Progress In Electromagnetics Research M, Vol. 19, 105–120, 2011

ACCELERATION TECHNIQUE OF FDTD MODEL WITH HIGH ACCURACY FOR NANOSTRUCTURE PHOTONICS

Y. Liu[1, 2, *], C. C. Chen[1], P. Wang[1], and H. Ming[1]

[1]Anhui Key Laboratory of Optoelectronic Science and Technology, University of Science and Technology of China, Hefei 230026, China

[2]School of Electronics and Information Engineering, Anhui University, Hefei 230601, China

Abstract—To accurately model nanophotonic structures, a conformal dispersive finite difference time domain (FDTD) method based on an effective permittivity technique is presented, which can describe exactly the behaviors of evanescent waves in the vicinity of curved interface. A mismatch between the numerical permittivity and the analytical value introduced by the discretization in FDTD is demonstrated, thus, very fine time-step size is always necessary for nanostructures modelling, which greatly increases the required overheads of CPU time as compared to usual FDTD simulations. To resolve this problem, the performance of parallel FDTD code is investigated on a Gigabit Ethernet, and the acceleration technique for parallel FDTD algorithm is presented, which is developed by means of the replicating computation based on overlapping grids, the OpenMP multithreading technique and the vectorization based on SSE instruction. The comparison of relevant numerical results shows that these methods are able to reduce the expense of the system communications and enhance the utilization ratio of the CPU effectively, which improves greatly the performance of parallel FDTD with high time-consuming.

Received 6 June 2011, Accepted 5 July 2011, Scheduled 12 July 2011

* Corresponding author: Yu Liu (ly2008@ustc.edu.cn).

1. INTRODUCTION

Recently, metallic nanostructures have been extensively studied due to their unique plasmonic properties [1–4]. It has been shown that metallic nanostructures can dramatically enhance the local electric field by concentrating electromagnetic energies into the subwavelength volumes that arise from the local surface plasmon resonance (SPR). SPR is a charge-density oscillation that exists at a metal-dielectric interface. The light waves that are trapped on metal surface by coherent oscillation of electrons are called surface Plasmon polaritons (SPPs). SPPs produce evanescent electric field normal to the metal surface and propagating wave along the metal-dielectric interface, which attracted growing interests in the past few years.

Modelling of light propagation in nanostructures leads to a solving of transmission problem for Maxwell's equations. They may be solved with various methods, such as the Discrete Dipole Approximation (DDA) [5], the Finite Element Method (FEM) [6] and Finite Difference Time Domain (FDTD) algorithm [7]. Among these methods, the FDTD algorithm is very popular, and has been used widely from microwave to optical problems [8, 9]. However, its applications in nanophotonic structures may appear challenging, especially for the treatment of the evanescent wave at the metal-dielectric interface. Besides, very fine meshes with respect to the wavelength are required in FDTD model for nanostructures [10], which causes a rapid decrease in the time-step size for the sake of the Courant stability condition. Moreover, to reduce the mismatch between the numerical constitutive parameters and the analytical value in the FDTD model, the tiny time-step size is always necessary. In view of the fact that the simulation of nanostructures is at sub-wavelength scale, and is electrically small problems, the required overhead of CPU time for these calculations increases faster than that of memory capacity. The conventional parallel FDTD method can overcome the bottlenecks of memory requirement effectively, nevertheless, with the quantity of computers increasing, its parallel efficiency deteriorates sharply in the problems with high overheads of time relative to memory requirement [11]. In the circumstances, lots of computing capability of the CPUs is wasted due to low parallel efficiency. Hence it is quite important to accelerate computing for FDTD model of nanostructures with high time-consuming.

This paper discusses the approach to increasing accuracy for modeling nanophotonic structures using the dispersive FDTD algorithm, which can describe well the propagation of the evanescent waves along the metal-dielectric interface of nanostructure. To improve

computational efficiency, the performance of parallel FDTD code is investigated on a Gigabit Ethernet, and the accelerating technique for FDTD computation is presented, which is developed by means of three methods: the replicating computation based on overlapping grids, the OpenMP multithreading technique and the vectorization based on SSE instruction. In order to evaluate the feasibility of the accelerating approach, a subwavelength nanophotonic structure is simulated, and various comparisons are made in different computing conditions to verify the efficiency and accuracy.

