An OcTree Multigrid Method for Quasi-Static Maxwell's Equations with Highly Discontinuous Coefficients

Eldad Haber and Stefan Heldmann [*]

February 12, 2006

Abstract

In this paper we develop an OcTree discretization for Maxwell's equations in the quasi-static regime. We then use this discretization in order to develop a multigrid method for Maxwell's equations with highly discontinuous coefficients. We test our algorithms and compare it to other multilevel algorithms.

1 Introduction

The solution of Maxwell's equations in the quasi-static regime is important in many practical settings such as geophysical prospecting, non-destructive testing and eddy current simulations. The equations read

$$\nabla \times \vec{E} + i\omega\mu\vec{H} = 0 \tag{1.1a}$$

$$\nabla \times \vec{H} - \sigma\vec{E} = \vec{s} \tag{1.1b}$$

where $\vec{E}$ is the electric field, $\vec{H}$ is the magnetic field, σ is the conductivity, μ is the magnetic permeability, ω is the frequency and $\vec{s}$ is a source term. We assume that the equations are given in a bounded box with some appropriate boundary conditions on $\vec{E}$ or $\vec{H}$ and that the conductivity σ and permeability

[*]Department of Mathematics and Computer Science, Emory University, Atlanta, GA.

μ have a small number of jumps with compact support. Our typical applications describe electromagnetic fields in metals berried in the ground in the frequencies $1-10^4$ Hz. For these problems σ range from $10^{-2}-10^4$ S/m and the relative μ range from $1-100$ (see [30]).

For ease of presentation we assume Perfectly Electric Conductor (PEC) boundary conditions which read

$$\vec{n} \times \vec{E} = 0 \tag{1.2}$$

although other boundary conditions can be considered.

It is worthwhile noting that equation (1.1a) also implies that

$$\nabla \cdot \mu\vec{H} = 0 \tag{1.3}$$

which corresponds to the fact that there are no magnetic sources. We also note that in the case of vanishing frequency Maxwell's equations degenerate to the electrostatic and magnetostatic equations

$$\nabla \cdot \sigma\nabla\phi = \nabla \cdot \vec{s} \tag{1.4a}$$

$$\nabla \times \mu^{-1}\nabla \times \vec{A} = \vec{s} - \sigma\nabla\phi \tag{1.4b}$$

$$\vec{E} = \nabla\phi \quad \text{and} \quad \vec{H} = \mu^{-1}\nabla \times \vec{A} \tag{1.4c}$$

where ϕ and $\vec{A}$ are the (scalar) electric and (vector) magnetic potentials.

The solution of Maxwell's equations is very challenging even for the static case. There are two main sources of difficulties. First, the **curl** operator has a nontrivial rich null space and second, in our applications the conductivity and magnetic permeability can have large jumps. Common algorithms for Maxwell's equations use finite element or finite volume/difference approximation, and have been extensionally studied in the last decade; see [13, 11, 14, 23, 16, 24] and references within. While some finite difference algorithms can effectively deal with highly discontinuous conductivity [1] to our best knowledge non of the finite difference algorithms can successfully deal with large jumps in magnetic permeability. In particular, the results in [14] suggest that geometric multigrid breaks for large jumps in the magnetic permeability.

Problems with discontinuous coefficients are in general challenging for multigrid methods. Typical approaches for jumps in coefficients require either algebraic multigrid or geometric multigrid with operator induced prolongation and restriction [18, 26, 6, 27, 28]. While approaches such as the one suggested in black box multigrid [6] work well for problems that evolve from scalar PDE's, they are not adequate for dealing with problems that evolve from systems of PDE's. A common AMG approach for systems of PDE's use smooth aggregation [27]. In the numerical experiment section of this paper we compare our method with the software package ML which uses smooth aggregation multigrid [3, 16]. Generally speaking, AMG codes tend to use large amounts of memory and by their generality under-utilize available information about the problem.

In order to generate an effective algorithm for Maxwell's equations and to deal with the above difficulties here we combine two approaches. First, we use our previous work [11, 12] that builds on the Helmhotz decomposition of the discrete fields, in order to stabilize the equations and avoid the null space of the **curl**. Second, we extend local refinement strategies to deal with jumping coefficients. In particular, we demonstrate the advantage of an OcTree discretization. The combination of the two strategies results in a highly efficient method for the solution of Maxwell's equations.

Local refinement can be used either in a finite element or a finite volume/difference framework. Finite elements are inherently built to deal with non-trivial geometries however, they require nontrivial meshing tools and (although $\mathcal{O}(n)$) significant amount of time in order to construct the stiffness matrix. Standard finite volume methods based on orthogonal staggered grids enable to generate the discrete systems quickly but require very fine meshing if one attempts to deal with complex geometries. A good compromise between the two are locally refined OcTree grids. Such grids are orthogonal but allow for better modeling of areas with complicated geometries and fast changing solutions. Although they may be slightly less accurate compared with finite elements (for the same number of elements/cells), the assembly of the stiffness matrix usually takes only a fraction of the time compared to problems on unstructured grids. Our interest lies in electromagnetic inverse problems with applications to geophysics and medical imaging, where the stiffness matrix has to be composed every iteration. It is therefore advantageous to use a refinement strategy that allows for fast computations of the stiffness matrix. Furthermore some of our data originates from medical imaging which naturally involves structured grids.

The use of OcTrees in order to locally refine grids in PDE's is not new. In particular it has been used in flow through porous media and fluid dynamics [7, 8] where cell center OcTrees with even and odd number of locally refined grids are considered. Recently, there has been renewed interest in local refinement and their applications to computational fluid dynamics and computer graphics [21, 20]. In particular, the work of Losasso et-al on OcTree discretization of Poisson equation demonstrates that second order accuracy can be obtained. A more general framework for Poisson equation was recently studied in [19]. In particular, they have demonstrated how to effectively deal with jumping coefficients in Poisson type problems. Other relevant work to ours is the recent work of [29] on the solution of Maxwell's equations in the hyperbolic regime where no large jumps in the coefficients are present.

In this paper we extend local refinement strategies for the solution of Maxwell's equations in the Quasi-Static regime. In particular we demonstrate the advantage of an OcTree discretization to problems with highly discontinuous coefficients. We show how an OcTree discretization can be used in order to develop an effective multigrid solution to Maxwell's equations with discontinuous coefficients.

The paper is organized as follows. In section 2 we review the discretization of the **div** and **grad** in 2 and 3D and discuss the discretization of the **curl** and its adjoint in 3D. We also review the matrices for material averaging and generate the discrete systems. In Section 3 we discuss a multigrid method for the problem. In particular, we use a coarsening strategy that allows us to deal with highly discontinuous coefficients. In Section 4 we perform numerical experiments that demonstrate the effectiveness of our approach. Finally, we summarize the paper in Section 5

2 OcTree discretization of Maxwell's equations

In this section we review and develop the discretization of Maxwell's equations on OcTrees. The discretization is based on mimicking finite volume methods and they are closely related to the work presented in [19, 17, 4, 5] which can be thought of extending Yee type discretization [31] to OcTrees.

