Opt Quant Electron (2012) 44:285–290
DOI 10.1007/s11082-012-9558-z

B-CALM: An open-source GPU-based 3D-FDTD with multi-pole dispersion for plasmonics

Pierre Wahl · Dany-Sebastien Ly-Gagnon · Christof Debaes · David A. B. Miller · Hugo Thienpont

Received: 30 September 2011 / Accepted: 25 January 2012 / Published online: 11 February 2012

Abstract Numerical calculations with finite-difference time-domain (FDTD) on metallic nanostructures in a broad optical spectrum require an accurate approximation of the permittivity of dispersive materials. In this paper, we present the algorithms behind B-CALM (Belgium-California Light Machine), an open-source 3D-FDTD solver operating on Graphical Processing Units with multi-pole dispersion models. Our modified architecture shows a reduction in computing times for multi-pole dispersion models. We benchmark B-CALM by computing the absorption efficiency of a metallic nanosphere on a broad spectral range with a six-poles Drude-Lorentz model and compare it with Mie theory.

Keywords FDTD · GPU · Dispersive · Multi-pole · Plasmonic

1 Introduction

Finite-difference time-domain (FDTD) simulations play a prominent role in numerical electromagnetic calculations (Taflove et al. 1995). Many problems in nanophotonics require three dimensional full-field simulations. When applying optimization schemes that require a full-field solution at each iteration step, such as genetic algorithms (Gondarenko and Lipson 2008) or adjoint optimization methods (Jensen and Sigmund 2005), the ability to find a solution is often limited by the speed of the full-field simulator. Also FDTD simulations are often limited by the available computational power. While the use of Graphical Processing Units (GPUs) to accelerate FDTD simulation have been reported before (Zunoubi 2010; Zunoubi et al. 2010), the results have been, to our knowledge, so far limited to the implementation of flat or single pole Drude dispersion material models in the microwave regime.

P. Wahl (✉) · C. Debaes · H. Thienpont
Brussels Photonics Team, Applied Physics and Photonics Department,
Vrije Universiteit Brussels, Pleinlaan 2, Brussels, Belgium
e-mail: pwahl@b-phot.org

P. Wahl · D.-S. Ly-Gagnon · D. A. B. Miller
Department of Electrical Engineering, Stanford University, Stanford, CA 940305, USA

At optical frequencies, the permittivity of metals can have more intricate features, which requires including multiple resonance to obtain an accurate material model (Palik and Ghosh 1998). Moreover, to our knowledge, no GPU-enabled FDTD simulator has been shared with the scientific community under an open source license. In this paper, we present an open-source GPU-based FDTD simulator called B-CALM that implements an algorithm to simulate multi-pole Drude-Lorentz materials and minimize thread divergence, enabling fast simulations of complex materials. As an example, we use B-CALM to simulate the absorption cross section of a gold nanosphere and compare the results with Mie theory. Compared with Mie theory, we obtain an error of less than 5% on a broad spectral range and an overall 40X speedup compared to Meep, a widely spread CPU-based FDTD simulator (Oskooi et al. 2010).

2 Drude–Lorentz model for dispersive media

A complex permittivity $\epsilon(\omega) = \epsilon'(\omega) + i\epsilon''(\omega)$ can be approximated over a broad range of wavelengths using the *Drude–Lorentz* multi-pole approximation. The permittivity of the dispersive material is then modeled as the sum of the spectral response of several damped harmonic oscillators or *poles*, following expression (1).

$$\epsilon_{DL}(\omega) = \epsilon_\infty + \sum_{m=0}^{P} \frac{{\omega_{pm}}^2}{\omega_m^2 - \omega^2 + i\omega\Gamma_m} \tag{1}$$

As the amount of poles P is increased, $\epsilon_{DL}(\omega)$ can be approximated more accurately and over a larger bandwidth. For example, to fit the relative permittivity of gold between wavelengths 300 and 1,200 nm, 6 poles are needed (Rakic et al. 1998). The exact fitting values of ω_m, ω_{pm} and Γ_m are tabulated in Table 1 and will be used throughout the rest of this paper.

3 Description of the GPU architecture

GPUs contain several hundreds of cores that can allow for massive parallization. The main difference with CPUs is that GPUs typically exploit the *same instruction multiple data* (SIMD) computer architecture (Nvidia 2009). This implies that all the cores execute identical commands at each clock cycle, but on data stored at different memory addresses. The instruction set is referred to as a *kernel*. So, when a kernel is executed, all the cores execute the same instruction but on different data in the GPU RAM (random access memory). The execution speed of FDTD simulations on GPUs is often limited by *memory bandwidth* and *divergence* (Nvidia 2009).

