ABSTRACT

IMPROVEMENT OF FDFD METHOD IN FORTRAN

John Michael Garofalo
Department of Electrical Engineering
Northern Illinois University, 2022
Dr. Veysel Demir, Director

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 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 shown that by implementing an enhanced divergence-based field equations in Fortran derived from the Maxwell equations, convergence rates of the algorithm can be increased, as well as increase the accuracy of radar cross section computations. This research improves the computational efficiency and accuracy of the FDFD method.

NORTHERN ILLINOS UNIVERSITY
DEKALB, ILLINOIS

MAY 2022

IMPROVEMENT OF FDFD METHOD IN FORTRAN

BY

JOHN MICHAEL GAROFALO

A PROJECT REPORT SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE

MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL ENGINEERING

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

DEDICATION

The research and work contained herein could not have been made possible without the unconditional love of my family, professors, and all like-minded kin I have had the pleasure of acquaintance in my brief journey to the current destination. Especially to my father, whose undying commitment to see me succeed, and infinite oversight of my well-being couldn’t have been made possible without. To all else I thank, and will remember in gratitude and cherish your memory, strength, and love forever.

ACKNOWLEDGEMENTS

I would like to thank all my professors in the Northern Illinois University Electrical Engineering and Physics Departments and for giving me a newfound perspective and vision, and especially Dr. Veysel Demir for his superior academic guidance and encouragement in the present subject. Of course, this could not have been made possible without all the great men and women who have come before, and continue to contribute to the forefront of our ever-expanding knowledge of science, mathematics, and engineering.

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, 1854

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.

1.2 Time Domain Vs. Frequency Domain

All computational electromagnetic algorithms use numerical methods to solve the Maxwell 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].

1.3 Finite Differences

Based on the seminal work by Yee [7], the Finite-Difference Time-Domain, and later the Finite-Difference Frequency-Domain by McCartin [8], 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 the Maxwell 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 Maxwell 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 and computational efficiency 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.

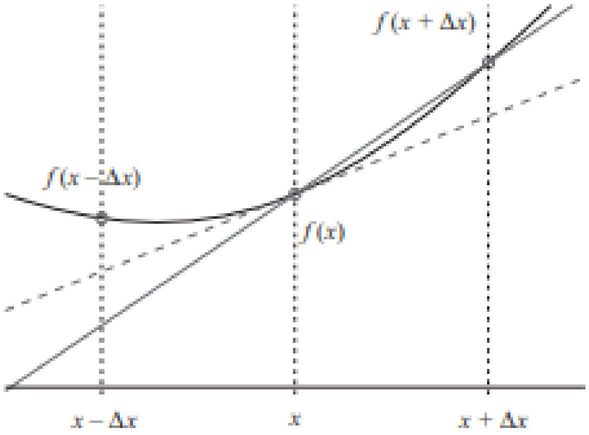

Figure 1: Forward-difference [3].

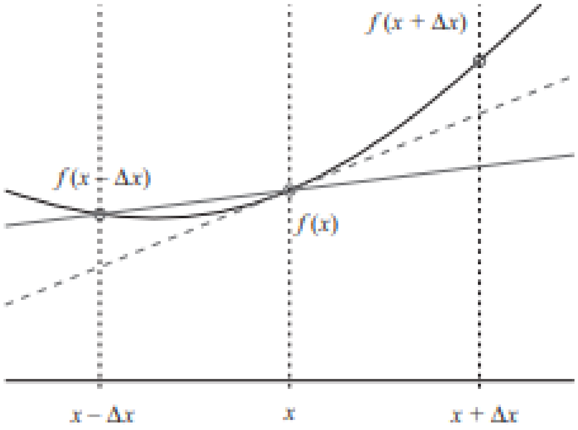

Figure 2: Backward-difference [3].

The backward and forward finite difference operators are applied to the discretized total-field Maxwell equations yielding six total equations for the electric and magnetic fields both in the x, y, and z directions.

Traditionally, once the set of discrete linear equations is obtained from the Maxwell equations, a linear matrix equation is formulated:

$$\mathrm{A}\mathbf{x} = \mathrm{F}$$

Equation 1: Linear Matrix Equation

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 coefficient matrix A is a sparse band matrix containing mostly zeros due to the nature of PDEs being loosely coupled in electromagnetic problems. These bands are vectorized directly from a three-dimensional problem space in the Fortran implementation of FDFD. 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 [5]. 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 also very easy to apply to non-uniform media, enabling material generality to be broad [5]. Geometric versatility is very good, allowing FDFD to model complex-shaped and inhomogeneous structures [5]. 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 [5].

1.4 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 $\varepsilon_{N.}$" 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].

1.5 Absorbing 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 [3]. 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 [3].

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.

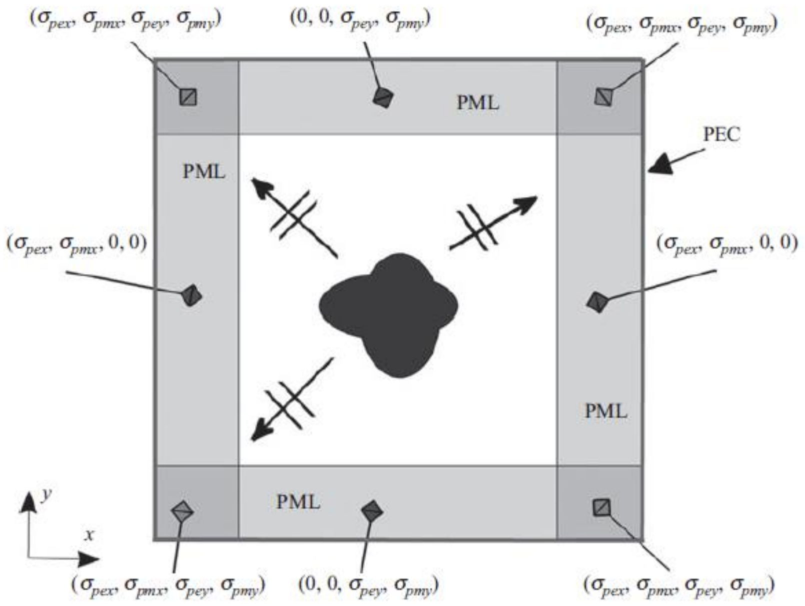
$(\sigma_{pex}, \sigma_{pmx}, \sigma_{pey}, \sigma_{pmy})$
$(0, 0, \sigma_{pey}, \sigma_{pmy})$
$(\sigma_{pex}, \sigma_{pmx}, \sigma_{pey}, \sigma_{pmy})$
PML
PEC
PML
$(\sigma_{pex}, \sigma_{pmx}, 0, 0)$
$(\sigma_{pex}, \sigma_{pmx}, 0, 0)$
PML
PML
y
x
$(\sigma_{pex}, \sigma_{pmx}, \sigma_{pey}, \sigma_{pmy})$
$(0, 0, \sigma_{pey}, \sigma_{pmy})$
$(\sigma_{pex}, \sigma_{pmx}, \sigma_{pey}, \sigma_{pmy})$

Figure 3: Diagram of PML [3]

Among all the ABCs in existence the perfectly matched layer (PML) as proposed by Berenger in [4] is the most effective ABC used to terminate the computational space in the finite difference method. The PML technique is based on a split-field formulation where each vector field component in the Maxwell equations is split into two orthogonal components. The problem space is surrounded by a matched material that has both electrical and magnetic PML conductivities used in the finite difference discrete Maxwell equations within the PML region to aid in effective annihilation of arbitrary scattered waveforms at the domain boundary.

The layer surrounding the problem domain can theoretically absorb all waves traveling toward the boundaries without any reflection [5]. 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 2: Retained Electric Conductivity and Magnetic Loss Ratio in PML Method [5].

in the direction normal to the computational boundary [4]. 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.

To reduce the reflections from the PML layer, both the electric and magnetic conductivities are chosen to increase from zero at the vacuum-PML interface to a value σ_{max} at the outer layer of the PML [5]. σ_{max} can be determined from [4] as

$$\sigma_{max} = -\frac{\left(n_{pml} + 1\right)\varepsilon_0 c ln\left(R(0)\right)}{2\delta_{pml}}$$

Equation 3: Power Increasing PML Loss Function [6]

where n_{pml} is the degree of the increment of loss of PML layers [5]. The loss is linear for n_{pml} = 1 and parabolic for n_{pml} = 2 within the PML layers [5]. 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 [5].

$$R(\varphi_0) = e^{-2\frac{\sigma cos(\varphi_0)}{\varepsilon_0 c}\delta_{pml}}$$

Equation 4: Theoretical Reflection Factor [6]

The conductivity inside the absorbing layers is determined by

$$\sigma(\rho) = \sigma_{max}\left(\frac{\rho}{\delta_{pml}}\right)^{n_{pml}}$$

Equation 5: Power Increasing PML Loss Function [6] where ρ is the distance from the vacuum-PML interface to a point inside the PML media [5]. Optimization of PML parameters for a problem space can result in increased accuracy of radar bi-static cross section (RCS) computation, and dramatic decreases in FDFD execution time.