2. ACCURATE MODELLING OF NANOPHOTONIC STRUCTURES

For modelling nanostructures with complex material properties, the frequency dispersion has to be taken into account. Although there are various dispersive FDTD methods available, due to its simplicity and efficiency, an auxiliary differential equations (ADE) method with Drude dispersion model is adopted in this paper [12]. The permittivity of metal at optical frequencies is described by the following:

$$\varepsilon(\omega) = \varepsilon_0 \left(1 - \frac{\omega_p^2}{\omega^2 - j\omega\gamma}\right) \tag{1}$$

where ω_p is the plasma frequency and γ is the collision frequency.

As a result of the strong field localisation at the interface, the evanescent waves play key roles in nanostructures and have to be modeled accurately. Usually, the evanescent waves decay exponentially over distances, thus they are concentrated in the close vicinity of the metal-dielectric interface, and will be even more sensitive to the way interfaces are implemented in FDTD. Therefore, special attention should be paid to the treatment of the interface for the FDTD model of nanostructures.

The FDTD method for homogeneous media is second order accurate in space, but the accuracy may be degraded in simulations including a material interface. It has been shown that the effective permittivity technique provides a second order accuracy [13]. In many situations, the average permittivity in the cell surrounding the electric field component node is used for the effective permittivity at the node. For a flat interface, the use of the arithmetic mean for the tangential electric field component node and harmonic mean for the perpendicular electric filed component node has been proposed [14].

For more general cases with a curved interface, conventionally, staircase approximations are often used in an orthogonal FDTD domain. However, the approximated shape introduces spurious

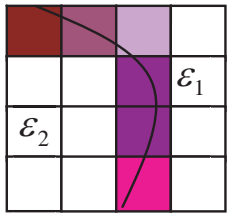

Figure 1. FDTD mesh showing different filling ratios of material for a curved interface.

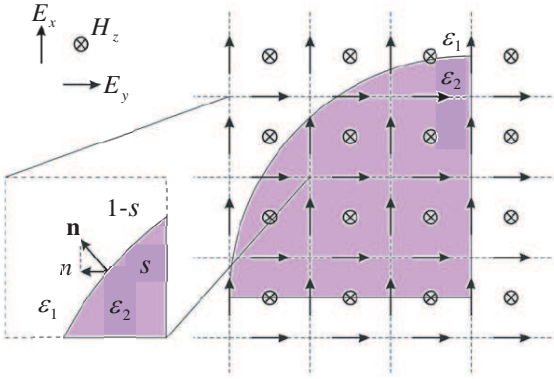

Figure 2. Effective permittivity technique for partially filled cells with curved interface.

numerical resonant modes which do not exist in actual structures, especially if the mesh is coarse and the dielectric contrast is large. This issue has limited the application of the FDTD method when the exact shape and permittivity values are important in determining the electromagnetic response.

To reduce the staircase error while keeping FDTD as simple as possible, using the concept of filling ratio, which is defined as the ratio of the area of material ε_2 to the area of a particular FDTD cell, (volume in three-dimensions), the curvature can be properly represented in FDTD domain as shown in Figure 1, where different colors indicate different filling ratios of material ε_2 for the partially filled cells. The accuracy of modeling can be significantly improved compared with staircase approximations, because much more information on the actual geometry is passed to the FDTD algorithm by using the effective permittivity.

Based on the principle proposed in literature [15], the effective

permittivity in a general form is given by

$$\varepsilon_{eff} = \varepsilon_{||}(1 - n^2) + \varepsilon_{\perp} n^2 \tag{2}$$

where n is the projection of the unit normal vector $\mathbf{n}$ along the field component as shown in Figure 2, $\varepsilon_{||}$ and $\varepsilon_{\perp}$ are parallel and perpendicular permittivities to the material interface, respectively and defined as

$$\varepsilon_{||} = s\varepsilon_2 + (1 - s)\varepsilon_1 \tag{3}$$

$$\varepsilon_{\perp} = [s/\varepsilon_2 + (1 - s)/\varepsilon_1]^{-1} \tag{4}$$

where s is the filling ratio of material ε_2 in a certain FDTD cell.

In order to take into account the frequency dispersion of the material, the electric flux density $\mathbf{D}$ is introduced into standard FDTD updating equations. At each time step, $\mathbf{D}$ is updated directly from $\mathbf{H}$, and $\mathbf{E}$ can be calculated from $\mathbf{D}$ through the following steps. Substitute (3) and (4) into (2) and using the expressions of ε_1 and ε_2 (1), the constitutive relation in the frequency domain reads.