Table 1 Exact fitting values of ω_m, ω_{pm} and Γ_m for gold between the wavelengths of 300 and 1,200 nm (Rakic et al. 1998)

Parameter	$m = 0$	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$
$\omega_m \times 10^{16}$	0	0.0630	0.1261	0.4510	0.6538	2.0235
$\omega_{pm} \times 10^{16}$	1.1959	0.2125	0.1372	0.3655	1.0634	2.8722
$\Gamma_m \times 10^{15}$	0.0805	0.3661	0.5241	1.3216	3.7887	3.3633

Divergence occurs when, within a kernel, different threads have to perform different instructions, e.g., through an *if*-statement. Both paths of the if-statement are then performed *sequentially*. This slows the program down significantly as the cores assigned to one path of the if-statement are busy waiting while the cores assigned to the other path (Nvidia 2009).

The *memory bandwidth* on a GPU is strongly related to the *coalescence* of the memory calls. Coalesced memory calls occur when cores request data that are co-located on the GPU-RAM (Nvidia 2009). If this is not the case, the program is significantly slowed down, as non-coalesced memory calls are also performed sequentially.

4 Mapping the FDTD algorithm on GPUs

The FDTD algorithm allows the calculation of propagating electromagnetic waves by alternately calculating the discretized electric and magnetic fields using a first-order spatial and temporal difference equation of Maxwell's equations (Taflove et al. 1995). For non-magnetic materials, the electric field update equation in a cell with the permittivity described by Eq. (1) is

$$\begin{aligned}\mathbf{E}^{n+1} = & \underbrace{\mathbf{E}_s + C_1\mathbf{E}^n + C_2\nabla\times\mathbf{H}^{\mathbf{n+1/2}}}_{<<<\text{Update}E_A>>>} \\ & + C_{pml}\underbrace{\left(\boldsymbol{\Psi}^{\boldsymbol{n}}_{\mathbf{E}_{\perp 1}} + \boldsymbol{\Psi}^{\boldsymbol{n}}_{\mathbf{E}_{\perp 2}}\right)}_{<<<\text{Update}E_A>>>} \\ & \underbrace{- C_3\mathbf{E}^{n-1} + \frac{1}{2}\sum_{p=1}^{P}\alpha_p\mathbf{J}^{\mathbf{n}}_{\mathbf{p}} + \xi_p\mathbf{J}^{\mathbf{n-1}}_{\mathbf{p}}}_{<<<\text{Update}E_B>>>},\end{aligned} \tag{2}$$

where n denotes the time-step, $\mathbf{E}$ the electric field, $\mathbf{E_s}$ the source term and $\mathbf{H}$ the magnetic field. C_{pml} is a scaling constant specific to the *Perfectly Matched Layer (PML)* while $\boldsymbol{\Psi}^{\boldsymbol{n}}_{\mathbf{E}_{\perp 1}}$ and $\boldsymbol{\Psi}^{\boldsymbol{n}}_{\mathbf{E}_{\perp 2}}$ are recursive accumulators only stored in the PML regions (Taflove et al. 1995). $C_1, C_2, C_3, \alpha_p, \xi_p$ are material-specific parameters and C_3, α_p, ξ_p are only used in dispersive materials. Finally, $\mathbf{J}^{\mathbf{n}}_{\mathbf{p}}$ and $\mathbf{J}^{\mathbf{n-1}}_{\mathbf{p}}$ denote recursive accumulators only stored for the electric field of dispersive materials.

The calculation of $\mathbf{E}^{\mathbf{n}}$ and $\boldsymbol{\Psi}^{\boldsymbol{n+1}}_{\mathbf{E}_{\perp 1,2}}$ requires $\boldsymbol{\Psi}^{\boldsymbol{n}}_{\mathbf{E}_{\perp 1,2}}$ and $\mathbf{H}$ fields of neighboring cells as illustrated in Fig. 1a. In contrast, the calculation of $\mathbf{J}^{\mathbf{n+1}}_{\mathbf{p}}$ requires $\mathbf{J}^{\mathbf{n}}_{\mathbf{p}}$, $\mathbf{J}^{\mathbf{n-1}}_{\mathbf{p}}$, $\mathbf{E}^n$ and $\mathbf{E}^{\mathbf{n-1}}$, which are associated to the calculated cell only and no fields associated to neighboring cells, as described in Fig. 1b.

4.1 Smart use of memory to minimize bandwidth

To minimize memory transfers, the material parameters $C_1, C_2, C_3, \alpha_p, \xi_p$, that remain constant throughout the simulation, are stored in a fast read-only *texture memory*. The slower and large *device memory* is only used to store the fields $\mathbf{E}$, $\mathbf{H}$ and the recursive accumulators for the C_{PML}'s and the dispersive materials. Also, fast shared memory is used as described in Fig. 1a and references (Zunoubi 2010; Micikevicius 2009), so that the electromagnetic fields in neighboring cells are only loaded once per update and that those loads are coalesced, such that multiple fields can be loaded on the same cycle which allows for a more efficient use of the memory bandwidth.

