# CEMPACK: A Benchmarking System for Hardware- and Software-Based Computational Electromagnetic Solvers

James P. Durbano, Fernando E. Ortiz, Ahmed S. Sharkawy, and Michael R. Bodnar
EM Photonics, Inc.
Newark, DE, USA
{durbano, ortiz, sharkaway, bodnar}@emphotonics.com

### Abstract

Much like the computer industry of 20 years ago, the electromagnetics community is experiencing great difficulty in quantitatively analyzing the performance of different computational electromagnetic solvers. With the recent advent of hardware-based solvers, this issue has become even more important as these tools are focused almost entirely on speed. Without a clear mechanism to evaluate different solvers, tools will be unfairly compared to one another, misrepresentations will abound, and ultimately the users of these products will suffer. To eliminate such problems, in this paper we propose the development of a computational electromagnetics benchmarking suite. We discuss the three questions that must be addressed by such a benchmark suite and describe several representative problems that should be included in the package. Ultimately, this suite will allow different hardware and software researchers/companies to provide understandable performance results and enable the direct comparison of tools over a wide range of problems.

## I. Introduction

Computational electromagnetic simulators are widely used in the early stages of research and development, in academic and industrial labs, in order to provide researchers with the necessary tools to better understand various aspects of science and technology. Currently, these simulators are implemented on both software and hardware platforms. Hardware-based implementations of computational electromagnetic algorithms, specifically the Finite-Difference Time-Domain (FDTD) algorithm, have been researched for the past 15 years. In the early 1990s, the capabilities and performance of such implementations were severely limited because of the expense associated with developing custom silicon and the immaturity of programmable hardware solutions, such as field-programmable gate arrays (FPGAs). However, by the turn of the century, FPGAs were capable of supporting entire algorithm implementations, rather than simple proof-of-concept prototypes. Several groups have been involved in this "modern" era of hardware-based FDTD solvers and two companies have been formed to market successful implementations [1-6]. Moreover, an entire chapter is dedicated to hardware acceleration in the most recent edition of the authoritative FDTD text by Allen Taflove, one of the pioneers of the FDTD method [7]. Despite all of the advances in this research area, a standard method to directly compare hardware implementations has not been developed. In [8] we proposed a method for measuring the computational performance of hardware-based solvers. However, in [7] the authors suggest a slightly different metric. In addition, a third metric is mentioned in [4].

The basis of each of these metrics involves measuring the number of discretization points that can be "updated" per second. However, this measurement is not as straightforward as it sounds. For example, are they 2D or 3D nodes? 3D nodes have more field components, and thus require more computations, as compared with 2D nodes. Is a node considered processed after updating its electric or magnetic fields or both? What about the absorbing boundary condition chosen? Berenger's perfectly-matched layer boundary conditions are more accurate, but more computationally intensive than others, such as second-order Mur boundaries [9, 10].

Without a consistent metric, misleading and confusing performance results will abound. For example, in Taflove's chapter, the metric cited is millions of cells per second (Mcels/s) [7]. However, the authors inaccurately compare the "cells" of one implementation to the "cells" of alternate implementations. If the developers of this technology cannot correctly distinguish among the metrics, one cannot expect end-users and other researchers to understand the relative performance.

The ability to quantitatively analyze the performance of different implementations is a problem that also plagued the computer industry. Specifically, hardware manufacturers wanted to compare their performance with those of competing companies. Comparisons included clock speed, floating-point operations per second (FLOPS), millions of instructions per second (MIPS), memory size/speed, cache size/speed, etc. Furthermore, because the requirements of different users varied, it was near impossible to make a statement such as "this is the best computer available." Rather, users needed to be able to evaluate machines under a variety of test problems and match up performance with the requirements of their application. Thus, benchmarking systems, such as SPEC and LINPACK were developed to help clarify system performance [11, 12].

Similarly, because discussing and quantifying the performance of both hardware and software solvers is more involved than simply quoting a single metric, such as cells per second or maximum problem size, we propose the development of a CEM-solver benchmarking suite, which we call CEMPACK. This will allow different hardware and software researchers/companies to provide understandable performance results and enable the direct comparison of tools over a wide range of problems. Section II discusses the shortcomings of related work in this area and how our proposed benchmark overcomes these weaknesses.

Section III discusses the requirements of the suite and proposes several simulations that should be included. Finally, in Section IV we draw concluding remarks and discuss future work in this area.

