Apply the Perfect Electromagnetic Conducting Boundary to the Time-Space Synchronized FDTD Algorithm

David Ge (dge893@gmail.com)

November 4, 2020

Abstract

Because the Time-Space-Synchronized FDTD algorithm generates the electric field and the magnetic field at the same space points and at the same time, it is extremely simple to apply the perfect electromagnetic conducting boundary conditions. Numerical results show excellent boundary behaviors.

Introduction

For a Yee style FDTD algorithm [1], it generates 6 components of 3D electric and magnetic fields; the 6 components are located at different space locations, and the electric field and the magnetic field are at different times. Thus, it is very difficult to apply the perfect electromagnetic conducting boundary conditions. Nayyeri et al [2] invented novel formula and methods to overcome the difficulty, and dedicated a whole paper for this problem. From the formulations they gave for 1-D and 2-D cases, we can see how complex the problem is. They did not give 3D formulations; one can image that that would be another daunting task.

I have derived a generic FDTD form [3] which may cover those FDTD algorithms based on the Taylor's series, including the Yee style algorithms. Numerous different FDTD algorithms can be derived from this generic FDTD form. One such FDTD algorithm is a Time-Space-Synchronized FDTD algorithm (TSS), which I first developed in 2016 [5].

Because the TSS generates the 6 field components at the same space points and at the same time, the difficulty Nayyeri et al [2] faced is gone. Thus, the applying of the perfect electromagnetic conducting boundary conditions is extremely simple. The numeric results show perfect boundary behaviors.

Apply the perfect electromagnetic conducting boundary

Let me quote what Nayyeri et al [2] says: "A perfect electromagnetic conductor (PEMC) is a nonreciprocal generalization of both a perfect electric conductor (PEC) and a perfect magnetic conductor (PMC), which was introduced by Lindell and Sihvola [2005a]."

PEC can be expressed as

$\vec{n} \times E = \vec{0}$	(1)

Where $\vec{n}$ denotes the unit vector norm to the boundary.

PMC can be expressed as

$\vec{n} \times H = \vec{0}$	(2)

Where

$$E = \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix} is\ electric\ field, H = \begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} is\ magnetic\ field$$

Suppose a boundary is a (x, z) plane, then the unit norm vector can be

$$\vec{n} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}$$

The boundary condition (1) and (2) become

$E_x = E_z = 0$	(3)
$H_x = H_z = 0$	(4)

The above conditions can be applied to the TSS directly.

Similarly, we can apply such boundary conditions for (x, y) plane and (y, z) plane.

Numeric Results

Numeric test are done using an example taken from John B. Schneider [4].

I was using Schneider's following source code in ricker.c for field source:

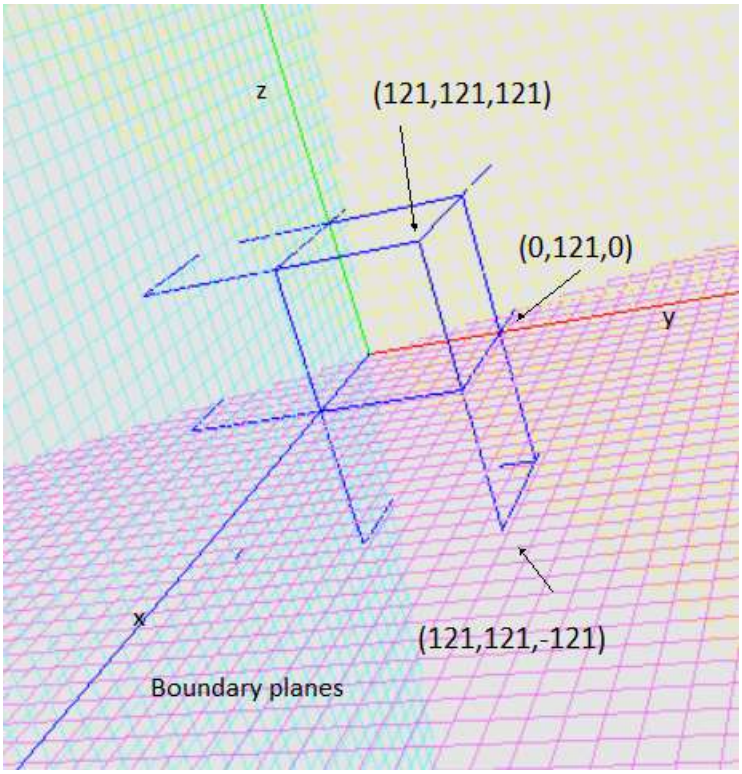
```
ricker.c:
              double ezInc(double time, double location) {
                     double arg;
                     arg = M_PI * ((cdtds * time - location) / ppw - 1.0);
                     arg = arg * arg;
                     return (1.0 - 2.0 * arg) * exp(-arg);
              }
```

