John Garofalo

Northern Illinois University

Master's Thesis Proposal

Department of Electrical Engineering

Spring 2021

Improvement of Finite-Difference Frequency-Domain Method

TABLE OF CONTENTS

ABSTRACT

The field of computational electrodynamics is rich in algorithms and methods for solving computationally based electromagnetic propagation scenarios in a variety of applications and phenomena. The current solving methods can be distinguished by their mathematical and physical approach on a computational grid in a time or frequency temporal domain. The Finite-Difference Frequency Domain (FDFD) method is one of such numerical methods for the solution of computational electromagnetics problems. Though this method provides some certain advantages compared to other numerical methods it could not find much use due to its computational inefficiency in its implementations. It is the goal of this work to show that by implementing an enhanced FDFD equation in FORTRAN based on Maxwell equations, convergence rates of the algorithm can be increased in tandem with appropriate numerical methods. Additionally, the implementation of improved periodic boundary conditions upon the computational lattice will improve FDFD utility on periodic structures. To showcase these improvements, test cases will be designed and simulated and compared with existing FDFD / FDTD algorithms. The goal of this research is to improve the computational efficiency and utility of the FDFD method.

1 INTRODUCTION

"Happy is the man who can recognize in the work of to-day a connected portion of the work of life and an embodiment of the work of Eternity. The foundations of his confidence are unchangeable, for he has been made a partaker of Infinity. He strenuously works out his daily enterprises because the present is given him for a possession.

Thus, ought man to be an impersonation of the divine process of nature, and to show forth the union of the infinite with the finite, not slighting his temporal existence, remembering that in it only is individual action possible, nor yet shutting out from his view that which is eternal, knowing that Time is a mystery which man cannot endure to contemplate until eternal Truth enlighten it."

— James Clerk Maxwell

1.1 Critical Dimensions

1.1.1 Computational Electromagnetics (CEM)

In the past two decades the field of electromagnetics has experienced a resurgence in research thanks to the increased computational power of new technology, and a variety of electromagnetic solving algorithms each utilizing different mathematical and computational techniques. Computational electromagnetics research offers an energizing way to initialize improved performance and minimize spatial constraints in practical realizations such as mobile phones, cellular networks, sensor systems, radar, and stealth systems by introducing the potential for replacement of critical components and perform material analysis in these modern electrical structures. Due to the complexity of these modern devices and propagation scenarios it is necessary to develop a theoretical and computational approach to understanding their implementation before

the realization process can take place. To accomplish this important task, we need efficient and accurate computational electromagnetic solving algorithms that can handle both large as well as small-scale simulations in a reasonable time frame.

1.1.1.2 Time Domain Vs. Frequency Domain

All computational electromagnetic algorithms use numerical methods to solve Maxwell's equations in either the time or frequency domain. Depending on the temporal domain a field propagator can be chosen (IE or DE) where harmonic time variation is assumed as the basis mode of operation. The most suitable selection of domain is dependent upon the type of problem space being modelled. Once an analytical formulation is found for a selected scenario, the discretization process in a particular domain may yield more accurate or efficient results.

The frequency domain technique tends to be more useful for problems with narrow bandwidth and high Q-factors [1]. Historically, the frequency domain has been favored for canonical problems that are useful for verifying numerical results in real-world applications. Time domain techniques are more often useful for broadband structures with complex and large geometries. The choice of excitation (transient or swept frequency) and output format (time or frequency) are relatively unimportant when considering the selection of temporal domain [1].

1.1.1.6 Finite Differences

Based on the seminal work by Yee [1], the Finite-Difference Time-Domain, and later the Finite-Difference Frequency-Domain by McCartin [2], finite difference methods were developed to handle complex electromagnetic propagation scenarios with ease and without imposing a heavy load on the system. Both FDTD and FDFD have matured greatly over the past decades, and are in daily use in the scientific community, academia, and engineering industry. However, the FDFD

method has been given relatively little attention in literature due to the computational power required by most problem spaces and is the focal point of this research.

The FD technique is based on discretizing Maxwell's curl equations in both the time and frequency domains and implementing linear field equations upon a special lattice known as a Yee grid that defines the computational space. The Yee lattice can be a single lattice or superimposed double lattice that allows for field values to occupy every edge and vertex of the Yee cell and allows support of both bi-isotropic and bianisotropic problem spaces.

The FDFD method is a differential equation method that relies on both of Maxwell's differential curl equations that describe Faraday's as well as Ampere's law. The FDFD method can utilize direct as well as iterative methods to solve varying sized systems of linear equations populating sparse matrices. The derivative operators in the finite-difference scheme are based on the forward, backward, and central differences with multiple orders of accuracy derived from their respective Taylor expansions. The convergence in the error of these differential operators on fields versus the analytical case is important for the accuracy of the electric and magnetic field values at any point in the domain, and so they must converge rapidly in as few iterations as possible. In most formulations of finite-difference methods, the second order accurate central difference operator is the optimum choice of finite difference operator to ensure rapid convergence to the analytic case. At a cost of computational speed and memory storage, higher order difference schemes can be used in the finite difference formulation. The central finite difference operators are applied to the discretized total-field Maxwellian curl equations, and pivot equations are formulated yielding six total equations for the electric and magnetic fields.

Once the set of discrete linear equations is obtained from Maxwell's equations, a linear matrix equation is formulated:

$$Ax = F \tag{1}$$

where A is the coefficient matrix, F is the excitation vector, and x is the unknown field vector containing the electric and magnetic field values being found. The resulting equation is a system whose size is $N = N_x \times N_y \times N_z$, where A is 6N x 6N, x is 6N x 1, and F is 6N x 1 since the fields are in three-dimensional space [7]. The coefficient matrix A is a sparse matrix containing mostly zeros due to the nature of PDEs being loosely coupled in electromagnetic problems. In one and two-dimensional problem spaces, direct matrix methods can be employed with reasonable amount of processing power and accuracy, however when the problem spaces become large and complex it is necessary to utilize iterative methods. The advantage of iterative methods being that they do not store all the values of the sparse coefficient matrix, only the non-zero elements.

The FDFD method has several benefits compared to the CEM previously mentioned methods. The FDFD method has no analytical load since there is no need for structure dependent Green's functions [7]. This enables the FDFD method to compute electromagnetic problems with complex geometric structures that would otherwise be extremely difficult to analyze with other methods. The FDFD method is very easy to apply to non-uniform media, enabling material generality to be broad [7]. Geometric versatility is very good, allowing FDFD to model complex-shaped and inhomogeneous structures [7]. The FDFD method handles dispersive, frequency dependent media very easily while being accurate and robust, being more stable than it's time domain counterparts like FDTD [7].

1.1.2 Desirable Model Attributes

In the development of a computer model in the context of computation electromagnetics, the task can be broken down in steps. Once a computer model has been developed, the

implementation must have very specific desirable characteristics relevant to the specific electromagnetic problems being modelled by the system. The computer model must be accurate, efficient, and have utility.

From Table II in [2], Accuracy is "… the quantitative degree to which the computed results conform to the mathematical and physical reality being modelled; accuracy, preferably of known and, better, yet, selectable, value is the single most important model attribute; determined by the physical modeling error ε_P and numerical modeling error ε_N." Efficiency is "… the relative cost of obtaining the needed results; determined by the *human* effort required to develop the computer input and interpret the output, and by the associated *computer* cost of running the model." Utility is "… the applicability of the computer model in terms of the problem size and complexity; utility also relates to ease of use, reliability of results obtained, etc."

Accuracy is most important since results that do not reflect reality cannot be trusted and may be harmful. Secondly, code that produces accurate results with an unacceptable computational cost has negligible benefit. Third, the breadth and depth of applicability in electromagnetic scenarios (i.e., excitation, material, geometric support) of a computer model determines its utility [2].