In the following we consider the discretizations of vector fields and scalar fields using three types of grid functions on an OcTree: cell, face and edge-centers. Figure 2 shows the three kind of discretizations for a small example. Dealing with several different grids might be confusing. To this end, in Table 2 we give a quick overview of the discretization of the incorporating quantities and operators we develop in this section.

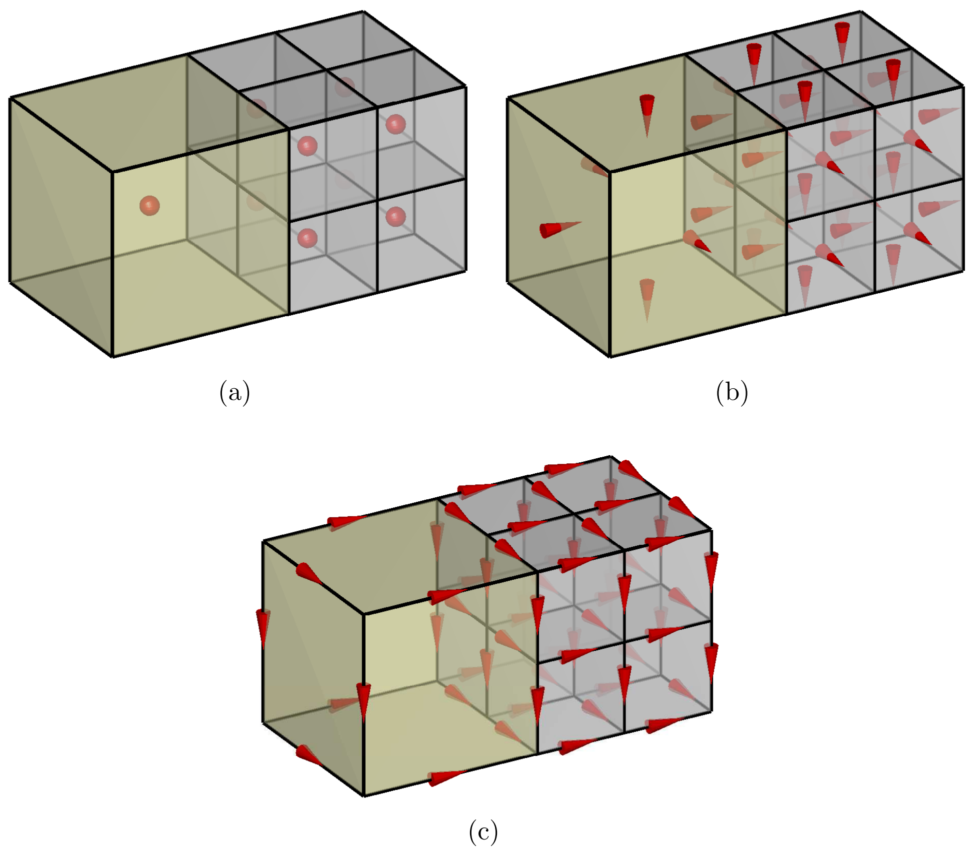


Figure 1: OcYree Discretization. (a) cell-center (b) face-center (c) edge-center

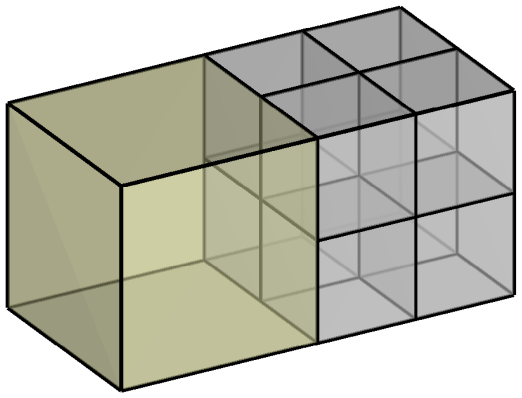
2.1 OcTree data structure

We consider a fine underlying orthogonal mesh of size $2^{m_1} \times 2^{m_2} \times 2^{m_3}$ with mesh size h. Our grid is composed of m square cells of different sizes. Each cell can have a different length which is a power of 2. To make the data structure easier and the discretization more accurate we allow only a factor of 2 between adjacent cells. The data is then stored as a sparse array. The size of each cell is stored in the upper left corner of the array. This allows us to quickly find neighbors which is the major operation in the discretization process. This data structure is closely related to the one suggested in [15]. An example of a small 3D grid is plotted in Figure 2.1.

Quantity	Symbol (cont-discrete)	Discretization
electric field	$\vec{E}$ $\mathbf{e}$	face-center
current density	$\vec{J}$ $\mathbf{j}$	face-center
magnetic potential	$\vec{A}$ $\mathbf{a}$	face-center
magnetic field	$\vec{H}$ $\mathbf{h}$	edge-center
electric potential	ϕ $\boldsymbol{\phi}$	cell-center
conductivity	σ	cell-center
magnetic permeability	μ	cell-center

Discrete Operator	Symbol	Mapping		
divergence	**DIV**	face-center	$\rightarrow$	cell-center
gradient	**GRAD**	cell-center	$\rightarrow$	face-center
curl	**CURL**	edge-center	$\rightarrow$	face-center
adjoint curl	$\overline{\mathbf{CURL}}$	face- center	$\rightarrow$	edge-center
face average	$\mathbf{A}_f$	cell-center	$\rightarrow$	face-center
edge average	$\mathbf{A}_e$	cell-center	$\rightarrow$	edge-center

Table 1: Discretization-Overview.



(a)

$$S(:,:,1) = \begin{pmatrix} 2 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

$$S(:,:,2) = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

(b)

Figure 2: (a) OcTree and (b) and its representation as $2 \times 4 \times 2$ (sparse) array

2.2 Discretization of the div and grad in 2 and 3D

Although our code is 3D it is worth while to follow the discretization in 2D for two main reasons. First, the discretization of the **div** and **grad** in

3D are simple extensions of the 2D case. Second, as we explore next, the discretization of the **curl** and its adjoint involves only 2D plains embedded in a 3D volume. We can therefore directly extend the 2D discretizations of the **grad** into the 3D discretization of the **curl**.

