Stretched-coordinate PMLs for Maxwell's equations in the discontinuous Galerkin time-domain method

Michael König,[*] Christopher Prohm, Kurt Busch, and Jens Niegemann

Institut für Theoretische Festkörperphysik and DFG-Center for Functional Nanostructures (CFN), Karlsruhe Institute of Technology (KIT), Germany

[*] *mkoenig@tfp.uni-karlsruhe.de*

Abstract: The discontinuous Galerkin time-domain method (DGTD) is an emerging technique for the numerical simulation of time-dependent electromagnetic phenomena. For many applications it is necessary to model the infinite space which surrounds scatterers and sources. As a result, absorbing boundaries which mimic its properties play a key role in making DGTD a versatile tool for various kinds of systems. Popular techniques include the Silver-Müller boundary condition and uniaxial perfectly matched layers (UPMLs). We provide novel instructions for the implementation of stretched-coordinate perfectly matched layers in a discontinuous Galerkin framework and compare the performance of the three absorbers for a three-dimensional test system.

OCIS codes: (000.3860) Mathematical methods in physics; (000.4430) Numerical approximation and analysis.

References and links

1. A. Taflove and S. C. Hagness, *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, 3rd ed. (Artech House, 2005).
2. J. S. Hesthaven and T. Warburton, "Nodal high-order methods on unstructured grids–I. Time-domain solution of Maxwell's equations," J. Comput. Phys. **181**, 186–221 (2002).
3. J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods—Algorithms, Analysis, and Applications* (Springer, 2007).
4. T. Lu, P. Zhang, and W. Cai, "Discontinuous Galerkin methods for dispersive and lossy Maxwell's equations and PML boundary conditions," J. Comput. Phys. **200**, 549–580 (2004).
5. J. Niegemann, M. König, K. Stannigel, and K. Busch, "Higher-order time-domain methods for the analysis of nano-photonic systems," Photonics Nanostruct. Fundam. Appl. **7**, 2–11 (2009).
6. M. König, K. Busch, and J. Niegemann, "The discontinuous Galerkin time-domain method for Maxwell's equations with anisotropic materials," Photonics Nanostruct. Fundam. Appl. **8**, 303–309 (2010).
7. A. Hille, R. Kullock, S. Grafström, and L. M. Eng, "Improving nano-optical simulations through curved elements implemented within the discontinuous Galerkin method computational," J. Comput. Theor. Nanosci. **7**, 1581–1586 (2010).
8. N. Feth, M. König, M. Husnik, K. Stannigel, J. Niegemann, K. Busch, M. Wegener and S. Linden, "Electromagnetic interaction of split-ring resonators: the role of separation and relative orientation," Opt. Express **18**, 6545–6554 (2010).
9. J. Niegemann, W. Pernice, and K. Busch, "Simulation of optical resonators using DGTD and FDTD," J. Opt. A: Pure Appl. Opt. **11**, 114015 (2009).
10. T. Hagstrom and S. Lau, "Radiation boundary conditions for Maxwell's equations: A review of accurate time-domain formulations," J. Comput. Math. **25**, 305–336 (2007).
11. J. P. Bérenger, "A perfectly matched layer for the absorption of electromagnetic waves," J. Comput. Phys. **114**, 185–200 (1994).

#139146 - $15.00 USD Received 3 Dec 2010; revised 17 Jan 2011; accepted 21 Jan 2011; published 24 Feb 2011

12. Z. S. Sacks, D. M. Kingsland, R. Lee, and J.-F. Lee, “A perfectly matched anisotropic absorber for use as an absorbing boundary condition,” IEEE Trans. Antennas Propag. **43**, 1460–1463 (1995).
13. W. C. Chew and W. H. Weedon, “A 3D perfectly matched medium from modified maxwell’s equations with stretched coordinates,” Microwave Opt. Technol. Lett. **7**, 599–604 (1994).
14. J.-P. Bérenger, *Perfectly Matched Layer (PML) for Computational Electromagnetics* (Morgan & Claypool Publishers, 2007).
15. P. Monk, *Finite Element Methods for Maxwell’s Equations* (Oxford University Press, 2003).
16. M. H. Carpenter and C. A. Kennedy, “Fourth-order 2N-storage Runge–Kutta schemes,” NASA Tech. Memo. 109112 (1994).
17. R. Diehl, K. Busch, and J. Niegemann, “Comparison of low-storage Runge-Kutta schemes for discontinuous Galerkin time-domain simulations of Maxwell’s equations,” J. Comput. Theor. Nanosci. **7**, 1572–1580 (2010).
18. R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems* (Cambridge University Press, 2002).

1. Introduction

Numerical simulations are essential tools to support electromagnetic and photonic experiments. Of all the available algorithms, the Finite-Difference Time-Domain (FDTD) method [1] is most often employed to obtain theoretical predictions. However, a couple of issues reduce the applicability of FDTD. As it relies on the structured Yee grid, the algorithm does not provide a natural way of dealing with material interfaces. Oblique and curved surfaces are subject to the stair casing effect and lead to significant error contributions. Together with the modest second-order accurate discretization in both time and space, this leads to demanding and challenging simulations for many systems.

The discontinuous Galerkin time-domain (DGTD) method is an emerging alternative to the established FDTD algorithm [2, 3]. It combines key features of finite volume and higher-order finite element methods to an explicit time-stepping scheme. Numerous extensions, often adapted from techniques originally created for FDTD, are available [4–7] and render DGTD a universal tool for complex electromagnetic systems [8, 9].

Limited computational resources (processor time, random access memory) invariably require the truncation of the physical world to the computational domain. The boundary of the computational domain is usually artificial, i.e., not present in the physical model, and requires special treatment in order not to contaminate the numerical results. A common problem is that the boundary should be transparent to outgoing radiation. In particular, there should be no reflections. Boundary conditions which approximate this behavior are called absorbing boundary conditions (ABCs). Two complementary approaches exist.

First, one can directly enforce physical boundary conditions which support outgoing radiation modes but suppress reflected ones. Strategies which follow this route are called analytical ABCs (AABCs). Even though higher-order schemes are available [10], in practice one usually employs lower order approximations of AABCs. While being relatively straightforward to implement, they suffer from mediocre performance for oblique incidence and small distances between the radiation source and the boundary [1].