1.2 Void Areas

1.2.1 FDFD Efficiency

In the FDFD method, there are many opportunities to improve the efficiency of the method. We can approach this problem by considering separate components of the FD algorithm such as whether a pre-conditioner is utilized on a coefficient matrix before iterative methods are applied, and which iterative methods are employed to reach convergence criteria, and what memory storage

methods are used in the computational process. Taken separately, these avenues can substantially increase FDFD efficiency by decreasing computation time and reducing memory requirements.

1.2.2 Periodic Boundary Conditions (PBC) in FDFD

Increased utility in FDFD can be achieved by implementing periodic boundary conditions (PBCs) within the problem space during the pre-conditioning of a coefficient matrix. The PBC algorithms used in methods like FDTD today include field transformation methods and direct field methods. There is an opportunity for improvement in FDFD PBCs by matching the efficiency and accuracy of the constant horizontal wavenumber approach (direct field method) in FDTD [3] with a new method in FDFD or implementing Bloch-Floquet boundary conditions on the coefficient matrix.

1.2.3 Metamaterial (MTM) in FDFD

We can evaluate these improvements my implementing a metamaterial device, periodic metasurfaces, and low radar cross-section structure into a FDFD problem space. This new process will illuminate how quicker, more efficient, and accurate simulations can be achieved in the Finite-Difference Frequency-Domain method while exploring desirable characteristics of the structures to show-case algorithm execution.

2 PROBLEM STATEMENT

2.1 Desired Outcomes

2.1.1 Improved Efficiency

It is of primary importance to improve algorithm convergence rate through the construction of a linear matrix equation, as well as pre-conditioning methods in tandem with iterative methods

in FDFD. The numerical methods required to accomplish complex computation for small and large problem spaces are iterative methods that avoid the need for the inversion operations seen in direct methods. Special pre-conditioners can be applied to the coefficient matrix before iterative methods are implemented that aid in convergence rate. In numerical linear algebra, these iterative methods are known as Krylov subspace methods which include such methods as BiCGSTAB, GMRES, and QMR that are applied to a conditioned or unconditioned coefficient matrix. BiCGSTAB is the iterative method of choice for FDFD applications. Once an FDFD equation has been formulated, we can establish how to most efficiently populate a coefficient matrix, pre-condition the matrix, and which iterative method to implement during the execution of the FDFD algorithm. A primary focus is to improve FDFD efficiency is by the implementation of a divergence-based equation for the electric and magnetic fields as developed similarly in [11] and methods described in [12] into the FDFD equation.

2.1.2 Improved Utility

When formulating a matrix FDFD equation it is important to establish an appropriate eigenvalue distribution of the sparse linear coefficient matrices, as well as apply periodic boundary conditions to the computational grid that conform to the geometry of the problem space. To increase the utility of FDFD we need reliable, representative values of the scattered fields based on the total field formulation for the electric and magnetic fields in a periodic problem space. This can be implemented with Bloch-Floquet boundary conditions imposed on each boundary of the computational grid which gives the benefit of unit cell conditions with the additional feature of being able to apply a phase shift as a function of the wave vector at the boundary in the frequency domain. Additional utility can be achieved by implementing RCS computation within the FDFD framework.

2.3 Hypothesis

The general hypothesis for this research is as follows: the optimization of numerical methods on sparse coefficient matrices and the implementation of improved periodic boundary conditions in the FDFD method can increase simulation efficiency of small-and large-scale problem spaces. The solution and verification of this hypothesis depends on an established FDFD code base in MATLAB or Fortran. With known numerical methods (iterative, pre-conditioner) used within the FDFD method we can impose more efficient computation within the computational grid along with periodic boundary conditions for arbitrary size problem spaces.

3 PROPOSED WORK

This work strives to integrate a unique coefficient matrix representation building from the divergence equation into an FDFD equation to enhance convergence rate of the algorithm in FORTRAN or MATLAB. Additionally, by implementing periodic boundary conditions into the FDFD method, we can verify that it accurately simulates periodic structures. Further verification of FDFD utility can be made by implementing RCS measurements.

REFERENCES

[1] T. Hubing, C. Su, H. Zeng and H. Ke, "Survey of Current Computational Electromagnetics Techniques and Software," Clemson Vehicular Electronics Laboratory Technical Report: CVEL-08-011.3, June 6, 2009

[2] E. K. Miller, “A Selective survey of Computational Electromagnetics,” IEEE Trans. Ant. &Prop., vol. 36, pp. 1281-1305, Sept. 1988.

[3] Elsherbeni, A. Z., & Demir, V. (2016). The finite-difference time-domain: Method for electromagnetics with MATLAB simulations. Edison, NJ: SciTech Publishing, an imprint of the IET.

[4] Balanis, C. A. (2016). Antenna theory: Analysis and design. Hoboken, NJ: Wiley.

[5] Kane Yee, "Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media," in *IEEE Transactions on Antennas and Propagation*, vol. 14, no. 3, pp. 302-307, May 1966, doi: 10.1109/TAP.1966.1138693.

[6] B. J. McCartin and J. F. Dicello, "Three-dimensional finite difference frequency domain scattering computation using the Control Region Approximation," in IEEE Transactions on Magnetics, vol. 25, no. 4, pp. 3092-3094, July 1989, doi: 10.1109/20.34378.

[7] Alkan, E. (2013). Double-Grid Finite-Difference Frequency-Domain (DG-FDFD) method for scattering from chiral objects. S.l.: Morgan & Claypool.

[8] Shin, Wonseok (2013). 3D Finite-Difference Frequency-Domain Method for Plasmonics and Nanophotonics [Doctor of Philosophy]. Massachusetts Institute of Technology.

[9] Lindell, I. V., & Sihvola, A. H. (2019). *Boundary conditions in electromagnetics*. Newark: John Wiley et Sons, Incorporated.

[10] Liesen, J., & Strakoš, Z. (2015). Krylov subspace methods: Principles and analysis. Oxford: Oxford University Press.

[11] Y. Saad, Krylov subspace methods for solving large unsymmetric linear systems, Math. Comp., 37 (1981), pp. 105–126.

[12] Song, Q., Sture, S., Zhang, Q., Fernandez, J. R., Ferronato, Massimiliano (2012). Preconditioning for Sparse Linear Systems at the Dawn of the 21st Century: History, Current Developments, and Future Perspectives. ISRN Applied Mathematics. International Scholarly Research Network

[13] M. A. Sharkawy, V. Demir and A. Elsherbeni, "An Efficient ILU Preconditioning for Highly Sparse Matrices Constructed using the FDFD Method," in *IEEE Antennas and Propagation Magazine*, vol. 49, no. 6, pp. 135-139, Dec. 2007, doi: 10.1109/MAP.2007.4455873.

[14] L. Grigori, “Sparse linear solvers: iterative methods, sparse matrix-vector multiplication, and preconditioning.” UC Berkeley. March 2015.

[15] Fokkema, R. Diederik (1996). Subspace Methods for Linear, Non-Linear and Eigen Problems [Doctor of Philosophy]. Universiteit Utrecht.

[16] V. Demir, E. Alkan, A. Z. Elsherbeni and E. Arvas, "An Algorithm for Efficient Solution of Finite-Difference Frequency-Domain (FDFD) Methods [EM Programmer's Notebook]," in IEEE Antennas and Propagation Magazine, vol. 51, no. 6, pp. 143-150, Dec. 2009, doi: 10.1109/MAP.2009.5433120.

[17] S. J. Lavdas, C. S. Lavranos and G. A. Kyriacou, "Periodic structures eigenanalysis incorporating the Floquet Field Expansion," 2011 International Conference on Electromagnetics in Advanced Applications, Torino, 2011, pp. 1253-1256, doi: 10.1109/ICEAA.2011.6046528.

[18] Bandala, Shiva (2010). A Simple Multigrid Approach for Faster Solution of FDFD Method [Master of Science]. Northern Illinois University.