To discretize the divergence operator we use the usual flux-balance approach. Consider a 2D cell shown in Figure 3. We discretize the flux,

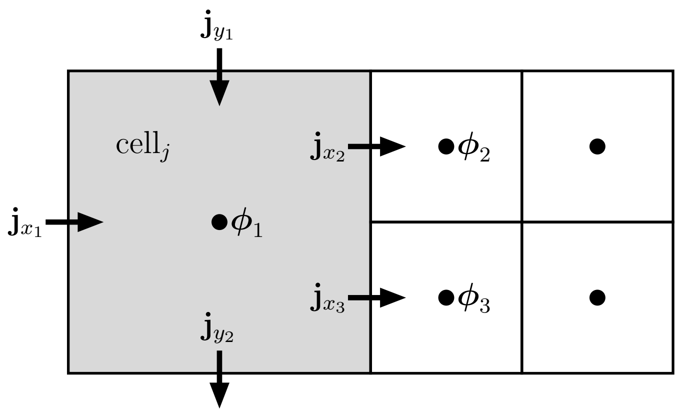


Figure 3: Discretization of the divergence

$\vec{J} = (J_x, J_y)^\top$ on the faces of the cell and using Gauss formula we write for the divergence of cell$_j$ with volume V_j

$$\frac{1}{V_j}\int_{\text{cell}_j} \nabla \cdot \ \vec{J}\, dV = \frac{1}{V_j}\int_{\text{faces cell}_j} \vec{J} \ \cdot d\vec{S} \approx \frac{1}{V_j}(2\mathbf{j}_{x_1} - \mathbf{j}_{x_2} - \mathbf{j}_{x_3} + 2(\mathbf{j}_{y_1} - \mathbf{j}_{y_2})).$$

It is easy to verify that this standard discretization of the divergence is second order accurate. Such mass-balance approximation can be formed for each cell in our grid, resulting in a discretization of the div operator. To this end, we collect all the fluxes on the faces in a vector

$$\mathbf{j} = (\mathbf{j}_x\ \mathbf{j}_y)^\top \text{ in 2D} \qquad \text{and} \qquad \mathbf{j} = (\mathbf{j}_x\ \mathbf{j}_y\ \mathbf{j}_z)^\top \text{ in 3D}$$

Then we define the discrete divergence operator as a $m \times m_f$ matrix **DIV** with m is the number of cells and m_f the number faces in our grid, such that

$$\left(\frac{1}{V_j}\int_{\text{cell}_j} \nabla \cdot \ \vec{J}\, dV\right)_{j=1}^{m} \approx \mathbf{DIV}\,\mathbf{j}.$$

It is useful to express this discretization in somewhat different form. Let $\mathbf{V} = \text{diag}\{V_1, \dots, V_m\}$ be a diagonal matrix which contains the relative volume of each cell (i.e. w.r.t the smallest cell). Let $\mathbf{F} = \text{diag}\{F_1, \dots, F_{m_f}\}$ be a diagonal matrix which contains the length of the large edges that bound each face on our grid. Finally, let $\mathbf{N} = (\mathbf{N}_1\ \mathbf{N}_2)$ in 2D or $\mathbf{N} = (\mathbf{N}_1\ \mathbf{N}_2\ \mathbf{N}_3)$ in 3D be a matrix of size $m \times m_f$ that contains the topology of the discrete divergence matrix, that is, it contains only nonzero values of ± 1. We set the sign based on the normal direction of each face w.r.t the each cell. Then, the div matrix can be written as

$$\mathbf{DIV} = \frac{1}{h}\mathbf{V}^{-1}\mathbf{N}\mathbf{F} \text{ in 2D} \qquad \text{and} \qquad \mathbf{DIV} = \frac{1}{h}\mathbf{V}^{-1}\mathbf{N}\mathbf{F}^2 \text{ in 3D} \tag{2.5}$$

There are various ways to define the discrete gradient operator. Here, the grad operator maps from cell-centers to the faces of our grid. Consider the arrangement in Figure 3. If we wish to obtain a second order discretization of the grad at the point where $\mathbf{j}_{x_2}$ is discretize we must use more points than the three points $\boldsymbol{\phi}_{1,2,3}$. However, as discussed in [20], second order discretization at every point would not result in a gradient which is a (scaled) transpose of the divergence. As we show next, our application require that the discrete divergence and gradient are (scaled) transposes of each other. We therefore use the same arguments presented in [20] and use only a first order approximation for the gradient operator at interfaces that do not conform. That is, we simply express the gradient of $\boldsymbol{\phi}$ at the point where $\mathbf{j}_{x_2}$ is discretized as

$$\frac{1}{F_j h}\left(\boldsymbol{\phi}_2 - \boldsymbol{\phi}_1\right)$$

where F_j is the (relative) edge length of the large cell. It is easy to verify that this discretization implies that

$$\mathbf{GRAD} = -\frac{1}{h}\mathbf{F}^{-1}\mathbf{N}^\top \tag{2.6}$$

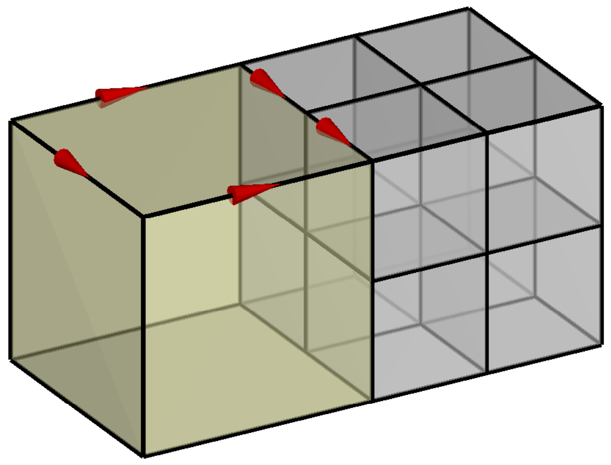
in 2 and 3D.

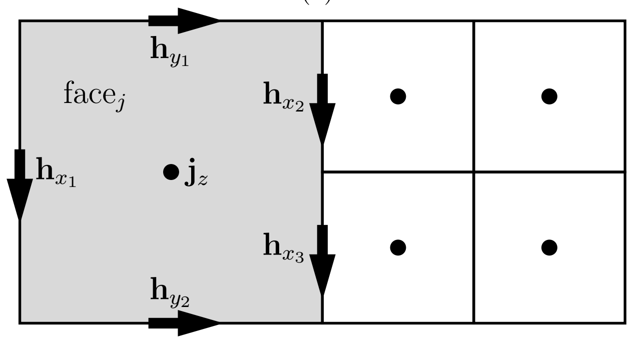
Note that the discretizations of the **div** and the **grad** in 3D are simple extensions of the 2D discretizations. The only differences are that in 2D each face can be composed of either 1 or 2 segments and in 3D each face is composed of either 1 or 4 segments and that we have to consider six faces in 3D rather than four faces in 2D.

2.3 Discretization of the curl

Similar to the discretization of the **div** we use integral identities in order to discretize the **curl**. We note that

$$\int_{\text{cell face}} \nabla \times \vec{H} \cdot d\vec{S} = \int_{\text{cell edges}} \vec{H} \cdot d\vec{\ell}$$


(a)



(b)

Figure 4: Discretization of the **curl**. (a) OcTree and (b) top 2D slice

Consider the upper face plotted in Figure 4. A straight forward discretization reads

$$\frac{1}{F_j^2}\int_{\text{face}_\text{j}} \nabla \times \vec{H} \cdot d\vec{S} \approx \frac{1}{F_j^2}\left(2\mathbf{h}_{x_1} - \mathbf{h}_{x_2} - \mathbf{h}_{x_3} - 2\left(\mathbf{h}_{y_1} - \mathbf{h}_{y_2}\right)\right)$$

where as above F_j is the length of the largest edge of the face. Similar to the above discretization we integrate over every cell-face in our mesh to obtain the discretization of the **curl**. Note that the discrete **curl** is a mapping from cell edges to cell faces.