As an alternative, Bérenger developed a novel technique in the mid 90’s [11]. His idea was to divide the computational domain into a physical region, which is surrounded by a specially tailored absorbing boundary layer. To this end, he introduced an unphysical splitting of the fields in Maxwell’s equations. By cleverly defining additional material parameters in the boundary layer, he was able to match the impedances of the materials in both the physical region and the boundary layer. Thus, the interface between both is perfectly transparent for electromagnetic waves of arbitrary frequency, polarization, and angle of incidence. Hence, he called the boundary layer a perfectly matched layer (PML). Within the PML, propagating waves are attenuated. Even though they are eventually reflected at the boundary of the computational domain, by then they have been sufficiently suppressed to provide just an insignificant contribution to the electromagnetic fields in the physical region (see Fig. 1).

#139146 - $15.00 USD Received 3 Dec 2010; revised 17 Jan 2011; accepted 21 Jan 2011; published 24 Feb 2011

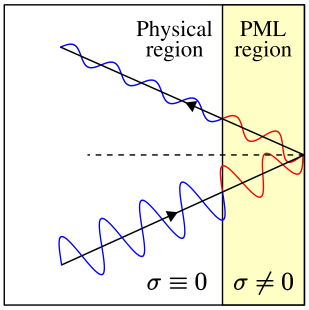

Fig. 1. Working principle of perfectly matched layers. Incident radiation is attenuated in the PML region. At the boundary of the computational domain the light wave is reflected and undergoes continued attenuation. Once it reenters the physical region, its amplitude (typically suppressed by multiple orders of magnitude) no longer presents a significant perturbation to the physical fields. Please note the absence of reflections at the interface between the physical and the PML region.

Over the years, Bérenger's original formulation was refined and generalized [1, 12–14]. Today, one usually implements PMLs either as an uniaxial anisotropic absorber (UPML) or interprets them in terms of a complex stretching of the coordinate axes. Directions how to include UPMLs in a DGTD framework have been known for some time [4, 5]. So far, however, corresponding instructions and performance characteristics for the stretched-coordinate formulation have not been reported in the literature.

In this paper we present a novel stretched-coordinate DGTD formulation to include PMLs. Even though it is mathematically equivalent to the UPML formulation, it is advantageous as its implementation is independent of material parameters. In particular, dispersive and nonlinear materials can be terminated without additional computational effort. After providing details on the numerical scheme we compare our implementation against a first-order accurate AABC (Silver-Müller boundary condition) [15] and UPMLs.

2. Stretched Coordinates in Maxwell's Equations

In a homogeneous medium plane waves propagate according to $\exp(i[\vec{\mathrm{k}}\cdot\vec{\mathrm{r}}-\omega t])$, where $\vec{\mathrm{k}}$ is the wave vector and ω the frequency of the wave. To achieve spatial damping either $\vec{\mathrm{k}}$ or $\vec{\mathrm{r}}$ has to be a complex-valued vector. In 1994, Chew and Weedon proposed a coordinate stretching to map real position vectors $\vec{\mathrm{r}}$ to the complex space [13]. We can include such a stretching into Maxwell's equations via the substitutions

$$\frac{\partial}{\partial x} \to \frac{1}{s_x}\frac{\partial}{\partial x}, \quad \frac{\partial}{\partial y} \to \frac{1}{s_y}\frac{\partial}{\partial y}, \quad \frac{\partial}{\partial z} \to \frac{1}{s_z}\frac{\partial}{\partial z},$$

where we have introduced complex stretching factors s_i. A choice commonly found in the FDTD literature [1] is given by

$$s_i(\omega) \equiv \kappa_i - \frac{\sigma_i}{\mathrm{i}\omega - \alpha_i}. \tag{1}$$

This particular choice includes three real-valued parameters which can be tuned for optimum performance. The main control parameter for the imaginary part of s_i is σ_i. For all $\sigma_i = 0$ we retrieve the non-absorbing physical space as indicated in Fig. 1. A non-zero value of α_i shifts

#139146 - $15.00 USD Received 3 Dec 2010; revised 17 Jan 2011; accepted 21 Jan 2011; published 24 Feb 2011

the pole from $\omega = 0$ to the complex frequency $\omega = -\mathrm{i}\alpha_i$. Hence, this particular choice of the stretching factor is referred to as complex frequency-shifted PML (CFS-PML). Finally, κ_i is the main contribution to the real part. Instead of adding absorption to the coordinate transform, $\kappa_i > 1$ effectively increases the width of the PML layer. This parameter plays an important role for FDTD, where one often resorts to an equidistant grid. For DGTD, however, this parameter is less relevant, since we can stretch the PML layer and the elements it consists of during the generation of the mesh. Thus, we can assume $\kappa_i \equiv 1$ without losing generality as compared to FDTD.

In the following we will show how the inclusion of the stretching factor Eq. (1) modifies Maxwell's equations. For brevity, we restrict our discussion to the necessary changes to the E_x component, for which the relevant Maxwell equation in dimensionless units ($\varepsilon_0 \equiv \mu_0 \equiv 1$) reads

$$-\mathrm{i}\omega\varepsilon E_x(\omega) = \frac{1}{s_y}\partial_y H_z(\omega) - \frac{1}{s_z}\partial_z H_y(\omega).$$

Please note that the previous equation is a frequency-domain equation, where we have used the sign convention

$$\begin{aligned} E_x(t) &\leftrightarrow \quad E_x(\omega) \\ \partial_t E_x(t) &\leftrightarrow -\mathrm{i}\omega E_x(\omega) \end{aligned} \tag{2}$$

to Fourier transform between time- and frequency-domain quantities. By splitting

$$\frac{1}{s_i} = 1 + \left(\frac{1}{s_i} - 1\right)$$

and introducing two auxiliary fields $G_{xy}^E(\omega)$ and $G_{xz}^E(\omega)$, we obtain

$$\begin{aligned} -\mathrm{i}\omega\varepsilon E_x(\omega) &= \quad \partial_y H_z(\omega) - \partial_z H_y(\omega) - G_{xy}^E(\omega) - G_{xz}^E(\omega), \\ G_{xy}^E(\omega) &= -\left(\frac{1}{s_y} - 1\right) \cdot \partial_y H_z(\omega), \\ G_{xz}^E(\omega) &= \quad \left(\frac{1}{s_z} - 1\right) \cdot \partial_z H_y(\omega). \end{aligned}$$

Now we insert the identity

$$\frac{1}{s_i} - 1 = \frac{\sigma_i}{\mathrm{i}\omega - (\sigma_i + \alpha_i)}$$

and multiply by the denominator, which yields

$$\begin{aligned} -\mathrm{i}\omega G_{xy}^E(\omega) &= \quad \sigma_y \partial_y H_z(\omega) - (\alpha_y + \sigma_y) \cdot G_{xy}^E(\omega), \\ -\mathrm{i}\omega G_{xz}^E(\omega) &= -\sigma_z \partial_z H_y(\omega) - (\alpha_z + \sigma_z) \cdot G_{xz}^E(\omega). \end{aligned}$$

