Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 223 (2007) 783–796
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

An octree multigrid method for quasi-static Maxwell's equations with highly discontinuous coefficients

Eldad Haber *, Stefan Heldmann

Department of Mathematics and Computer Science, Emory University, 400 Dowman Drive Suite W 401, Atlanta, GA 30322, United States

Received 13 February 2006; received in revised form 8 August 2006; accepted 3 October 2006
Available online 5 December 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

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

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

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 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 ranges from 10^{-2} to 10^4 S/m and the relative μ range from 1 to 100 (see [31]). For ease of presentation we assume perfectly magnetic conductor (PMC) boundary conditions which read

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

although other boundary conditions can be considered.

* Corresponding author.
E-mail address: haber@mathcs.emory.edu (E. Haber).

0021-9991/$ - see front matter © 2006 Published by Elsevier Inc.
doi:10.1016/j.jcp.2006.10.012

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 approximations, and have been extensionally studied in the last decade; see [11,13,14,16,24,25] and references within. While some finite volume algorithms can effectively deal with highly discontinuous conductivity [1] to our best knowledge none of the finite volume 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 dependent prolongation and restriction [6,19,27–29]. While approaches such as the one suggested in black box multigrid [6] work well for problems that evolve from scalar PDEs, they are not adequate for dealing with problems that evolve from systems of PDEs. A common AMG approach for systems of PDEs use smooth aggregation [28]. 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 Helmholtz decompositions 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 PDEs 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 [21,22]. 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 [20]. 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 [30] 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 [4,5,17,20] which can be thought of extending Yee type discretization [32] 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. Fig. 1 shows the three kind of discretizations for a small example.

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 3D mesh 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 are 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 Fig. 2.

2.2. Discretization of the *div* and *grad*

We are interested in the solution of Maxwell's equation (1.1b) where the **div** and **grad** operators do not show explicitly. However, as we see in Section 2.5 we need to obtain a discrete approximation to the **div** in order to deal with the null space of the **curl**. The gradient operator is never used explicitly in our approach and we introduce it for the sake of completeness.

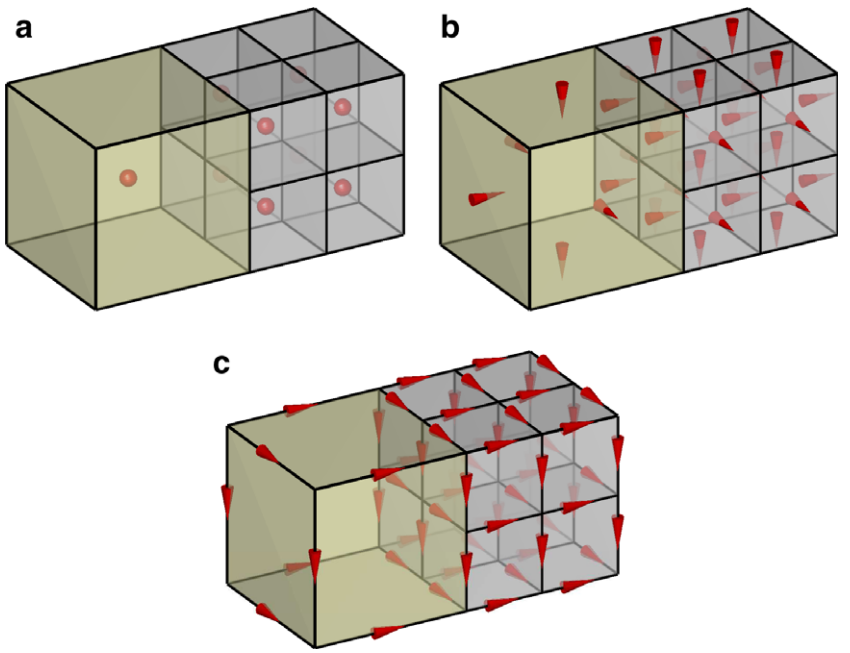

Fig. 1. OcTree discretization: (a) cell-center, (b) face-center, (c) edge-center.

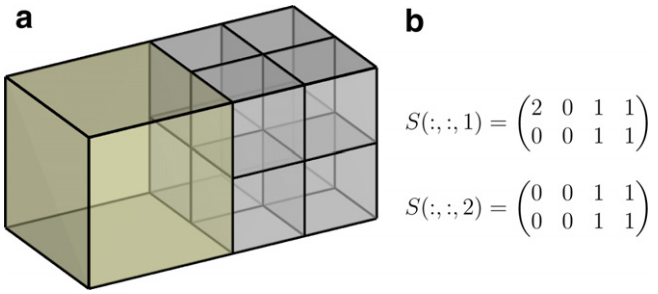

Fig. 2. (a) OcTree and (b) and its representation as $2 \times 4 \times 2$ (sparse) array. The notation $S(:,:,i)$ implies the ith plain in the 3D array S.