1.6 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” [10]. This paper proved to be the launching ramp for all modern iterative methods used in CEM today. Krylov’s motivation was applied in the analysis of oscillations of mechanical systems and constructed a method for computing the minimal polynomial of a matrix [11]. His method is based on the following: Given $A \in \mathbb{F}^{NxN}$ and a non-zero vector $v \in \mathbb{F}^{N}$, consider the Krylov sequence generated by A and v,

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

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,…, $A^{d-1}v$, $A^{d}v$ are linearly dependent. We always have d ≥ 1 since v is non-zero, and d ≤ N since the N+1 vectors v, Av,…,$A^{N}v \in \mathbb{F}^{N}$ must be linearly dependent.

By construction, there exists scalars γ_0,…, γ_{d-1} with

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

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

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

The polynomial p(λ) is called the minimal polynomial of v with respect to A. Its degree d = d(A,v) is called the grade of v with respect to A [11].

Later, the characterization of Krylov subspace methods as projection methods evolved as pioneered by Saad in the 1980s [12, 13]. 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

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

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, and GMRES among others. Depending on the solving algorithm, as well as the architecture of the system that the simulations are being performed, other methods may be preferable. In this research the Bi-CGSTAB(2) method of Fokkema is utilized due its efficiency and ability to perform computations without transposition, inversion, and the expensive nature of memory [17].

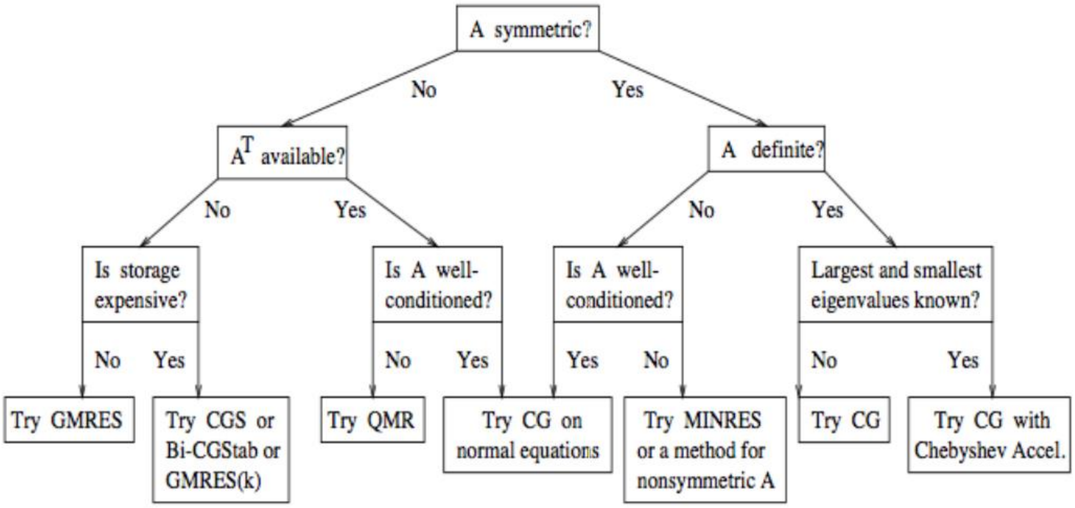
A symmetric?
No
Yes
A^T available?
A definite?
No
Yes
No
Yes
Is storage expensive?
Is A well-conditioned?
Is A well-conditioned?
Largest and smallest eigenvalues known?
No
Yes
No
Yes
Yes
No
No
Yes
Try GMRES
Try CGS or Bi-CGStab or GMRES(k)
Try QMR
Try CG on normal equations
Try MINRES or a method for nonsymmetric A
Try CG
Try CG with Chebyshev Accel.

Figure 4: Krylov Subspace Decision Tree [14]

For most linear systems of equations arising from realistic electromagnetic problems, the Bi-CGSTAB algorithm of van der Vorst [15] to solve these equations is most attractive due to irregular convergence behavior found in other methods such as rounding errors resulting in severe cancellation effects in the solution [15]. The Bi-CGSTAB, and Bi-CGSTAB(2) does not suffer from these negative effects and is much more efficient for electromagnetic problems.

2 WORKS DONE

This work integrates a unique coefficient array representation building from discretized Maxwell differential equations to enhance convergence rate of the FDFD algorithm in Fortran. We have shown that this implementation accurately and efficiently simulates canonical dielectric or perfectly conducting surfaces while appropriately annihilating waveforms at the domain boundaries.

We define our metric for FDFD accuracy as a comparison to the analytic bi-static RCS calculations of [16]. The equivalence principle states that "each point on a primary wavefront can be considered to be a new source of a secondary spherical wave and that a secondary wavefront can be constructed as the envelope of these secondary spherical waves [18]." This principle is based on the uniqueness theorem which states "a field in a lossy region is uniquely specified by the sources within the region plus the tangential components of the electric field over the boundary, or the tangential components of the magnetic field over the boundary, or the former over part of the boundary and the latter over the rest of the boundary [18]." Since the numerical computations are confined to the scattered field region, Surface Equivalence can model far-fields with high accuracy in the FDFD method.

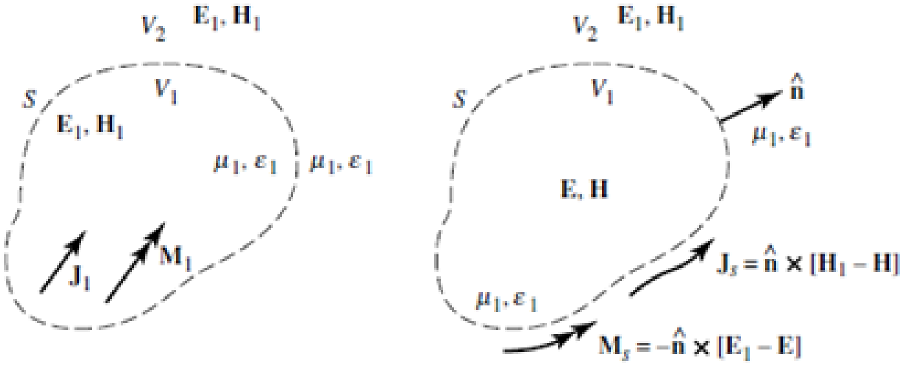
V_2
$\mathbf{E}_1, \mathbf{H}_1$
S
V_1
$\mathbf{E}_1, \mathbf{H}_1$
μ_1, ε_1
μ_1, ε_1
$\mathbf{J}_1$
$\mathbf{M}_1$
V_2
$\mathbf{E}_1, \mathbf{H}_1$
S
V_1
$\hat{\mathbf{n}}$
μ_1, ε_1
$\mathbf{E}, \mathbf{H}$
$\mathbf{J}_s = \hat{\mathbf{n}} \times [\mathbf{H}_1 - \mathbf{H}]$
μ_1, ε_1
$\mathbf{M}_s = -\hat{\mathbf{n}} \times [\mathbf{E}_1 - \mathbf{E}]$

Figure 5: Representation of the Surface Equivalence Principle [18].

2.1 Accuracy and Efficiency Improvement

2.1.1 Divergence Re-Enforcing Field Equations

In all applications of the finite-difference method the solution to the wave equations as applied to a specific set of boundaries in the problem space need to satisfy the Maxwell equations in both space and time as well as their inverted counterparts. Re-enforcing the Maxwell divergence equations at every node in the Yee lattice in the derivation of the updating equations, with or without symmetry breaking of the total field formulation, allows us to calculate field values more accurately and efficiently using BiCGSTAB(2) in linear isotropic mediums.

$$\vec{\mathbf{D}}_{\mathbf{T}} = \varepsilon \vec{\mathbf{E}}_{\mathbf{T}}$$

Equation 6**:** Electric Linear Isotropic Constitutive Relation

$$\vec{\mathbf{B}}_{\mathbf{T}} = \mu \vec{\mathbf{H}}_{\mathbf{T}}$$

Equation 7**:** Magnetic Linear Isotropic Constitutive Relation

$$\vec{\mathbf{H}}_{\mathbf{T}} = \vec{\mathbf{H}}_{\mathbf{scat}} + \vec{\mathbf{H}}_{\mathbf{inc}}$$

Equation 8: Total Field Formulation for Magnetism

$$\vec{\mathbf{E}}_{\mathbf{T}} = \vec{\mathbf{E}}_{\mathbf{scat}} + \vec{\mathbf{E}}_{\mathbf{inc}}$$

Equation 9: Total Field Formulation for Electricity

$$\vec{\nabla} \cdot \mu \vec{\mathbf{H}}_{\mathbf{T}} = 0$$

Equation 10: Gauss's Law for Magnetism

$$\vec{\nabla} \cdot \varepsilon \vec{\mathbf{E}}_{\mathbf{T}} = 0$$

Equation 11: Gauss's Law for Electric Displacement and Zero Free Charges

$$\nabla \times \vec{\mathbf{E}}_{\mathbf{T}} + \vec{\nabla}\left(\vec{\nabla} \cdot \varepsilon \vec{\mathbf{E}}_{\boldsymbol{T}}\right) = -j\omega\mu\vec{\mathbf{H}}_{\mathbf{T}}$$

Equation 12: Faraday's Law of Induction with Divergence Re-Enforced [9]

$$\nabla \times \vec{\mathbf{H}}_{\mathbf{T}} + \vec{\nabla}\left(\vec{\nabla} \cdot \mu \vec{\mathbf{H}}_{\mathbf{T}}\right) = j\omega\varepsilon\vec{\mathbf{E}}_{\mathbf{T}}$$

Equation 13: Ampere's Circuital Law with Divergence Re-Enforced [9]

In the derivation of improved Finite-Difference Frequency-Domain equations it is important to consider causality as well as symmetry in the emergent behavior of electric and magnetic fields in the total field formulation, as well as cost effective derivative operators to apply in the discretization process.