Applying the transformation rule Eq. (2) finally leads to the time-domain formulation

$$\begin{aligned} \varepsilon \partial_t E_x(t) &= \quad \partial_y H_z(t) - \partial_z H_y(t) - G_{xy}^E(t) - G_{xz}^E(t), \\ \frac{1}{\sigma_y}\partial_t G_{xy}^E(t) &= \quad \partial_y H_z(t) - \frac{\alpha_y + \sigma_y}{\sigma_y} \cdot G_{xy}^E(t), \\ \frac{1}{\sigma_z}\partial_t G_{xz}^E(t) &= -\partial_z H_y(t) - \frac{\alpha_z + \sigma_z}{\sigma_z} \cdot G_{xz}^E(t). \end{aligned} \tag{3}$$

Similar equations for the other components of the electric field follow from cyclic permutations of the coordinate axes. Expressions for the magnetic field can be obtained by straightforward substitutions. Hence, Eq. (3) represents Maxwell's equations in a stretched coordinate system.

#139146 - $15.00 USD Received 3 Dec 2010; revised 17 Jan 2011; accepted 21 Jan 2011; published 24 Feb 2011

3. The Discontinuous Galerkin Method

The derivation of the DG discretization for the stretched-coordinate formulation does not provide any conceptual difficulties. We briefly sketch the derivation and refer the reader to the literature [2, 3, 5] for more details on the DG method. We start by reformulating Maxwell's equations with stretched-coordinate ADEs as the conservation law

$$\mathscr{Q}\partial_t \mathbf{q} + \vec{\nabla} \cdot \vec{\mathrm{F}} + \mathbf{S} = 0. \tag{4}$$

In contrast to the Maxwell problem in an unstretched space, the state vector $\mathbf{q}$ must be extended to include the auxiliary fields, i.e.,

$$\mathbf{q} = \left(E_x, E_y, E_z, H_x, H_y, H_z, G^E_{xy}, G^E_{xz}, G^E_{yx}, G^E_{yz}, G^E_{zx}, G^E_{zy}, G^H_{xy}, G^H_{xz}, G^H_{yx}, G^H_{yz}, G^H_{zx}, G^H_{zy}\right)^T.$$

Please note that $G^{E/H}_{ij}$ denotes the auxiliary field which results from the modification of the j-derivative for the i-component of E or H, respectively. The material matrix $\mathscr{Q}$ is modified as well and reads

$$\mathscr{Q} = \mathrm{diag}\left(\varepsilon, \varepsilon, \varepsilon, \mu, \mu, \mu, \frac{1}{\sigma_y}, \frac{1}{\sigma_z}, \frac{1}{\sigma_x}, \ldots, \frac{1}{\sigma_y}\right).$$

Finally, we define the components

$$\begin{aligned}
\mathbf{F}_x(\mathbf{q}) &= \left(0, H_z, -H_y, 0, -E_z, E_y, 0, 0, H_z, 0, 0, -H_y, 0, 0, -E_z, 0, 0, E_y\right)^T \\
\mathbf{F}_y(\mathbf{q}) &= \left(-H_z, 0, H_x, E_z, 0, -E_x, -H_z, 0, 0, 0, H_x, 0, E_z, 0, 0, 0, -E_x, 0\right)^T \\
\mathbf{F}_z(\mathbf{q}) &= \left(H_y, -H_x, 0, -E_y, E_x, 0, 0, H_y, 0, -H_x, 0, 0, 0, E_y, 0, -E_x, 0, 0\right)^T
\end{aligned}$$

of the flux vector $\vec{\mathrm{F}} = (\mathbf{F}_x, \mathbf{F}_y, \mathbf{F}_z)^T$ and specify the source vector as

$$\mathbf{S} = \left(G^E_{xy} + G^E_{xz}, \ldots, G^H_{zx} + G^H_{zy}, \frac{\alpha_y + \sigma_y}{\sigma_y} G^E_{xy}, \ldots, \frac{\alpha_y + \sigma_y}{\sigma_y} G^H_{zy}\right)^T.$$

Please note that $\vec{\mathrm{F}}$ is a vector with three components, where each component is a state vector itself. Therefore, the divergence is defined in the usual fashion as $\vec{\nabla} \cdot \vec{\mathrm{F}} \equiv \partial_x \mathbf{F}_x + \partial_y \mathbf{F}_y + \partial_z \mathbf{F}_z$.

With the notation in place, we briefly go through the essential steps of the DG discretization. First of all, we divide our computational domain into elements, e.g., tetrahedrons for three-dimensional systems. Then, we restrict ourselves to a single element, where we multiply Eq. (4) by a test function L_i and integrate over the volume of the element. Specifically, we choose L_i to be a Lagrange polynomial on the local element. Details on the polynomials and the discretization can be found in Ref. [3]. Subsequently applying Gauss's law leads to

$$\int_V \left(\mathscr{Q}\partial_t \mathbf{q} \cdot L_i - \vec{\mathrm{F}} \cdot \vec{\nabla} L_i + \mathbf{S} \cdot L_i\right) \mathrm{d}^3 r = -\int_{\partial V} \left(\hat{\mathrm{n}} \cdot \vec{\mathrm{F}}\right) \cdot L_i \, \mathrm{d}^2 r \equiv 0,$$

where $\hat{\mathrm{n}}$ is the outwardly directed normal vector on the surface of the element. Employing a mathematical trick, we replace $\vec{\mathrm{F}}$ by $\vec{\mathrm{F}}^*$, the numerical flux, and reverse the integration by parts. This leads to

$$\int_V \left(\mathscr{Q} \cdot \partial_t \mathbf{q} + \vec{\nabla} \cdot \vec{\mathrm{F}} + \mathbf{S}\right) \cdot L_i \, \mathrm{d}^3 r = \int_{\partial V} \hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^*\right) \cdot L_i \, \mathrm{d}^2 r \equiv 0,$$