One difference is that the source is applied at point (1, 0, 0), instead of (0, 0, 0) as Schneider does. No special purpose of this change; I made the change for other purposes and forgot to change it back.

The other differences are that I used much larger space domain, 6 meter, and many more space grids, $(2 \times 121 + 1)^3$ grids; these changes are for stress testing my TSS code, resulting in getting about 400 G-Byte data for one test.

John Schneider's example uses absorbing boundary condition. We can compare both boundary conditions.

There are 6 boundary planes located at $m\Delta_x, m = \pm 121, n\Delta_y, n = \pm 121, p\Delta_z, p = \pm 121$. See the following figure.



Because the huge data magnitude differences generated using the ABC boundary conditions and using the PEMC boundary conditions, it is not easy to use graphics to show the comparisons. I am using following table to show the comparisons. The values in the table are calculated in the following ways.

The space digitization is represented by

$$(x, y, z) = \left(m\Delta_x, n\Delta_y, p\Delta_z\right);\ m, n, p = 0, \pm 1, \pm 2, \dots, \pm 121$$

The fields are summarized on every radius *r*; a radius is calculated by

$$r = \max(|m|, |n|, |p|)$$

Field energy is calculated by

$$Field\ Energy = \frac{1}{2}(\varepsilon E \cdot E + \mu H \cdot H)$$

Field divergences are calculated by

$$Divergence\ E = averge\ of\ \nabla \cdot E\ at\ radius\ r$$

$$Divergence\ H = average\ of\ \nabla \cdot H\ at\ radius\ r$$

Simulation accuracy is calculated in the following way.

$$Simulation\ Accuracy\ not\ near\ source\ or\ boundary = \frac{|Divergence\ E| + |Divergence\ H|}{Field\ Energy} = v10^a$$

$$1 \le v < 10$$

$$-\infty < a < +\infty$$

A large negative a value indicates small divergences. A large positive a value indicates a large divergence.

I choose to list data at following radius to show the most important characteristics of the simulations related to the boundary conditions applied.

$$r = 0 \; and \; r = 1: the \; spaces \; are \; near/at \; the \; source$$

$$r = 60: the \; spaces \; are \; not \; near \; the \; source \; and \; boundary$$

$$r = 120: the \; spaces \; are \; near \; the \; boundary$$

$$r = 121: the \; spaces \; are \; at \; the \; boundary$$

Time	radius	ABC Boundary Conditions				PEMC Boundary Conditions			
		Field Energy	Divergence E	Divergence H	a	Field Energy	Divergence E	Divergence H	a
0	0	0	0.013031	0		0	0.013031	0	
	1	4.15903e-018	6.264e-005	0	13	4.1590e-018	6.264e-005	0	13
	60	0	0	0		0	0	0	
	120	0	0	0		0	0	0	
	121	0	0	0		0	0	0	
150	0	3.42416e-021	0.0122838	0	19	3.4241e-021	0.0122838	0	19
	1	7.35994e-019	5.90567e-005	2.5785e-018	14	7.3599e-019	5.905e-005	2.5785e-018	14
	60	1.46676e-013	3.36549e-018	8.933e-021	-5	1.4667e-013	3.391e-018	8.94006e-021	-5
	120	3.25515e-012	1.20449e-018	3.10688e-021	6	7.4645e-014	1.341e-019	3.57412e-022	-5
	121	7.90733e-012	0.00240231	6.67844e-021	9	4.5567e-015	0.00117323	2.52309e-006	12
300	0	7.83942e-019	0.0122838	2.6669e-037	17	3.480e-021	0.0122838	5.03748e-037	19
	1	3.59205e-017	5.90567e-005	2.57947e-018	12	7.3564e-019	5.905e-005	2.57868e-018	14
	60	7.86636e-012	2.6853e-017	8.49538e-020	-5	1.142e-013	5.524e-018	1.56684e-020	-5
	120	1.31149e+026	0.593555	0.00391796	-25	1.467e-012	1.156e-017	3.00822e-020	-5
	121	7.53798e+027	6.01305e+014	0.0587374	-13	4.8109e-013	0.0214704	5.13184e-005	11
500	0	4.96832e+015	0.10297	2.50351e-016	-16	5.8903e-018	0.0122838	2.17797e-036	20
	1	1.89559e+018	0.131181	0.000133985	-17	1.0984e-016	5.905e-005	2.58112e-018	11
	60	2.54129e+047	2.25174e+011	6.65632e+008	-36	4.7607e-013	1.225e-017	3.82358e-020	-4
	120	3.68501e+091	3.13481e+032	2.18994e+030	-59	6.0503e-013	1.958e-017	5.23297e-020	-4
	121	2.11755e+093	2.93431e+047	3.39666e+031	-46	1.1553e-013	0.0130144	3.50257e-005	11