$$\begin{aligned} &\{\omega^4 - 2\gamma j\omega^3 - [\gamma^2 + (1 - s)\omega_p^2]\omega^2 + \gamma(1 - s)\omega_p^2 j\omega\}D \\ &= [\omega^4 - 2\gamma j\omega^3 - (\gamma^2 + \omega_p^2)\omega^2 + \gamma\omega_p^2 j\omega + s(1 - s)(1 - n^2)\omega_p^4]\varepsilon_0 E \end{aligned} \tag{5}$$

Using the inverse Fourier transformation, i.e., $j\omega \to \partial/\partial t$, we obtain the constitutive relation in the time domain as

$$\begin{aligned} &\left\{\frac{\partial^4}{\partial t^4} + 2\gamma\frac{\partial^3}{\partial t^3} + [\gamma^2 + (1 - s)\omega_p^2] \times \frac{\partial^2}{\partial t^2} + \gamma(1 - s)\omega_p^2\frac{\partial}{\partial t}\right\} D \\ &= \left[\frac{\partial^4}{\partial t^4} + 2\gamma\frac{\partial^3}{\partial t^3} + (\gamma^2 + \omega_p^2)\frac{\partial^2}{\partial t^2} + \gamma\omega_p^2\frac{\partial}{\partial t} + s(1 - s)(1 - n^2)\omega_p^4\right]\varepsilon_0 E \end{aligned} \tag{6}$$

Discretizing (6) by using central finite difference operators in time and a central average operator with respect to time [16], we can obtain the updating equation between $\mathbf{E}$ and $\mathbf{D}$ for FDTD. The computations of $\mathbf{H}$ and $\mathbf{D}$ are performed using Yee's standard updating equations in the free space. Note that if the plasma frequency is equal to zero, then (6) reduces to the updating equation in the free space.

In addition to the corrections at the material interface, a few other aspects are important to the accurate FDTD modeling of nanostructures. The properties of the evanescent waves are known to be extremely sensitive to minor changes of material parameters. Since the discretization introduces a mismatch between numerical and analytical permittivity, when modelling nanostructures, especially when the evanescent waves are involved, the FDTD discrete resolution has a significant impact on the accuracy of simulation results [17].

The numerical permittivity for lossy materials, which is derived from the dispersive FDTD method, is as follows:

$$\hat{\varepsilon} = \varepsilon_0 \left[1 - \frac{\omega_p^2 (\Delta t)^2 \cos^2(\omega \Delta t/2)}{2\sin(\omega \Delta t/2)[2\sin(\omega \Delta t/2) - j\gamma \Delta t \cos(\omega \Delta t/2)} \right] \quad (7)$$

It is clear that (7) approaches the analytical permittivity described by the Drude model in (1) when $\Delta t \to 0$, yet for a finite Δt used in an actual simulation, $\hat{\varepsilon}$ presents a slight deviation from analytical value. This small perturbation does not affect the propagating waves significantly. However, a subwavelength nanostructure is critically dependent on the reconstruction of the evanescent waves, and this reconstruction is in turn critically dependent on the material constitutive parameters. A typical example is the degradation of simulated resolution for the super-lens due to small variations (within 0.03%) of numerical material parameters [18]. This means that even a slight mismatch may cause a crucial discrepancy between simulated and theoretical results. Therefore, in order to ensure that the numerical material parameters correspond to the required values, and describe better the evanescent waves in nanostructures, very fine time-steps size for FDTD is always necessary, which greatly increases the required overheads of CPU time as compared to usual FDTD simulations. This makes the acceleration technique growing important in the actual simulation.

3. NUMERICAL EXAMPLE

3.1. Environment of Parallel FDTD Computing

The personal computer (PC) cluster, on which the proposed parallel FDTD will run, consists of 15 PCs. Each computer in the cluster has a Quad-core CPU of AMD Phenom II x4 965 and 4 GB memory, interconnected by a 1000 Mbps high speed Ethernet network, and runs an operation system of Windows. The software support is composed of Microsoft Visual Studio .net 2005 in conjunction with Intel C++ and MPICH.NT 1.2.5. The former includes compilers and debuggers which support OpenMP multithreading and auto-vectorization, the latter provides MPI communication library.

3.2. Simulation of Plasmon Nanowaveguide

Consider a T-junction plasmon waveguide indicated by white circles in Figure 3, which is made of ordered arrays of closely spaced gold nanospheres (GNS). The diameter d and intercenter spacing a of the

GNS are 50 nm and 75 nm, respectively. Surface-plasmon resonances can be excited on each GNS and light can be guided along the array due to near field interactions between adjacent GNSs.