[19] Saad, Y. (2003). Iterative Methods for Sparse Linear Systems. doi:10.1137/1.9780898718003

[20] Saad, Y. (2011). Numerical Methods for Large Eigenvalue Problems. doi:10.1137/1.9781611970739

[21] Chen, K. (2005). Matrix Preconditioning Techniques and Applications. doi:10.1017/cbo9780511543258

APPENDIX

A.1.1 Finite Differences

The calculus of the finite difference method was originally developed to numerically approximate partial differential equations with finite difference equations. The derivatives of the finite difference method are found from expressions for any order derivative of a function via their respective Taylor expansions to any order accuracy in the error term. In the FDFD method, the derivatives employed are the forward, backward, and central finite differences. The forward and backward derivatives being first order accurate, and the central derivative being second order accurate in the error term.

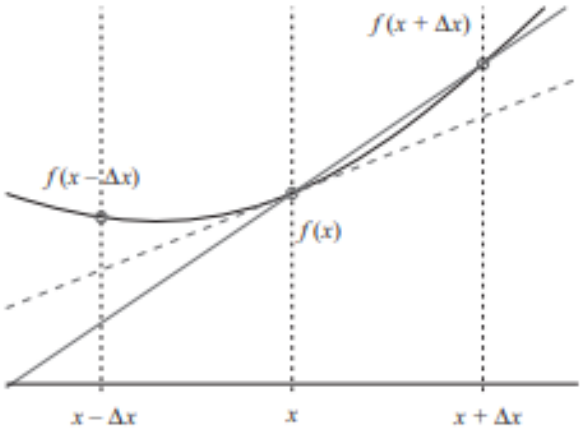


Figure 2: Forward-difference.

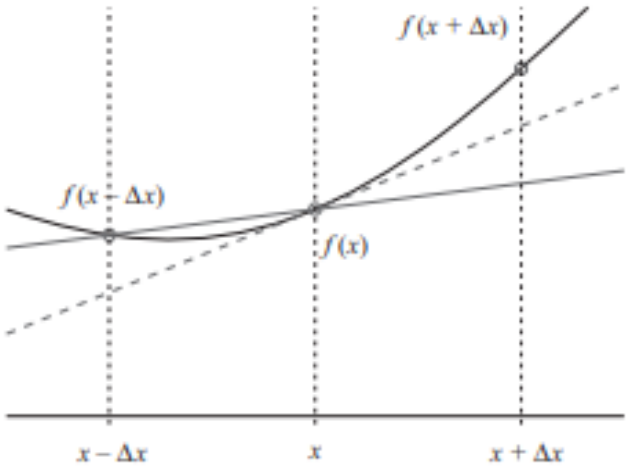


Figure 2: Backward-difference.

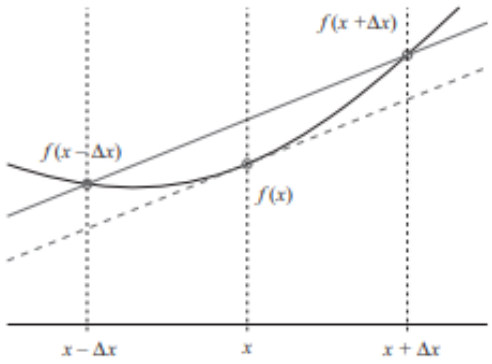


Figure 3: Central-difference.

When utilizing finite difference equations to approximate PDEs, it is important to use the finite difference equation that yields the quickest convergence to the analytical case. For the FDFD technique, this is almost always the second order accurate central difference equation due to its accuracy and rapid convergence.

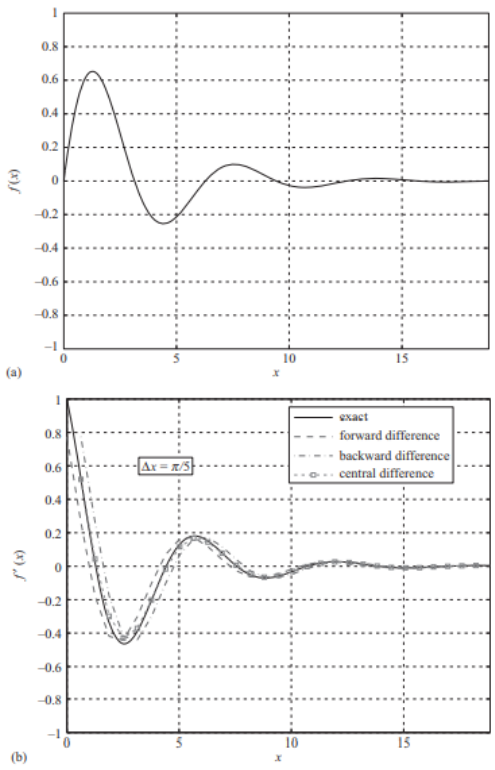


Figure 4: Finite difference convergence plot.

The derivation for the finite difference equations begins at the general Taylor expansion of a function. By finding the Taylor expansions of shifted functions we can arrive at expressions for the forward, backward, and central finite difference equations.

$$f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x} + O(\Delta x),$$

Equation 1: Forward-difference formula.

$$f'(x) = \frac{f(x) - f(x-\Delta x)}{\Delta x} + O(\Delta x).$$

Equation 2: Backward-difference formula.

$$f(x+\Delta x) = f(x) + \Delta x f'(x) + \frac{(\Delta x)^2}{2} f''(x) + \frac{(\Delta x)^3}{6} f'''(x) + \frac{(\Delta x)^4}{24} f''''(x) + \cdots.$$

Equation 3: Taylor expansion of forward shift.

$$f(x-\Delta x)=f(x)-\Delta x f'(x)+\frac{(\Delta x)^2}{2}f''(x)-\frac{(\Delta x)^3}{6}f'''(x)+\frac{(\Delta x)^4}{24}f''''(x)+\cdots.$$

Equation 4: Taylor expansion of backward shift.

$$f(x+\Delta x)-f(x-\Delta x)=2\Delta x f'(x)+\frac{2(\Delta x)^3}{6}f'''(x)+\cdots.$$

Equation 5: Subtraction of forward and backward shifted Taylor expansions.

$$f'(x)=\frac{f(x+\Delta x)-f(x-\Delta x)}{2\Delta x}-\frac{(\Delta x)^2}{6}f'''(x)+\cdots=\frac{f(x+\Delta x)-f(x-\Delta x)}{2\Delta x}+O\left((\Delta x)^2\right)$$

Equation 6: Central-difference formula.

In the time domain, this derivation of the central difference formula holds. However, the time domain central difference equation has an artifact of two in the denominator due to spatial values of the fields being at half-integer steps on the Yee cell. In the Taylor expansion, the space delta term is redefined for half-cell steps at integer nodes due to the computational indexing scheme.

A.1.2 Computational Grid

All CEM problems require a mesh in a computational space to solve a particular problem space on discrete unit cells. Yee developed the FDTD method in [9] using what is known today as a Yee grid. The Yee grid (or lattice) is a staggard, uniform cube that quantifies electric and magnetic field values (and associated permittivity and permeability) that satisfy Maxwell's curl equations. This allows the problem space to be approximated in a stair-stepped fashion. The cell

is staggard because when the curl operator is approximated using a difference formula, the derivative is evaluated between the sample locations used in the difference formula [11].

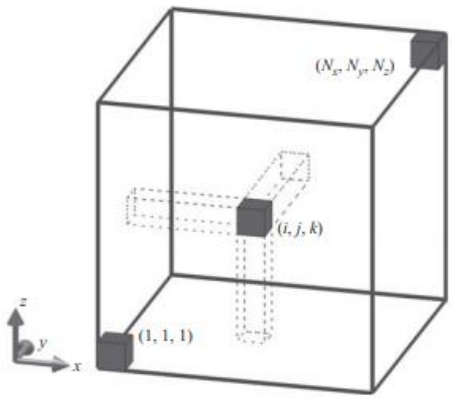


Figure 5: A three-dimensional computational space composed of (N_x x N_y x N_z) Yee cells.