#139146 - $15.00 USD Received 3 Dec 2010; revised 17 Jan 2011; accepted 21 Jan 2011; published 24 Feb 2011

the strong formulation of Maxwell's equations. In a final step, we expand the fields in terms of the Lagrange polynomials of order p to obtain

$$\begin{aligned}
\varepsilon\partial_t \tilde{\mathrm{E}}_x =& \quad \mathscr{D}_y \cdot \tilde{\mathrm{H}}_z - \mathscr{D}_z \cdot \tilde{\mathrm{H}}_y - \left(\tilde{\mathrm{G}}_{xy} + \tilde{\mathrm{G}}_{xz}\right) + \mathscr{M}^{-1}\mathscr{F} \cdot \left[\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^*\right)\right]_{E_x}, \\
\frac{1}{\sigma_y}\partial_t \tilde{\mathrm{G}}_{xy} =& \quad \mathscr{D}_y \cdot \tilde{\mathrm{H}}_z - \left(\alpha_y + \sigma_y\right) \cdot G_{xy} + \mathscr{M}^{-1}\mathscr{F} \cdot \left[\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^*\right)\right]_{G_{xy}}, \\
\frac{1}{\sigma_z}\partial_t \tilde{\mathrm{G}}_{xz} =& -\mathscr{D}_z \cdot \tilde{\mathrm{H}}_y - \left(\alpha_z + \sigma_z\right) \cdot G_{xz} + \mathscr{M}^{-1}\mathscr{F} \cdot \left[\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^*\right)\right]_{G_{xz}}.
\end{aligned} \tag{5}$$

Similar equations hold for the other fields. The spatial dependencies of the fields have been absorbed into the mass matrix $\mathscr{M}$, the derivative matrices $\mathscr{D}_i$, and the face matrix $\mathscr{F}$, which act on vectors of expansion coefficients denoted by a tilde. The definition of these matrices is given in Ref. [5]. As with the divergence, the operation $\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^*\right)$ returns a vector with the dimension of the state vector. Hence, the subscripts of this term refer to the expansion coefficients of the corresponding component of $\mathbf{q}$. Equation (5) represents a coupled system of first order differential equations. As such, the time-integration is easily and efficiently accomplished using a fourth-order five stage low-storage Runge-Kutta scheme [16, 17].

In order to incorporate Eq. (5) into a DG computer code, we still require an expression for the numerical flux $\vec{\mathrm{F}}^*$. In comparison to the standard Maxwell problem, additional, non-vanishing components were added to $\mathbf{F}_x$, $\mathbf{F}_y$, and $\mathbf{F}_z$ due to the presence of spatial derivatives in the auxiliary equations. For this reason we have to find a modified expression for the numerical flux.

4. The Split-Flux Formulation

The numerical flux is the remaining key ingredient of the DG discretization. It is responsible for the exchange of information between adjacent elements and, as a result, crucial for the accuracy and stability of the whole numerical scheme. For Maxwell's equations, the upwind flux [2,3,18]

$$\left[\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^*\right)\right]_{\vec{\mathrm{E}}} = \hat{\mathrm{n}} \times \frac{Z^+ \cdot \Delta\vec{\mathrm{H}} - \hat{\mathrm{n}} \times \Delta\vec{\mathrm{E}}}{Z^+ + Z^-} \tag{6}$$

has been found most beneficial because it efficiently damps unphysical modes which may be excited during a simulation. In Eq. (6) we have used the impedance $Z^\pm = \sqrt{\mu^\pm/\varepsilon^\pm}$ and the field differences $\Delta\vec{\mathrm{E}} = \vec{\mathrm{E}}^+ - \vec{\mathrm{E}}^-$, $\Delta\vec{\mathrm{H}} = \vec{\mathrm{H}}^+ - \vec{\mathrm{H}}^-$. Quantities with a "-" represent values in the local element while a "+" indicates values from the corresponding neighbor.

Ultimately, our new flux should be compatible with the established formulation. In particular, the flux components relevant for the electric and magnetic field components should correspond to the upwind case. Hence, we propose to use

$$\begin{aligned}
\left[\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^*\right)\right]_{\vec{\mathrm{E}}} &= \hat{\mathrm{n}} \times \frac{Z^+ \cdot \Delta\vec{\mathrm{H}} - \hat{\mathrm{n}} \times \Delta\vec{\mathrm{E}}}{Z^+ + Z^-}, \\
\left[\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^*\right)\right]_{G^E_{ij}} &= \sum_{k=1}^{3} \varepsilon_{ijk} \cdot n_j \left(\frac{Z^+ \cdot \Delta\vec{\mathrm{H}} - \hat{\mathrm{n}} \times \Delta\vec{\mathrm{E}}}{Z^+ + Z^-}\right)_k
\end{aligned}$$

as the numerical flux for the stretched-coordinate formulation of PMLs. Please note the use of the Levi-Civita symbol ε_{ijk}. Analog expressions hold for the magnetic field components and the respective auxiliary equations. An interesting analogy to Bérenger's original split field formulation surfaces in the identity

$$\left[\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^*\right)\right]_{E_x} \equiv \left[\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^*\right)\right]_{G_{xy}} + \left[\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^*\right)\right]_{G_{xz}}.$$

#139146 - $15.00 USD Received 3 Dec 2010; revised 17 Jan 2011; accepted 21 Jan 2011; published 24 Feb 2011

Instead of splitting the electromagnetic fields, our stretched-coordinate formulation relies on a splitting of the numerical flux into two contributions. Hence, the term “split-flux formulation” seems appropriate.

To validate our numerical scheme we consider a small test system not unlike the one presented later in section 7.1. We continuously illuminate a spherical object ($\varepsilon = 2.25$) of diameter 2 (dimensionless units) with a plane wave of wavelength $\lambda = 2.5$. Scattered radiation is absorbed by a single layer of stretched-coordinate PMLs. The computational domain is terminated by a perfect electric conductor. After simulating for $100,000$ time units, i.e., $40,000$ optical cycles, we conclude that our implementation is numerically stable. Correctness is later shown by the performance comparisons in section 7.2.

5. Briefly on Other Absorbing Boundaries

In the upcoming sections we want to compare our new stretched-coordinate formulation against established absorbing boundaries from the literature. To allow a better understanding we quickly review two techniques.

5.1. *Silver-Müller Boundary Condition*

The Silver-Müller (SM) boundary condition (BC) [15] is a first-order AABC. In contrast to volume absorbers like PMLs, the SMBC is an actual boundary condition. Most conveniently, it can be implemented in a DG code by modifying $\Delta\vec{\mathrm{E}}$ and $\Delta\vec{\mathrm{H}}$ for elements with missing neighbors [5]. As a result, this boundary condition does not require additional storage or computational time. Since it is basically a finite distance approximation of the radiation condition, its accuracy, however, crucially depends on the distance between the radiation source, e.g., a scatterer, and the boundary. As a result, there is a trade-off between accuracy and computational overhead due to the simulation of additional space.

