^-}{\alpha_{z_{k+1/2}}^+} G_{z,i,j,k+1/2}^n \right] \quad (23)$$

where $\alpha_{\eta_m}^\pm$, (for $\eta = x, y, \text{ or } z$), is given by

$$\alpha_{\eta_m}^\pm = 1 \pm \Delta t \sigma_{\eta_m} / 2\varepsilon_0 \quad (24)$$

After computing $G_{z,i,j,k+1/2}^{n+1}$ and $D_{z,i,j,k+1/2}^{n+1}$ from (22) and (23), respectively, $E_{z,i,j,k+1/2}^{n+1}$ can be obtained from (11) and (12). It is important to note that (22) and (23) can also be applied in the inner FDTD computational domain by setting the APML conductivity profiles ($\sigma_\eta(\eta = x, y, z)$) to zero. Similar expressions can be obtained for the other field components.

C. Parallelization Strategy

In the presented parallel algorithm, the computational domain is spatially partitioned into adjacent non-overlapping sub-domains using 2-D topology, in which the computational domain is divided into sub-domains along two directions. Fig.

1 shows a typical 2-D decomposition when 4 processors are used. To update the field components at the sub-domain boundaries, data from the neighboring sub-domains are needed. In this paper, the inter-processor communication among the neighboring processors is carried out by using the MPI library [7]. Fig. 2 shows the data need to be exchanged between neighboring sub-domains. For the communication purpose, ghost layers located at the edges of the sub-domains are used as shown in Fig. 2. It is important to note that as the APML field equations involve the same number of inter-processor communication operations as the conventional FDTD equations, as can be seen from (7), (22), and (23), the APML finite-difference equations can be used for the total computation domain by properly choosing the APML parameters. This makes the parallel FDTD algorithm easier to implement.

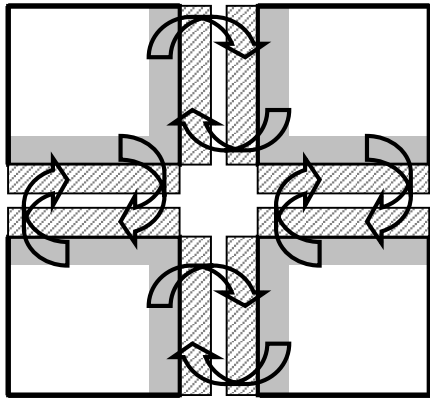


Fig.1: Computational domain partitioning using 2-D topology. Shaded and gray layers represent ghost layers and internal edges at the sub-domains, respectively.

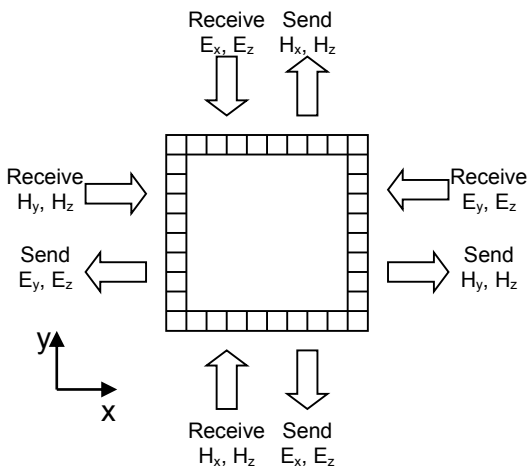


Fig. 2: Communications at the boundaries of a sub-domain for the 2-D topology.

Finally, the steps for the proposed parallel algorithm can be summarized as:

1. MPI initialization.
2. Reading of simulation parameters.
3. Creation of the 2-D topology.
4. At each time step perform the following:
 - 4.1 Exchange **H**-fields with the neighbor sub-domains by using the MPI library functions.
 - 4.2 Update the **E**-fields and other auxiliary variables in each sub-domain.
 - 4.3 Exchange **E**-fields with the neighbor sub-domains by using the MPI library functions.
 - 4.4 Update the **H**-fields in each sub-domain.
5. MPI finalization.