Below are my observations of the simulation data.

- The radius 0 and 1 represent spaces near the field source. The large a values indicate the effects of the field source.
- The radius 120 represents spaces near the boundary.
    - At time step 150, for the ABC, $a = 6$, it indicates that the ABC acted as a field source. It indicates that the ABC did not fully absorb the energy at the boundary.
    - At time step 300, and 500, for the ABC, the field energy reached $10^{26} \; and \; 10^{91}$. I believe it indicates that the simulation had become unstable due to the failure of the ABC boundary conditions.

    - At time step 150, for PEMC, $a = -5$, it indicates that there is not a field source at the space points near the boundary. That is, PEMC was indeed working perfectly.
    - For time step 300 and 500, PEMC kept the perfect performance, that is, $a = -5 \; and \; a = -4$. The field energy was low. The simulation was stable.
- The radius 121 represents spaces at the boundary.
    - At time step 150, for the ABC, $a = 9$, and for the PEMC, $a = 12$. It indicates that the boundary conditions acted as field sources. We expect such field sources will absorb the field energy. That is, the "source" is a drainage.
    - At time step 300, for the ABC, the field energy reached 10^{27}. We know that the ABC is not supposed to absorb all the energy. But this huge energy value probably suggests that I did not coded the ABC properly, I don't know. Now we do know is that if not coded properly, the boundary conditions can inject large amount of energy, instead of absorbing energy.
    - At time step 300, for the PEMC, the field energy was still normal, at a level of 10^{-13}. It indicates that the PEMC worked as expected.
    - At time step 500, for the PEMC, the field energy was still normal, at a level of 10^{-13}. Actually at time step 600 (not shown), the energy is at 10^{-14}. It looks that the simulation can keep going on forever without a problem.
- The radius 60 represents spaces away from the boundary and from the field source. For all the time steps, the a values are large negative values, indicating that the field divergences are small. It indicates that 1). The TSS algorithm is correct; 2). My coding is correct; 3). The above observations are real.

Conclusions

By applying PEMC conditions to the TSS algorithm, we can get accurate and stable field simulations. It is extremely simple to apply PEMC conditions to the TSS algorithm. I expect the same simplicity can be true for dealing with dielectric boundaries in simulations using the TSS.

References

[1] K. S. Yee, Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media, IEEE Trans. Antennas Propagat., Vol. 14, 302-307 (1966)

[2] Vahid Nayyeri, Mohammad Soleimani, and Mojtaba Dehmollaian, Modeling of the perfect electromagnetic conducting boundary in the finite difference time domain method, RADIO SCIENCE, VOL. 48, 453–462, doi:10.1002/rds.20051, 2013

[3] David Ge, A Generic FDTD Form for Maxwell Equations, https://www.researchgate.net/publication/344868091_A_Generic_FDTD_Form_for_Maxwell_Equations

[4] John B. Schneider, Understanding the Finite-Difference Time-Domain Method, www.eecs.wsu.edu/~schneidj/ufdtd, 2010.

[5] David Ge, Time-Space Synchronized FDTD Method, https://github.com/DavidGeUSA/TSS/blob/master/TSS.pdf