Benchmark of Acceleware vs XFdtd for Field Simulations of Microstrip Patch Antenna

Tendayi Kamucheka [1], Zhijun Gui [2], Miaoqing Huang [1], Hugh Churchill [3], and Magda El-Shenawee [2]

[1] Department of Computer Science Computer Engineering, University of Arkansas, Fayetteville, AR 72701
[2] Department of Electrical Engineering, University of Arkansas, Fayetteville, AR 72701
[3] Department of Physics, University of Arkansas, Fayetteville, AR 72701
tfkamuch@uark.edu

Abstract—We benchmark the accuracy and the speed of the Acceleware FDTD library vs XFdtd in simulating a microstrip patch antenna. The benchmark is based on the sampled electric and magnetic fields. The results show that Acceleware performs almost three times faster than XFdtd. The relative mean error is less than 2.0%.

Index Terms—Acceleware, FDTD, microstrip antenna, XFdtd.

I. INTRODUCTION

The finite-difference time-domain (FDTD) method is a well-established and widely used technique in solving electromagnetic (EM) problems [1]. Fast FDTD computations including those in many FDTD commercial solvers were achieved by utilizing graphics processing units (GPUs) [2]–[5]. However, it is necessary to verify the accuracy of the simulation of the electric (E) and magnetic (H) fields before fabricating real devices. In this work, the simulations on two established software packages are presented, Acceleware FDTD library and Remcom's XFdtd. The Acceleware FDTD library is a bundle of C/C++ functions that can be used to build applications to solve EM problems [6]. In contrast, Remcom's XFdtd is a full-featured EM simulation solver software package [7]. In this paper, the Acceleware FDTD library is benchmarked against XFdtd regarding the simulation of the E- and H-fields of a patch antenna. The hardware and implementation are described below.

II. DESIGN and SIMULATIONS

For the benchmark, a rectangular microstrip patch antenna is simulated used in both XFdtd and the Acceleware application. Fig. 1 shows the dimensions of the antenna in yee-cells. Each yee-cell has dimensions, $\Delta x = 1 \times 10^{-3}$, $\Delta y = 9.9 \times 10^{-4}$, and $\Delta z = 9.41 \times 10^{-4}$ meters. Since XFdtd presents a more polished interface and features, the modeling effort starts in the XFdtd package. The antenna design and material properties are provided in [8]. Initially, a voltage source with a Gaussian time pulse of a frequency range from 0 to 20 GHz is implemented. The S-parameters are extracted from the package to identify the antenna resonant frequency at 6.564 GHz. Then the excitation is substituted for a sinusoid with frequency 6.564 GHz.
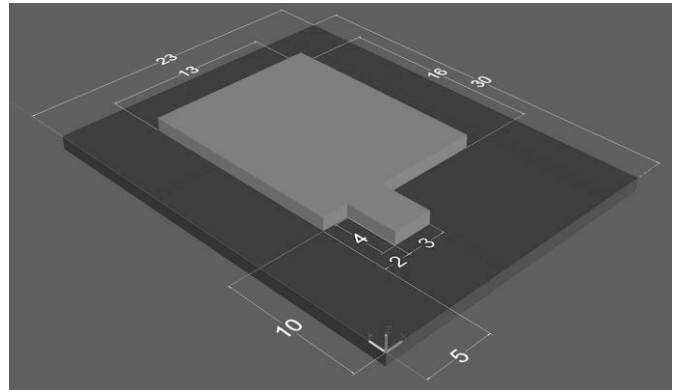
Fig. 1. Microstrip patch antenna model. All dimensions are given in yee-cells. ($\Delta x = 1 \times 10^{-3}$, $\Delta y = 9.9 \times 10^{-4}$, and $\Delta z = 9.41 \times 10^{-4}$ meters).

Once the design is completed in XFdtd, the same antenna specification is modeled in Acceleware C/C++. A custom voltage source with the same frequency and sinusoidal time signal is implemented. Convolutional perfectly matched layer (CPML) is modeled based on [9], [10]. Uniform kappa and alpha values are used as 1 and 0 respectively, over seven layers of CPML. The matching layer covers all sides except the bottom one (in the z-direction), which is set to be a Perfect Electrical Conductor (PEC) and is used as the ground plane. Following the work in [1], [11], the time-step value Δt is calculated as,

$$\Delta t \leq \frac{1}{c \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}}}, \qquad (1)$$

where, c is the speed of light and Δx, Δy, and Δz are the yee-cell dimensions for free space in meters.

A key part of the benchmark is to compare the E- and H-field data at the same yee-cell vertices in the two simulations. A 1:1 mapping between spatial coordinates of the yee-cells is ensured in both models between XFdtd and Acceleware. In XFdtd, the near-field sensors are used to record the field data from the selected locations. The same locations are used in the C/C++ code to extract the E- and H-field data at those locations. In total, there are three sets of locations, each set containing at least four spatial coordinates. Each coordinate yields six data points, three for the E-field components and three for the H-field components in x-, y-, and z-directions.
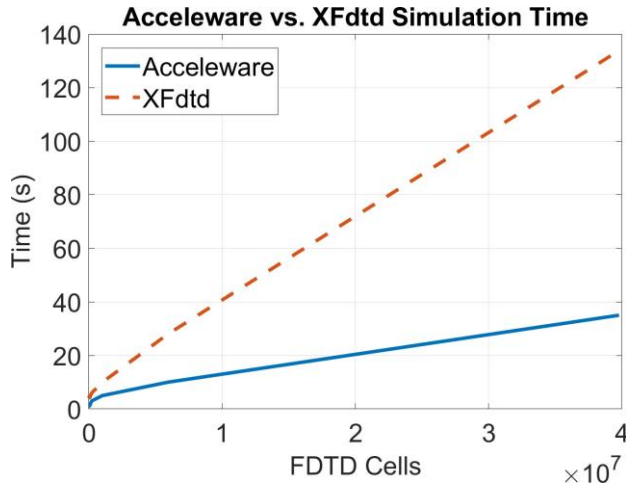
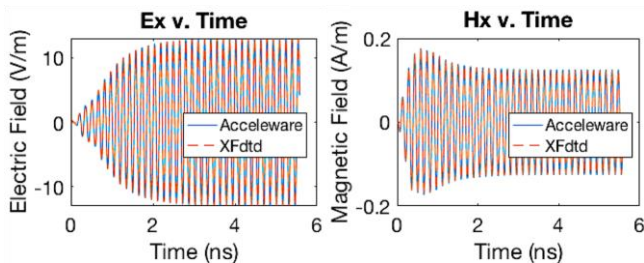Fig. 2. Plot showing simulation times vs the number of FDTD cells.