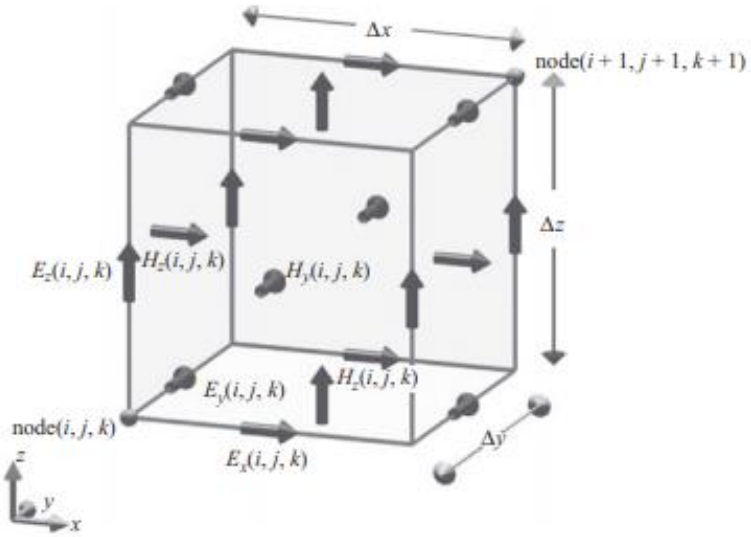


Figure 6: Arrangement of field values on a Yee cell indexed as (i, j, k).

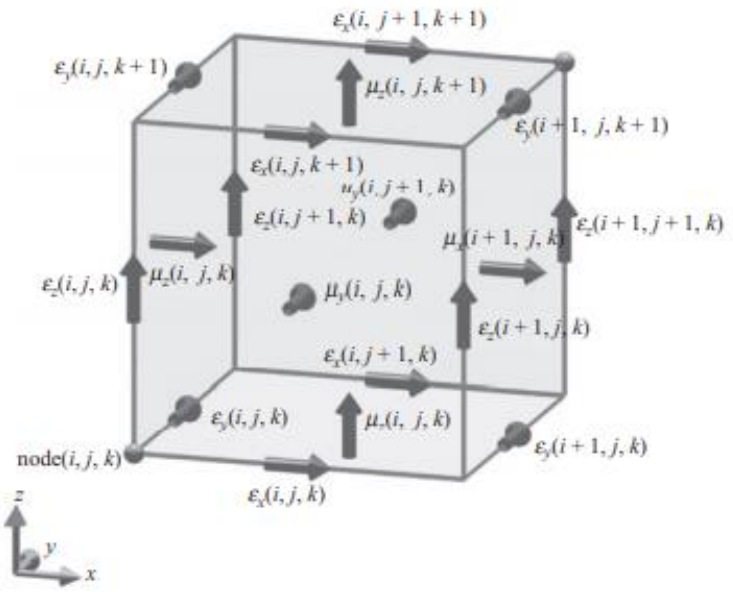


Figure 7: Arrangement of permittivity and permeability values on a Yee cell indexed as (i, j, k).

This can be extended to the FDFD method identically. The electric field values occupy edges, while the magnetic field values occupy the faces of the cells. The permittivity and permeability of the problem space occupy the edges and faces, respectively.

A.1.3 Maxwell's Curl Equations

The foundation of CEM is solving for Maxwell's equations in either time or frequency domain. The FDFD method is based on the Fourier transform of time-harmonic Maxwell's equations. The curl equations are of utility and are the basis for solving all electromagnetic problems cast in the finite difference method.

$$\nabla x E = -i\omega B - M$$

Equation 7: Faraday's Law of Induction in Frequency Domain

$$\nabla x H = i\omega E + J$$

Equation 8: Ampere's Circuital Law in Frequency Domain

These equations coupled with specific constitutive relations that represent isotropic, anisotropic, or bianisotropic materials faithfully describe all electromagnetic-matter interaction in the known universe at the macroscopic (non-quantum) scale.

The divergence equations can also be used in the FDFD method to improve convergence rate and accuracy by implementing the free-space electric field divergence (Gauss's Law) into the FDFD matrix equation as formulated in [12]. This method utilized the properties of the continuity equation for charge density and electric current density.

A.1.4 Total Field Formulation

To get an accurate depiction of any electromagnetic scenario in the computational realm, it is necessary to utilize the total field/scattered field formulation. With this formulation we can find expressions for the scattered fields as well as the incidence and use them in the curl equations to find a general relationship in three dimensions between the excitations and the unknown scattered fields resulting from field-matter interaction. The decomposition is as follows:

$$E_{total} = E_{scat} + E_{inc}$$

Equation 9: Total Electric Field

$$H_{total} = H_{scat} + H_{inc}$$

Equation 10: Total Magnetic Field

In the FDFD method only the scattered fields are computed on the Yee lattice, and no incident waves propagate within the grid [11]. If a total field value is needed at some point in the grid it is found in post processing [11].

A.1.5 Discretization

For any computational model, naturally, it is necessary to discretize the analytical model of an electromagnetic propagation scenario so that numerical methods can be applied. For the discretization of Maxwell's equations, we reformulate the equations upon the computational grid (Yee lattice) in a framework that allows for granular approximation of electromagnetic-material interactions. The PDEs are decoupled into their constituent components and solved individually on a nodal basis with material parameters applied to the fields. These nodes correspond to vectorizable, array indices that can be computed and stored discretely in computer memory. In this process error is incurred but is minimized in the iterative process of the FDFD method. The discrete (non-PML) electric and magnetic field equations are expressed as follows:

$$E_{scat,x}(i,j,k) = \frac{1}{j\omega\varepsilon_{xy}(i,j,k)\Delta y} H_{scat,z}(i,j,k) - \frac{1}{j\omega\varepsilon_{xy}(i,j,k)\Delta y} H_{scat,z}(i,j-1,k)$$
$$- \frac{1}{j\omega\varepsilon_{xz}(i,j,k)\Delta z} H_{scat,y}(i,j,k) + \frac{1}{j\omega\varepsilon_{xz}(i,j,k)\Delta z} H_{scat,y}(i,j,k-1)$$
$$- \frac{\left(\varepsilon_{xi}(i,j,k) - \varepsilon_o\right)}{\varepsilon_{xi}(i,j,k)} E_{inc,x}(i,j,k)$$

$$E_{scat,y}(i,j,k) = \frac{1}{j\omega\varepsilon_{yz}(i,j,k)\Delta z} H_{scat,x}(i,j,k) - \frac{1}{j\omega\varepsilon_{yz}(i,j,k)\Delta z} H_{scat,x}(i,j,k-1)$$
$$- \frac{1}{j\omega\varepsilon_{yx}(i,j,k)\Delta x} H_{scat,z}(i,j,k) + \frac{1}{j\omega\varepsilon_{yx}(i,j,k)\Delta x} H_{scat,z}(i-1,j,k)$$
$$- \frac{\left(\varepsilon_{yi}(i,j,k) - \varepsilon_o\right)}{\varepsilon_{yi}(i,j,k)} E_{inc,y}(i,j,k)$$

$$E_{scat,z}(i,j,k) = \frac{1}{j\omega\varepsilon_{zx}(i,j,k)\Delta x} H_{scat,y}(i,j,k) - \frac{1}{j\omega\varepsilon_{zx}(i,j,k)\Delta x} H_{scat,y}(i-1,j,k)$$
$$- \frac{1}{j\omega\varepsilon_{zy}(i,j,k)\Delta y} H_{scat,x}(i,j,k) + \frac{1}{j\omega\varepsilon_{zy}(i,j,k)\Delta y} H_{scat,x}(i,j-1,k)$$
$$- \frac{\left(\varepsilon_{zi}(i,j,k) - \varepsilon_o\right)}{\varepsilon_{zi}(i,j,k)} E_{inc,z}(i,j,k)$$