5.2. *Uniaxial Perfectly Matched Layers*

Another way to implement the PML concept is the uniaxial perfectly matched layer approach (UPML) [4, 5]. In essence, the computational domain is enclosed by a surface layer which is filled with an anisotropic material. In particular, the material parameters are given by

$$\underline{\varepsilon} = \underline{\Lambda}\varepsilon, \qquad \underline{\mu} = \underline{\Lambda}\mu, \qquad \underline{\Lambda} = \operatorname{diag}\left(\frac{s_y s_z}{s_x}, \frac{s_x s_z}{s_y}, \frac{s_x s_y}{s_z}\right).$$

The common choice of s_i is $[1,4,5]$

$$s_i = 1 - \frac{\sigma_i}{\mathrm{i}\omega},$$

which is identical to Eq. (1) for $\alpha = 0$. Common DG implementations rely on auxiliary differential equations. For example, the modified equations for E_x read

$$\begin{aligned} \varepsilon\partial_t \tilde{\mathrm{E}}_x &= \mathscr{D}_y \cdot \tilde{\mathrm{H}}_z - \mathscr{D}_z \cdot \tilde{\mathrm{H}}_y + \varepsilon\left(\sigma_x - \sigma_y - \sigma_z\right) \cdot \tilde{\mathrm{E}}_x - \tilde{\mathrm{P}}_x + \mathscr{M}^{-1}\mathscr{F} \cdot \left[\hat{\mathrm{n}} \cdot \left(\vec{\mathrm{F}} - \vec{\mathrm{F}}^*\right)\right]_{E_x}, \\ \partial_t \tilde{\mathrm{P}}_x &= \left(\sigma_x^2 - \sigma_x\sigma_y - \sigma_x\sigma_z + \sigma_y\sigma_z\right) \cdot \varepsilon\tilde{\mathrm{E}}_x - \sigma_x\tilde{\mathrm{P}}_x. \end{aligned} \tag{7}$$

In this formulation, UPMLs can be used to terminate dielectric materials. For a correct treatment of dispersive materials additional auxiliary equations are required. In any case, however, the numerical flux is not changed by the UPML formulation as compared to the standard Maxwell case.

#139146 - $15.00 USD Received 3 Dec 2010; revised 17 Jan 2011; accepted 21 Jan 2011; published 24 Feb 2011

6. Computational Effort

Apparently, the inclusion of CFS-PMLs in Maxwell's equations (5) leads to the rather significant overhead of two auxiliary fields per electromagnetic field component. In comparison, the UPML formulation Eq. (7) requires only one auxiliary field per field component. Please note, however, that the auxiliary fields in both formulations can be omitted in regions where no complex stretching is present, i.e., $\sigma_i = 0$.

Furthermore, the stretched-coordinate formulation features matrix-vector products in the auxiliary equations. Hence, the computational effort to calculate the right-hand sides of Eq. (5) is significantly higher than that of the corresponding scalar operations in Eq. (7). On the other hand, the stretched-coordinate formulation intrinsically supports dispersive materials, which are usually implemented via auxiliary polarization currents. The UPML formulation requires additional auxiliary fields to combine dispersive materials and PML regions. In total, the computational costs of the stretched-coordinate formulation exceed those of the UPML method. Nevertheless, it offers a systematic treatment of dispersive materials and allows to employ complex frequency-shifting, which may be important for quite a few systems of interest.

7. Performance Comparison

The previous sections have discussed three different methods to absorb outgoing radiation. Both Silver-Müller absorbing boundaries and UPMLs are established methods while our newly developed stretched-coordinate implementation has yet to prove its value. To this end, we conduct a series of numerical experiments in order to assess the performance of each method.

7.1. Test Configuration

Our physical test system consists of a vacuum with a radiating point dipole located at the origin. As no scatterers are present, outgoing radiation should propagate toward the infinity of space. We model this physical problem by a cubic computational domain centered around the origin. The computational domain consists of the actual domain of interest and a surrounding boundary layer. The domain of interest is given by $\vec{\mathrm{r}} \in [-2.0, 2.0]^3$ in dimensionless units. The boundary layer has a variable thickness of $0.5 \cdot l$, where l is number of layers. A sketch of the computational domain can be found in Fig. 2.

To avoid mesh-induced perturbations of the outgoing waves, regular meshes are used. To this end, we divide the whole computational domain into small cubes of edge length $\Delta x = 0.5$. In turn, each of these cubes is made up of five tetrahedrons (see Fig. 2). We further minimize the mesh anisotropy by dividing the computational domain into eight sectors $\pm x > 0$, $\pm y > 0$, $\pm z > 0$. Each cube is assigned to one of these sectors depending on the x-, y-, and z-coordinates of its center. The exact configuration of the tetrahedrons which make up a cube is the same for all cubes in a sector. The respective configurations for the other sectors follow by symmetry operations.

We include the dipole source by means of the total-field/scattered-field method [1, 5]. The injection surface surrounds the origin and was chosen to be as spherical as possible with the given mesh (Fig. 2) to minimize numerical errors. The time-dependence of the dipole is given by

$$\vec{\mathrm{j}}(t) = \begin{cases} \hat{\mathrm{e}}_z \cdot \sin(\omega_0 t) \cdot \exp\left(-\frac{(t-t_0)^2}{2w^2}\right), & 0 \leq t \leq 2t_0 \\ 0, & \text{otherwise} \end{cases}$$

where $\hat{\mathrm{e}}_z$ represents the unit vector in z-direction. The excitation parameters in dimensionless units are given by $\omega_0 = 0.4 \cdot 2\pi$, $w = 1$, and $t_0 = 5$. We simulate this system for a total of 15 time units. During the simulation the time-dependence of E_z is recorded for each time step t_i at

#139146 - $15.00 USD Received 3 Dec 2010; revised 17 Jan 2011; accepted 21 Jan 2011; published 24 Feb 2011

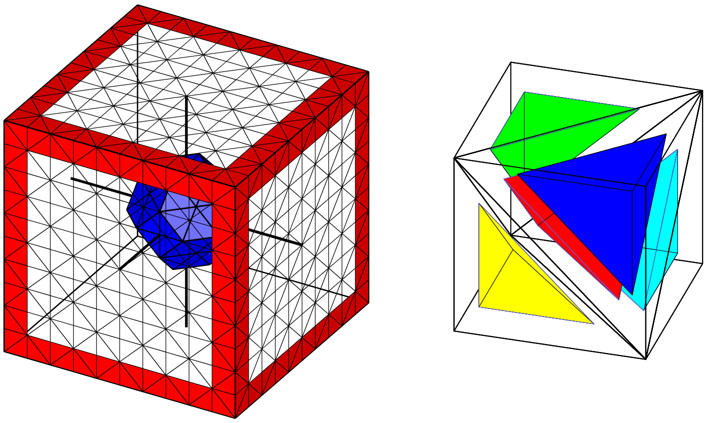

Fig. 2. Setup of the test system. The left panel shows a surface mesh of the computational domain. The red triangles indicate the extent of the boundary layer. The blue surface is used for the injection of radiation as generated by an oscillating dipole located at the center of the system. The volume mesh consists of a number of tetrahedrons, a few of which are depicted in the right panel. As outlined in the text, five tetrahedrons make up a cube. For better visibility, all tetrahedrons have been colored and shrunken.