As in the case of the **div** it is useful to express this discretization in a different form. Let $\mathbf{E} = \text{diag}\{E_1, \ldots E_{m_e}\}$, where m_e is the total number of edges, be a diagonal matrix which contains the relative edge sizes and let

$$\mathbf{T} = \begin{pmatrix} \mathbf{0} & -\mathbf{T}_{zy} & \mathbf{T}_{yz} \\ \mathbf{T}_{zx} & \mathbf{0} & -\mathbf{T}_{xz} \\ -\mathbf{T}_{yx} & \mathbf{T}_{xy} & \mathbf{0} \end{pmatrix}$$

where $\mathbf{T}_{ij},\ i,j \in \{x,y,z\}$ are again difference matrices which contain the value ± 1 and 0. The sign of the entry is determined by the direction of the edge with respect to the face the **curl** is integrated on. Using these matrices the **curl** matrix can be written as

$$\mathbf{CURL} = \frac{1}{h}\mathbf{F}^{-2}\,\mathbf{T}\,\mathbf{E} \tag{2.7}$$

In order to discretize the adjoint of the **curl** we note that it is a difference matrix that maps the edges of the faces to the face-centers and therefore operates along the tangential plains of the faces. Plotting a single plain, we see that we are in the same situation as in computing the **grad** in 2D (cf. Figure 4). Thus, an $\mathcal{O}(h^2)$ approximation would not result in a discrete transpose of the **curl**. However, if we require only an $\mathcal{O}(h)$ approximation then we are able to obtain a discrete approximation to the adjoint of the **curl** which is a scaled transpose of the discretization proposed above. Thus, similar to the gradient we approximate the adjoint by

$$\overline{\mathbf{CURL}} = \mathbf{E}^{-1}\mathbf{T}^{\top} \tag{2.8}$$

It is easy to verify that this approximation is identical to the one made in the 2D discretization of the **grad**.

2.4 Discretization of material averaging

It is common to have material properties given (or discretized) in cell centered. However, since we need to approximate quantities such as $\sigma\vec{E}$ and

$\mu\vec{H}$ we require to average material properties on faces and edges. As discussed in [11] harmonic averaging is required for averaging at the faces while arithmetic averaging is required for the edges. Similar to the discretization of the differential operators we set $\mathbf{A}_f$ to be an $m_f \times m$ averaging matrix that perform arithmetic averaging of cells at cell-faces and we let $\mathbf{A}_e$ as an $m_e \times m$ averaging matrix from cells to cell-edges. The material properties can be expressed as

$$\mathbf{S}(\sigma) = \text{diag}\left((\mathbf{A}_f\sigma^{-1})^{-1}\right) \tag{2.9a}$$

$$\mathbf{M}(\mu) = \text{diag}\left(\mathbf{A}_e\mu\right) \tag{2.9b}$$

2.5 Assembly of the linear system and its properties

Given the above differential operators we can easily assemble a discrete version of Maxwell's equations. Discretizing $\vec{E}$ on cell-faces and $\vec{H}$ on cell edges we obtain the following system of equations for the discrete grid functions $\mathbf{e}$ and $\mathbf{h}$

$$\overline{\mathbf{CURL}}\,\mathbf{e} + i\omega\mathbf{M}(\mu)\mathbf{h} = 0 \tag{2.10a}$$

$$\mathbf{CURL}\,\mathbf{h} - \mathbf{S}(\sigma)\mathbf{e} = \mathbf{s}. \tag{2.10b}$$

Although it is possible (at least in principle) to directly solve the discrete system (2.10) it is not our preferable strategy. The system is unfavorable for iterative methods due to the large null space of the discrete **curl** operator. To stabilize the system we use the properties of our discretization. There are two main properties that we require for our stabilization.

a. As noted in [5] the matrices $\mathbf{T}$ and $\mathbf{N}$ span the whole space, that is $\text{span}(\mathbf{T})\bigcup\text{span}(\mathbf{N}) = \mathcal{R}^n$.

b. $\mathbf{T}$ and $\mathbf{N}$ are orthogonal to each other, $\mathbf{NT} = \mathbf{0}$, which also implies that $\mathbf{DIV}\,\mathbf{CURL} = \mathbf{0}$

To use these properties we first eliminate the magnetic field from the equations, obtaining

$$\left(\frac{1}{h^2}\mathbf{T}\mathbf{M}(\mu)^{-1}\mathbf{T}^\top + i\omega\mathbf{F}^2\mathbf{S}(\sigma)\right)\mathbf{e} = -i\omega\mathbf{F}^2\mathbf{s} \tag{2.11}$$

This system still suffers from the null space of the **curl**. Assume now a discrete Helmholtz decomposition and set

$$\mathbf{e} = \mathbf{a} + \frac{1}{h}\mathbf{N}^\top \boldsymbol{\phi} \tag{2.12}$$

$$0 = \frac{1}{h}\mathbf{N}\mathbf{a} \tag{2.13}$$

where $\mathbf{a}$ is a (discrete) vector potential and $\boldsymbol{\phi}$ is a (discrete) scalar potential. Substituting in (2.11) we obtain

$$\begin{pmatrix} \frac{1}{h^2}\mathbf{T}\mathbf{M}(\mu)^{-1}\mathbf{T}^\top + i\omega\mathbf{F}^2\mathbf{S}(\sigma) & \frac{i\omega}{h}\mathbf{F}^2\mathbf{S}(\sigma)\mathbf{N}^\top \\ \frac{1}{h}\mathbf{N} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \boldsymbol{\phi} \end{pmatrix} = \begin{pmatrix} -i\omega\mathbf{F}^2\mathbf{s} \\ 0 \end{pmatrix} \tag{2.14}$$

Although the system (2.14) is well conditioned (see [23]) it is still difficult to solve. The (1,1) block is almost singular and the system is indefinite. Following our work [11] we obtain a strongly diagonally dominant system by the following steps.

1. Since $\mathbf{N}\mathbf{a} = 0$ we add $h^{-2}\mathbf{N}^\top\mathbf{M}_c(\mu)\mathbf{N}$ to the (1,1) block, where $\mathbf{M}_c(\mu) = \mathrm{diag}\,(\mu_1, \dots, \mu_m)$.
2. Multiplying the first row in (2.14) by $h\mathbf{N}$ (using $\mathbf{N}\mathbf{T} = \mathbf{0}$), multiplying the second row by $\mathbf{N}\mathbf{N}^\top\mathbf{M}_c^{-1}$ and subtracting.