By means of the accurate modeling method discussed above, we use the traditional MPI-FDTD described in literature [19] to investigate the optical response of the T-junction. The excitation pulse is a sine modulated Gaussian pulse, with central angular frequency equal to 3.65×10^{15} rad/s. The spatial mesh size is equal to 2.5 nm. The whole discretized space along x, y, z direction is $272 \times 332 \times 128$, and is divided into eight subspaces by using 2-D domain decomposition $(2 \times 4 \times 1)$.

Figure 3 shows the snapshots of electric field distribution along the T-junction at two successive time steps, respectively.

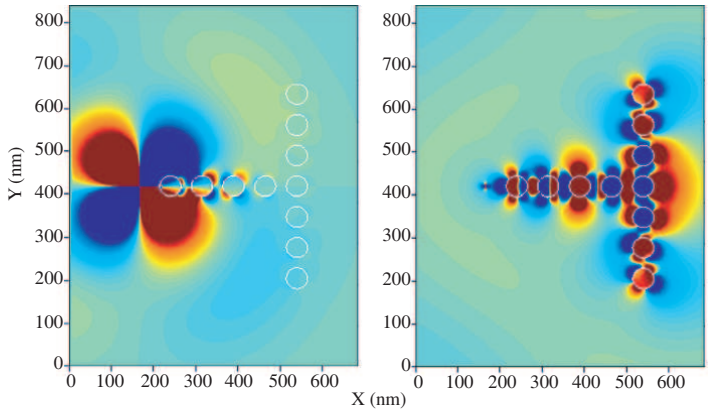

Figure 3. Electric field distribution along a T-junction plasmon waveguide.

Although the excitation wavelength is much larger than the diameter or spacing of the nanospheres, the accurate modelling methods, mentioned above, describe exactly the behaviors of evanescent waves. The plasmon resonances and sub-diffraction light propagation are clearly visible. Light is well confined along the horizontal arm and split into the two vertical arms, which shows a good agreement with the solution in literature [20].

In order to validate the accuracy enhancement of the model, we have also performed calculations using several different methods. Figure 4 shows the comparison of the first resonant frequency of the plasmonic waveguide calculated using different methods. It can be seen, although the adopted spatial mesh ($\Delta = 2.5$ nm) is relatively coarse for the approximation of the actual nanosphere geometry, the

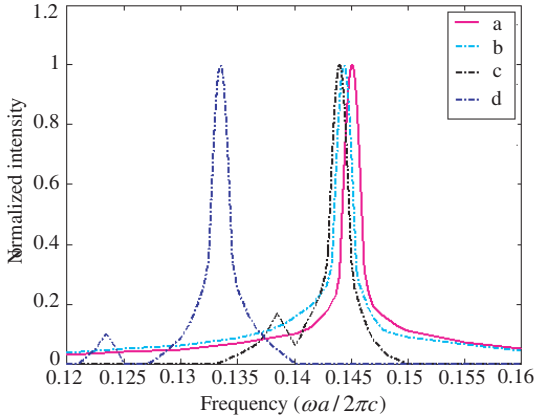

Figure 4. Comparison of the first resonant frequency calculated using different methods, a: dipole model, b: accurate modelling method ($\Delta = 2.5\,\text{nm}$), c: FDTD with the staircase ($\Delta = 1.5625\,\text{nm}$), d: FDTD with the staircase ($\Delta = 2.5\,\text{nm}$).

proposed accurate modelling method shows excellent agreement with the results calculated using the analytical dipole model [21], on the contrary, with the same spatial grids, the FDTD with the staircase approximation not only leads to a shift of the resonant frequency, but also introduces a spurious numerical resonant mode. It is also shown in Figure 4 that although the resonant frequency can be corrected using finer meshes, the spurious resonant mode still remains due to the discontinuity for the treatment of the interface in FDTD with the staircase.

We investigate the performance of parallel codes for the traditional MPI-FDTD. To simplify the implementation of the algorithm, when the quantity of the FDTD subspaces is even number, the two dimensional domain decomposition ($2 \times x \times 1$) is adopted, and when the quantity of the subspaces is odd number, the one dimensional domain decomposition ($1 \times x \times 1$) is adopted. The data transference between subspaces utilizes the standard non-blocking mode based on MPI library.