three different observation points

$$\vec{r}_A = \big(1.8, 0.0, 0.0\big), \qquad \vec{r}_B = \big(1.8, 1.8, 0.0\big), \qquad \vec{r}_C = \big(1.8, 1.8, 1.8\big).$$

These points lie in close vicinity to the boundary layer. Hence, any distortions of the outgoing waves due to the PMLs should be quite pronounced. The accuracy of the absorbing boundary can be obtained by a simple comparison with a reference solution. Even though an analytical reference is readily available, it is not suitable for our purpose. Each numerical simulation introduces numerical errors via the spatial and temporal discretizations, which might overshadow the influence of the boundary condition.

Instead, we perform another simulation on the much larger computational domain $\vec{r} \in [-10.0, 10.0]^3$ terminated by a perfect electric conductor (PEC boundary condition). The mesh is generated in the same fashion as outlined earlier. This system, too, is simulated for 15 time units with the same time step as the smaller system. The light wave as emitted by the source propagates to the boundaries of the system where it is reflected. The reflected wave travels toward the center of the computational domain. As the speed of light in dimensionless units is given by 1, a travel distance of 10 units from the center to the boundary guarantees that the field at the observation points is not influenced by the reflected wave. Thus, the enlarged system provides a reference solution which incorporates the same spatial and temporal discretization errors as our performance test system.

Finally, we define the error of the absorbing boundary implementation as

$$\mathrm{Error}\big(\vec{r}_p\big) = \frac{\max\limits_{t_i \le 15} \left| E_z^{\mathrm{abs}}\big(\vec{r}_p, t_i\big) - E_z^{\mathrm{ref}}\big(\vec{r}_p, t_i\big) \right|}{\max\limits_{t_i \le 15} \left| E_z^{\mathrm{ref}}\big(\vec{r}_p, t_i\big) \right|}, \qquad p = A, B, C.$$

This error measure represents the largest relative deviation between the reference simulation (E_z^{ref}) and the one with an absorber (E_z^{abs}), where we have taken the maximum over all time

#139146 - $15.00 USD Received 3 Dec 2010; revised 17 Jan 2011; accepted 21 Jan 2011; published 24 Feb 2011

steps t_i. To get a representative error measure for the whole computational domain, we define the average error

$$\mathscr{E} = \frac{1}{3}\Big[\mathrm{Error}\big(\vec{\mathrm{r}}_A\big) + \mathrm{Error}\big(\vec{\mathrm{r}}_B\big) + \mathrm{Error}\big(\vec{\mathrm{r}}_C\big)\Big].$$

7.2. Results

Using the setup described in the previous section we performed a number of simulations for various parameters. For one boundary layer ($l = 1$), parameter scans for five different configurations of absorbing boundaries were conducted. These configurations are classified as

- Only a Silver-Müller boundary condition without PMLs,
- UPMLs combined with a PEC boundary condition,
- UPMLs combined with a Silver-Müller boundary condition,
- Stretched-coordinate PMLs with a PEC boundary condition,
- Stretched-coordinate PMLs with a Silver-Müller boundary condition.

With the exception of the first (parameter-free) configuration, we varied the parameter σ in 51 steps. More precisely, σ was obtained via the formula

$$\sigma = \frac{R}{2d}, \tag{8}$$

where d is the thickness of the PML layer and R is a parameter which was varied in the range $[0.0, 20.0]$ in steps of $\Delta R = 0.4$. Equation (8) is also used in the FDTD community, where it is usually combined with a polynomial grading. Such a grading however, was previously shown to have no beneficial impact on the PML error in a DGTD code [5]. Thus, we did not employ a grading in our study. The stretched-coordinate formulation also allows us to tune the complex frequency-shift α, which we have varied in steps of $\Delta\alpha = 0.05$ in the ranges $[0.0, 1.0]$ (PEC) and $[0.0, 2.0]$ (Silver-Müller), respectively.

These calculations were repeated for the polynomial orders $p = 3, 4, 5$, where higher orders represent a finer spatial discretization. Additional simulations were performed for $p = 3$ with Silver-Müller boundary conditions, where we have investigated the influence of additional layers of PMLs, i.e., $l = 2, 3$. The results are depicted in Figs. 3 through 6.

Figure 3 shows the performance of the stretched-coordinate PML formulation in comparison to UPMLs. The left panel shows the results for the computational domain being terminated by PEC boundary conditions, while a Silver-Müller boundary condition was used in the right panel. The averaged error $\mathscr{E}(R, \alpha)$ is mapped on a logarithmic color scale. For better visibility each order of magnitude has been assigned a distinct color. Moreover, black lines represent isocontours of the error in the R-α-plane. Thick lines indicate the orders of magnitude while thin lines are isocontours defined by

$$\log_{10}\mathscr{E}(R, \alpha) \equiv 0.1 \cdot n,$$

where n is an arbitrary integer. The parameter set with the minimum error is marked by a black cross. For comparison with the UPML formulation we have added another isocontour (thick blue line). Its level is given by the error from the optimum parameter set as determined from the corresponding UPML simulations. Similarly, the error of the system without an absorbing layer, but with a Silver-Müller boundary condition has also been included as a thick blue isocontour. Please note that even though the last simulation does not require a boundary layer, we have kept

#139146 - $15.00 USD Received 3 Dec 2010; revised 17 Jan 2011; accepted 21 Jan 2011; published 24 Feb 2011