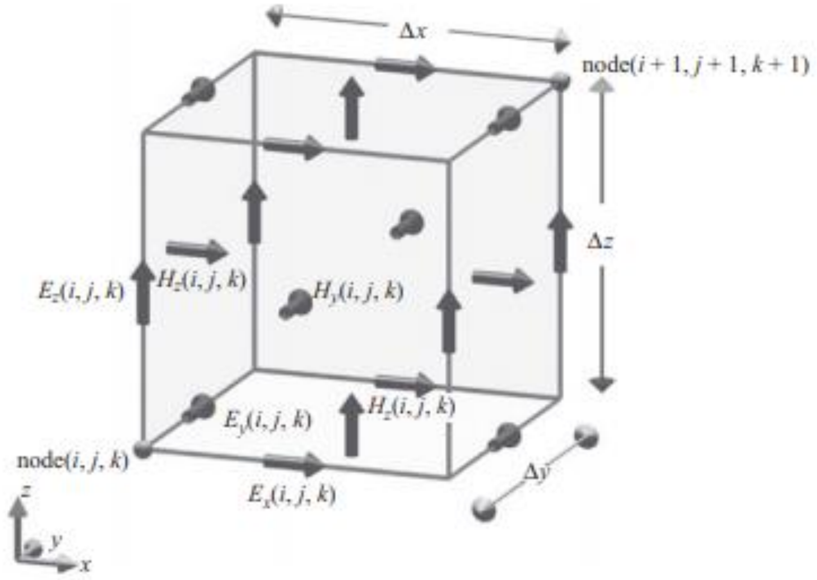

Figure 6: Arrangement of field values on a Yee cell indexed as (i, j, k) [6].

We can find coefficients with the Maxwell electric curl equation. For instance, in the x-direction we have:

$$\nabla \times \vec{\mathbf{E}}_{\mathbf{T}} + diag\left(\vec{\nabla}\left(\vec{\nabla} \cdot \varepsilon \vec{\mathbf{E}}_{\mathbf{T}}\right)\right) = -j\omega\mu\vec{\mathbf{H}}_{\mathbf{T}}$$

$$\nabla \times \vec{\mathbf{E}}_{\mathbf{inc}} + \nabla \times \vec{\mathbf{E}}_{\mathbf{scat}} + diag\left(\vec{\nabla}\left(\vec{\nabla} \cdot \varepsilon_{xi} \vec{\mathbf{E}}_{\mathbf{inc}}\right)\right) + diag\left(\vec{\nabla}\left(\vec{\nabla} \cdot \varepsilon_{xi} \vec{\mathbf{E}}_{\mathbf{scat}}\right)\right) = -j\omega\mu\vec{\mathbf{H}}_{\mathbf{inc}} - j\omega\mu\vec{\mathbf{H}}_{\mathbf{scat}}$$

$$-j\omega\mu_0\vec{\mathbf{H}}_{\mathbf{inc}} + diag\left(\vec{\nabla}\left(\vec{\nabla} \cdot \varepsilon_{xi} \vec{\mathbf{E}}_{\mathbf{inc}}\right)\right) + \nabla \times \vec{\mathbf{E}}_{\mathbf{scat}} + diag\left(\vec{\nabla}\left(\vec{\nabla} \cdot \varepsilon_{xi} \vec{\mathbf{E}}_{\mathbf{scat}}\right)\right) = -j\omega\mu_{xi}\vec{\mathbf{H}}_{\mathbf{inc}} - j\omega\mu_{xi}\vec{\mathbf{H}}_{\mathbf{scat}}$$

$$\nabla \times \vec{\mathbf{E}}_{\mathbf{scat}} + diag\left(\vec{\nabla}\left(\vec{\nabla} \cdot \varepsilon_{xi} \vec{\mathbf{E}}_{\mathbf{inc}}\right)\right) + diag\left(\vec{\nabla}\left(\vec{\nabla} \cdot \varepsilon_{xi} \vec{\mathbf{E}}_{\mathbf{scat}}\right)\right) + j\omega\mu_{xi}\vec{\mathbf{H}}_{\mathbf{scat}} = j\omega\mu_0\vec{\mathbf{H}}_{\mathbf{inc}} - j\omega\mu_{xi}\vec{\mathbf{H}}_{\mathbf{inc}}$$

$$\frac{\partial}{\partial x}\frac{\partial}{\partial x}\varepsilon_{xi}E_{x,scat} + \frac{\partial}{\partial y}E_{z,scat} - \frac{\partial}{\partial z}E_{y,scat} + \frac{\partial}{\partial x}\frac{\partial}{\partial x}\varepsilon_{xi}E_{x,inc} + j\omega\mu_{xi}H_{x,scat} = j\omega(\mu_0 - \mu_{xi})H_{x,inc}$$

The diagonal terms in the permeability dyadic become off-diagonal terms in the derivative.

$$\frac{\varepsilon_{xi}}{j\omega\mu_{xi}}\frac{\partial}{\partial x}\frac{\partial}{\partial x}E_{x,scat}+\frac{1}{j\omega\mu_{xy}}\frac{\partial}{\partial y}E_{z,scat}-\frac{1}{j\omega\mu_{xz}}\frac{\partial}{\partial z}E_{y,scat}+\frac{\varepsilon_{xi}}{j\omega\mu_{xi}}\frac{\partial}{\partial x}\frac{\partial}{\partial x}E_{x,inc}+$$

$$H_{x,scat}=\frac{\mu_0-\mu_{xi}}{\mu_{xi}}H_{x,inc}$$

Discretizing, we have:

<u>X-Direction</u>:

$$\frac{\varepsilon_{xi}(i,j,k)}{j\omega\mu_{xi}(i,j,k)\Delta x}\cdot\Big(E_{x,scat}(i+2,j,k)-E_{x,scat}(i+1,j,k)+E_{x,scat}(i,j,k)\Big)+$$

$$\frac{1}{j\omega\mu_{xy}(i,j,k)\Delta y}\cdot\Big(E_{z,scat}(i,j+1,k)-E_{z,scat}(i,j,k)\Big)-$$

$$\frac{1}{j\omega\mu_{xz}(i,j,k)\Delta z}\cdot\Big(E_{y,scat}(i,j,k+1)-E_{y,scat}(i,j,k)\Big)+$$

$$\frac{\varepsilon_{xi}(i,j,k)}{j\omega\mu_{xi}(i,j,k)\Delta x}\cdot\Big(E_{x,inc}(i+2,j,k)-E_{x,inc}(i+1,j,k)+E_{x,inc}(i,j,k)\Big)+$$

$$H_{x,scat}(i,j,k)=\frac{\mu_0(i,j,k)-\mu_{xi}(i,j,k)}{\mu_{xi}(i,j,k)}H_{x,inc}(i,j,k)$$

Our coefficients for magnetic incidence in the x-direction become:

$\mathbf{C_{hxhx}}=\dfrac{\mu_0(i,j,k)-\mu_{xi}(i,j,k)}{\mu_{xi}(i,j,k)}$

$\mathbf{C_{hxei}}=\dfrac{\varepsilon_{xi}(i,j,k)}{j\omega\mu_{xi}(i,j,k)\Delta x}$

$\mathbf{C_{hxex}}=\dfrac{\varepsilon_{xi}(i,j,k)}{j\omega\mu_{xi}(i,j,k)\Delta x}$

$\mathbf{C_{hxey}}=\dfrac{1}{j\omega\mu_{xz}(i,j,k)\Delta z}$

$\mathbf{C_{hxez}}=\dfrac{1}{j\omega\mu_{xy}(i,j,k)\Delta y}$

Similarly, we can find the coefficient in the remaining incidence directions using the same method in the y-direction and z-direction, respectively.