## II. Related Work and Original Contribution

CEM methods have been extensively used for benchmarking computer systems, particularly parallel machines, because the algorithms stress many performance-critical components simultaneously, including the interconnection network and the memory subsystem [13-15]. For example, the temporal and spatial locality of the data dependencies tests the efficiency of the caching subsystem. Also, the depth and bandwidth of the main memory are challenged by substantial problem sizes and large, non-cacheable data sets. Rather than use CEM techniques to benchmark computers, as described above, we are proposing the development of a suite of electromagnetic simulation problems to test the performance of various CEM solver implementations. This will provide users with an unbiased assessment of the capabilities of various hardware and software solvers, in terms of functionality, speed, and accuracy.

CEM code and technique standardization has been a topic of great importance to the electromagnetics community, as evidenced through numerous papers and an IEEE standards group [16-25]. Although these works discuss topics ranging from the definition and need for standardization to the actual enumeration of specific benchmark problems, the recurring theme is validation. Specifically, much of this work has been focused on determining which method best suits the problem at hand, in order to provide the most accurate results. Of these papers, the work most similar to that contained in this paper is by Archambeault et al. [19]. In their 2001 paper, these researchers provided a set of problems that could be used to assist scientists concerned with electromagnetic compatibility. Unfortunately, the proposed set of problems was limited in scope, focusing primarily on printed circuit board analysis and device packaging. Although our paper presents benchmark problems in a similar fashion, they encompass a much broader spectrum of problem types in order to reach a more diverse audience, including engineers focused on radar design and nanophotonics. Previous work in this area is very limited and only consists of self-published benchmarks from software vendors, where they describe specific features and capabilities of their solvers [14, 26].

Similarly, the proposed IEEE Standards, 1597.1 and 1597.2, when fully realized, will cover the validation and standardization of computational electromagnetics models, methods, and numerical codes [27]. However, the standard does not include ‘performance’ as a key metric for evaluating CEM tools, which is certainly an important criterion for many simulation users. Additionally, the standard is neither finalized nor available in “pre-release” form for users to begin using. Our work aims to provide a suite of simple problems that test, not only accuracy, but also performance, and can be quickly realized on a variety of platforms.

As shown, previous work in this area is focused on CEM code validation or encompasses a narrow application band, while the proposed IEEE standards are not readily available nor consider CEM performance. Thus, there is a clear need to develop both a benchmarking system that is capable of fully characterizing a given CEM solver and also a clear method for comparing performance among different solvers. The CEMPACK benchmark suite, proposed in the next section, attempts to fill this void.

# III. Benchmarking Suite

As outlined above, it is critical that the electromagnetics community develop a consistent metric to describe the performance of various CEM solvers. Moreover, only measuring processing throughputs, although important, does not provide enough information when comparing different implementations. For example, a user does not care how fast the solver is if it cannot solve their particular problem. Thus, the user wants to know the answer to three main questions: 1) Can it solve my problem? 2) How long will it take? 3) How accurate will the answer be? Therefore, the proposed metric must answer these three questions. In this section, we propose the Computational Electromagnetics Package (CEMPACK). CEMPACK (pronounced SEM-pack) consists of several synthetic benchmark problems that can be used to characterize CEM-solver implementations. The problems are "synthetic" in that they do not necessarily correspond to physically useful problems or scenarios. Rather, they attempt to stress various aspects of a solver implementation, including maximum problem size, absorbing boundary conditions, and various source types. We now address each of these questions.

## Can It Solve My Problem?

The fastest computational platform available is of no use to researchers if it cannot solve the problem at hand. Thus, the benchmark suite must test a variety of problem types. For example, several source types, including uniform plane waves, point sources, and spatially and/or temporally modulated sources, should be exercised. In addition, the suite should test platform capabilities in various media, including inhomogeneous, dispersive, and non-linear materials. The suite should also examine performance with a range of problem sizes, because a very fast, accurate solver is not useful to many if it can only solve relatively small problems.

## How Long Will It Take?

One of the most important requirements of this benchmarking suite is that actual runtimes be reported in terms of wall clock time, rather than metrics such as cells per second, nodes per second, or voxels per second [7, 8, 28]. These metrics, although useful, can be confusing as different vendors use these terms interchangeably, but measure the performance differently. Furthermore, if a particular solver implementation is capable of solving the same problem using fewer nodes, due to adaptive meshing or exploitation of symmetry, the metric may provide misleading information. For example, if two solvers are each capable of updating 10 million Yee cells per second [7], but Solver A utilizes adaptive meshing and Solver B does not, Solver A can solve the problem faster despite being “equivalent” to Solver B on paper. Because the end user is ultimately concerned with how much time the simulation requires, the benchmarking suite should simply measure wall clock time.