Computations are performed for different numbers of PCs (from 4 to 15 PCs). Figure 5 shows the benchmarking results of the normalized run time versus the number of PCs. For the fixed size problem in this paper, the traditional MPI-FDTD employing thirteen PCs reaches the parallel speedup limitation. Thus, the computation time, on the contrary, will increase as the employed computers are more in quantity

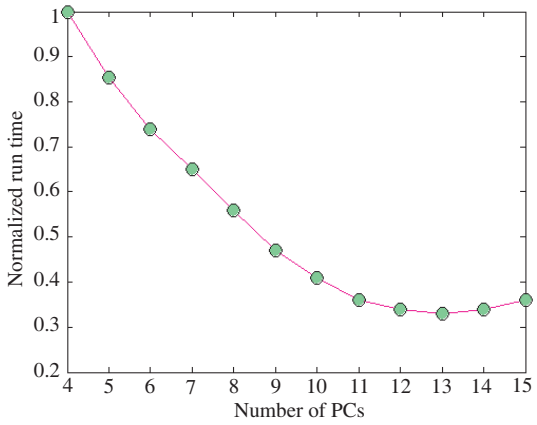

Figure 5. The speedup curve for the traditional MPI-FDTD.

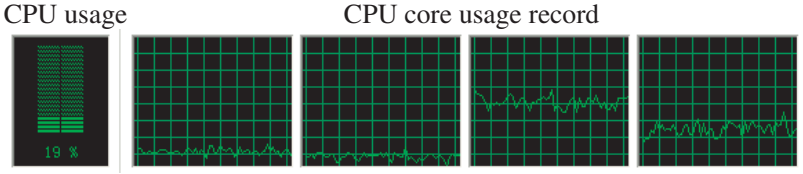

Figure 6. CPU usage for the traditional MPI-FDTD.

than thirteen PCs.

By using Windows Task Manager, we monitor the CPU usage of a computer in the PC cluster when the parallel performance degrades with fifteen PCs employed. It can be seen from Figure 6 that the CPU usage is merely around 20%, and the CPU core usage is under 65%. Lots of computing capability of the CPU is wasted. In order to accelerate FDTD computation, it is quite necessary to enhance the utilization ratio of the CPU for the parallel system.

4. ACCELERATION TECHNIQUE AND ANALYSIS

Usually, the performance degradation for parallel FDTD system is because the PC cluster spends less time computing but essentially more time communicating between subspaces. To increase computational efficiency, we need to solve this problem in two ways: reducing the expense of the system communications and enhancing the utilization ratio of the CPU.

4.1. Replicating Computation Based on Overlapping Grids

Compared with the commodity cluster, the PC cluster based on the Ethernet network has larger communication delays. If only a layer of field components at the subspace boundaries are transmitted, the communication time for the message latency dominates relative to the CPU time when the size of the subspace is decreased to the critical point. We can reduce number of communications by replicating computations on overlapping grids of neighbouring subspaces. If sending two or three layers instead of one, we can advance simulation two or three time-steps before another communication, as shown in Figure 7. Since the bandwidth is far better than the short message latency for Ethernet, this scheme can decrease communication overheads availably.

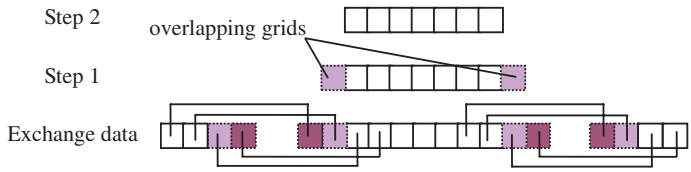

Figure 7. Principle of replicating computation for parallel FDTD.

Figure 8 shows the benchmarking results of the MPI-FDTD with replicating computations. The speedup curve maintains still a relatively stable drop when the number of PCs is fifteen. This is because fewer numbers of communications lessens the influence of the network overheads, which brings about the result that the accelerating ability depends on, to a great extent, the quantity of PCs operated. Since the CPU is less idling for waiting for the completion of communications, the utilization ratio of the CPU is enhanced, as shown in Figure 9, the CPU core usage increases to around 90%, when using the same fifteen PCs.

4.2. OpenMP Multithreading Technique

Utilizing the replicating computation technique, we have enhanced the CPU core usage availably, yet the utilization ratio of the CPU is low (about 30%), and three of the four CPU-cores are almost idle. This happens because the parallel program based the MPI-FDTD goes still single-threaded, which can not give directly support to the multi-core CPUs. In order to exploit the multi-core parallelism, OpenMP is used in conjunction with MPI, which is a parallel programming standard for shared-memory mode, and consists of a set of compiler directives and a