<u>Y-Direction</u>:

$$\frac{\varepsilon_{yi}(i,j,k)}{j\omega\mu_{yi}(i,j,k)\Delta y}\cdot\Big(E_{y,scat}(i,j+2,k)-E_{y,scat}(i,j+1,k)+E_{y,scat}(i,j,k)\Big)+$$
$$\frac{1}{j\omega\mu_{yx}(i,j,k)\Delta x}\cdot\Big(E_{z,scat}(i+1,j,k)-E_{z,scat}(i,j,k)\Big)-$$
$$\frac{1}{j\omega\mu_{yz}(i,j,k)\Delta z}\cdot\Big(E_{x,scat}(i,j,k+1)-E_{x,scat}(i,j,k)\Big)+$$
$$\frac{\varepsilon_{yi}(i,j,k)}{j\omega\mu_{yi}(i,j,k)\Delta y}\cdot\Big(E_{y,inc}(i,j+2,k)-E_{y,inc}(i,j+1,k)+E_{y,inc}(i,j,k)\Big)+$$
$$H_{y,scat}(i,j,k)=\frac{\mu_0(i,j,k)-\mu_{yi}(i,j,k)}{\mu_{yi}(i,j,k)}H_{y,inc}(i,j,k)$$

Our coefficients for magnetic incidence in the y-direction become:

$\mathbf{C_{hyhy}}=\dfrac{\mu_0(i,j,k)-\mu_{yi}(i,j,k)}{\mu_{yi}(i,j,k)}$

$\mathbf{C_{hyex}}=\dfrac{1}{j\omega\mu_{yz}(i,j,k)\Delta z}$

$\mathbf{C_{hyei}}=\dfrac{\varepsilon_{yi}(i,j,k)}{j\omega\mu_{yi}(i,j,k)\Delta y}$

$\mathbf{C_{hyey}}=\dfrac{\varepsilon_{yi}(i,j,k)}{j\omega\mu_{yi}(i,j,k)\Delta y}$

$\mathbf{C_{hyez}}=\dfrac{1}{j\omega\mu_{yx}(i,j,k)\Delta x}$

<u>Z-Direction</u>:

$$\frac{\varepsilon_{zi}(i,j,k)}{j\omega\mu_{zi}(i,j,k)\Delta z}\cdot\Big(E_{z,scat}(i+2,j,k)-E_{z,scat}(i+1,j,k)+E_{z,scat}(i,j,k)\Big)+$$

$$\frac{1}{j\omega\mu_{zx}(i,j,k)\Delta x}\cdot\Big(E_{y,scat}(i+1,j,k)-E_{y,scat}(i,j,k)\Big)-$$

$$\frac{1}{j\omega\mu_{zy}(i,j,k)\Delta y}\cdot\Big(E_{x,scat}(i,j+1,k)-E_{x,scat}(i,j,k)\Big)+$$

$$\frac{\varepsilon_{zi}(i,j,k)}{j\omega\mu_{zi}(i,j,k)\Delta z}\cdot\Big(E_{z,inc}(i+2,j,k)-E_{z,inc}(i+1,j,k)+E_{z,inc}(i,j,k)\Big)+$$

$$H_{z,scat}(i,j,k)=\frac{\mu_0(i,j,k)-\mu_{zi}(i,j,k)}{\mu_{zi}(i,j,k)}H_{z,inc}(i,j,k)$$

Our coefficients for magnetic incidence in the z-direction become:

$\mathbf{C_{hzhz}}=\dfrac{\mu_0(i,j,k)-\mu_{zi}(i,j,k)}{\mu_{zi}(i,j,k)}$

$\mathbf{C_{hzex}}=\dfrac{1}{j\omega\mu_{zy}(i,j,k)\Delta y}$

$\mathbf{C_{hzey}}=\dfrac{1}{j\omega\mu_{zx}(i,j,k)\Delta x}$

$\mathbf{C_{hzei}}=\dfrac{\varepsilon_{zi}(i,j,k)}{j\omega\mu_{zi}(i,j,k)\Delta z}$

$\mathbf{C_{hzez}}=\dfrac{\varepsilon_{zi}(i,j,k)}{j\omega\mu_{zi}(i,j,k)\Delta z}$

We use the Maxwell magnetic curl equation using the same method to find the other coefficients. For instance, in the x-direction we have:

$$\nabla \times \vec{\mathbf{H}}_{\mathbf{T}} + diag\left(\vec{\nabla}\left(\vec{\nabla} \cdot \mu \vec{\mathbf{H}}_{\mathbf{T}}\right)\right) = j\omega\varepsilon \vec{\mathbf{E}}_{\mathbf{T}}$$

$$\nabla \times \vec{\mathbf{H}}_{\mathbf{inc}} + \nabla \times \vec{\mathbf{H}}_{\mathbf{scat}} + diag\left(\vec{\nabla}\left(\vec{\nabla} \cdot \mu_0 \vec{\mathbf{H}}_{\mathbf{inc}}\right)\right) + diag\left(\vec{\nabla}\left(\vec{\nabla} \cdot \mu_{xi} \vec{\mathbf{H}}_{\mathbf{scat}}\right)\right) = j\omega\varepsilon_{xi}\vec{\mathbf{E}}_{\mathbf{inc}} + j\omega\varepsilon_{xi}\vec{\mathbf{E}}_{\mathbf{scat}}$$

$$j\omega\varepsilon_0\vec{\mathbf{E}}_{\mathbf{inc}} + \nabla \times \vec{\mathbf{H}}_{\mathbf{scat}} + diag\left(\vec{\nabla}\left(\vec{\nabla} \cdot \mu_0 \vec{\mathbf{H}}_{\mathbf{inc}}\right)\right) + diag\left(\vec{\nabla}\left(\vec{\nabla} \cdot \mu_{xi} \vec{\mathbf{H}}_{\mathbf{scat}}\right)\right) = j\omega\varepsilon_{xi}\vec{\mathbf{E}}_{\mathbf{inc}} + j\omega\varepsilon_{xi}\vec{\mathbf{E}}_{\mathbf{scat}}$$

$$\nabla \times \vec{\mathbf{H}}_{\mathbf{scat}} + diag\left(\vec{\nabla}\left(\vec{\nabla} \cdot \mu_0 \vec{\mathbf{H}}_{\mathbf{inc}}\right)\right) + diag\left(\vec{\nabla}\left(\vec{\nabla} \cdot \mu_{xi} \vec{\mathbf{H}}_{\mathbf{scat}}\right)\right) + j\omega\varepsilon_{xi}\vec{\mathbf{E}}_{\mathbf{scat}} = j\omega\varepsilon_{xi}\vec{\mathbf{E}}_{\mathbf{inc}} - j\omega\varepsilon_0\vec{\mathbf{E}}_{\mathbf{inc}}$$

$$\mu_{xi}\frac{\partial}{\partial x}\frac{\partial}{\partial x}H_{x,scat} + \frac{\partial}{\partial y}H_{z,scat} - \frac{\partial}{\partial z}H_{y,scat} + \mu_{xi}\frac{\partial}{\partial x}\frac{\partial}{\partial x}H_{x,inc} - j\omega\varepsilon_{xi}E_{x,scat} = j\omega(\varepsilon_{xi} - \varepsilon_0)E_{x,inc}$$

The diagonal terms in the permeability dyadic become off-diagonal terms in the derivative.

$$\frac{\mu_{xi}}{j\omega\varepsilon_{xi}} \cdot \frac{\partial}{\partial x}\frac{\partial}{\partial x}H_{x,scat} + \frac{1}{j\omega\varepsilon_{xy}} \cdot \frac{\partial}{\partial y}H_{z,scat} - \frac{1}{j\omega\varepsilon_{xz}} \cdot \frac{\partial}{\partial z}H_{y,scat} + \frac{\mu_{xi}}{j\omega\varepsilon_{xi}} \cdot \frac{\partial}{\partial x}\frac{\partial}{\partial x}H_{x,inc} - E_{x,scat} = \frac{(\varepsilon_{xi} - \varepsilon_0)}{\varepsilon_{xi}}E_{x,inc}$$

Discretizing, we have:

X-Direction:

$$\frac{\mu_{xi}(i,j,k)}{j\omega\varepsilon_{xi}(i,j,k)\Delta x}\cdot\Big(H_{x,scat}(i,j,k)-H_{x,scat}(i-1,j,k)+H_{x,scat}(i-2,j,k)\Big)+$$

$$\frac{1}{j\omega\varepsilon_{xy}(i,j,k)\Delta y}\cdot\Big(H_{z,scat}(i,j,k)-H_{z,scat}(i,j-1,k)\Big)-$$

$$\frac{1}{j\omega\varepsilon_{xz}(i,j,k)\Delta z}\cdot\Big(H_{y,scat}(i,j,k)-H_{y,scat}(i,j,k-1)\Big)+$$

$$\frac{\mu_{xi}(i,j,k)}{j\omega\varepsilon_{xi}(i,j,k)\Delta x}\cdot\Big(H_{x,scat}(i,j,k)-H_{x,scat}(i-1,j,k)+H_{x,scat}(i-2,j,k)\Big)+$$

$$E_{x,scat}(i,j,k)=\frac{\varepsilon_{xi}(i,j,k)-\varepsilon_0(i,j,k)}{\varepsilon_{xi}(i,j,k)}E_{x,inc}(i,j,k)$$

Our coefficients for electric incidence in the x-direction become:

$\mathbf{C_{exex}}=\dfrac{\varepsilon_{xi}(i,j,k)-\varepsilon_0(i,j,k)}{\varepsilon_{xi}(i,j,k)}$

$\mathbf{C_{exhi}}=\dfrac{\mu_{xi}(i,j,k)}{j\omega\varepsilon_{xi}(i,j,k)\Delta x}$

$\mathbf{C_{exhx}}=\dfrac{\mu_{xi}(i,j,k)}{j\omega\varepsilon_{xi}(i,j,k)\Delta x}$

$\mathbf{C_{exhy}}=\dfrac{1}{j\omega\varepsilon_{xy}(i,j,k)\Delta y}$

$\mathbf{C_{exhz}}=\dfrac{1}{j\omega\varepsilon_{xz}(i,j,k)\Delta z}$

Similarly, we can find the coefficient in the remaining incidence directions using the same method in the y-direction and z-direction, respectively.