## How Accurate Will The Answer Be?

Ultimately, a fast solver will not be used if it does not provide accurate results. Thus, it is important that the suite report the validity of benchmark results. However, because the needs of different users/applications vary, the necessary accuracy also varies. Whereas some applications may tolerate errors on the order of $10^{-2}$, others may require accuracy to $10^{-5}$ or better. Because some solvers, such as those based on ADI-FDTD and fixed-point formats, may trade accuracy for speed, it is critical to evaluate how much accuracy is lost [1, 28-31]. If the numerical error is acceptable, these tools are highly desirable as they can potentially provide results faster than alternative platforms. Thus, the benchmark suite should test the accuracy of solvers against known analytic and experimental solutions whenever possible.

Now that we have discussed the three questions that must be addressed by a proposed benchmark suite, we present our initial problem suite. Each problem is briefly described and then several variations on the problem are presented in order to test numerous solver features.

## A. Introduction to the Suite

Each of the test cases listed below is well defined, but not "over" defined. For example, it is not necessary (nor fair) to mandate the exact absorbing boundary condition used when different boundary conditions, with significantly different computational requirements, can prove equally effective for the same problem. Similarly, there are no pre-set accuracy requirements embedded with this suite. For example, CEMPACK does not require simulations to be performed within 1% accuracy of their analytic counterparts. Rather, our intention is that multiple simulations be run with varying criteria such that a performance vs. accuracy curve can be generated. Thus, tool providers could run the same problem many different times and describe their results. This will provide more information to the user who can examine the simulation results that are most applicable to his particular problem. Furthermore, temporally and spatially modulated waveforms are not mathematically described, as different vendors implement them differently. Once again, multiple simulations can be run with various source types in order to provide a wealth of information to the users and tool vendors can describe their particular source configurations.

Ultimately, the proposed suite is designed to be general. Certainly, this may not be desirable to every user and could allow tool vendors to perform simulations that show their tool in the best possible conditions. However, this will allow a variety of simulation tool developers to quickly report results and provide information that is of interest to researchers, rather than conform to a pre-set problem that does not reflect the needs of individual users.

## B. Plane Wave Propagation in Homogeneous Media

Consider the propagation of uniform plane waves in unbounded free space. This problem is designed to stress the absorbing boundary conditions that surround the computational region. It also tests the plane-wave source. Accuracy can be verified by placing detectors at the beginning and end of the computational space, since the wave at the end is simply a time-delayed version of the initial wave. Note that “simulation time” is specified in the table in microseconds, rather than timesteps, to ensure that all platforms simulate wave propagation for the exact same length of time. Specifying timesteps is misleading, as a larger discretization grid will allow light to travel farther in the same number of timesteps.

| | |
|---|---|
| Computational Space | 250 mm x 250 mm x 500 mm |
| Source | On-axis (500 mm direction), uniform plane wave of 2.4 GHz |
| Materials | Free space |
| Geometry | N/A |
| Boundaries | Absorbing |
| Simulation time | 10 μs |

*Variation 1*: Instead of directing the plane wave entirely along an axis, launch the wave at several oblique angles. This tests the ability of the software to launch oblique waves, as well as the boundaries to absorb off-axis waves.

*Variation 2*: Launch a spatially modulated plane wave, which tests support of this feature.

*Variation 3*: Launch a temporally modulated plane wave, which tests support of this feature.

*Variation 4*: Use a linear gallium arsenide (GaAs) as the background material, with the properties described in Section 9.7.2 of [7], which tests solver support for dispersive materials.

*Variation 5*: Use a non-linear Corning glass as the background material, with the properties described in Section 9.6.7 of [7], which tests solver support for non-linear materials.

## C. Dielectric Sphere in Free Space

In this problem, a plane wave is launched at a dielectric sphere. Because this problem has an analytic solution (e.g., Mie Theory), it can be used to verify solver accuracy [32]. Furthermore, the problem size can be easily scaled by simply increasing the frequency of the incident wave (which necessitates a sampling rate change). Also, this problem allows solvers that support non-uniform meshing to sample the sphere and the surrounding free space at different rates to minimize computations.

| | |
|---|---|
| Computational Space | 250 mm x 250 mm x 500 mm |
| Source | On-axis (500 mm direction), uniform plane wave of 2.4 GHz |
| Materials | Free space, Glass |
| Geometry | Glass sphere (radius 62.5 mm) |
| Boundaries | Absorbing |
| Simulation time | 10 μs |

*Variation 1*: Increase source frequency to 15 GHz, 20 GHz, and 30 GHz in order to test support for and performance of larger problem sizes.

*Variation 2*: Increase simulation time to 200 μs to test algorithm stability.