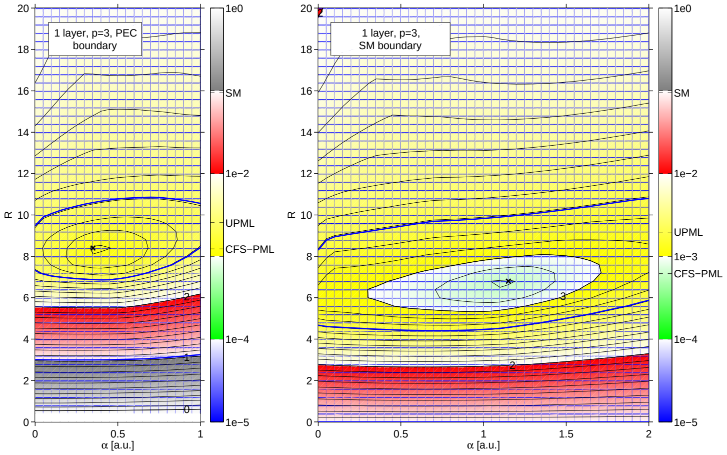

Fig. 3. Performance of the stretched-coordinate formulation in comparison to UPMLs and Silver-Müller boundary conditions for $p = 3$ and one layer of PMLs. The left panel shows the average error $\mathscr{E}$ for a test system terminated using the PEC boundary condition. The right-hand side shows corresponding results with the PEC replaced by a Silver-Müller boundary condition. Please note the logarithmic scales of the false color plots. For a description of the isocontours please refer to section 7.2.

the size of the computational domain the same for all simulations in Fig. 3. This guarantees a fair comparison of the different techniques. Finally, the error of the SMBC and the minimum errors of the UPML and the CFS-PML calculations are included in the respective colormaps. This allows a quick quantitative overview on the performance of the various methods. The panels in Figs. 4 through 6 are to be understood in this fashion, too.

The results for one layer of PMLs and $p = 3, 4, 5$ show quite a few similarities. For fixed α, the variation of R leads to strong variations of the error. If R is too small, the PMLs do not sufficiently absorb radiation, which is consequently (partly) reflected at the outer boundary of the computational domain (see Fig. 1). If R is too high, then the PMLs absorb radiation too fast, i.e., the mesh cannot properly resolve the abrupt spatial change of the electromagnetic fields. For extreme values of R we would just observe a steep drop from a finite value to zero. This situation would be comparable to a perfect electric conductor, inside of which electromagnetic fields cannot exist. Hence, high values of R lead to a numerically induced reflection from the PML interface. The optimum value of R balances both effects.

Furthermore, both UPMLs and our stretched-coordinate formulation considerably outperform the simple SMBC. However, combining the SM condition with PMLs helps to reduce the error. At the same time, the error becomes less sensitive on the PML parameters. Since the SMBC is easily implemented via the numerical flux, its usage is essentially for free and does not introduce any computational overhead over PEC boundaries. Hence, we recommend to always accompany PMLs with a Silver-Müller boundary condition. For this case, table 7.2 lists optimum parameter values R for various polynomial orders p.

It is also obvious that the additional freedom in choosing α significantly reduces the error. For $p = 5$ and Silver-Müller termination, the best error for the stretched-coordinate formulation is one order of magnitude smaller than the corresponding UPML error (see Fig. 5). It should be noted, though, that this impressive result strongly depends on the ratio between the wavelength and the thickness of the PML. In experiments with other pulse parameters (increased ω_0) we

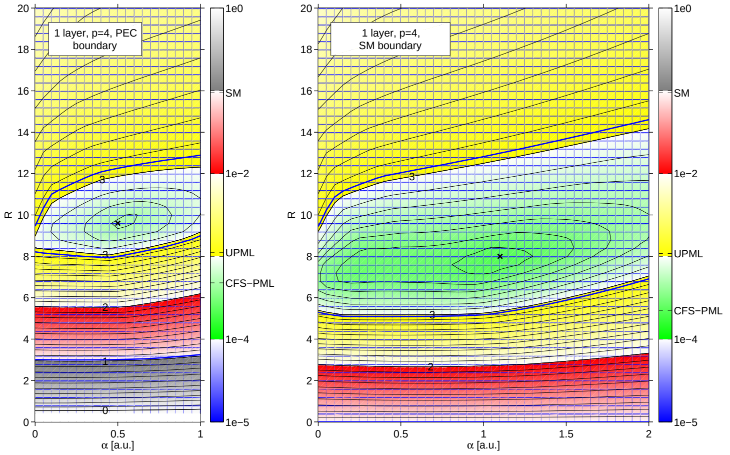

Fig. 4. Performance of the stretched-coordinate formulation in comparison to UPMLs and Silver-Müller boundary conditions for $p = 4$ and one layer of PMLs. For a more detailed explanation please refer to the caption of Fig. 3.

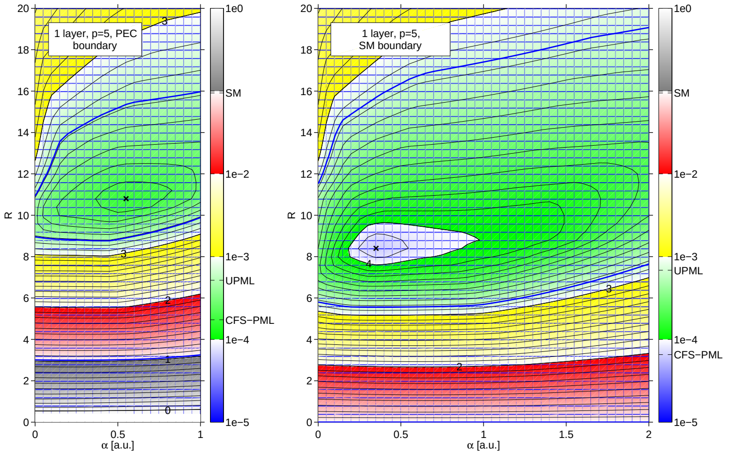

Fig. 5. Performance of the stretched-coordinate formulation in comparison to UPMLs and Silver-Müller boundary conditions for $p = 5$ and one layer of PMLs. For a more detailed explanation please refer to the caption of Fig. 3.

observe that $\alpha \neq 0$ is only advantageous if the PML is small as compared to the wavelength of the incident light. This result is consistent with the FDTD literature [1]. We also observe this behavior in Fig. 6, where we have increased the size of the boundary region to two and three layers. The optimum value for α is very close to 0 and the accuracy gains of the stretched-coordinate formulation over the UPML formulation are quite small. Since the optimum value of α crucially depends on the incident wavelength to PML width ratio, it is difficult to give general recommendations.