<u>Y-Direction</u>:

$$\frac{\mu_{yi}(i,j,k)}{j\omega\varepsilon_{yi}(i,j,k)\Delta y}\cdot\Big(H_{y,scat}(i,j,k)-H_{y,scat}(i-1,j,k)+H_{y,scat}(i-2,j,k)\Big)+$$

$$\frac{1}{j\omega\varepsilon_{yx}(i,j,k)\Delta x}\cdot\Big(H_{z,scat}(i,j,k)-H_{z,scat}(i-1,j,k)\Big)-$$

$$\frac{1}{j\omega\varepsilon_{yz}(i,j,k)\Delta z}\cdot\Big(H_{x,scat}(i,j,k)-H_{x,scat}(i,j,k-1)\Big)+$$

$$\frac{\mu_{yi}(i,j,k)}{j\omega\varepsilon_{yi}(i,j,k)\Delta y}\cdot\Big(H_{y,scat}(i,j,k)-H_{y,scat}(i-1,j,k)+H_{y,scat}(i-2,j,k)\Big)+$$

$$E_{y,scat}(i,j,k)=\frac{\varepsilon_{yi}(i,j,k)-\varepsilon_0(i,j,k)}{\varepsilon_{yi}(i,j,k)}E_{y,inc}(i,j,k)$$

Our coefficients for electric incidence in the y-direction become:

$$\mathbf{C_{eyey}}=\frac{\varepsilon_{yi}(i,j,k)-\varepsilon_0(i,j,k)}{\varepsilon_{yi}(i,j,k)}$$

$$\mathbf{C_{eyhx}}=\frac{1}{j\omega\varepsilon_{yz}(i,j,k)\Delta z}$$

$$\mathbf{C_{eyhi}}=\frac{\mu_{yi}(i,j,k)}{j\omega\varepsilon_{yi}(i,j,k)\Delta y}$$

$$\mathbf{C_{eyhy}}=\frac{\mu_{yi}(i,j,k)}{j\omega\varepsilon_{yi}(i,j,k)\Delta y}$$

$$\mathbf{C_{eyhz}}=\frac{1}{j\omega\varepsilon_{yx}(i,j,k)\Delta x}$$

Z-Direction:

$$\frac{\mu_{zi}(i,j,k)}{j\omega\varepsilon_{zi}(i,j,k)\Delta z}\cdot\Big(H_{z,scat}(i,j,k)-H_{z,scat}(i-1,j,k)+H_{z,scat}(i-2,j,k)\Big)+$$

$$\frac{1}{j\omega\varepsilon_{zx}(i,j,k)\Delta x}\cdot\Big(H_{y,scat}(i,j,k)-H_{y,scat}(i-1,j,k)\Big)-$$

$$\frac{1}{j\omega\varepsilon_{zy}(i,j,k)\Delta y}\cdot\Big(H_{x,scat}(i,j,k)-H_{x,scat}(i,j-1,k)\Big)+$$

$$\frac{\mu_{zi}(i,j,k)}{j\omega\varepsilon_{zi}(i,j,k)\Delta z}\cdot\Big(H_{z,scat}(i,j,k)-H_{z,scat}(i-1,j,k)+H_{z,scat}(i-2,j,k)\Big)+$$

$$E_{z,scat}(i,j,k)=\frac{\varepsilon_{zi}(i,j,k)-\varepsilon_0(i,j,k)}{\varepsilon_{zi}(i,j,k)}E_{z,inc}(i,j,k)$$

Our coefficients for electric incidence in the z-direction become:

$\mathbf{C_{ezez}}=\dfrac{\varepsilon_{zi}(i,j,k)-\varepsilon_0(i,j,k)}{\varepsilon_{zi}(i,j,k)}$

$\mathbf{C_{ezhx}}=\dfrac{1}{j\omega\varepsilon_{zy}(i,j,k)\Delta y}$

$\mathbf{C_{ezhi}}=\dfrac{1}{j\omega\varepsilon_{zx}(i,j,k)\Delta x}$

$\mathbf{C_{ezhy}}=\dfrac{1}{j\omega\varepsilon_{zx}(i,j,k)\Delta x}$

$\mathbf{C_{ezhz}}=\dfrac{\mu_{zi}(i,j,k)}{j\omega\varepsilon_{zi}(i,j,k)\Delta z}$

2.2 Verification

The verification of efficiency in the divergence-based implementation can be shown with convergence plots displaying the number of matvec subroutine calls in the Bi-CGSTAB subroutine versus the natural logarithm of the ratio of the complex L-2 norm of the residual. Once a desired tolerance has been met, multiplication stops and the iterative procedure is finished. The verification of accuracy in the divergence-based implementation is shown with the root mean square error between the analytical and non-divergence RCS as well as between the analytical and divergence based RCS

.