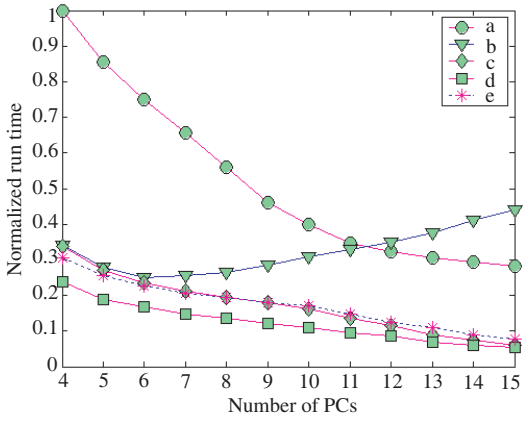

Figure 8. The speedup curve in different circumstances, a: MPI-FDTD with replicating computations, b: MPI-OpenMP FDTD without replicating computations, c: MPI-OpenMP FDTD with replicating computations, d: Auto-vectorized FDTD, e: Vectorized FDTD with incomplete data alignment.

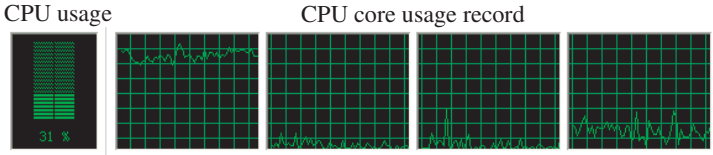

Figure 9. CPU usage for the MPI-FDTD with replicating computations.

library of support functions. In previous studies, we have implemented this hybrid algorithm [19]. Its principle is to insert OpenMP compiler directives (**#pragma omp parallel for**) into the nested loops which update EM fields in the traditional MPI-FDTD program. The loop iterations are split among threads, and go multithreaded. Thus, the computing time of the FDTD subspace is reduced.

Figure 8 shows the benchmarking results of the MPI-OpenMP hybrid algorithm. A noteworthy fact is that the algorithm without replicating computations, employing merely six PCs, reaches the parallel speedup limitation. This happens because the multithreading technique can further reduce the computational time of each subspace, whereas it cannot lower the communication overheads between subspaces, which cause the parallel program to reach more rapidly the speedup limitation. Figure 10 and Figure 11 show the CPU

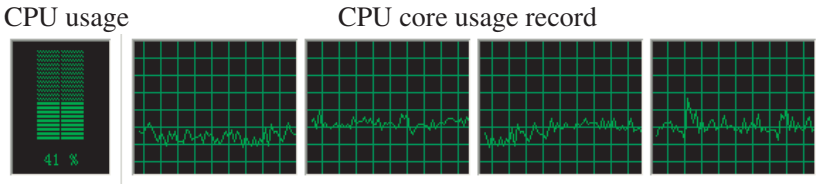

Figure 10. CPU usage for the MPI-OpenMP FDTD without replicating computations.

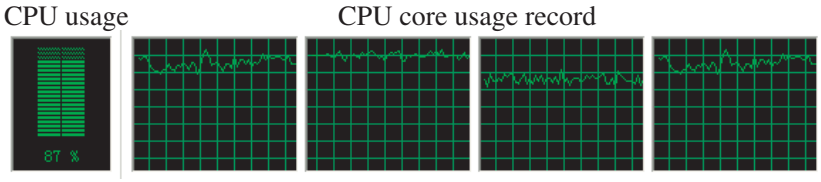

Figure 11. CPU usage for the MPI-OpenMP FDTD with replicating computations.

usage in two situations. Although the algorithm without replicating computations goes multithreaded, the utilization ratio of the CPU is still low (under 45%) due to high communication overheads. In order to accelerate FDTD computation, the parallel system has to match the CPU capability with the communication overheads. It can be seen from Figure 11, due to reducing number of communications for the algorithm with replicating computations, the CPUs work almost at full load.

4.3. Vectorization Based on SSE Instruction

By using two techniques mentioned above, the utilization ratio of the CPUs increases to around 90%. However, in FDTD simulation, the vast majority of run time will be spent in the "for-loop" procedure. We need to optimize further the FDTD loop to reduce the overheads from these complex loop structures.

Vectorization is the technique used to optimize loop performance on the x86-CPUs, which introduce the 128 bit streaming SIMD (Single-Instruction-Multiple-Data) extensions (SSE) registers inside CPU, supporting floating point operations on four single-precision or two double-precision numbers in one step [22]. Because a single SSE instruction processes multiple data elements in parallel, all these extensions are very useful to the parallel FDTD applications. Recently,

as an advanced version of SSE, Advanced Vector Extensions (AVX) is released by Intel Corporation in 2011. Intel AVX is a new 256 bit instruction set extension to SSE, and greatly improves the performance of the vectorized applications due to wider vectors, new extensible syntax and rich functionality.