This yields the following system of equations

$$\begin{pmatrix} \frac{1}{h^2}\left(\mathbf{T}\mathbf{M}^{-1}\mathbf{T}^\top + \mathbf{N}^\top\mathbf{M}_c^{-1}\mathbf{N}\right) + i\omega\mathbf{F}^2\mathbf{S} & \frac{i\omega}{h}\mathbf{F}^2\mathbf{S}\mathbf{N}^\top \\ \frac{1}{h}\mathbf{N}\mathbf{F}^2\mathbf{S} & \frac{1}{h^2}\mathbf{N}\mathbf{F}^2\mathbf{S}\mathbf{N}^\top \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \boldsymbol{\phi} \end{pmatrix} = -\begin{pmatrix} i\omega\mathbf{F}^2\mathbf{s} \\ \frac{1}{h}\mathbf{N}\mathbf{F}^2\mathbf{s} \end{pmatrix} \tag{2.15}$$

We note that this reformulation of obtaining a Poisson equation for the potential is similar to the pressure Poisson equation that has been extensively studied in computational fluid dynamics [25, 10].

It is clear that as long as $\omega \ll h^{-1}$ the system is diagonally dominant. This fact was used in [11] in order to use block diagonal preconditioners. Furthermore, the work in [1] developed a multigrid preconditioner of the form

$$\begin{pmatrix} \frac{1}{h^2}\left(\mathbf{T}\mathbf{M}^{-1}\mathbf{T}^\top + \mathbf{N}^\top\mathbf{M}_c^{-1}\mathbf{N}\right) & * \\ & \frac{1}{h^2}\mathbf{N}\mathbf{F}^2\mathbf{S}\mathbf{N}^\top \end{pmatrix}. \tag{2.16}$$

where $*$ is either zero (which corresponds to block Jaccobi) or $-\frac{i\omega}{h}\mathbf{F}^2\mathbf{S}\mathbf{N}^\top$ (which corresponds to a block Gauss Seidel). As shown in [1], under the assumption of constant coefficients, it is possible to verify that the condition number of the preconditioned system is h-independent and therefore, iterative methods tend to quickly converge.

However, in order to have an effective solution to the preconditioned problem one has to solve two types of PDE's. First, the (2,2) block corresponds to the electrostatic problem. Second, the (1,1) block corresponds to the magnetostatic problem. In the work of [1] jumps in the electric conductivity were considered and black box multigrid [6] was used to solve the discretization of the scalar PDE. However, multigrid algorithms for the magnetostatic problem with jumping coefficients tend to be inefficient. In the next Section we develop multigrid algorithms for both problems and combine them to precondition the system (2.15).

3 Multigrid

In this Section we develop a multigrid preconditioner for the system (2.16). We compare two different approaches.

M1. A multigrid preconditioner based on the system (2.16) where the electro and magneto static problems are solved separately. This was previously suggested in [1].

M2. A multigrid method applied directly to the system (2.15).

Both preconditioners requires the combination of two different multigrid solvers. First, we require to develop a multigrid solver for the discrete electrostatic problem

$$\frac{1}{h^2}\mathbf{N}\mathbf{F}^2\mathbf{S}\mathbf{N}^\top\boldsymbol{\phi} = \mathbf{b}_1. \tag{3.17}$$

Second, we require to develop a multigrid method for the discrete magnetostatic problem

$$\frac{1}{h^2}\left(\mathbf{T}\mathbf{M}^{-1}\mathbf{T}^\top + \mathbf{N}^\top\mathbf{M}_c^{-1}\mathbf{N}\right)\mathbf{a} = \mathbf{b}_2. \tag{3.18}$$

If there where no jumps in the coefficients and the grid was regular then, we could use any standard multigrid method to solve both problems. However, our grid does not have to be regular and furthermore, even for a regular grid, we assume that the matrices $\mathbf{M}, \mathbf{M}_c$ and $\mathbf{S}$ contain very large and very small entries which correspond to large jumps in the coefficients. This make standard multigrid methods ineffective.

As stated before, while methods for jumping coefficients exist for the electrostatic problem (see [18]) no effective method known to us exist for the magnetostatic problem. Our approach combines AMG ideas with semi-geometric multigrid. Rather than building the coarse grid from the matrix, we use the underlying given finest grid in order to build coarser grids that make "physical sense". The coarse grids are generated by a process of local coarsening rather than a global coarsening. This approach allows to interpolate only where the coefficients are smooth and does not interpolated over discontinuities. The idea of not to coarsen over discontinuities appears first in [28]. However, the work in [28] uses only tensor product grids and is developed only for the scalar electrostatic equation. Using tensor product grids can result in very fine coarse grids with bad aspect ratios especially if the discontinuities in the coefficients lie diagonally to the grid. In our experience, unless special smoothers are used (line relaxation) on coarser grids standard multigrid methods fail. Our approach builds on the ideas presented in [28] but uses the OcTree structure to avoid bad aspect ratios.

Local Coarsening

We now describe our coarsening process in more detail. Let S_h represent the fine OcTree grid and assume first that no discontinuities in material properties are present. The process of coarsening proceeds from the leafs up. If some of the leafs of the tree are fine then they are coarsen first. Coarsening in that form proceeds until we have a uniform (coarse) grid and the coarsening process continues in the usual way.

In the case of discontinuities in material properties we leave the cells with discontinuous coefficients unchanged and coarsen only cells where no discontinuities take place. This is demonstrated in Figure 5.

It is important to note that our coarse grid should be "fine enough" in order to resolve the main geometrical features of the coefficients. In our numerical experiments we have observed that this results in coarse grid approximations that can have a few hundreds to a few thousands of unknowns. Nevertheless, using the either the latest direct solution techniques for linear systems or a preconditioned Krylov method with an inexact factorization, such problems can be solved in negligible time compared with the relaxation process. Furthermore, since we are interested in very large scale problems,

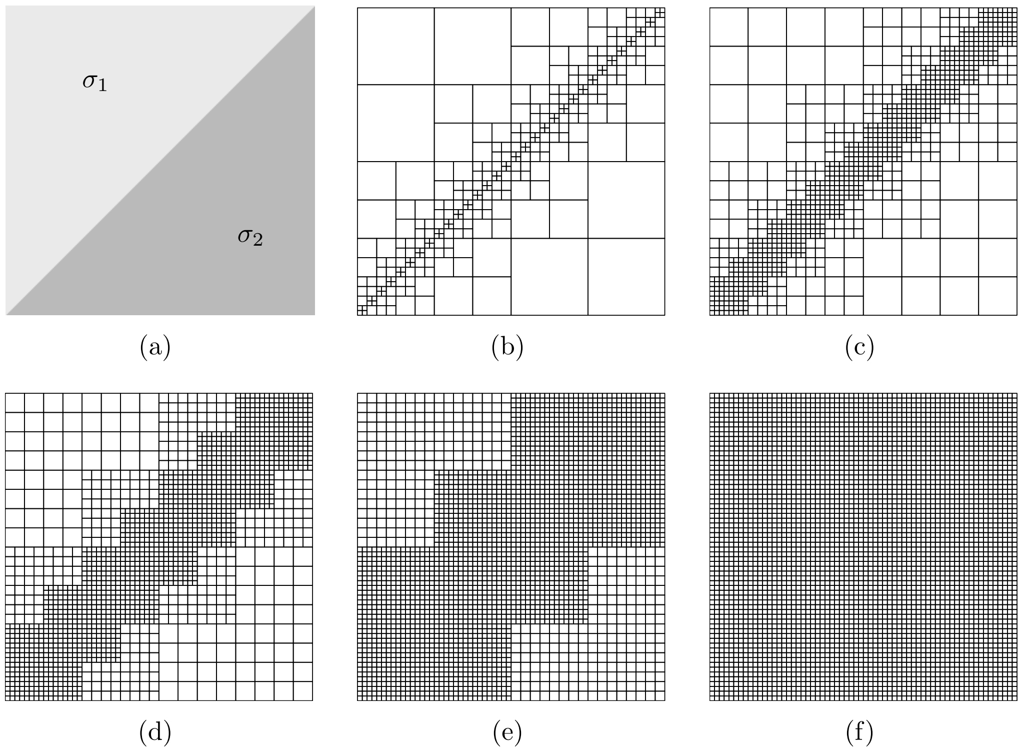


Figure 5: Multigrid refinement. Starting from the coarsest grid (b), we refine the coarsest cells and continue until we obtain the finest grid.

typically with millions of unknowns, the final reduction in size is substantial.