2.2.1 Perfectly Conducting Sphere

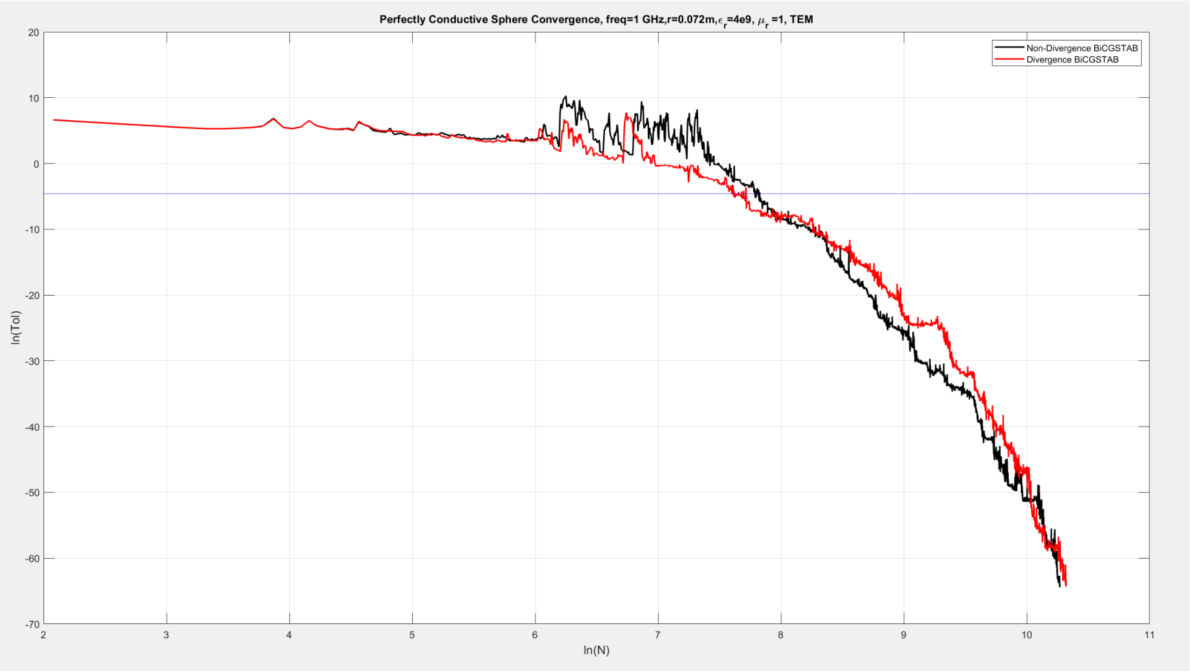

Figure 7: Perfectly Conducting Sphere Convergence BiCGSTAB

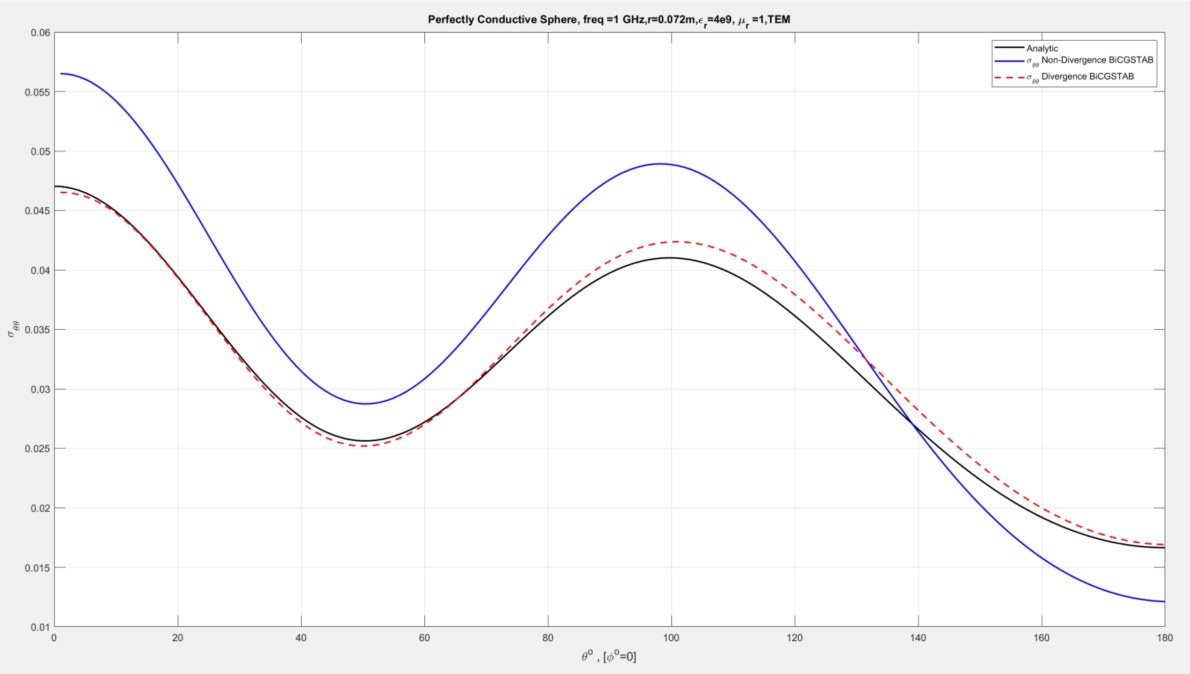

Figure 8: Perfectly Conducting Sphere RCS Comparison

2.2.2 Dielectric Sphere

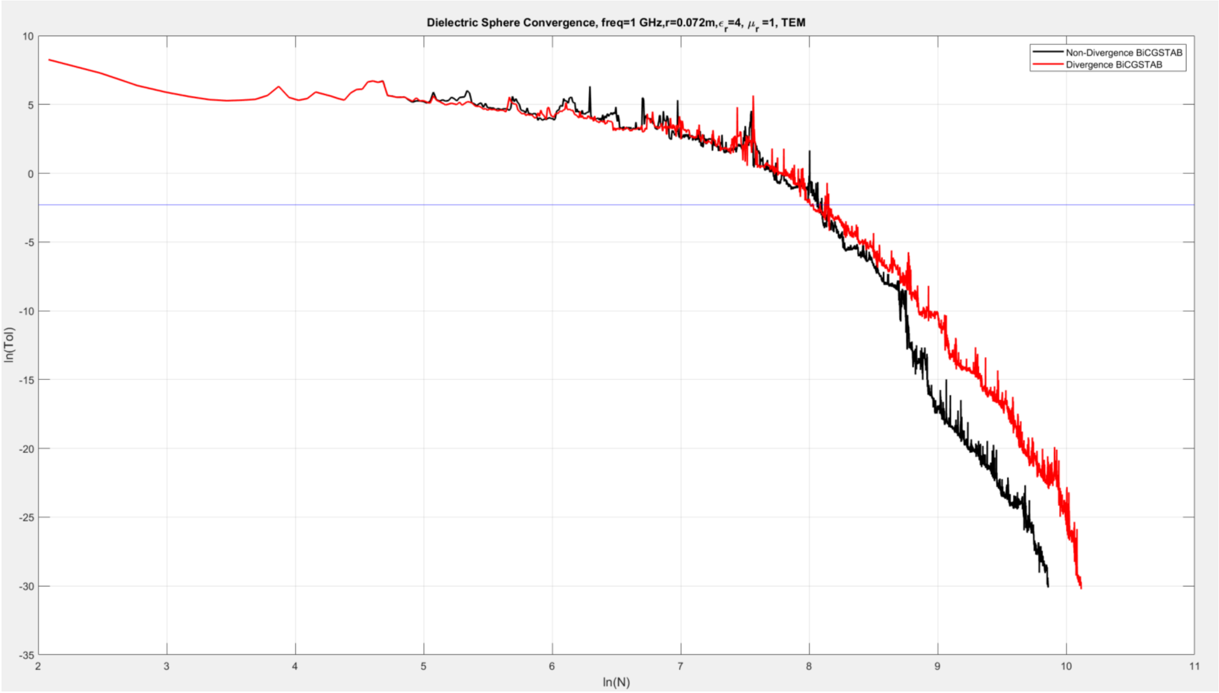

Figure 9: Dielectric Sphere Convergence BiCGSTAB

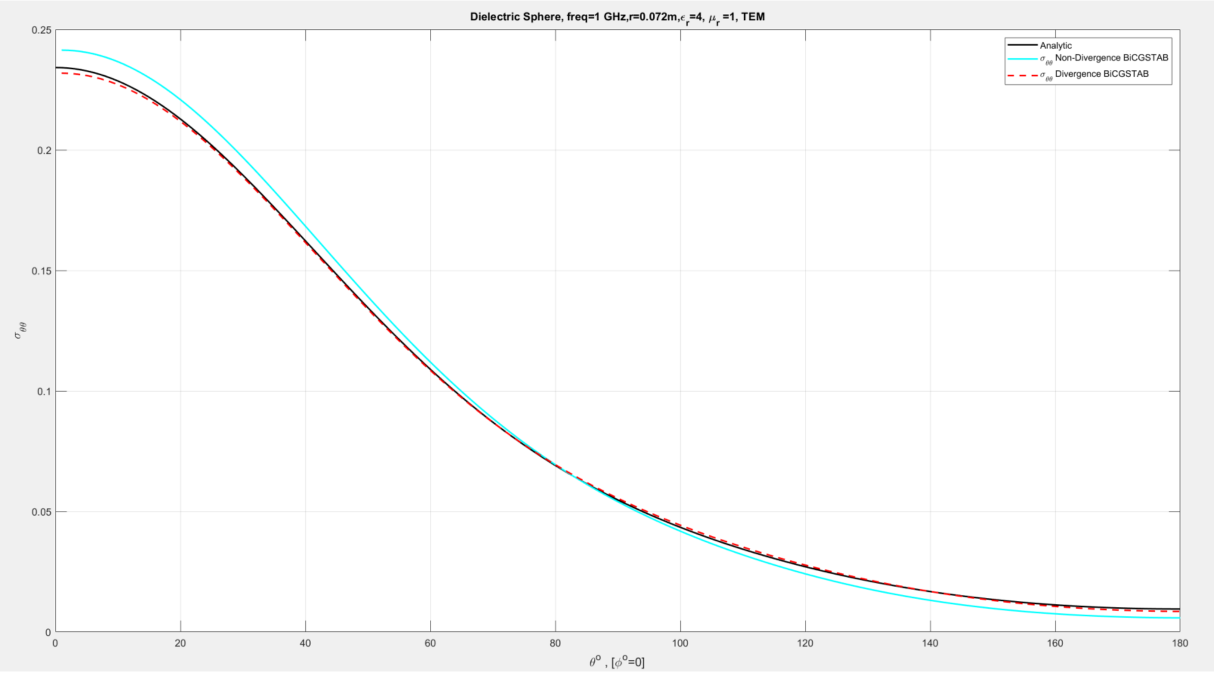

Figure 10: Dielectric Sphere RCS Comparison

2.2.3 Comparisons

For meaningful comparisons of efficiency and accuracy of the FDFD method the number of matrix-vector multiplications performed by the subroutine matvec when called by Bi-CGSTAB subroutine is used as the metric for efficiency of the algorithm. Once the specified tolerance has been met, multiplication in Krylov subspace stops, yielding a reliable update. The root mean square (RMS) error between the analytic and non-divergence as well as between the analytic and divergence re-enforcing radar cross sections is used as the metric for accuracy.