A final lesson we can learn from Figs. 3 through 6 is that we can reduce spurious reflections by improving the spatial discretization. Either we use higher-order Lagrange polynomials or we increase the thickness of the PML layer. Both approaches effectively distribute more degrees of

#139146 - $15.00 USD Received 3 Dec 2010; revised 17 Jan 2011; accepted 21 Jan 2011; published 24 Feb 2011

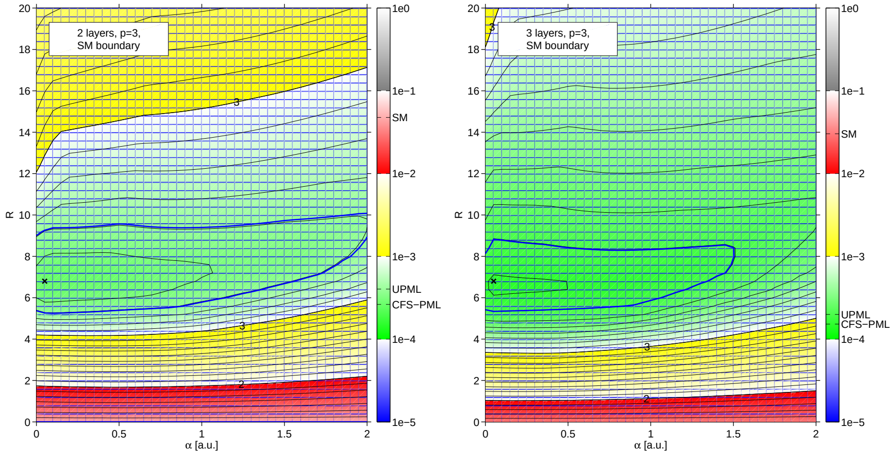

Fig. 6. Comparison of the PML performance for two and three layers and $p = 3$. For a more detailed explanation please refer to the caption of Fig. 3.

Table 1. **Optimal R-parameters for one layer of stretched-coordinate PMLs for various orders p. The computational domain is terminated using a Silver-Müller boundary condition. Please note that the optimal value of the complex frequency-shift α depends on the excitation and, thus, cannot be easily tabularized.**

Order p	**Optimal R**	**Minimal error $\mathscr{E}$**
3	6.8	$6.2 \cdot 10^{-4}$
4	8.0	$2.2 \cdot 10^{-4}$
5	8.4	$6.6 \cdot 10^{-5}$

freedom in the PML region. As a result, the evanescent decay of the incident radiation is much better resolved, and thus reflections due to an insufficient resolution are diminished.

7.3. Computational Efficiency

As we have seen in the previous section, stretched-coordinate PMLs can significantly suppress the numerical errors introduced by the boundary of the computational domain. In this section we provide some numbers on the associated computational overhead.

To this end, we measure the time required to simulate the test systems known from our previous studies. All systems are terminated by Silver-Müller boundary conditions. All simulations are performed with third-order polynomials ($p = 3$) on a single core of an Intel® Core™ 2 Quad CPU (Q9300) running at 2.50 GHz. CPU times are recorded for four systems. The first one does not comprise any PMLs. The next ones feature single layers of UPMLs and stretched-coordinate CFS-PMLs, respectively. The last one is surrounded by two layers of UPMLs. The PML parameter values correspond to the best values in Fig. 3 and Fig. 6. Please note that for these optimal parameters the error introduced by one layer of CFS-PMLs is approximately the same as the error caused by two layers of UPMLs.

The resulting CPU times can be found in table 7.3. As compared to the single layer UPML formulation, the stretched-coordinate formulation requires 19% more time for the same physical system. However, it provides more accurate results. On the other hand, two layers of UPMLs

Received 3 Dec 2010; revised 17 Jan 2011; accepted 21 Jan 2011; published 24 Feb 2011

Table 2. **Computational effort required for the simulation of various systems. The table features a short system description (all systems terminated by SMBCs), the total number of elements N_{tot} in the mesh, the number of PML elements N_{PML} in the mesh, and the CPU time required to evolve the respective system for 15 time units (see section 7.2). The next column compares the CPU time against the system without PMLs and against the system with one layer of UPMLs. Finally, the last column provides an error comparison for the different boundary conditions.**

System	$\mathbf{N_{tot}}$	$\mathbf{N_{PML}}$	Time	Relative time	Error
No PMLs	5,000	—	132 s	100% / —	$9.5 \cdot 10^{-2}$
UPML, 1 layer	5,000	2,440	155 s	+17% / 100%	$2.0 \cdot 10^{-3}$
CFS-PML, 1 layer	5,000	2,440	185 s	+40% / +19%	$6.2 \cdot 10^{-4}$
UPML, 2 layers	8,640	6,080	279 s	— / +80%	$4.0 \cdot 10^{-4}$

require 80% more time than the single layer computation, while providing only slightly more accurate results than the CFS-PML simulations. It should be noted, though, that in these systems an unusually large fraction of the mesh elements is filled with PMLs. In practice, the relative number of PML elements is often below 10%, if one resorts to one layer of PMLs.

We conclude that stretched-coordinate PMLs are computationally more efficient than UPMLs if 1) the ratio of wavelength to PML thickness is large, i.e., we have a setup where the complex frequency-shift α provides an advantage, and 2) one actually requires low PML errors. In most other situations UPMLs should be sufficient.

8. Conclusion

We have presented a material-independent, stretched-coordinate based implementation of perfectly matched layers for the discontinuous Galerkin time-domain method for electrodynamics. The numerical scheme employs auxiliary differential equations and introduces a split-flux formulation to connect neighboring elements. A thorough investigation indicates that stretched-coordinate PMLs are best used in conjunction with Silver-Müller absorbing boundary conditions. In our test case, the new formulation yields a reflectivity which is up to one order of magnitude smaller than that of the established UPML formulation. The reason behind this is the increased flexibility due to the complex frequency shift α, which helps to avoid reflections for low-frequency waves. For many practically relevant systems, e.g., metamaterials, the ratio of feature size to wavelength is considerably smaller than one. These systems will benefit most from thin PMLs, and thus our novel stretched-coordinate formulation.

Acknowledgements

We acknowledge financial support by the Center for Functional Nanostructures (CFN) of the Deutsche Forschungsgemeinschaft (DFG) within subproject A1.2. The project METAMAT acknowledges financial support by the Bundesministerium für Bildung und Forschung. The research of MK is supported through the Karlsruhe School of Optics & Photonics (KSOP) at the Karlsruhe Institute of Technology (KIT) and by the Studienstiftung des Deutschen Volkes. The research of CP and KB is supported through the DFG Priority Program SPP 1391 ”Ultrafast Nanooptics” under grant No. Bu 1107/7-1. The Young Investigator Group of JN received financial support by the “Concept for the Future” of the KIT within the framework of the German Excellence Initiative.

#139146 - $15.00 USD Received 3 Dec 2010; revised 17 Jan 2011; accepted 21 Jan 2011; published 24 Feb 2011