Fig. 3. Plot showing accuracy of the simulated E-x & H-x fields at yee-cell coordinates (13, 11, 3).

All XFdtd simulations are run on a Ryzen Threadripper 2990WX 32-core CPU with 128 GB system memory, and a Nvidia GTX 1050 GPU, running on Windows 10. We test the Acceleware application on an Intel Xeon Silver 4110 8-core CPU with 128 GB of system memory, and a Nvidia Tesla V100 GPU, running on CentOS 7. Field recording instruments are disabled in all simulation time tests.

III. SIMULATION RESULTS

Fig. 2 shows the performance achieved by the Acceleware code vs XFdtd packages. Upon increasing the number of cells in the FDTD simulation, the execution times were measured for the time-marching loop. Each test was executed five times and the mean execution time was recorded. The results show that the Acceleware code performs significantly faster. However, this could be due to the use of the Tesla V100 compared with the GTX 1050 in the XFdtd simulation.

It should be noted that when benchmarking field data, data recorded in XFdtd was used as the reference. Furthermore, to observe the difference between the two datasets, we calculated percentage error following the formulation of equation 2,

$$Error = \frac{|A - X|}{X} \times 100\%, \qquad (2)$$

where, X is peak values of $|XFdtd|$ and A is peak values of $|Acceleware|$. Fig. 3 shows E- and H-field data simulation data in the x-direction at yee-cell coordinates (13, 11, 3) in space. The results show that the Acceleware code is in good accuracy

agreement with the XFdtd package. The highest error calculated for all locations is less than 5%. Table I shows the mean and standard deviation of error at twelve locations in space for each field component averaged over 3000 time-steps.

TABLE I: Table showing summary of Mean and Standard deviation of error

Field	E[x]	E[y]	E[z]	H[x]	H[y]	H[z]
Mean	0.46%	1.09%	0.55%	1.37%	0.28%	1.00%
Stdev	0.36%	0.58%	0.45%	0.71%	0.27%	0.60%

IV. CONCLUSION

We have shown that the Acceleware FDTD library-based code results are in good accuracy agreement with the commercial package XFdtd. The cases presented here demonstrate a mean error less than 2.0%. Furthermore, we have also demonstrated that with a state-of-the-art GPU, the Acceleware code achieves high-performance simulations. The Acceleware code performed almost three times faster than XFdtd. It should be noted that the hardware platform is not the same across the comparison. In the future, we plan to use identical hardware configurations in similar benchmarks. We are also working on modeling more complex configurations using the Acceleware FDTD library package.

REFERENCES

[1] A. Taflove and S. C. Hagness, Computational Electrodynamics: The Finite-Difference Time-Domain Method, 3rd ed., Norwood: Artech House, 2005.

[2] S. Krakiwsky, L. Turner, and M. Okoniewski, "Acceleration of Finite-Difference Time-domain (FDTD) using Graphics Processor Units (GPU)," in 2004 IEEE MTT-S International Microwave Symposium Digest (IEEE Cat. No. 04CH37535), IEEE, vol. 2, 2004, pp. 1031-1034. [Online]. Available: http://ieeexplore.ieee.org/document/1339160/

[3] T. P. Stefan´ski, S. Benkler, N. Chavannes, and N. Kuster, "Parallel Implementation of the Finite-difference Time-domain Method in Open Computing Language," in Proceedings - 2010 12th International Conference on Electromagnetics in Advanced Applications, ICEAA'10, 2010, pp. 557-560.

[4] S. Adams, J. Payne, and R. Boppana, "Finite Difference Time Domain (FDTD) Simulations using Graphics Processors," in Department of Defense - Proceedings of the HPCMP Users Group Conference 2007; High Performance Computing Modernization Program: A Bridge to Future Defense, DoD HPCMP UGC, 2007, pp. 334-338.

[5] D. L. Markovich, K. S. Ladutenko, and P. A. Belov, "Performance of FDTD method CPU implementations for simulation of electromagnetic processes," Progress in Electromagnetics Research, vol. 139, pp. 655-670, 2013.

[6] Acceleware, "Acceleware FDTD Library." [Online]. Available: https://www.accelew are.com, https://www.acceleware.com/fdtd-solvers

[7] Remcom, "XFdtd." [Online]. Available: https://www.remcom.com/xfdtd-3d-em-simulation-software

[8] ——, "XFdtd User Guide," Tech. Rep., 2018.

[9] Acceleware, "Convolutional Perfectly Matched Layer Whitepaper," Tech. Rep., 2007.

[10] A. Elsherbeni and V. Demir, "The finite-difference time-domain method for electromagnetics with MATLAB simulations, aces series on computational electromagnetics and engineering," 2015.

[11] Remcom, "XFdtd Reference Manual," Tech. Rep., 2018.