Although our code is 3D it is worth while to follow the discretization in 2D for simplicity. To discretize the divergence operator we use the usual flux-balance approach. Consider a 2D cell shown in Fig. 3. We discretize the flux, $\vec{J} = (J_x, J_y)^{\mathrm{T}}$ on the faces of the cell and using Gauss formula we write for the divergence of cell_j with volume V_j

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

It is easy to verify that this standard midpoint discretization of the integral 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, **j**, and 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

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

It is useful to express this discretization in somewhat different form. Let $\mathbf{V} = \mathrm{diag}\{V_1, \ldots, V_m\}$ be a diagonal matrix which contains the relative volume of each cell (i.e. w.r.t the smallest cell). Let $\mathbf{F} = \mathrm{diag}\{F_1, \ldots, F_{m_f}\}$ be a diagonal matrix which contains the relative 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

$$\mathbf{DIV} = \frac{1}{h}\mathbf{V}^{-1}\mathbf{NF} \text{ in 2D} \quad \text{and} \quad \mathbf{DIV} = \frac{1}{h}\mathbf{V}^{-1}\mathbf{NF}^2 \text{ in 3D}. \tag{2.3}$$

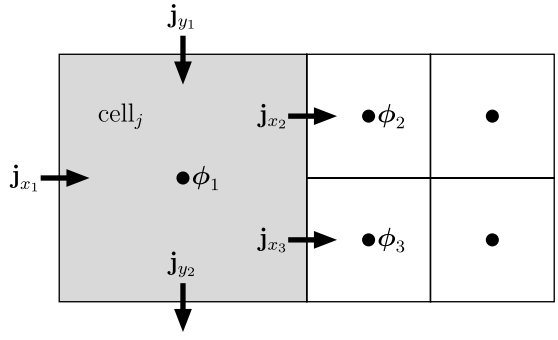

Fig. 3. Discretization of the divergence.

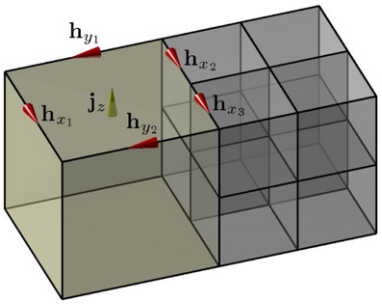

Fig. 4. Discretization of edge variables.

There are various ways to define the discrete gradient operator. Here, the grad operator maps from cell-centers to the faces of our grid. Using the usual 2-point formula [21,8], the gradient is a scaled transposed of the **DIV** and reads

$$\mathbf{GRAD} = -\frac{1}{h}\mathbf{F}^{-1}\mathbf{N}^{\mathrm{T}}. \tag{2.4}$$

While other higher order approximations can be used [18,8] they are not required in our application.[1]

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 from Stokes theorem that

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

Consider the upper face plotted in Fig. 4. A straight forward second order discretization reads

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

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 $\mathbf{E} = \mathrm{diag}\{E_1, \ldots, E_{m_e}\}$, 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 values ± 1 and 0. The sign of the entry is determined by the direction of the edge with respect to the orientation of the face the **curl** is integrated on. Using these matrices the **curl** matrix can be written as

$$\mathbf{CURL} = \frac{1}{h}\mathbf{F}^{-2}\mathbf{T}\mathbf{E}. \tag{2.5}$$

Unlike the discretization of the **curl** the discretization of its adjoint is not trivial. The adjoint maps from cell-faces to cell-edges and it is difficult to keep second order accuracy everywhere. Here, similar to the discretization of the 2-point gradient we set

[1] we show next that only the null space of the discrete **curl** is needed.

$$\overline{\mathbf{CURL}} = \mathbf{E}^{-1}\mathbf{T}^{\mathrm{T}}. \tag{2.6}$$

Although such approximation of the adjoint is only $\mathcal{O}(1)$, it still yields a second order accurate solution of the Maxwell's equations for the edge variables. To see that, note that Maxwell's equations can be derived from a variational principle [32] of finding the extremum point of the functional

$$\mathscr{F}[\vec{H}] = \int_{\Omega} \sigma^{-1}|\nabla \times \vec{H}|^2 + \mathrm{i}\omega\mu|\vec{H}|^2 - 2(\nabla \times \vec{H}) \cdot (\sigma^{-1}\vec{s})\, \mathrm{d}\mathbf{x}$$

such that (2.5) yields a second order accurate discretization of $\mathscr{F}$. Although the adjoint of the **curl** does not appear directly in the functional, (2.6) arises naturally in the discretized variational form.