Equations 11-13: Discrete 3-D Scattered Electric Field Components

$$H_{scat,x}(i,j,k) = \frac{1}{j\omega\mu_{xz}(i,j,k)\Delta z}E_{scat,y}(i,j,k+1) - \frac{1}{j\omega\mu_{xz}(i,j,k)\Delta z}E_{scat,y}(i,j,k)$$
$$- \frac{1}{j\omega\mu_{xy}(i,j,k)\Delta y}E_{scat,z}(i,j+1,k) + \frac{1}{j\omega\mu_{xy}(i,j,k)\Delta y}E_{scat,z}(i,j,k)$$
$$+ \frac{(\mu_o - \mu_{xi}(i,j,k))}{\mu_{xi}(i,j,k)}H_{inc,x}(i,j,k)$$

$$H_{scat,y}(i,j,k) = \frac{1}{j\omega\mu_{yx}(i,j,k)\Delta x}E_{scat,z}(i+1,j,k) - \frac{1}{j\omega\mu_{yx}(i,j,k)\Delta x}E_{scat,z}(i,j,k)$$
$$- \frac{1}{j\omega\mu_{yz}(i,j,k)\Delta z}E_{scat,x}(i,j,k+1) + \frac{1}{j\omega\mu_{yz}(i,j,k)\Delta z}E_{scat,x}(i,j,k)$$
$$+ \frac{(\mu_o - \mu_{yi}(i,j,k))}{\mu_{yi}(i,j,k)}H_{inc,y}(i,j,k)$$

$$H_{scat,z}(i,j,k) = \frac{1}{j\omega\mu_{zy}(i,j,k)\Delta y}E_{scat,x}(i,j+1,k) - \frac{1}{j\omega\mu_{zy}(i,j,k)\Delta y}E_{scat,x}(i,j,k)$$
$$- \frac{1}{j\omega\mu_{zx}(i,j,k)\Delta x}E_{scat,y}(i+1,j,k) + \frac{1}{j\omega\mu_{zx}(i,j,k)\Delta x}E_{scat,y}(i,j,k)$$
$$+ \frac{(\mu_o - \mu_{zi}(i,j,k))}{\mu_{zi}(i,j,k)}H_{inc,z}(i,j,k)$$

Equations 14-16: Discrete 3-D Scattered Magnetic Field Components

A.1.7 Boundary Conditions

For all CEM problems it is necessary to establish boundary conditions upon the computational grid that the propagation scenario is being evaluated. A boundary is a surface on which secondary electromagnetic sources are induced by the primary fields so that the fields beyond the surface vanish [13]. Boundary conditions can exist on a cell-by-cell basis on the Yee lattice in the FDFD method and can be implemented as periodic or unit cell boundary conditions. Periodic boundary conditions differ from unit cell boundary conditions in that we can implement a phase shift at each boundary whereas with unit cell boundary conditions no phase shift takes place. A boundary surface is different from a media interface because the fields beyond the interface do not vanish (i.e., there is a transmission wave). Mathematically, boundary conditions

are required to ensure existence and uniqueness of solutions for a particular electromagnetic problem [13].

A.1.7.1 Absorbing Boundary Conditions

For most practical applications of FDFD only open problem spaces are considered. Consequently, it is necessary to surround the problem domain with regions that annihilate scattered waves resulting from interactions in the problem space. These are what are known as Absorbing Boundary Conditions (ABCs). The ideal ABC completely absorbs the scattered fields caused by a scatterer so that the problem domain can be truncated to replicate an infinite region without causing reflections at the edge, effectively reducing computational cost necessary to perform a simulation.

A.1.7.2 Perfectly Matched Layer

Among all the ABCs in existence the perfectly matched layer (PML) as proposed by Berenger in [14] is the most effective ABC used to terminate the computational space in the finite difference method. The PML technique is based on a novel split-field formulation where each vector field component in Maxwell's equation is split into two orthogonal components. The problem space is surrounded by a matched material that has both electrical and magnetic conductivity used in the usual finite difference discrete Maxwell's equations withing the PML region.

$$E_x = E_{xy} + E_{xz} \qquad H_x = H_{xy} + H_{xz}$$

$$E_y = E_{yz} + E_{yx} \qquad H_y = H_{yz} + H_{yx}$$

$$E_z = E_{zx} + E_{zy} \qquad H_z = H_{zx} + H_{zy}$$

Equations 17-22: Decomposed PML Field Components

The layer surrounding the problem domain can theoretically absorb all waves traveling toward the boundaries without any reflection [11]. The PML uses an anisotropic lossy layer which only retains the electric conductivity and magnetic loss with the ratio

$$\frac{\sigma^e}{\sigma^m} = \frac{\varepsilon_0}{\mu_0}$$

Equation 23: Retained Electric Conductivity and Magnetic Loss Ratio in PML Method

in the direction normal to the computational boundary [14]. The PML layers are applied to the outer region of the computational space and given an appropriate thickness to minimize the reflection magnitudes back into the problem space. The PML layer is ended with a perfect electric conductor (PEC). Due to the nature of numerical discretization, however, there may still be reflections back into the problem space [11].

To reduce the reflections from the PML layer, both the electic and magnetic conductivities are chosen to increase from zero at the vacuum-PML interface to a value σ_{MAX} at the outler layer of the PML [11]. σ_{MAX} can be determined from [14] as

$$\sigma_{MAX} = -\frac{\varepsilon_0 c(n+1)\ln[R(0)]}{2\delta_{PML}}$$

Equation 24: Equation Determining σ_{max}

where n is the degree of loss of increment of the loss of the PML layers [11]. The loss is linear for n = 1 and parabolic for n = 2 within the PML layers [11]. δ_{PML} is the PML layer thickness, c is the speed of light in vacuum, and R(0) is the theoretical reflection factor at normal incidence [11].

The conductivity inside the the absorbing layers is determined by

$$\sigma(h) = \sigma_{MAX}\left(\frac{h}{\delta_{PML}}\right)^n$$

Equation 25: Equation Determining Conductivity Inside PML Layer

where h is the distance from the vacuum-PML interface to a point inside the PML media [11]. These relation match the PML layers and the main computational space to absorb all waves propagating from the computational domain toward the PEC boundary [11].