Prolongation and coarse grid operator

We use linear restriction and its adjoint as a prolongation. The (local) stencils for the face variables and cell centers are as presented in [26]. The main difference between classical prolongation and restriction operators to the one we use is that our restriction is only local. As stated before, this implies that some of the cells are not coarsened but others do. The prolongation matrix is therefore built from an identity part (for the unchanged entries in the matrix) and the usual linear prolongation for cells that are coarsened.

Let $\mathbf{P}_s^\top$ be the restriction matrix for the scalar electrostatic system and let $\mathbf{P}_v^\top$ be the restriction matrix for the vector magnetostatic system. The restriction matrix for the full Maxwell system (2.15) is simply set to

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_v & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_s \end{pmatrix}$$

Finally, we then use the Galerkin coarse grid approximation to generate the coarse grid operator setting

$$\mathbf{A}_H = \mathbf{P}^\top \mathbf{A}_h \mathbf{P}$$

Smoothing
For scalar problems we use the pointwise symmetric Gauss-Seidel for smoothing. For the magnetostatic problem or for the full Maxwell system, where the coefficients are highly discontinuous there is strong coupling between the different modes. Therefore, we use a box relaxation method as described in [26]. Consider a computational cell as shown in Figure 6. Each cell contains magnetic vector fluxes and one scalar potential that can be relaxed simultaneously. The relaxation is done cell-by-cell.

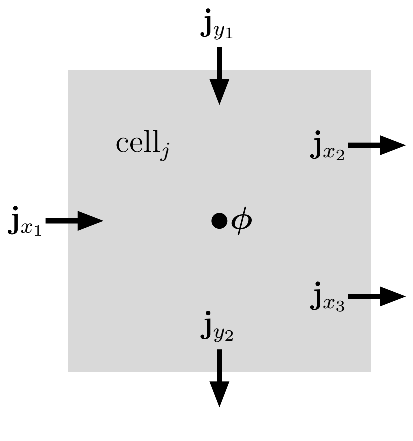


Figure 6: Computational cell and unknowns that relax simultaneously

Multigrid Cycle
The above components are combined into a multigrid V or W-cycle. We have

implemented either one of the classical cycles or an F-cycle with self tune refinement based on τ_h^{2h} criteria presented in [26].

Multigrid as a Preconditioner

For the electro and magneto-static problems, the above multigrid components are integrated as preconditioners into a PCG routine. For Maxwell's equations we use a BiCGStab [2] routine. Each BiCGStab iteration requires two applications of the preconditioner. Here we have used either a multigrid cycle for system (2.15) or similar to the work of [1] we use only a single multigrid cycle for the system (2.16). The multigrid preconditioner for (2.16) requires the solution of the electro and magnetostatic problems, which can be done independently while the multigrid cycle for equation (2.15) treats the whole system simultaneously.

4 Numerical experiments

In this section we describe numerical experiments that demonstrate the effectiveness of our approach. This section is structured as follows. We first describe a model problem, the material properties and the experiments we perform. Second, we demonstrate that we are able to solve the electrostatic problem for these cases using our multigrid method. Next, we demonstrate that we are able to solve the magnetostatic problem and finally we combine both in order to obtain an effective preconditioner for Maxwell's equations. For the electrostatic and the magnetostatic problem we compare our results to two other solution techniques. First, we compare our method to a PCG method with ILU(0) and second, we compare our method to the algebraic multigrid package ML [22].

4.1 Electromagnetic simulations

Modelling highly conductive and highly permeable objects is a common task in geophysical applications [9]. Some of the targets are metallic objects with conductivity that can vary from 10^1 to 10^6 S/m and relative magnetic permeability that varies from 1 to 10^2. They are buried in the ground which can have a conductivity of $\sigma_{\rm bg} = 1$ S/m and a relative magnetic permeability of $\mu_{\rm bg} = 1$. Such targets can have irregular shapes. It is therefore a great challenge to accurately model the response of these objects to an external magnetic field. As a model problem we simulate the electromagnetic response that evolves from the shape presented in Figure 4.1. The shape is made from a cone with a conductivity of σ_1 and magnetic permeability of μ_1. Inside this cone there is a cylinder. We set its electromagnetic properties to $\sigma_2 = 10\sigma_1$ and $\mu_2 = 2\mu_1$. Our goal is to solve the electromagnetic problem for a range of σ_1's, μ_1's and for low frequencies ω.

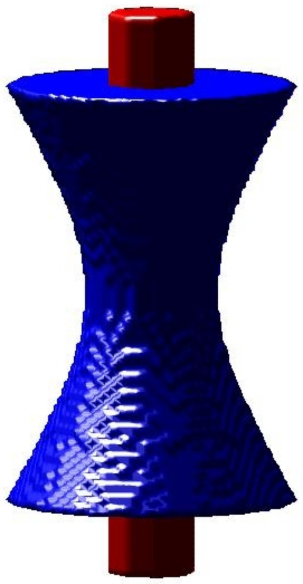


Figure 7: Electromagnetic model problem

In order to evaluate the performance of our multigrid method we solve the different problems setting the finest grid to $N = 16^3,\ 32^3,\ 64^3$ and 128^3 uniform cells.

The largest problem we solve (for $N = 128^3$) corresponds to $6,390,144$ magnetic potential unknowns and $2,097,152$ electric potential unknowns. Thus, the reformulated Maxwell's system has over 8 million degrees of freedom. The coarsest OcTree grid for this resolution has only 4,546 cells. In Figure 8 we use a mosaic plot for the grids generated by our approach where the finest grid is 32^3. As can be seen from these plots the reduction in the number of cells is substantial.