2.4. Discretization of material averaging

It is common to have material properties given (or discretized) at cell centers. 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 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 performs arithmetic averaging from cell-centers to cell-faces and we define $\mathbf{A}_e$ as an $m_e \times m$ averaging matrix from cell-centers to cell-edges. Then, the material properties can be expressed as

$$\mathbf{S}(\sigma) = \mathrm{diag}((\mathbf{A}_f\sigma^{-1})^{-1}), \tag{2.7a}$$

$$\mathbf{M}(\mu) = \mathrm{diag}(\mathbf{A}_e\mu), \tag{2.7b}$$

where the inverse in (2.7a) is meant componentwise.

Furthermore, to avoid inaccurate discretizations of derivatives, special attention is given such that we do not have discontinuous material properties where the grid is rapidly changing.

2.5. Assembly of the linear system and its properties

Given the above differential operators we can 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 grid functions **e** and **h**

$$\overline{\mathbf{CURL}}\,\mathbf{e} + \mathrm{i}\omega\mathbf{M}(\mu)\mathbf{h} = 0, \tag{2.8a}$$

$$\mathbf{CURL}\,\mathbf{h} - \mathbf{S}(\sigma)\mathbf{e} = \mathbf{s}. \tag{2.8b}$$

Although it is possible (at least in principle) to directly solve the discrete system (2.8b) 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 we take advantage from for our stabilization.

(a) As noted in [5] the matrices **T** and **N** span the whole space, that is $\mathrm{span}(\mathbf{T}) \cup \mathrm{span}(\mathbf{N}) = \mathscr{R}^n$.
(b) It is possible to verify that **T** and **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

$$\left(\frac{1}{h^2}\mathbf{T}\mathbf{M}(\mu)^{-1}\mathbf{T}^{\mathrm{T}} + \mathrm{i}\omega\mathbf{F}^2\mathbf{S}(\sigma)\right)\mathbf{e} = -\mathrm{i}\omega\mathbf{F}^2\mathbf{s}. \tag{2.9}$$

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

$$\mathbf{e} = \mathbf{a} + \frac{1}{h}\mathbf{N}^{\mathrm{T}}\boldsymbol{\phi}, \tag{2.10}$$

$$0 = \frac{1}{h}\mathbf{N}\mathbf{a}, \tag{2.11}$$

where $\mathbf{a}$ is a (discrete) vector potential and $\boldsymbol{\phi}$ is a (discrete) scalar potential. Substituting in (2.9) we obtain

$$\begin{pmatrix} \frac{1}{h^2}\mathbf{T}\mathbf{M}(\mu)^{-1}\mathbf{T}^{\mathrm{T}} + \mathrm{i}\omega\mathbf{F}^2\mathbf{S}(\sigma) & \frac{\mathrm{i}\omega}{h}\mathbf{F}^2\mathbf{S}(\sigma)\mathbf{N}^{\mathrm{T}} \\ \frac{1}{h}\mathbf{N} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \boldsymbol{\phi} \end{pmatrix} = \begin{pmatrix} -\mathrm{i}\omega\mathbf{F}^2\mathbf{s} \\ 0 \end{pmatrix}. \tag{2.12}$$

Although the system (2.12) is well conditioned (see [24]) 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 $h^{-2}\mathbf{N}^{\mathrm{T}}\mathbf{M}_c(\mu)\mathbf{N}$ to the (1,1) block, where $\mathbf{M}_c(\mu) = \mathrm{diag}(\mu_1, \ldots, \mu_m)$.
2. Multiplying the first row in (2.12) by $h\mathbf{N}$ (using $\mathbf{N}\mathbf{T} = \mathbf{0}$), multiplying the second row by $\mathbf{N}\mathbf{N}^{\mathrm{T}}\mathbf{M}_c^{-1}$ and subtracting.