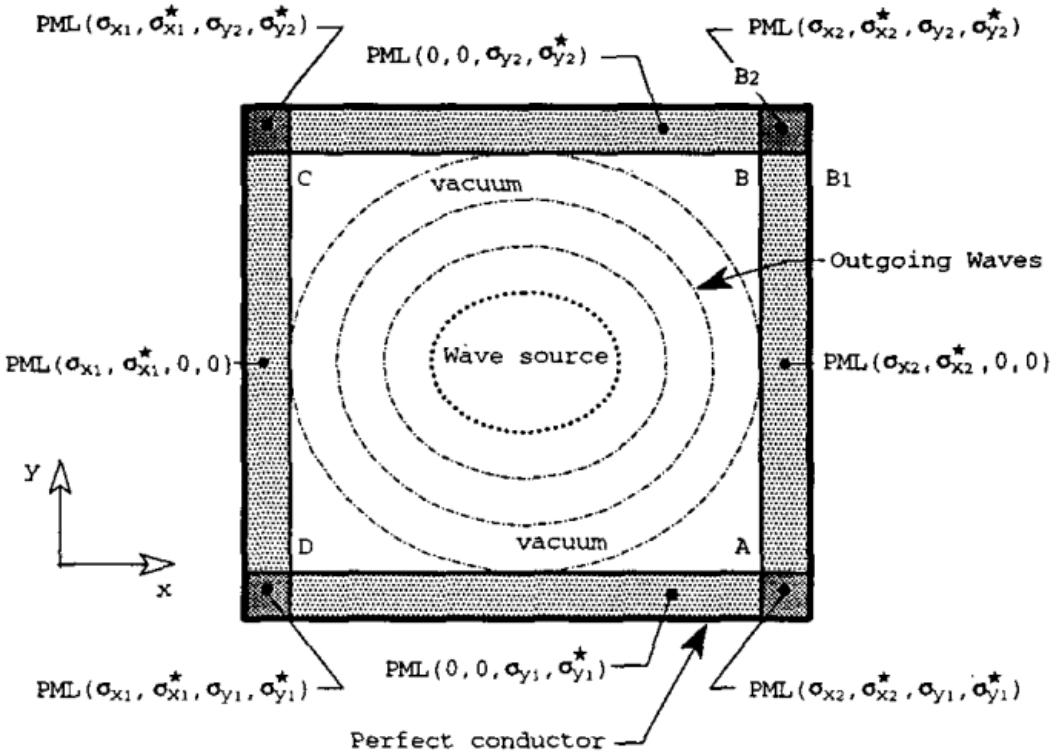


Figure 8: A Problem Domain with PML Regions

A.1.7.3 Periodic Boundary Conditions

In the analysis of periodic isotropic or anisotropic structures such as periodic metamaterials it is useful to implement periodic boundary conditions upon the Yee lattice in the FDFD problem space. Periodic boundary conditions can improve the accuracy of an FDFD solution for complex problems. The most useful method for implementing PBCs in FDFD is to use Bloch-Floquet theory

which states that in a periodic system, for a given mode of steady-state frequency, the fields at one cross section differ from that one period (or integer multiple periods) away by only a complex exponential constant [25]. The first approach is the replacement of an infinite periodic structure by a unit cell. Periodic boundary conditions are imposed on periodic surfaces of the unit cell. PBCs are then applied to rotation matrices of the electric and magnetic field [25].

The implementation of a Floquet Field Expansion into the FDFD formulation can deliver more accurate simulations of infinite periodic structures. The Floquet Theorem is a discrete spatial Fourier transform. The electric and magnetic fields can be expressed as

$$E_{(x,y,z)} = \sum_{n=-\infty}^{n=\infty} C_{n_E} e^{-j\beta_n z}$$

$$\boldsymbol{H_{(x,y,z)}} = \sum_{\boldsymbol{n=-\infty}}^{\boldsymbol{n=\infty}} \boldsymbol{C_{n_H} e^{-j\beta_n z}}$$

where $C_{n_{E,H}}$ are called Floquet coefficients

$$C_{n_E} = \frac{1}{p}\int_0^p E_{(x,y,z)} e^{j\beta_n z} dz$$

$$C_{n_H} = \frac{1}{p}\int_0^p H_{(x,y,z)} e^{j\beta_n z} dz$$

Where β_n is the Floquet wavenumber which is equal to

$$\beta_n = \beta_0 + \frac{2n\pi}{p}$$

and n is the number of spatial harmonics of the Floquet series [25].

A.1.9 Pre-Conditioner

The terms “pre-conditioning” refers to “the art of transforming a problem that appears intractable into another whose solution can be approximated rapidly” [20]. While iterative methods are currently the solvers of choice for large sparse linear systems of equations, it is well known that pre-conditioners are required for the acceleration and convergence of the system. There are many options for choosing an appropriate pre-conditioner, but for CEM in general we need only concern ourselves with algebraic pre-conditioners. The application of pre-conditioners to symmetric positive definite matrices is superior to direct (inversion) methods for large three-dimensional simulations because it requires much less memory.

Pre-conditioning literally means transforming system into an equivalent mathematical problem which is expected to converge faster using an iterative solver. To demonstrate, consider a non-singular matrix M, pre-multiplying by M^{-1} yields

$$M^{-1}A\boldsymbol{x} = M^{-1}\boldsymbol{b}$$

If the matrix $M^{-1}A$ is better conditioned than A for a Krylov subspace method, then M^{-1} is the pre-conditioner, and the above equations denotes the left pre-conditioned system. We can also apply M^{-1} to the right side of the equation:

$$AM^{-1}\boldsymbol{y} = \boldsymbol{b}, \qquad \boldsymbol{x} = M^{-1}\boldsymbol{y}$$

producing a right pre-conditioned system. Also, if the pre-conditioner can be written in the factorized form:

$$M^{-1} = M_2^{-1}M_1^{-1}$$

a split preconditioning can also be used:

$$M_1^{-1}AM_2^{-1}\mathbf{y} = M_1^{-1}\boldsymbol{b}, \qquad \mathbf{x} = M_2^{-1}\mathbf{y}$$

thus, potentially exploiting the advantages of both left and right formulations. There are three basic requirements for obtaining a good pre-conditioner [20]:

(i) The pre-conditioned matrix should have a clustered eigenspectum away from zero.

(ii) The pre-conditioner should be cheap to compute as possible.

(iii) Its application to a vector should be cost effective.

The development and implementation of efficient pre-conditioners are crucial for improving the performance and robustness of Krylov subspace methods in the solution of large sparse linear systems. Algebraic pre-conditioner is used because they can effectively be used as a black-box tool in FDFD requiring knowledge of the system matrix only. A classification of current algebraic pre-conditioners from [20] is as follows:

(i) Incomplete factorizations

(ii) Approximate inverses

(iii) Algebraic multigrid

(iv) Parallel preconditioners

All categories in this research are considered for the FDFD method with most attention given to incomplete factorizations such as in [21] and approximate inverses such as in [12] and algebraic multigrid. While each type of pre-conditioner is at various stages of maturity in terms of research and application, each can be applied to a computational electromagnetic problem.

A.1.10 Krylov Subspace Methods

In 1931 Alexi Krylov published his seminal paper "On the numerical solution of the equation by which the frequency of small oscillations is determined in technical problems" [15]. This paper proved to be the jumping board for all modern iterative methods used in CEM today. Krylov's motivation was application in the analysis of oscillations of mechanical systems and constructed a method for computing the minimal polynomial of a matrix [16]. His method is based on the following: Given $A \in \mathbb{F}^{NxN}$ and a non-zero vector $v \in \mathbb{F}^N$, conside the Krylove sequence generated by A and v,

$$v, Av, A^2v, \ldots$$

There then exists a uniquely defined integer d = d(A,v), so that the vectors v,…, A^{d-1}v are linearly independent, and the vectors v,…, A^{d-1}v, A^dv are linearly dependent. We always have d ≥ 1 since v is non-zero, and d ≤ N since the N+1 vectors v, Av,…,A^Nv ∈ $\mathbb{F}^N$ must be linearly dependent.

By construction, there exists scalars $\gamma_0,\ldots, \gamma_{d-1}$ with

$$A^d v = \sum_{j=0}^{d-1} \gamma_j A^j v.$$

Here A^dv is either the zero vector, or A^dv is a nontrivial linear combination of the linear independent vectors v,..., A^{d-1}v. We can write

$$p(A)v = 0, where\ p(\gamma) \equiv \gamma^d - \sum_{j=0}^{d-1} \gamma_j \lambda^j.$$

The polynomial p(λ) is called the minmal polynomial of v with respect to A. Its degree d = d(A,v) is called the grade of v with respect to A [16].