## D. Rectangular Waveguide

Here, a broadband pulse is launched into a single-mode, rectangular waveguide. The frequency of the guided mode can be calculated analytically and compared with the simulation results for accuracy computations. This problem also tests the performance of the solver when techniques such as non-uniform meshing cannot be employed and the ability of the solver to support Gaussian beams.

| | |
|---|---|
| Computational Space | 1.667 cm x 1.071 cm x 20 cm |
| Source | On-axis (20 cm direction), broadband pulse with center frequency = 12 GHz and a 4 GHz bandwidth |
| Materials | Free space, PEC |
| Geometry | Hollow metal parallelepiped |
| Boundaries | Absorbing |
| Simulation time | 500 ps |

*Variation 1*: Arbitrarily increase length of waveguide (and simulation time) in order to determine maximum supported problem size. This problem can also be used to generate Speed vs. Problem Size curves.

In this section, we presented the three questions that a CEM benchmark suite must address (Can it solve my problem? How long will it take? How accurate will the answer be?) and several benchmark problems that should be included in such a suite. Furthermore, we presented several variations associated with each test case that stress a variety of solver features and discussed the importance of each benchmark problem. However, this benchmark suite is not yet complete and, in the next section, we discuss the work that remains.

## IV. Conclusion and Future Work

Much like the computer industry of 20 years ago, the electromagnetics community is experiencing great difficulty in quantitatively analyzing the performance of different CEM simulation tools. With the recent advent of hardware-based solvers, this issue has become even more important as these tools are focused almost entirely on speed. Without a clear mechanism to evaluate different solver platforms, tools will be unfairly compared to one another, misrepresentations will abound, and ultimately the users of these products will suffer.

Furthermore, only measuring processing throughputs, although important, does not provide enough information when comparing different implementations (a fast solver does not help if it cannot solve the intended problem). Because discussing and quantifying the performance of both hardware and software solvers is more involved than simply quoting a single metric, such as cells per second or maximum problem size, in this paper we proposed the development of a computational electromagnetic benchmarking suite. This will allow different hardware and software researchers/developers to provide understandable performance results and enable the direct comparison of tools over a wide range of problems. This will greatly benefit users as they purchase commercial products and researchers as they compare their new methods/approaches against established implementations.

In order for such a suite to be beneficial, it is vital that those in the community embrace the benchmark. We encourage others to respond to this paper and suggest additional problems that should be included. Although we have attempted to cover an array of problem types that stress various solver features, other researchers and developers will undoubtedly suggest important test problems. For example, the authors are most familiar with the FDTD method and almost certainly overlooked test problems that stress various features of other methods, such as convergence for an eigenvalue solver. Further, we encourage all developers, hardware- and software-based, to solve as many of these problems as possible and post the results in a white paper on their websites. We also encourage independent researchers to verify these results by performing direct comparisons between the tools provided by various software and hardware vendors. Only when a clear, well defined suite of problems have been defined, will legitimate comparisons between tools exist. It is important that the electromagnetics community support such a suite so that users can effectively compare and contrast various solvers and new features without ambiguities and misrepresentations.

# References

[1] W. Chen, P. Kosmas, M. Leeser, and C. Rappaport, "An FPGA Implementation of the Two-Dimensional Finite-Difference Time-Domain (FDTD) Algorithm," presented at The Twelfth ACM International Symposium on Field-Programmable Gate Arrays (FPGA), 2004.

[2] J. P. Durbano, F. E. Ortiz, J. R. Humphrey, and D. W. Prather, "FPGA-Based Acceleration of the 3D Finite-Difference Time-Domain Method," presented at 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM), 2004.

[3] R. N. Schneider, L. E. Turner, and M. M. Okoniewski, "Application of FPGA technology to accelerate the Finite-Difference Time-Domain (FDTD) method," presented at The Tenth ACM International Symposium on Field-Programmable Gate Arrays, Monterey, CA, 2002.

[4] H. Kawaguchi, K. Takahara, and D. Yamauchi, "Design Study of Ultrahigh-Speed Microwave Simulator Engine," *IEEE Transactions on Magnetics*, vol. 38, pp. 689-692, 2002.

[5] "EM Photonics, Inc.," www.emphotonics.com.

[6] "Acceleware," www.acceleware.com.

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

[8] J. P. Durbano, F. E. Ortiz, J. R. Humphrey, D. W. Prather, and M. S. Mirotznik, "Hardware implementation of a three-dimensional Finite-Difference Time-Domain algorithm," *IEEE Antennas and Wireless Propagation Letters*, vol. 2, pp. 54-57, 2003.