This yields the following system of equations

$$\begin{pmatrix} \frac{1}{h^2}\left(\mathbf{T}\mathbf{M}^{-1}\mathbf{T}^{\mathrm{T}} + \mathbf{N}^{\mathrm{T}}\mathbf{M}_c^{-1}\mathbf{N}\right) + \mathrm{i}\omega\mathbf{F}^2\mathbf{S} & \frac{\mathrm{i}\omega}{h}\mathbf{F}^2\mathbf{S}\mathbf{N}^{\mathrm{T}} \\ \frac{1}{h}\mathbf{N}\mathbf{F}^2\mathbf{S} & \frac{1}{h^2}\mathbf{N}\mathbf{F}^2\mathbf{S}\mathbf{N}^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ \boldsymbol{\phi} \end{pmatrix} = -\begin{pmatrix} \mathrm{i}\omega\mathbf{F}^2\mathbf{s} \\ \frac{1}{h}\mathbf{N}\mathbf{F}^2\mathbf{s} \end{pmatrix}. \tag{2.13}$$

We note that this reformulation as a Poisson-like equation for the potential is similar to the pressure Poisson equation that has been extensively studied in computational fluid dynamics [26,10].

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

$$\begin{pmatrix} \frac{1}{h^2}\left(\mathbf{T}\mathbf{M}^{-1}\mathbf{T}^{\mathrm{T}} + \mathbf{N}^{\mathrm{T}}\mathbf{M}_c^{-1}\mathbf{N}\right) & * \\ & \frac{1}{h^2}\mathbf{N}\mathbf{F}^2\mathbf{S}\mathbf{N}^{\mathrm{T}} \end{pmatrix}. \tag{2.14}$$

where $*$ is either zero (which corresponds to block Jacobi) or $-\frac{\mathrm{i}\omega}{h}\mathbf{F}^2\mathbf{S}\mathbf{N}^{\mathrm{T}}$ (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 PDEs. 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 (2.13).

3. Multigrid

In this section we develop a multigrid preconditioner for the system (2.14). We compare two different approaches.

M1. A multigrid preconditioner based on the system (2.14) where the electro- and magnetostatic problems are solved separately. This was previously suggested in [1].
M2. A multigrid method applied directly to the system (2.13).

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

$$\frac{1}{h^2}\mathbf{N}\mathbf{F}^2\mathbf{S}\mathbf{N}^{\mathrm{T}}\boldsymbol{\phi} = \mathbf{b}_1. \tag{3.15}$$

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

$$\frac{1}{h^2}(\mathbf{T}\mathbf{M}^{-1}\mathbf{T}^{\mathrm{T}} + \mathbf{N}^{\mathrm{T}}\mathbf{M}_c^{-1}\mathbf{N})\mathbf{a} = \mathbf{b}_2. \tag{3.16}$$

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 [19]) no effective method, known to us, exists 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 [29]. However, the work in [29] 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 [29] but uses the OcTree structure to avoid bad aspect ratios.

3.1. 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 Fig. 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

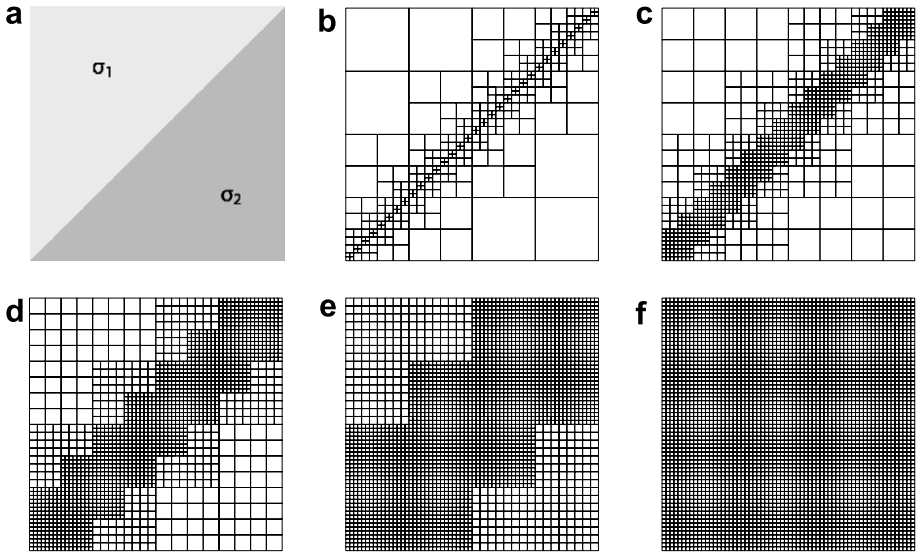

Fig. 5. Multigrid refinement. Starting from the coarsest grid (b), we refine the coarsest cells and continue until we obtain the finest grid.

factorization, such problems can be solved in negligible time compared with the relaxation process. Furthermore, since we are interested in very large scale problems, typically with millions of unknowns, the final reduction in size is substantial.