Figure 12 shows the serial execution of the simplified FDTD iterations on the left without vectorization where the lower parts of the SIMD registers are not utilized. The vectorized version on the right shows four components of the A and B arrays are added in parallel for each iteration utilizing the full width of the SIMD registers.

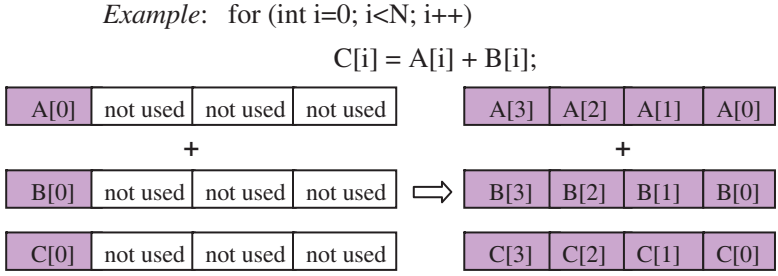

Figure 12. Addition of 128 bit SIMD registers (A, B and C).

To get top performance for vectorized codes, the SSE instructions demand that the address of data should be aligned in memory. This is a very tedious work. Fortunately, the modern compilers provide a set of compiler directives for data alignment, and it is easy to implement. For example, Intel C++ can achieve auto-vectorization by virtue of the compiler directive "/arch:SSE2".

Auto-vectorization can be used in conjunction with other parallelization techniques such as OpenMP discussed earlier. Due to favorable parallelization features in the FDTD loops, the auto-vectorized version of FDTD can obtain obvious performance improvement. In most cases, the performance increases over 30% relative to that without vectorization, as shown in Figure 8. However, data alignment has an important effect on the performance of the vectorized codes. We achieve the incomplete data alignment manually by using Intel C++ libraries for the same codes, and it can be seen from Figure 8 that the performance is hardly increased, even drops somewhat. Therefore, compared with other methods for data alignment, auto-vectorization is easy to achieve, and more recommendable. Moreover, it is worth noting that the great acceleration potential can be got by virtue of full use of CPU's

computing power. Figure 8 shows that if four PCs are exploited fully, the acceleration capability exceeds even fifteen PCs where the traditional MPI-FDTD is performed. In view of the fact that a small-scale parallel system is more easily operated, it is important to enhance the utilization ratio of the CPU and reduce the scale of parallel system for a fixed size problem.

5. CONCLUSIONS

We have developed a conformal dispersive FDTD method for accurately modelling nanophotonic structures. The conformal scheme is based on the effective permittivity technique. Since the traditional orthogonal FDTD grid is maintained in simulations, its main advantage is to keep FDTD as simple as possible while improving the modeling accuracy. Moreover, to reduce the mismatch between the numerical permittivity and the analytical value in the FDTD model, we have demonstrated that the tiny time-step size is always necessary for nanostructures modeling. The numerical simulations for the plasmon nanowaveguide show that the accurate modeling methods can describe well the propagation of the evanescent waves in the vicinity of curved interface. However, the FDTD model with high accuracy means higher overhead of CPU time, in order to accelerate the parallel FDTD computation, we have solved this problem by using three schemes: the replicating computation based on overlapping grids, the OpenMP multithreading technique and the vectorization based on SSE instruction. The comparison of relevant numerical results shows that these methods are able to reduce the expense of the system communications and enhance the utilization ratio of the CPU effectively.

Since the multi-core CPUs which support directly multithreading and vectorization from hardware are widely employed on modern PCs, the proposed acceleration techniques are increasingly valuable to improve the efficiency of the parallel FDTD. Furthermore, with the gradual improvement of the hardware acceleration techniques for Graphics Processing Units (GPU), it would be interesting to extend the CPU computing to the CPU-GPU collaborative computing in the future, which can accelerate the FDTD simulations more strongly.

ACKNOWLEDGMENT

This work was supported by the National Key Basic Research Program of China No. 2011CB301802 and China Postdoctoral Science Foundation No. 20100470851.