[9] G. Mur, "Absorbing Boundary Conditions for the Finite-Difference Approximation of the Time-Domain Electromagnetic-Field Equations," *IEEE Transactions on Electromagnetic Compatibility*, vol. 23, pp. 377-382, 1981.

[10] J.-P. Berenger, "Three-Dimensional Perfectly Matched Layer for the Absorption of Electromagnetic Waves," *Journal of Computational Physics*, vol. 127, pp. 363-379, 1996.

[11] "Standard Performance Evaluation Corporation (SPEC)," www.spec.org.

[12] "Top 500 Supercomputer Sites," http://www.top500.org/lists/linpack.php.

[13] "FDTD Cube Grid Benchmark," http://lc.cray.com/fdtd/, 1999.

[14] M. Martínez-Vázquez and D. Sánchez-Hernández, "Benchmarking of Commercial Electromagnetic Simulators for the Design of Printed Antennas," http://www.rc.fer.hr/cost260/davos/p0063.pdf.

[15] U. Anderson, P. Ekman, and P. Oster, "Performance and performance counters on the Itanium 2 - A benchmarking case study," https://rparco.urz.tu-dresden.de/Parco2003/up/1page-abstract/1p-abstract-103.pdf.

[16] B. Archambeault and C. Brench, "Proposed standard EMI modeling problems," presented at IEEE International Symposium on EMC, 1994.

[17] B. Archambeault, "Evaluating tools which predict the shielding effectiveness of metal enclosures using a set of proposed standard EMI modeling problems," presented at IEEE International Symposium on EMC, 1998.

[18] B. Archambeault, "Modeling and simulation validation for EMC applications," presented at IEEE International Symposium on EMC, 1999.

[19] B. Archambeault, S. Pratapneni, L. Zhang, D. C. Wittwer, and J. Chen, "A Proposed Set of Specific Standard EMC Problems To Help Engineers Evaluate EMC Modeling Tools," presented at IEEE International Symposium on EMC, 2001.

[20] B. Archambeault, "Concerns and approaches for accurate EMC simulation validation," presented at IEEE International Symposium on EMC, 2003.

[21] B. Archambeault and S. Connor, "Various approaches to validating full wave simulations," presented at IEEE InternationalSymposium on EMC, 2004.

[22] A. L. Drozd, "Progress on the Development of Standards and Recommended Practices for CEM Computer Modeling and Code Validation," presented at IEEE International Symposium on EMC, 2003.

[23] A. L. Drozd and B. Archambeault, "Recent Developments in the Formation of Standards and Recommended Practices for Validating Computational Electromagnetics Models and Numerical Codes," presented at IEEE International Symposium on EMC, 2004.

[24] A. Duffy, A. Drozd, B. Archambeault, and D. Coleby, "Measuring Similarity For Validation of Computational Electromagnetic Modelling (sic)," presented at IEEE International Symposium on EMC, 2004.

[25] H. Jordan, S. Bokhari, S. Staker, J. Sauer, M. ElHelbawy, and M. Piket-May, "Experience with ADI-FDTD techniques on the Cray MTA supercomputer," *Proc. of the SPIE - Commercial Applications for High-Performance Computing*, vol. 4528, pp. 68-76, 2001.

[26] "IE3D Benchmark Simulation Examples," http://www.bay-technology.com/IE3D_Benchmark.pdf.

[27] "P1597.1 - IEEE Standard for Validation of Computational Electromagnetics (CEM) Computer Modeling and Simulation & P1597.2 - IEEE Recommended Practice for Computational Electromagnetics (CEM) Computer Modeling and Simulation Applications," IEEE EMC Society Standards Development Committee.

[28] H. Songoro, "On the Real-World Performance of ADI-FDTD," http://www.semcad.com/publications/hs_poster_05.pdf, 2005.

[29] S. G. García, T.-W. Lee, and S. C. Hagness, "On the Accuracy of the ADI-FDTD Method," *IEEE Antennas and Wireless Propagation Letters*, vol. 1, pp. 31-34, 2002.

[30] T. Namiki, "A new FDTD algorithm based on alternating-direction implicit method," *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, pp. 2003-2007, 1999.

[31] F. Zheng, Z. Chen, and J. Zhang, "A finite-difference time-domain method without the Courant stability conditions," *IEEE Microwave and Guided Wave Letters*, vol. 9, pp. 441-443, 1999.

[32] G. Mie, "Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen (Contributions on optical properties of turbid solutions, with special reference to colloid metallic solutions)." *Ann. Physik*, vol. 25, pp. 377-445, 1908.