3.2. 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 [27]. 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 $\mathbf{P}_s^{\mathrm{T}}$ be the restriction matrix for the scalar electrostatic system and let $\mathbf{P}_v^{\mathrm{T}}$ be the restriction matrix for the vector magnetostatic system. The restriction matrix for the full Maxwell system (2.13) 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

$$\mathbf{A}_H = \mathbf{P}^{\mathrm{T}}\mathbf{A}_h\mathbf{P}.$$

3.3. 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 [27]. Consider a computational cell as shown in Fig. 6. Each cell contains magnetic vector fluxes and one scalar potential that can be relaxed simultaneously. The relaxation is done cell-by-cell.

3.4. 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 [27].

3.5. Multigrid as a preconditioner

For the electro- and magnetostatic 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 (2.13) or similar to the work of [1] we use only a single multigrid cycle for the system (2.14). The multigrid precondi-

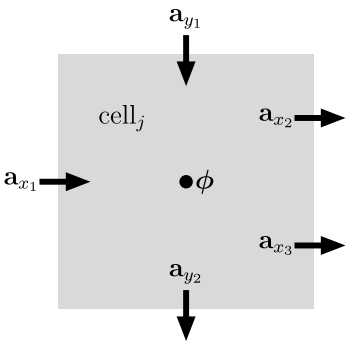

Fig. 6. Computational cell and unknowns that relax simultaneously.

tioner for (2.14) requires the solution of the electro- and magnetostatic problems, which can be done independently while the multigrid cycle for Eq. (2.13) 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 electro- and magnetostatic problems 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 [23].

4.1. Electromagnetic simulations

Modeling 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 $\sigma_{bg} = 1$ S/m and a relative magnetic permeability of $\mu_{bg} = 1$. 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 Fig. 7. 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.

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 4546 cells. In Fig. 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.

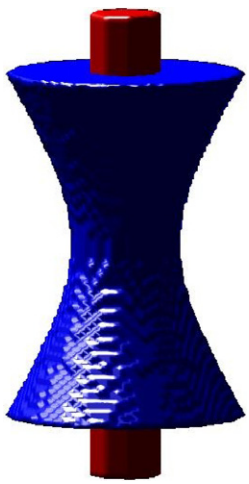

Fig. 7. Electromagnetic model problem.

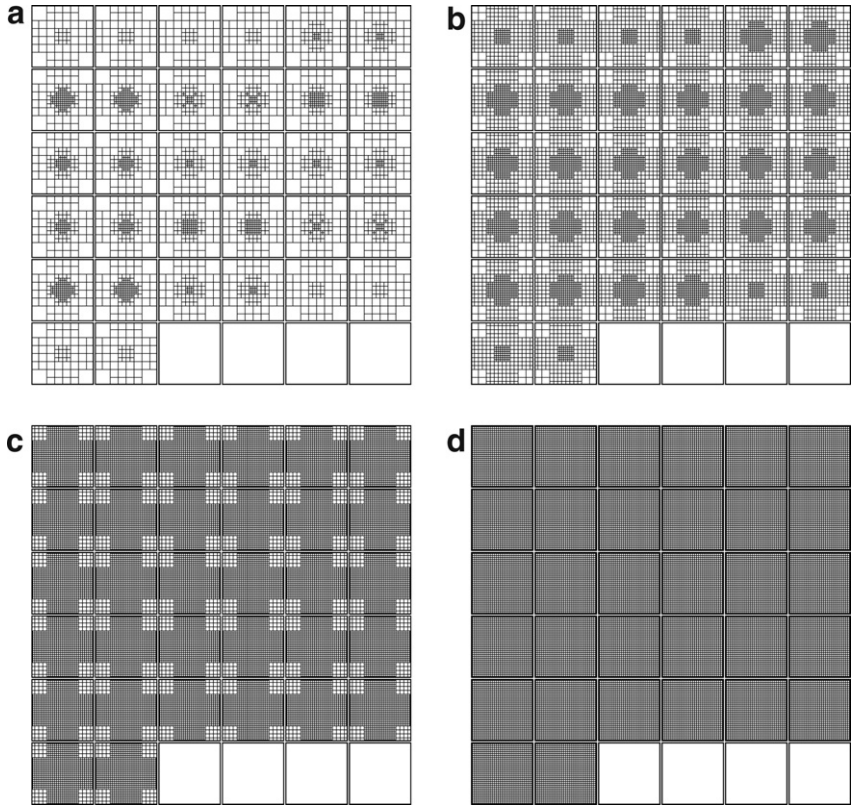

Fig. 8. Grid hierarchy for 32^3 grid. The images (a)–(d) are mosaic plots of the 3D grids.