Later, the characterization of Krylov subspace methods as projection methods evolved as pioneered by Saad in the 1980s [17, 18]. Krylov's method can be restated in this more commonly used formulation. For each matrix A and vector v, the nested sequence of Krylov subspaces defined by

$$\mathcal{K}_n(A, v) \equiv span\{v, Av, \ldots, A^{n-1}v\}, for\ n\ =\ 1,2,\ldots$$

will eventually stop to grow and become invariant under A. If d is the grade of v with respect to A, then

$$A^d v \in \mathcal{K}_d(A, v)$$

and hence,

$$A\mathcal{K}_d(A, v) \subseteq \mathcal{K}_d(A, v)$$

For technical reasons we define

$$\mathcal{K}_0(A, v) = 0$$

This mathematical foundation is the basis for the most used iterative methods used in CEM today such as, Bi-CG, Bi-CGSTAB, GMRES, QMR, CG, CG-S, MINRES, among others. Depending on the solving algorithm, as well as the architecture of the system simulations are being performed other methods may be preferable. In this research the preferred method is the Bi-CGSTAB method is utilized due its efficiency and ability to perform computations without transposition, inversion, and the expensive nature of memory.

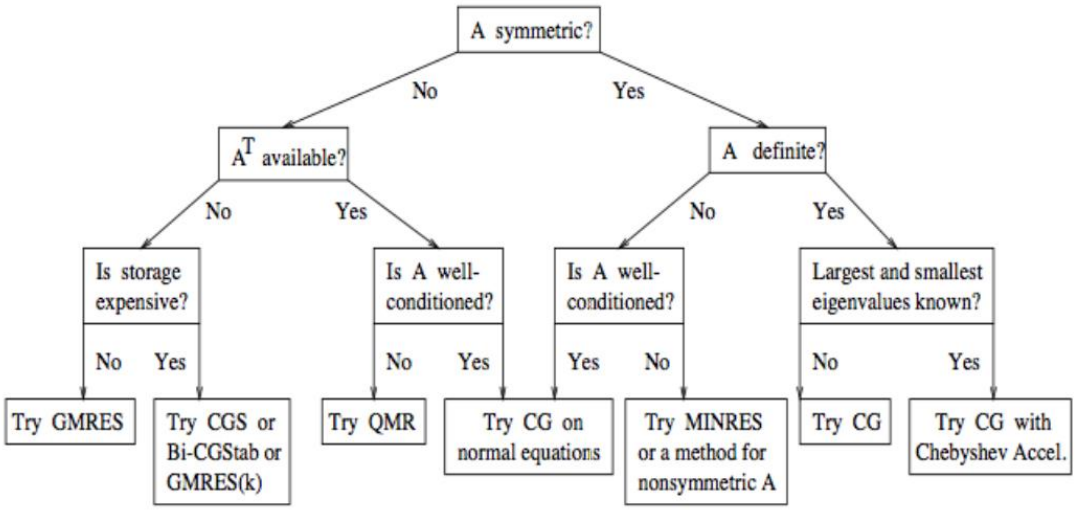
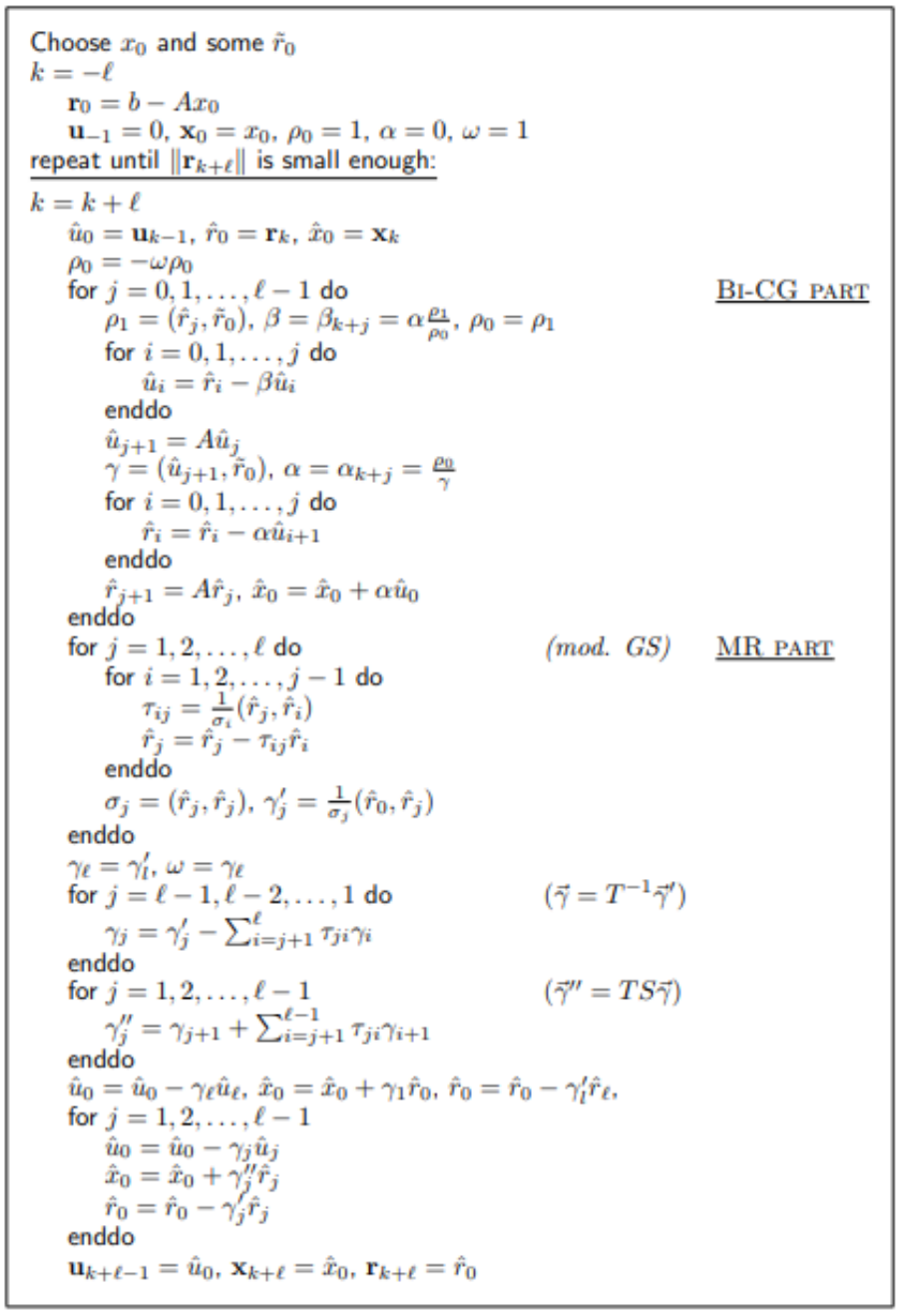


Figure 9: Krylov Subspace Decision Tree [22]

For most linear systems of equations arising from realistic electromagnetic problems, the Bi-CGSTAB algorithm of van der Vorst [19] to solve these equations is most attractive due to irregular convergence behavior found in other methods such as rounding errors causing resulting in severe cancellation effects in the solution [19]. The Bi-CGSTAB method does not suffer from these negative effects and is much more efficient for electromagnetic problems.