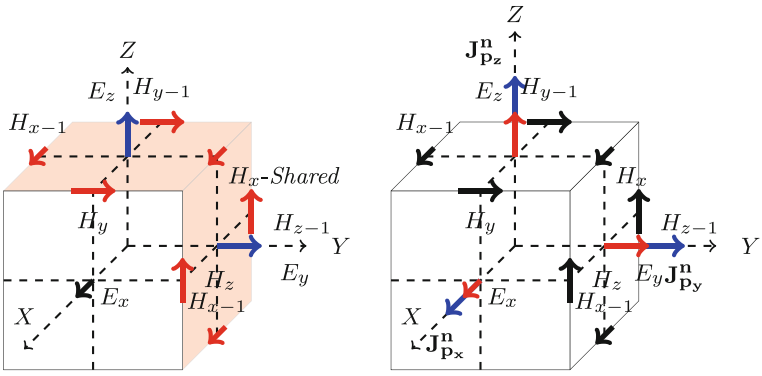

Fig. 1 Data needed for the update of a dispersive cell. **a** The field update of the electric field **E** and $\mathbf{\Psi^{n+1}}_{\mathbf{E}_{\perp 1,2}}$ (in *blue*) requires the neighboring **H** fields (in *red*) to be loaded. Some of those fields are shared. **b** The update of $\mathbf{J^n_p}$ (in *blue*) requires no knowledge of the neighboring fields; only of the electric field in the same direction (in *red*). (Color figure online)

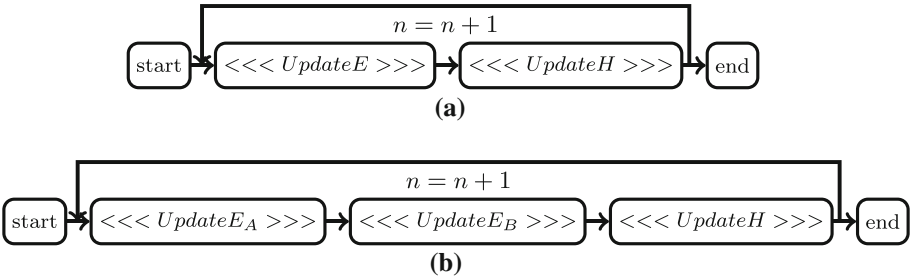

Fig. 2 Comparison of the original approach with the split kernel approach. **a** Original approach with two kernels. **b** Modified approach with split of kernels

4.2 An additional kernel to alleviate thread divergence

In previous GPU implementations of FDTD (Zunoubi 2010), the update equations are split into two kernels, one for the electric and one for the magnetic field update equation as illustrated in Fig. 2a. However, as $\mathbf{J^n_p}$ only needs to be updated for dispersive materials, the electric field update creates a slow diverging path, since, in general, not all cells are dispersive. The situation is exacerbated for materials with a higher number of poles, as the calculation is done sequentially for the diverging paths. To alleviate this issue, B-CALM separates the electric field update equation into two separate kernels (labeled <<<UpdateE$_A$>>> and <<<UpdateE$_B$>>> in Fig. 2b). The cost per field is one extra read and write operation to the device memory to store the intermediate result as both kernels are called sequentially. As <<<UpdateE$_B$>>> is responsible for $2(P + 1)$ read-write operations per field, [1] a split kernel is faster as soon as the number of poles $P > 1$. Also, we expect complicated models for non-linear effects that usually do not need neighboring fields in their updates to perform significantly better on GPUs, by adopting the splitting of the kernel in their update.

[1] This corresponds to $\mathbf{E}^n$, $\mathbf{E^{n-1}}$ for each field and $\mathbf{J^n_p}$, $\mathbf{J^{n-1}_p}$ per pole per field.

5 Results

We used B-CALM to calculate the absorption efficiency of a metallic sphere of 80 nm diameter under plane wave illumination and compare it with Mie theory (Ishimaru 1999) using the modeled values of the permittivity. We used a uniform mesh of 0.5 nm and 208 × 208 × 400 cells with two symmetry planes and a 15-cell wide PML. We calculate the absorption efficiency of the sphere by first integrating the Poynting vector on each face of a closed box surrounding the sphere. Then we subtract the output flux from the input flux through the box surrounding the sphere.

As shown in Fig. 3, the absorption efficiency obtained using B-CALM matches its analytical value obtained by Mie theory with a relative error below 5% over the entire simulated spectrum. The simulation speed is 1.6*e*10 cells/min for a six-poles dispersion model on an NVIDIA C-2075 GPU, which is 40 times faster than with Meep (Oskooi et al. 2010) on a Intel(R) Xeon(R) CPU X5650 processor. In Fig. 4 the speed difference between Meep and B-CALM is compared for different simulation spaces. The speedup is somewhat reduced for smaller grid sizes as the parallelization overhead becomes dominant, but B-CALM remains 3 times faster even for very small grids.