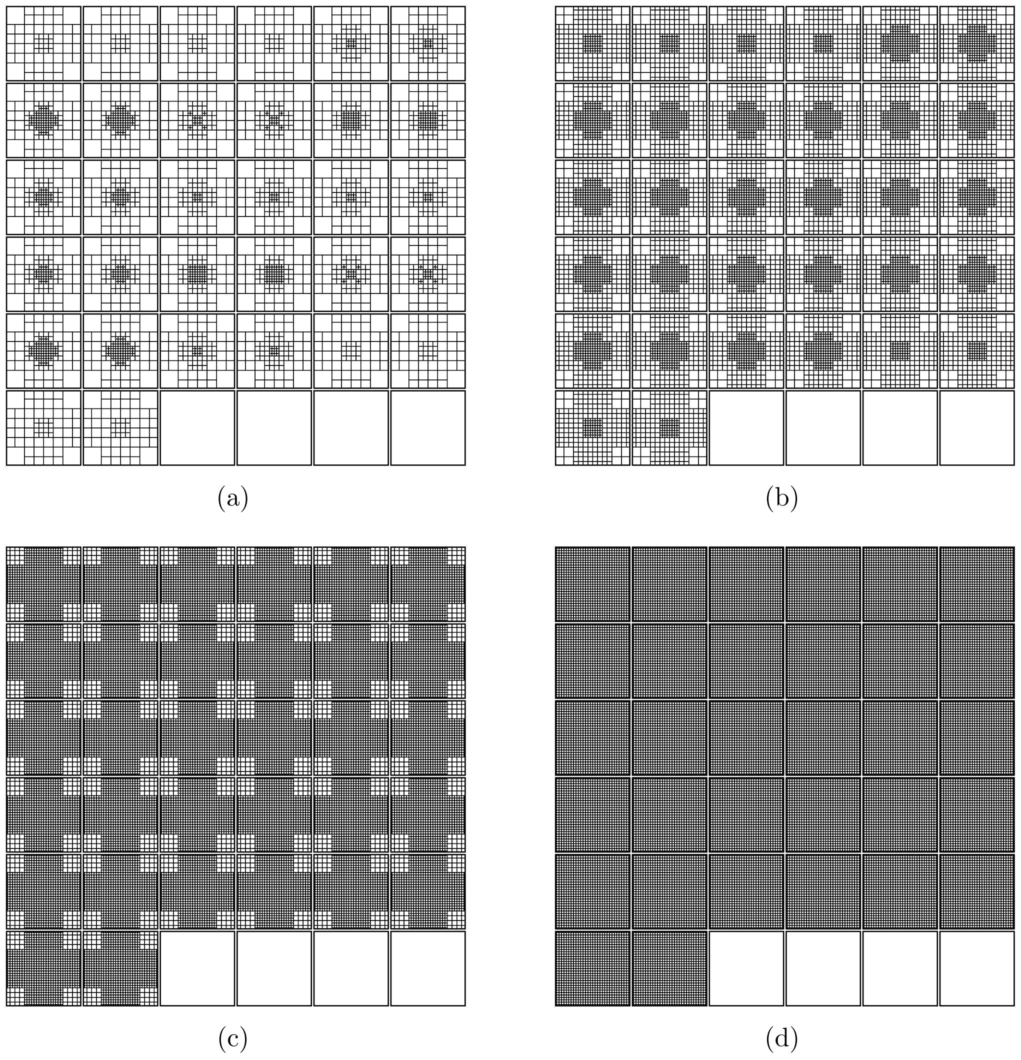


Figure 8: Grid hierarchy for 32^3 grid. The images (a)-(d) are mosaic plots of the 3D grids.

4.2 Experiments with the electrostatic problem

For the electrostatic problem we solve the scalar PDE $\nabla \cdot \sigma \nabla \phi = b$ on the different grids. We use a standard multigrid V(2,1) cycle as a preconditioner for a Conjugate Gradient method and stop the iteration when the relative residual is less than 10^{-6}. As we can see from the table there is a slight

$\sigma_1/\sigma_{\text{bg}}$	N	iterations
10^0	16^3	6
10^0	32^3	7
10^0	64^3	9
10^0	128^3	11
10^2	16^3	6
10^2	32^3	7
10^2	64^3	9
10^2	128^3	11
10^4	16^3	6
10^4	32^3	7
10^4	64^3	9
10^4	128^3	11
10^6	16^3	6
10^6	32^3	7
10^6	64^3	9
10^6	128^3	11

Table 2: Results for the electrostatic problem based on grid refinement as demonstrated in Figure 8

increase in the number of iterations as the grids get finer. Nevertheless, jumps in conductivity do not affect our method and the number of iterations stay similar for jumps that range from 1 to 10^6. Still, although our method does exhibit grid dependency it is highly effective and compares well with other solvers for the same problem [18, 6].

4.3 Experiments with the magnetostatic problem

For the magnetostatic problem we solve the system of PDEs $\nabla\times\mu^{-1}\nabla\times\vec{A} = \mathbf{b}$ on the different grids which range from 16^3 to 128^3. Again, we use a standard multigrid V(2,1) cycle as a preconditioner for a Conjugate Gradient method and stop the iteration when the relative residual is less than 10^{-6}.

Similar to our target application, we set $\mu/\mu_{\rm bg}$ between 1 and 10^2. We then use our multigrid method as a preconditioner to a PCG algorithm. The results are presented in Table 3.

$\mu/\mu_{\rm bg}$	N	iterations
10^0	16^3	5
10^0	32^3	5
10^0	64^3	5
10^0	128^3	6
10^1	16^3	5
10^1	32^3	5
10^1	64^3	6
10^1	128^3	6
10^2	16^3	6
10^2	32^3	6
10^2	64^3	6
10^2	128^3	7

Table 3: Results for the magnetostatic problem

Unlike the electrostatic problem we see that our method here is h-independent. However, a small increase in the number of iterations is observed for larger values of μ. To our knowledge, no other 3D multigrid method performs as well for such problems. To compare, we have experimented with the multigrid package ML package [22]. The results of a V(2,1) cycle for large

μ/μ_{bg}	N	iterations
10^0	16^3	8
10^0	32^3	9
10^0	64^3	13
10^1	16^3	13
10^1	32^3	18
10^1	64^3	25
10^2	16^3	41
10^2	32^3	52
10^2	64^3	77

Table 4: Results for the solution of the magnetostatic problem using ML

jumps in μ are given in Table 4. Note that by construction, the complexity of our method is 1 while the complexity of ML is roughly 1.5.

Comparison of our method to PCG with ILU(0) as a preconditioner is presented in Table 5. We see that ILU(0) is not an effective method for problems with jumping coefficients.

μ/μ_{bg}	N	iterations
10^2	16^3	161
10^2	32^3	295
10^2	64^3	502

Table 5: Results for the solution of the magnetostatic problem using ILU(0) preconditioner

4.4 Experiments with Maxwell's equations

We now combine the two solvers above into a preconditioner for the discrete reformulation of Maxwell's equations (2.15). We use the same grid as above but since the system (2.15) is nonsymmetric we use BiCGStab [2] for the solution of the system. We experiment with a range of μ_1/μ_{bg}, $\sigma_1/\sigma_{\text{bg}}$ and frequencies ω. The results are presented in Table 6. Once again we see a slight increase in the number of iterations for finer grids and larger jumps.