FDFD	NMATVEC	RCS RMS Error	Tolerance
Non-Divergence	1638	0.00500	0.01
Divergence	1345	0.00083	0.01

Table 1: Perfectly Conducting Sphere Comparison

FDFD	NMATVEC	RCS RMS Error	Tolerance
Non-Divergence	1284	0.0034	0.1
Divergence	1389	0.0015	0.1

Table 2: Dielectric Sphere Comparison

2.2.4 Conclusions

The future for the field of computational electromagnetics resides in our ability to create robust, efficient, and accurate algorithms that can simulate and solve canonical problems, complex systems, and devices quickly and accurately. This leads to interesting opportunities to improve upon the finite difference formulations with more advanced numerical methods that operate in parallel with more advanced hardware. By extending these algorithms to easy-to-understand end user experiences in the ever more competitive field of design environments, we can streamline design and conceptual understanding of advanced propagation scenarios that can facilitate production in industry and physical understanding in education.

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

[4] Jean-Pierre Berenger, A perfectly matched layer for the absorption of electromagnetic waves, Journal of Computational Physics, Volume 114, Issue 2, 1994, Pages 185-200.

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

[6] 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.

[7] 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.

[8] 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.

[9] K. Beilenhoff, W. Heinrich and H. L. Hartnagel, "Improved finite-difference formulation in frequency domain for three-dimensional scattering problems," in IEEE Transactions on Microwave Theory and Techniques, vol. 40, no. 3, pp. 540-546, March 1992, doi: 10.1109/22.121730.

[10] A. N. Krylov, On the numerical solution of the equation by which the frequency of small oscillations in determined in technical problems, Izv. Akad. Nauk SSSR, Ser.Fiz.-Mat., 4 (1931), pp. 491–539. In Russian. Title translation as in [216].

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

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

[13] Y. Saad, The Lanczos biorthogonalization algorithm and other oblique projection
methods for solving large unsymmetric systems, SIAM J. Numer. Anal., 19 (1982),
pp. 485–506.

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

[15] H.A. van der Vorst, Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 631–644

[16] V. Demir, A. Elsherbeni, D. Worasawate and E. Arvas, "A graphical user interface (GUI) for plane-wave scattering from a conducting, dielectric, or chiral sphere," in IEEE Antennas and Propagation Magazine, vol. 46, no. 5, pp. 94-99, Oct. 2004, doi: 10.1109/MAP.2004.1388838.