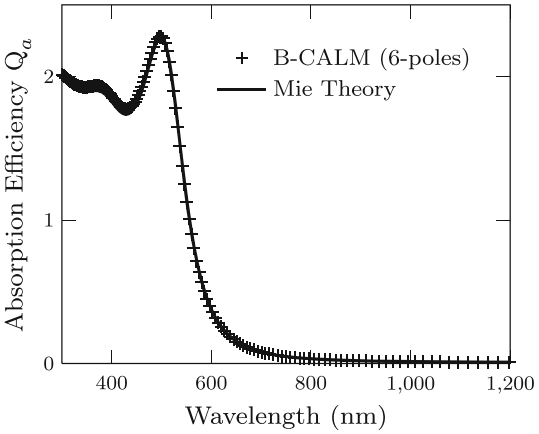

Fig. 3 Simulated absorption efficiency of gold nanosphere of 80 nm diameter. The simulated absorption cross section matches the theoretical value over a broad wavelength range

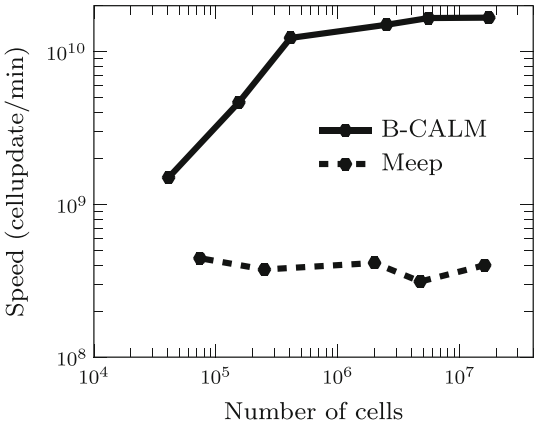

Fig. 4 Simulation speed for a metallic nanosphere on different grid sizes

6 Conclusion

We have shown that dispersive materials with complex wavelength dependence can be accurately simulated using GPU-based FDTD while preserving the speed and the low-cost advantage of GPUs. B-CALM, our GPU-accelerated open-source 3D-FDTD simulator allows us to quickly simulate complex metal structures over a broad wavelength range. B-CALM currently supports any user-defined sources, variable grid size, PML and is structured to easily allow the implementation of non-linearity and anisotropy. B-CALM and its user-friendly interface can be freely downloaded at http://b-calm.sf.net.

Acknowledgments The authors acknowledge the support of the Belgian American Education Foundation, the Methusalem and Hercules Foundation and the Interconnect Focus Center, one of six research centers funded under the Focus Center Research Program (FCRP), a Semiconductor Research Corporation entity.

References

Gondarenko, A., Lipson, M.: Low modal volume dipole-like dielectric slab resonator. Opt. Express **16**(22), 17689–17694 (2008)

Ishimaru, A.: Wave Propagation and Scattering in Random Media. Wiley-IEEE Press, New York, NY (1999)

Jensen, J.S., Sigmund, O.: Topology optimization of photonic crystal structures: a high-bandwidth low-loss T-junction waveguide. JOSA B **22**(6), 1191–1198 (2005)

Micikevicius, P.: In: Proceedings of 2nd Workshop on General Purpose Processing on Graphics Processing Units, pp. 79–84. ACM (2009)

Nvidia, C.: C Programming Guide v3.2. Nvidia Corp (2010)

Oskooi, A.F., Roundy, D., Ibanescu, M., Bermel, P., Joannopoulos, J.D., Johnson, S.G.: MEEP: A flexible free-software package for electromagnetic simulations by the FDTD method. Comput. Phys. Commun. **181**(3), 687–702 (2010). doi:10.1016/j.cpc.2009.11.008

Palik, E.D., Ghosh, G.: Handbook of Optical Constants of Solids. Academic press, New York, NY (1998)

Rakic, A.D., Djurišic, A.B., Elazar, J.M., Majewski, M.L.: Optical properties of metallic films for vertical-cavity opto- electronic devices. Applied Optics **37**(22), 5271–5283 (1998)

Taflove, A., Hagness, S.C., et al.: Computational Electrodynamics: The Finite-Difference Time-Domain Method. Artech House, Norwood, MA (1995)

Zunoubi, M.R.: Analysis of 3-dimensional electromagnetic fields in dispersive media using CUDA. Prog. Electromagn. Res. **16**, 185–196 (2010)

Zunoubi, M.R., Payne, J., Roach, W.P.: CUDA implementation of TEz-FDTD solution of Maxwell's equations in dispersive media. Antennas Wirel. Propag. Lett. IEEE **9**, 756–759 (2010)