TABLE I: FDTD PARALLEL SYSTEM CHARACTERISTICS.

CPU	Pentium IV 2.20 GHz
Memory	512 Mbyte
Processor number	4, 8, or 16
Communication software	Message passing interface
Network interface	100Mbps Ethernet
Operating system	Windows XP
Compiler	C++

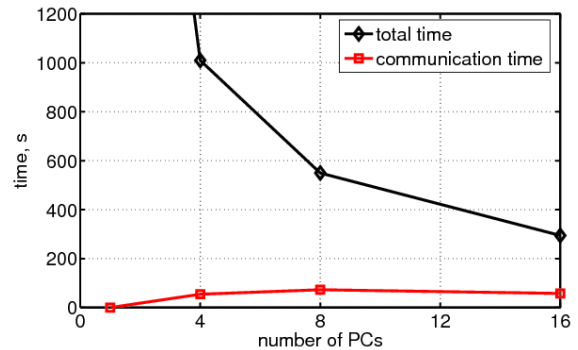


Fig. 3: Total simulation time and communication time of the proposed parallel algorithm

III. SIMULATION STUDY

The performance of the proposed parallel algorithm was studied for a three dimensional radiation problem. In this study, a *z*-polarized modulated Gaussian pulse with a carrier frequency of 20 PHz was excited at the center of $240\Delta \times 240\Delta \times 40\Delta$ computational domain, where $\Delta = \Delta x = \Delta y = \Delta z = 1 \times 10^{-10}m$. The computational domain was entirely composed of linear Lorentz material ($M = 2$) with a dielectric permittivity given by

$$\epsilon_r(\omega) = \epsilon_\infty + \frac{\Delta\epsilon\omega_0^2}{\omega_0^2 + j2\delta\omega - \omega^2} \quad (25)$$

where $\epsilon_\infty = \epsilon_r(\infty) = 1.0$, $\Delta\epsilon = \epsilon_s - \epsilon_\infty$, with $\epsilon_s = \epsilon_r(0) = 2.25$, $\omega_0 = 4 \times 10^{16}rad/s$ is the resonance radial frequency, and $\delta = 0.28 \times 10^{16}s^{-1}$ is the damping constant [17]. In this case, the coefficients of (3) are

$$a_0 = (\epsilon_\infty + \Delta\epsilon)\omega_0^2, a_1 = 2\epsilon_\infty\delta, a_2 = \epsilon_\infty,$$

$$b_0 = \omega_0^2, b_1 = 2\delta, \text{ and } b_2 = 1 \quad (26)$$

The computational domain was truncated by eight additional PML layers with a quadratic conductivity profile and with a theoretical reflection coefficient of 10^{-5} , as defined in [3]. The simulation time was carried out for the first 2500 time steps and the time step was taken as $\Delta_t = \Delta/(\sqrt{3}c/\sqrt{\epsilon_\infty})$, where c is the speed of light in vacuum. The parallel system used in this study was composed of 16 PCs interconnected through 100Mbps ethernet. Table I shows the characteristics of the proposed parallel system. Fig. 3 shows the total simulation time and the communication time of the proposed parallel algorithm.

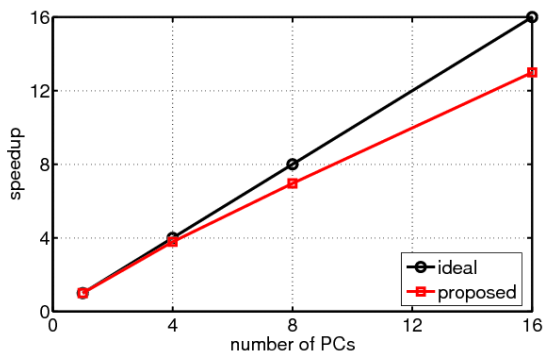


Fig. 4: Speed-up of the proposed parallel algorithm.