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

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

APPENDIX

A.1 Fortran Code

```fortran
subroutine setCoefficients
use global
implicit none
integer mi, mj, mk

allocate(Cexhy(Nx,Ny,Nz), Cexhz(Nx,Ny,Nz), Cexex(Nx,Ny,Nz));
allocate(Ceyhz(Nx,Ny,Nz), Ceyhx(Nx,Ny,Nz), Ceyey(Nx,Ny,Nz));
allocate(Cezhx(Nx,Ny,Nz), Cezhy(Nx,Ny,Nz), Cezez(Nx,Ny,Nz));

allocate(Chxey(Nx,Ny,Nz), Chxez(Nx,Ny,Nz), Chxhx(Nx,Ny,Nz));
allocate(Chyez(Nx,Ny,Nz), Chyex(Nx,Ny,Nz), Chyhy(Nx,Ny,Nz));
allocate(Chzex(Nx,Ny,Nz), Chzey(Nx,Ny,Nz), Chzhz(Nx,Ny,Nz));

allocate(Chxex(Nx,Ny,Nz), Chyey(Nx,Ny,Nz), Chzez(Nx,Ny,Nz));
allocate(Cexhx(Nx,Ny,Nz), Ceyhy(Nx,Ny,Nz), Cezhz(Nx,Ny,Nz));
allocate(Chxei(Nx,Ny,Nz), Chyei(Nx,Ny,Nz), Chzei(Nx,Ny,Nz));
allocate(Cexhi(Nx,Ny,Nz), Ceyhi(Nx,Ny,Nz), Cezhi(Nx,Ny,Nz));


allocate(tmpx(Nx,Ny,Nz), tmpy(Nx,Ny,Nz), tmpz(Nx,Ny,Nz));

allocate(Bex(Nx,Ny,Nz), Bey(Nx,Ny,Nz), Bez(Nx,Ny,Nz));
allocate(Bhx(Nx,Ny,Nz), Bhy(Nx,Ny,Nz), Bhz(Nx,Ny,Nz));

allocate(Bexi(Nx,Ny,Nz), Beyi(Nx,Ny,Nz), Bezi(Nx,Ny,Nz));
allocate(Bhxi(Nx,Ny,Nz), Bhyi(Nx,Ny,Nz), Bhzi(Nx,Ny,Nz));

tmpx = (0.0,0.0); tmpy = (0.0,0.0); tmpz = (0.0,0.0);

Cexhy = 1/(j*w*s*dz*eps_xz); Cexhz = 1/(j*w*s*dy*eps_xy); Cexex = -(eps_xi - eps_o) /
eps_xi;
Ceyhz = 1/(j*w*s*dx*eps_yx); Ceyhx = 1/(j*w*s*dz*eps_yz); Ceyey = -(eps_yi - eps_o) /
eps_yi;
Cezhx = 1/(j*w*s*dy*eps_zy); Cezhy = 1/(j*w*s*dx*eps_zx); Cezez = -(eps_zi - eps_o) /
eps_zi;

Chxey = 1/(j*w*s*dz*mu_xz); Chxez = 1/(j*w*s*dy*mu_xy); Chxhx = (mu_xi - mu_o) / mu_xi;
Chyez = 1/(j*w*s*dx*mu_yx); Chyex = 1/(j*w*s*dz*mu_yz); Chyhy = (mu_yi - mu_o) / mu_yi;
Chzex = 1/(j*w*s*dy*mu_zy); Chzey = 1/(j*w*s*dx*mu_zx); Chzhz = (mu_zi - mu_o) / mu_zi;

! divergence re-enforcing coefficients

Cexhx = mu_xi/(j*w*s*dx*eps_xi); Cexhi = mu_xi/(j*w*s*dx*eps_xi);
Ceyhy = mu_yi/(j*w*s*dy*eps_yi); Ceyhi = mu_yi/(j*w*s*dy*eps_yi);
Cezhz = mu_zi/(j*w*s*dz*eps_zi); Cezhi = mu_zi/(j*w*s*dz*eps_zi);

Chxex = eps_xi/(j*w*s*dx*mu_xi); Chxei = eps_xi/(j*w*s*dx*mu_xi);
Chyey = eps_yi/(j*w*s*dy*mu_yi); Chyei = eps_yi/(j*w*s*dy*mu_yi);
Chzez = eps_zi/(j*w*s*dz*mu_zi); Chzei = eps_zi/(j*w*s*dz*mu_zi);

Bex = Cexex * Einc_x; Bey = Ceyey * Einc_y; Bez = Cezez * Einc_z;
Bhx = Chxhx * Hinc_x; Bhy = Chyhy * Hinc_y; Bhz = Chzhz * Hinc_z;
```

```
! divergence re-enfocing incidences

Bexi = Cexhi * Einc_x; Beyi = Ceyhi * Einc_y; Bezi = Cezhi * Einc_z;
Bhxi = Chxei * Hinc_x; Bhyi = Chyei * Hinc_y; Bhzi = Chzei * Hinc_z;

! divergence re-enfocing equations

Bex(3:nx,2:ny,2:nz) = Bex(3:nx,2:ny,2:nz) - &
(-Cexhz(3:nx,2:ny,2:nz)*Bhz(3:nx,2:ny,2:nz)+Cexhz(3:nx,2:ny,2:nz)*Bhz(3:nx,1:ny-1,2:nz)
&
+Cexhy(3:nx,2:ny,2:nz)*Bhy(3:nx,2:ny,2:nz)-Cexhy(3:nx,2:ny,2:nz)*Bhy(3:nx,2:ny,1:nz-1)
&

-Cexhi(3:nx,2:ny,2:nz)*Bhxi(3:nx,2:ny,2:nz)+Cexhi(3:nx,2:ny,2:nz)*Bhxi(2:nx-1,2:ny,2:nz)-
Cexhi(3:nx,2:ny,2:nz)*Bhxi(1:nx-2,2:ny,2:nz) &
+Cexhx(3:nx,2:ny,2:nz)*Bhx(3:nx,2:ny,2:nz)-Cexhx(3:nx,2:ny,2:nz)*Bhx(2:nx-
1,2:ny,2:nz)+Cexhx(3:nx,2:ny,2:nz)*Bhx(1:nx-2,2:ny,2:nz));

Bey(2:nx,3:ny,2:nz) = Bey(2:nx,3:ny,2:nz) - &
(-Ceyhx(2:nx,3:ny,2:nz)*Bhx(2:nx,3:ny,2:nz)+Ceyhx(2:nx,3:ny,2:nz)*Bhx(2:nx,3:ny,1:nz-1)
&
+Ceyhz(2:nx,3:ny,2:nz)*Bhz(2:nx,3:ny,2:nz)-Ceyhz(2:nx,3:ny,2:nz)*Bhz(1:nx-1,3:ny,2:nz)
&

-Ceyhi(2:nx,3:ny,2:nz)*Bhyi(2:nx,3:ny,2:nz)+Ceyhi(2:nx,3:ny,2:nz)*Bhyi(2:nx,2:ny-1,2:nz)-
Ceyhi(2:nx,3:ny,2:nz)*Bhyi(2:nx,1:ny-2,2:nz) &
+Ceyhy(2:nx,3:ny,2:nz)*Bhy(2:nx,3:ny,2:nz)-Ceyhy(2:nx,3:ny,2:nz)*Bhy(2:nx,2:ny-
1,2:nz)+Ceyhy(2:nx,3:ny,2:nz)*Bhy(2:nx,1:ny-2,2:nz));

Bez(2:nx,2:ny,3:nz) = Bez(2:nx,2:ny,3:nz) - &
(-Cezhy(2:nx,2:ny,3:nz)*Bhy(2:nx,2:ny,3:nz)+Cezhy(2:nx,2:ny,3:nz)*Bhy(1:nx-1,2:ny,3:nz)
&
+Cezhx(2:nx,2:ny,3:nz)*Bhx(2:nx,2:ny,3:nz)-Cezhx(2:nx,2:ny,3:nz)*Bhx(2:nx,1:ny-1,3:nz)
&

-Cezhi(2:nx,2:ny,3:nz)*Bhzi(2:nx,2:ny,3:nz)+Cezhi(2:nx,2:ny,3:nz)*Bhzi(2:nx,2:ny,2:nz-1)-
Cezhi(2:nx,2:ny,3:nz)*Bhzi(2:nx,2:ny,1:nz-2) &
+Cezhz(2:nx,2:ny,3:nz)*Bhz(2:nx,2:ny,3:nz)-Cezhz(2:nx,2:ny,3:nz)*Bhz(2:nx,2:ny,2:nz-
1)+Cezhz(2:nx,2:ny,3:nz)*Bhz(2:nx,2:ny,1:nz-2));

Bvec(0*Nxyz+1:1*Nxyz) = reshape(Bex, (/Nxyz/));
Bvec(1*Nxyz+1:2*Nxyz) = reshape(Bey, (/Nxyz/));
Bvec(2*Nxyz+1:3*Nxyz) = reshape(Bez, (/Nxyz/));

end subroutine setCoefficients
```

```
subroutine matvec(n,x,y)
use global
implicit none
integer n; complex*16 x(nx,ny,nz,3), y(nx,ny,nz,3);


! divergence re-enforcing matvec

tmpx(1:Nxm1-1,1:Nym1,1:Nzm1) = Chxez(1:Nxm1-1,1:Nym1,1:Nzm1)*(x(1:Nxm1-1,2:Ny,1:Nzm1,3)-
x(1:Nxm1-1,1:Nym1,1:Nzm1,3))        &
+ Chxey(1:Nxm1-1,1:Nym1,1:Nzm1)*(-x(1:Nxm1-1,1:Nym1,2:Nz,2)+x(1:Nxm1-1,1:Nym1,1:Nzm1,2))
&

+ Chxex(1:Nxm1-1,1:Nym1,1:Nzm1)*(x(3:Nx,1:Nym1,1:Nzm1,1)-
x(2:Nxm1,1:Nym1,1:Nzm1,1)+x(1:Nxm1-1,1:Nym1,1:Nzm1,1)) &
- Chxei(1:Nxm1-1,1:Nym1,1:Nzm1)*(-x(3:Nx,1:Nym1,1:Nzm1,1)+x(2:Nxm1,1:Nym1,1:Nzm1,1)-
x(1:Nxm1-1,1:Nym1,1:Nzm1,1));

tmpy(1:Nxm1,1:Nym1-1,1:Nzm1) = Chyex(1:Nxm1,1:Nym1-1,1:Nzm1)*(x(1:Nxm1,1:Nym1-1,2:Nz,1)-
x(1:Nxm1,1:Nym1-1,1:Nzm1,1))        &
+ Chyez(1:Nxm1,1:Nym1-1,1:Nzm1)*(-x(2:Nx,1:Nym1-1,1:Nzm1,3)+x(1:Nxm1,1:Nym1-1,1:Nzm1,3))
&

+ Chyey(1:Nxm1,1:Nym1-1,1:Nzm1)*(x(1:Nxm1,3:Ny,1:Nzm1,2)-
x(1:Nxm1,2:Nym1,1:Nzm1,2)+x(1:Nxm1,1:Nym1-1,1:Nzm1,2)) &
- Chyei(1:Nxm1,1:Nym1-1,1:Nzm1)*(-x(1:Nxm1,3:Ny,1:Nzm1,2)+x(1:Nxm1,2:Nym1,1:Nzm1,2)-
x(1:Nxm1,1:Nym1-1,1:Nzm1,2));

tmpz(1:Nxm1,1:Nym1,1:Nzm1-1) = Chzey(1:Nxm1,1:Nym1,1:Nzm1-1)*(x(2:Nx,1:Nym1,1:Nzm1-1,2)-
x(1:Nxm1,1:Nym1,1:Nzm1-1,2))        &
+ Chzex(1:Nxm1,1:Nym1,1:Nzm1-1)*(-x(1:Nxm1,2:Ny,1:Nzm1-1,1)+x(1:Nxm1,1:Nym1,1:Nzm1-1,1))
&

+ Chzez(1:Nxm1,1:Nym1,1:Nzm1-1)*(x(1:Nxm1,1:Nym1,3:Nz,3)-
x(1:Nxm1,1:Nym1,2:Nzm1,3)+x(1:Nxm1,1:Nym1,1:Nzm1-1,3)) &
- Chzei(1:Nxm1,1:Nym1,1:Nzm1-1)*(-x(1:Nxm1,1:Nym1,3:Nz,3)+x(1:Nxm1,1:Nym1,2:Nzm1,3)-
x(1:Nxm1,1:Nym1,1:Nzm1-1,3));

y(3:Nx,2:Ny,2:Nz,1) = x(3:Nx,2:Ny,2:Nz,1) &
- (Cexhz(3:Nx,2:Ny,2:Nz)*(-tmpz(3:Nx,2:Ny,2:Nz)+tmpz(3:Nx,1:Nym1,2:Nz))  &
+  Cexhy(3:Nx,2:Ny,2:Nz)*(tmpy(3:Nx,2:Ny,2:Nz)-tmpy(3:Nx,2:Ny,1:Nzm1))   &

+  Cexhx(3:Nx,2:Ny,2:Nz)*(tmpy(3:Nx,2:Ny,2:Nz)-tmpy(2:Nxm1,2:Ny,2:Nz)+tmpy(1:Nxm1-
1,2:Ny,2:Nz)) &
-  Cexhi(3:Nx,2:Ny,2:Nz)*(-tmpy(3:Nx,2:Ny,2:Nz)+tmpy(2:Nxm1,2:Ny,2:Nz)-tmpy(1:Nxm1-
1,2:Ny,2:Nz)));

y(2:Nx,3:Ny,2:Nz,2) = x(2:Nx,3:Ny,2:Nz,2) &
- (Ceyhx(2:Nx,3:Ny,2:Nz)*(-tmpx(2:Nx,3:Ny,2:Nz)+tmpx(2:Nx,3:Ny,1:Nzm1))  &
+  Ceyhz(2:Nx,3:Ny,2:Nz)*(tmpz(2:Nx,3:Ny,2:Nz)-tmpz(1:Nxm1,3:Ny,2:Nz))   &

+  Ceyhy(2:Nx,3:Ny,2:Nz)*(tmpz(2:Nx,3:Ny,2:Nz)-tmpz(2:Nx,2:Nym1,2:Nz)+tmpz(2:Nx,1:Nym1-
1,2:Nz)) &
-  Ceyhi(2:Nx,3:Ny,2:Nz)*(-tmpz(2:Nx,3:Ny,2:Nz)+tmpz(2:Nx,2:Nym1,2:Nz)-tmpz(2:Nx,1:Nym1-
1,2:Nz)));

y(2:Nx,2:Ny,3:Nz,3) = x(2:Nx,2:Ny,3:Nz,3) &
- (Cezhy(2:Nx,2:Ny,3:Nz)*(-tmpy(2:Nx,2:Ny,3:Nz)+tmpy(1:Nxm1,2:Ny,3:Nz))  &
+  Cezhx(2:Nx,2:Ny,3:Nz)*(tmpx(2:Nx,2:Ny,3:Nz)-tmpx(2:Nx,1:Nym1,3:Nz))   &

+  Cezhz(2:Nx,2:Ny,3:Nz)*(tmpx(2:Nx,2:Ny,3:Nz)-
tmpx(2:Nx,2:Ny,2:Nzm1)+tmpx(2:Nx,2:Ny,1:Nzm1-1)) &
-  Cezhi(2:Nx,2:Ny,3:Nz)*(-tmpx(2:Nx,2:Ny,3:Nz)+tmpx(2:Nx,2:Ny,2:Nzm1)-
tmpx(2:Nx,2:Ny,1:Nzm1-1)));

end subroutine matvec
```