Choose x_0 and some $\tilde{r}_0$
$k = -\ell$
$\mathbf{r}_0 = b - Ax_0$
$\mathbf{u}_{-1} = 0$, $\mathbf{x}_0 = x_0$, $\rho_0 = 1$, $\alpha = 0$, $\omega = 1$
repeat until $\|\mathbf{r}_{k+\ell}\|$ is small enough:

$k = k + \ell$
$\hat{u}_0 = \mathbf{u}_{k-1}$, $\hat{r}_0 = \mathbf{r}_k$, $\hat{x}_0 = \mathbf{x}_k$
$\rho_0 = -\omega\rho_0$
for $j = 0, 1, \ldots, \ell - 1$ do — BI-CG PART
  $\rho_1 = (\hat{r}_j, \tilde{r}_0)$, $\beta = \beta_{k+j} = \alpha\frac{\rho_1}{\rho_0}$, $\rho_0 = \rho_1$
  for $i = 0, 1, \ldots, j$ do
    $\hat{u}_i = \hat{r}_i - \beta\hat{u}_i$
  enddo
  $\hat{u}_{j+1} = A\hat{u}_j$
  $\gamma = (\hat{u}_{j+1}, \tilde{r}_0)$, $\alpha = \alpha_{k+j} = \frac{\rho_0}{\gamma}$
  for $i = 0, 1, \ldots, j$ do
    $\hat{r}_i = \hat{r}_i - \alpha\hat{u}_{i+1}$
  enddo
  $\hat{r}_{j+1} = A\hat{r}_j$, $\hat{x}_0 = \hat{x}_0 + \alpha\hat{u}_0$
enddo
for $j = 1, 2, \ldots, \ell$ do — *(mod. GS)* — MR PART
  for $i = 1, 2, \ldots, j - 1$ do
    $\tau_{ij} = \frac{1}{\sigma_i}(\hat{r}_j, \hat{r}_i)$
    $\hat{r}_j = \hat{r}_j - \tau_{ij}\hat{r}_i$
  enddo
  $\sigma_j = (\hat{r}_j, \hat{r}_j)$, $\gamma'_j = \frac{1}{\sigma_j}(\hat{r}_0, \hat{r}_j)$
enddo
$\gamma_\ell = \gamma'_l$, $\omega = \gamma_\ell$
for $j = \ell - 1, \ell - 2, \ldots, 1$ do — $(\vec{\gamma} = T^{-1}\vec{\gamma}')$
  $\gamma_j = \gamma'_j - \sum_{i=j+1}^{\ell} \tau_{ji}\gamma_i$
enddo
for $j = 1, 2, \ldots, \ell - 1$ — $(\vec{\gamma}'' = TS\vec{\gamma})$
  $\gamma''_j = \gamma_{j+1} + \sum_{i=j+1}^{\ell-1} \tau_{ji}\gamma_{i+1}$
enddo
$\hat{u}_0 = \hat{u}_0 - \gamma_\ell\hat{u}_\ell$, $\hat{x}_0 = \hat{x}_0 + \gamma_1\hat{r}_0$, $\hat{r}_0 = \hat{r}_0 - \gamma'_l\hat{r}_\ell$,
for $j = 1, 2, \ldots, \ell - 1$
  $\hat{u}_0 = \hat{u}_0 - \gamma_j\hat{u}_j$
  $\hat{x}_0 = \hat{x}_0 + \gamma''_j\hat{r}_j$
  $\hat{r}_0 = \hat{r}_0 - \gamma'_j\hat{r}_j$
enddo
$\mathbf{u}_{k+\ell-1} = \hat{u}_0$, $\mathbf{x}_{k+\ell} = \hat{x}_0$, $\mathbf{r}_{k+\ell} = \hat{r}_0$

Figure 9: Bi-CGSTAB Algorithm [23]

A.1.10 FDFD Algorithm

The solution to any efficient FDFD problem space utilizes both iterative methods as well as pre-conditioning technique. With the most efficient iterative method generally being Bi-GCSTAB for solving systems composed of Maxwell's equations, pre-conditioning can be very costly and requires extensive apriori research into the physical system being described in the problem space. Along with iterative methods and pre-conditioning techniques, the solution to an FDFD problem is based on solving a system of linear equations of the form:

$$A\mathbf{x} = \boldsymbol{b}$$

where A is the coefficient matrix, x is the vector of unknown fields, and b is the right-side excitation vector. An iterative solver begins with an initial guess x_0 and tried to minimize the residual $r_k = b - Ax_k = b - b'$ as the iteration proceed, where x_k is the solution at the k^{th} iteration. As the residual minimizes, the vector x_k converges to the solution vector x.

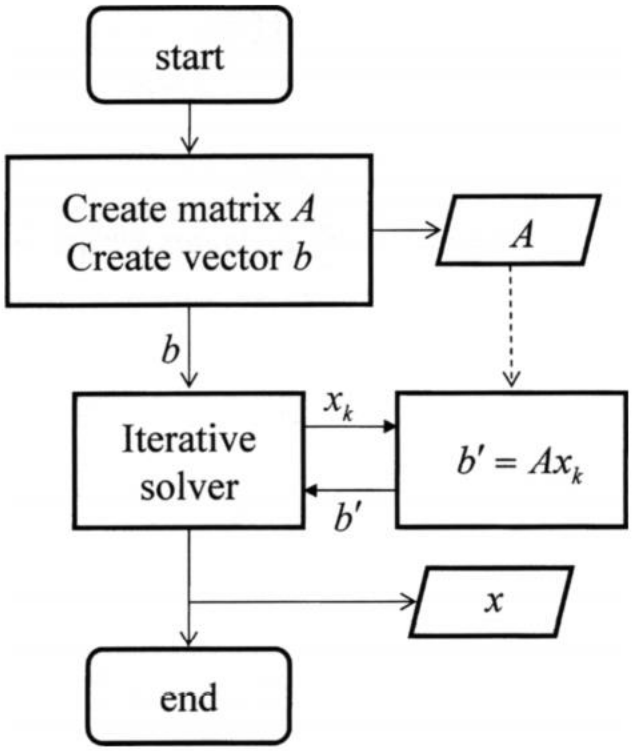


Figure 10: Procedure of Calling the Iterative Solver

The iteration procedure continues until the convergence criteria is met. Although a linear system can be expressed by a linear matrix equation, during the iterative procedure, it is not necessary to use the actual matrix A to store the coefficients and multiply it by x_k to calculate b'. It is only required to obtain b' for a given x_k. The $A\mathbf{x}_k$ matrix-vector product is replaced by a multi-step algorithm as described in [24].

In the multi-step algorithm, the total number of coefficients that need to be stored reduced to 24N, which is a significant reduction from 39N. By recognizing that the electric scattered fields have four coefficients in the equation, there are only two coefficient pairs differing only by their signs. This reduced the total number of field coefficients to 12N. The new algorithm coupled with the reduction of coefficients and three-dimensional storage format has the following advantages [4]:

(i) The number of coefficients to be stored is reduced by half, for a total of 12N coefficients instead of 24N.

(ii) The use of one-dimensional row and column arrays is eliminated. The amount of memory needed per coefficient will drop to 16 bytes from 24 bytes. The total saving becomes almost 80% relative to an FDFD method not storing modified coefficients.

(iii) By using FORTAN, we do not need to complete the matrix-vector product within a for loop and instead use array operations. The electric fields are also stored in three-dimensional arrays.

```
subroutine matvec(n,x,y)
use global
implicit none
integer n;
complex*16 x(Nx,Ny,Nz,3), y(Nx,Ny,Nz,3);
tmpx(:,1:Ny-1,1:Nz-1) = &
    Chxez(:,1:Ny-1,1:Nz-1) &
        *(+x(:,2:Ny,1:Nz-1,3) &
        -x(:,1:Ny-1,1:Nz-1,3))&
    + Chxey(:,1:Ny-1,1:Nz-1) &
        *(-x(:,1:Ny-1,2:Nz,2) &
        +x(:,1:Ny-1,1:Nz-1,2));
...
...
y(1:Nx,2:Ny,2:Nz,1) = &
        x(1:Nx,2:Ny,2:Nz,1) &
    -(Cexhz(1:Nx,2:Ny,2:Nz) &
        *(-tmpz(1:Nx,2:Ny,2:Nz) &
        +tmpz(1:Nx,1:Ny-1,2:Nz)) &
    + Cexhy(1:Nx,2:Ny,2:Nz) &
        *(+tmpy(1:Nx,2:Ny,2:Nz) &
        -tmpy(1:Nx,2:Ny,1:Nz-1)));
...
...
end subroutine matvec
```

Figure 11: 4D matvec Subroutine in FORTRAN.