The performance of the proposed parallel algorithm was studied according to Speedup and efficiency factors. The speedup was calculated as

$$S(P) = T(1)/T(P) \quad (27)$$

where $T(1)$ is the time needed to solve the problem using one processor and $T(P)$ is the time needed to solve the same problem using P processors. The efficiency was calculated as

$$E(P) = S(P)/P \quad (28)$$

Figs. 4 and 5 show, respectively, the speedup and the efficiency of the proposed parallel algorithm. For the purpose of comparison, the ideal speedup and efficiency were also shown in Figs. 4 and 5. As can be seen from Fig. 4, almost linear speedup was obtained when the parallel code was run on less than four processors. Beyond this, the efficiency of the parallel system decreases. This is due to the fact that as the number of processors increases, the size of each sub-domain will be too small and hence the communication time becomes comparable to the computational time in the sub-domain. It is important to note that the performance of the parallel system can be improved further by using 3-D topology, which involves dividing the computational domain in the x , y , and z -directions [8].

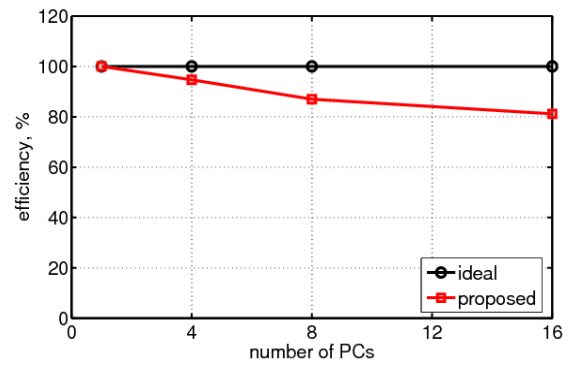


Fig. 5: Efficiency of the proposed parallel algorithm.

Finally, it should be noted that in the above example, the scalability of the proposed parallel algorithm was measured for a fixed problem size. Nevertheless, for some problems, parallel systems can also be used to solve larger problems. For such problems, the performance of the parallel algorithm is measured as the problem size scales proportionally to the number of processors. In this case, the computational problem size is kept constant per processor, while the number of processors increases. In the present study, the sub-domain size is kept fixed at $240 \times 240 \times 40$ per processor. Table II shows the scalability of the proposed parallel algorithm. As can be seen from these results, although the problem size is increased, there is a slight change in the total simulation time, which is due to the communication time between the processors. Hence, the problem size can be increased as the number of processors is increased.

TABLE II: SCALABILITY OF THE PROPOSED PARALLEL ALGORITHM FOR SCALED PROBLEM SIZE.

P	P_x	P_y	N_x	N_y	Computation time
1	1	1	240	240	3814.8
4	2	2	480	480	4083.0
8	4	2	960	480	4131.6
16	4	4	960	960	4275.6

IV. CONCLUSIONS

In this paper, full-wave parallel FDTD algorithm is presented for modeling electromagnetic wave propagation in dispersive open region problems. In the presented work, the problem geometry is divided into non-overlapping sub-domains using the 2-D topologies. It has been observed that the proposed parallel algorithm not only speed up computations but also increases the maximum solvable problem size. It is important to note that the presented formulations can be used for modeling electromagnetic waves interactions with human tissues like mobile phone radiations effect on human head. Finally, it should be noted that the simulations can be accelerated dramatically by using the graphical processing unit

(GPU) and employing the compute unified device architecture (CUDA) parallel programming model [18], and this issue is under investigations.