$N = 16^3$		
$\sigma_1/\sigma_{\text{bg}}$	μ_1/μ_{bg}	iterations
10^2	10^0	3.5
10^2	10^1	2.5
10^2	10^2	4.5
10^4	10^0	3.5
10^4	10^1	2.5
10^4	10^2	4.5

$N = 32^3$		
$\sigma_1/\sigma_{\text{bg}}$	μ_1/μ_{bg}	iterations
10^2	10^0	3.5
10^2	10^1	3
10^2	10^2	5
10^4	10^0	3.5
10^4	10^1	3
10^4	10^2	5

$N = 64^3$		
$\sigma_1/\sigma_{\text{bg}}$	μ_1/μ_{bg}	iterations
10^2	10^0	3.5
10^2	10^1	3.5
10^2	10^2	5.5
10^4	10^0	3.5
10^4	10^1	3.5
10^4	10^2	5.5

$N = 128^3$		
$\sigma_1/\sigma_{\text{bg}}$	μ_1/μ_{bg}	iterations
10^2	10^0	3.5
10^2	10^1	3.5
10^2	10^2	5.5
10^4	10^0	3.5
10^4	10^1	3.5
10^4	10^2	5.5

Table 6: BiCGSTAB iterations for Maxwell's equations. Each iteration requires 2 preconditioning steps ($\omega = 10^3$)

The results demonstrate that we have developed a highly effective multigrid method for Maxwell's equations with jumping coefficients.

5 Conclusions

In this paper we have developed an OcTree method for the numerical solution of 3D quasi-static Maxwell's equations. We have discussed the discretization of the problem and proposed a multigrid preconditioner. Our preconditioner is highly effective in solving problems that have jumpy coefficients. As for all multigrid methods there are many parameters that need to be ”fine tuned”. We have found that W-cycles tend to be much more robust compared to V-cycles. The number of smoothing steps had less effect on the final results. Further analysis is needed in order to understand the sensitivity

References

[1] D. Aruliah and U. Ascher. Multigrid preconditioning for time-harmonic maxwell's equations in 3D. *SIAM J. on Sci. Comp.*, 24:702–718, 2003.

[2] R. Barrett, M. Berry, T.F. Chan, J. Demmel, , J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.* SIAM, Philadelphia, 1994.

[3] P. Bochev, J. Hu, A. Robinson, and R. Tuminaro. Towards robust 3d z-pinch simulations: discretization and fast solvers for magnetic diffusion in heterogeneous conductors. *Elec. Trans. Numer. Analysis*, 14:23–34, 2002.

[4] A. Bossavit and L. Kettunen. Yee-like schemes on staggered cellular grids: a synthesis between and fem approaches. *COMPUMAG*, 1999. Short contribution.

[5] M. Clemens and T. Weiland. Numerical algorithms for the FDITD and FDFD simulation of slowly varying electromagnetic fields. *Int. J. Num. Modelling*, 1999. To appear.

[6] J. E. Dendy. Black box multigrid. *Journal of Computational Physics*, 48:366–386, 1982.

[7] M. Edwards. Elimination of adaptive grid interface errors in the discrete cell centered pressure equation. *Journal of Computational Physics*, 126:356–372, 1996.

[8] R.E. Ewing, R.D. Lazarov, and P.S. Vassilevski. Local refinement techniques for elliptic problems on cell-centered grids i, error analysis. *Math. Comp.*, 56:437461, 1991.

[9] N. Geng, C.E. Baum, and L. Carin. On the low-frequency natural response of conducting and permeable targets. *IEEE Trans. Geoscience and Remote Sensing*, 37:pp. 347–359, 1999.

[10] P. M. Gresho and R. L. Sani. On pressure boundary conditions for the incompressible Navier-Stokes equations. *Internat. J. Numer. Methods Fluids*, 7:1111–1145, 1987.

[11] E. Haber and U. Ascher. Fast finite volume simulation of 3D electromagnetic problems with highly discontinuous coefficients. *SIAM J. Scient. Comput.*, 22:1943–1961, 2001.

[12] E. Haber, U. Ascher, D. Aruliah, and D. Oldenburg. Fast simulation of 3D electromagnetic using potentials. *J. Comput. Phys.*, 163:150–171, 2000.

[13] E. Haber, U. Ascher, and D. Oldenburg. On optimization techniques for solving nonlinear inverse problems. *Inverse problems*, 16:1263–1280, 2000.

[14] R. Hiptmair. Multigrid method for Maxwell's equations. *SIAM J. Numer. Anal*, 36:204–225, 1998.

[15] G. R. Hjaltason and H. Samet. Speeding up construction of quadtrees for spatial indexing. *The VLDB Journal*, 11:109–137, 2002.

[16] J.J. Hu, R.S. Tuminaro, P.B. Tuminaro, C.J. Garasi, and A.C. Robinson. Towards an h-independent algebraic multigrid method for maxwell's equation. *SIAM J. on Sci. Comp.*, To Appear:XXX, 2006.

[17] J.M. Hyman and M. Shashkov. Mimetic discretizations for Maxwell's equations. *J. Comp. Phys.*, 151:881–909, 1999.

[18] S. Knapek. Matrix-dependent multigrid homogenization for diffusion problems. *SIAM J. Sci. Comp.*, 20(2):515–533, 1999.

[19] K. Lipnikov, J. Morel, and M. Shashkov. Mimetic finite difference methods for diffusion equations on non-orthogonal amr meshes. *Journal of Computational Physics*, 199:589–597, 2004.

[20] F. Losasso, R. Fedkiw, and S. Osher. Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids*, 35:457–462, 2006.

[21] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure. *SIGGRAPH*, 23:457–462, 2004.

[22] J. Hu M. Sala and R. S. Tuminaro. Ml 3.1 smoothed aggregation user's guide. *Technical Report SAND2004-4819*, Sandia National Laboratories, 2004.

[23] P. Monk. *Finite Element methods for Maxwell's equations.* Oxsford University Press, 2003.

[24] R. Reitzinger and J. Schoberl. An algebraic multigrid method for finite element discretizations with edge elements. *Numer. Linear Algebra Appl.*, 9:223–238, 2002.

[25] D. Sidilkover and U. Ascher. A multigrid solver for the steady state navier-stokes equations using the pressure-poisson formulation. *Comp. Appl. Math. (SBMAC)*, 14:21–35, 1995.

[26] U. Trottenberg, C. Oosterlee, and A. Schuller. *Multigrid*. Academic Press, 2001.

[27] P. Vanek, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. Technical Report UCD-CCM-036, 1, 1995.

[28] J.W.L Wan. Interface preserving coarsening for elliptic problems with highly discontinuous coefficients. *Numer. Lin. Alg. Appl.*, 7:727–741, 2000.

[29] Z.J. Wang, A.J. Przekwasb, and Y. Liuc. A fv-td electromagnetic solver using adaptive cartesian grids. *Journal of Computational Physics*, 148:17–29, 2002.

[30] S.H. Ward and G.W. Hohmann. Electromagnetic theory for geophysical applications. *Electromagnetic Methods in Applied Geophysics*, 1:131–311, 1988. Soc. Expl. Geophys.

[31] K.S. Yee. Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media. *IEEE Trans. on antennas and propagation*, 14:302–307, 1966.