REFERENCES

1. Andrew, T. L., H.-Y. Tsai, and R. Menon, "Confining light to deep subwavelength dimensions to enable optical nanopatterning," *Science*, Vol. 324, 917–921, 2009.
2. Suyama, T. and Y. Okuno, "Surface plasmon resonance absorption in a multilayered thin-film grating," *Journal of Electromagnetic Waves and Applications*, Vol. 23, No. 13, 1773–1783, 2009.
3. Politano, A., R. G. Agostino, E. Colavita, et al., "Electronic properties of self-assembled quantum dots of sodium on Cu(111) and their interaction with water," *Surface Science*, Vol. 601, 2656–2659, 2007.
4. Politano, A., R. G. Agostino, E. Colavita, et al., "High resolution electron energy loss measurements of Na/Cu(111) and H_2O/Na/Cu(111): Dependence of water reactivity as a function of Na coverage," *The Journal of Chemical Physics*, Vol. 126, 244712, 2007.
5. Draine, B. T. and P. J. Flatau, "Discrete-dipole approximation for periodic targets: theory and tests," *Journal of Optical Society of America A*, Vol. 25, No. 11, 2693–2703, 2008.
6. Jiao, D. and J. M. Jin, "Time-domain finite-element modeling of dispersive media," *IEEE Microwave and Wireless Components Letters*, Vol. 11, No. 5, 220–222, 2001.
7. Liu, Y., Z. Liang, and Z. Yang, "Computation of electromagnetic dosimetry for human body using parallel FDTD algorithm combined with interpolation technique," *Progress In Electromagnetics Research*, Vol. 82, 95–107, 2008.
8. Garcia, S. G., F. Costen, M. Fernandez Pantoja, L. D. Angulo, and J. Alvarez, "Efficient excitation of waveguides in Crank-Nicolson FDTD," *Progress In Electromagnetics Research Letters*, Vol. 17, 27–38, 2010.
9. Zhang, Y.-Q. and D.-B. Ge, "A unified FDTD approach for electromagnetic analysis of dispersive objects," *Progress In Electromagnetics Research*, Vol. 96, 155–172, 2009.
10. Mohammadi, A. and M. Agio, "Dispersive contour-path finite-difference time-domain algorithm for modelling surface plasmon polaritons at flat interfaces," *Optics Express*, Vol. 14, No. 23, 11330–11338, 2006.
11. Wang, A.-Q., L.-X. Guo, and C. Chai, "Numerical simulations of electromagnetic scattering from 2D rough surface: Geometric modeling by nurbs surface," *Journal of Electromagnetic Waves*

and Applications, Vol. 24, No. 10, 1315–1328, 2010.

12. Wei, B., S.-Q. Zhang, Y.-H. Dong, and F. Wang, "A general FDTD algorithm handling thin dispersive layer," *Progress In Electromagnetics Research B*, Vol. 18, 243–257, 2009.
13. Hirono, T., Y. Yoshikuni, and T. Yamanaka. "Effective permittivities with exact second-order accuracy at inclined dielectric interface for the two-dimensional finite-difference time-domain method," *Applied Optics*, Vol. 49, No. 7, 1080–1096, 2010.
14. Hwang, K.-P. and A. C. Cangellaris, "Effective permittivities for second-order accurate FDTD equations at dielectric interfaces," *IEEE Microwave and Wireless Components Letters*, Vol. 11, No. 4, 158–160, 2001.
15. Mohammadi, A., H. Nadgaran, and M. Agio, "Contour-path effective permittivities for the two-dimensional finite-difference time-domain method," *Optics Express*, Vol. 13, No. 25, 10367–10381, 2005.
16. Taflove, A. and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method,* 3nd Edition, Artech House, Norwood, MA, 2005.
17. Zhao, Y., P. Belov, and Y. Hao, "Accurate modelling of left-handed metamaterials using a finite-difference time-domain method with spatial averaging at the boundaries," *Journal of Optics A: Pure and Applied Optics*, Vol. 9, S468–S475, 2007.
18. Chen, J. J., T. M. Grzegorczyk, B.-I. Wu, and J. A. Kong, "Limitation of FDTD in simulation of a perfect lens imaging system," *Optics Express*, Vol. 13, No. 26, 10840–10845, 2005.
19. Liu, Y., Z. Liang, and Z. Q. Yang, "A novel FDTD approach featuring two-level parallelization on PC cluster," *Progress In Electromagnetics Research*, Vol. 80, 393–408, 2008.
20. Teixeira, F. L., "Time-domain finite-difference and finite-element methods for Maxwell equations in complex media," *IEEE Transactions on Antennas and Propagation*, Vol. 56, No. 8, 2150–2166, 2008.
21. Brongersma, M. L., J. W. Hartman, and H. A. Atwater, "Electromagnetic energy transfer and switching in nanoparticle chain arrays below the diffraction limit," *Physical Review B*, Vol. 62, No. 24, R16356–R16359, 2000.
22. Bik, A., M. Girkar, P. Grey and X.-M. Tian, "Efficient exploitation of parallelism on Pentium III and Pentium 4 processor-based systems," *Intel Technology Journal*, Q1, 1–9, 2001.