REFERENCES

- [1] K.S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media", *IEEE Transaction on Antennas and Propagation*, vol. 14, no. 3, pp. 302-307, May 1966.
- [2] A. Taflove and S.C. Hangess, *Computational electrodynamics: The Finite-Difference Time-Domain Method*, 2nd ed., Norwood, MA: Artech-House, 2000.
- [3] J.P. Berenger, "A perfectly matched layer for the absorption of electromagnetic waves", *Journal of Computational Physics*, vol. 114, no. 2, pp. 185-200, Oct. 1994.
- [4] W.C. Chew, and W.H. Weedon, "A 3-D perfectly matched medium from modified Maxwell's equation with stretched coordinates," *Microwave and Optical Technology Letters*, vol. 7, no. 13, pp. 599-604, Sep. 1994.
- [5] S.D. Gedney, "An anisotropic perfectly matched layer absorbing medium for the truncation of FDTD lattices", *IEEE Transactions on Antennas and Propagation*, vol. 44, no. 12, pp. 1630-1639, Dec. 1996.
- [6] S. A. Cummer, "A simple, nearly perfectly matched layer for general electromagnetic media," *IEEE Microwave Wireless Component Letters*, vol. 13, no. 3, pp. 128-130, Mar. 2003.
- [7] W. Gropp, E. Lusk, and A. Skjellum, *Using MPI: Potable parallel Programming with the Message-Passing Interface*, 2nd ed., Cambridge, MA, MIT Press, 1999.
- [8] H. Hoteit, R. Sauleau, B. Philippe, P. Coquet, and J.P. Daniel, "Vector and parallel implementations for the FDTD analysis of millimeter wave planar antennas", *International Journal of High Speed Computing*, vol. 10, no. 2, pp. 209-234, Jun. 1999.
- [9] C. Guiffaut, and K. Mahdjoubi, "A parallel FDTD algorithm using the MPI library", *IEEE Antennas and Propagation Magazine*, vol. 43, no. 2, pp. 94-103, Apr. 2001.
- [10] D.-H. Sheen, K. Tuncay, C.-B. Baag, P. J. Ortoleva, "Parallel implementation of a velocity-stress staggered-grid finite-difference method for 2-D poroelastic wave propagation", *Computers and Geosciences*, vol. 32, no. 8, pp. 1182-1191, Oct. 2006.
- [11] O. Ramadan, "Three Dimensional MPI Parallel Implementation of the PML algorithm for truncating finite-difference time-domain grids," *Parallel Computing*, vol. 33, no. 2, pp. 109-115, Mar. 2007.
- [12] O. Ramadan and O. Akaydin, "Efficient parallel PML algorithms for truncating finite difference time domain simulations," *Electrical Engineering*, vol. 90, no. 3, pp. 175-180, Feb. 2008.
- [13] J.S. Ayubi-Moak, S.M. Goodnick, D. Stanzione, G. Speyer, and P. Sotirelis, "Improved parallel 3D FDTD simulator for photonic crystal", DoD HPCMP Users Group Conference, 14-17 July 2008, Seattle, WA, pp. 319-326.
- [14] J. Li, L.-X. Guo, H. Zeng, and X.-B. Han, "Message-passing-interface-based parallel FDTD investigation on the EM scattering from a 1-D rough sea surface using uniaxial perfectly matched layer absorbing boundary", *Journal of the Optical Society of America A*, vol. 26, no 6, pp. 1494-1502, Jun 2009.
- [15] O. Ramadan, "An Efficient MPI-Based Parallel Wave-Equation FDTD Algorithm for Dispersive Electromagnetic Applications," *The 5th International Conference on Information Technology, (ICIT'11)*, Amman, Jordan, May 11-3, 2011
- [16] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications*, 3rd ed., Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [17] R. M. Joseph, S. C. Hagness, and A. Taflove, "Direct time integration of Maxwell's equations in linear dispersive media with absorption for scattering and propagation of femtosecond electromagnetic pulses," *Optics Letters*, vol. 16, no. 18, pp. 1412-1414, Sep. 1991.
- [18] Nvidia, CUDA C programming guide version 4.0, Nvidia Corporation, Santa Clara, CA, 2011.