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 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 [6,19]. For comparison, Table 1 also shows the results for the ML package.

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 (regular) grids which range from 16^3 to 128^3. Again, we use a standard multigrid V(2, 1) cycle based on OcTree grid refinement 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 μ/μ_{bg} between 1 and 10^2.

To compare, we have experimented with the multigrid package ML package [23] and PCG with ILU(0) as preconditioner. The results for large jumps in μ are given in Table 2. 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. Note that by construction, the complexity of our method is 1 while the complexity of ML is roughly 1.5. Comparing our method to PCG with ILU(0) as a preconditioner, we see that ILU(0) is not an effective method for problems with jumping coefficients.

Table 1
Results for the electrostatic problem based on OcTree grid refinement as demonstrated in Fig. 8 and the ML package

σ_1/σ_{bg}	N	OcTree #iterations	ML #iterations
10^0	16^3	6	7
10^0	32^3	7	7
10^0	64^3	9	8
10^0	128^3	11	10
10^2	16^3	6	9
10^2	32^3	7	10
10^2	64^3	9	12
10^2	128^3	11	15
10^4	16^3	6	9
10^4	32^3	7	11
10^4	64^3	9	14
10^4	128^3	11	18
10^6	16^3	6	9
10^6	32^3	7	11
10^6	64^3	9	15
10^6	128^3	11	20

Table 2
Results for the magnetostatic problem

μ/μ_{bg}	N	OcTree #iterations	ML #iterations	ILU(0) #iterations
10^0	16^3	5	8	–
10^0	32^3	5	9	–
10^0	64^3	5	13	–
10^0	128^3	6	–	–
10^1	16^3	5	13	–
10^1	32^3	5	18	–
10^1	64^3	6	25	–
10^1	128^3	6	–	–
10^2	16^3	6	41	161
10^2	32^3	6	52	295
10^2	64^3	6	77	502
10^2	128^3	7	–	–

Table 3
BiCGStab iterations for Maxwell's equations

σ_1/σ_{bg}	μ_1/μ_{bg}	#Iterations	σ_1/σ_{bg}	μ_1/μ_{bg}	#Iterations
$N = 16^3$			$N = 32^3$		
10^2	10^0	3.5	10^2	10^0	3.5
10^2	10^1	2.5	10^2	10^1	3
10^2	10^2	4.5	10^2	10^2	5
10^4	10^0	3.5	10^4	10^0	3.5
10^4	10^1	2.5	10^4	10^1	3
10^4	10^2	4.5	10^4	10^2	5
$N = 64^3$			$N = 128^3$		
10^2	10^0	3.5	10^2	10^0	3.5
10^2	10^1	3.5	10^2	10^1	3.5
10^2	10^2	5.5	10^2	10^2	5.5
10^4	10^0	3.5	10^4	10^0	3.5
10^4	10^1	3.5	10^4	10^1	3.5
10^4	10^2	5.5	10^4	10^2	5.5

Each iteration requires two preconditioning steps ($\omega = 10^3$).

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 2.13. We use the same grid as above but since the system (2.13) is non-symmetric we use BiCGStab [2] for the solution of the system. We experiment with a range of μ_1/μ_{bg}, σ_1/σ_{bg} and frequencies ω. The results are presented in Table 3. Once again we see a slight increase in the number of iterations for finer grids and larger jumps. 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, U. Ascher, Multigrid preconditioning for time-harmonic maxwell's equations in 3D, SIAM J. Sci. Comp. 24 (2003) 702–718.
[2] R. Barrett, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, 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, R. Tuminaro, Towards robust 3d z-pinch simulations: discretization and fast solvers for magnetic diffusion in heterogeneous conductors, Electron. Trans. Numer. Anal. 14 (2002) 23–34.
[4] A. Bossavit, L. Kettunen, Yee-like schemes on staggered cellular grids: a synthesis between and fem approaches, COMPUMAG, 1999, short contribution.
[5] M. Clemens, T. Weiland, Numerical algorithms for the FDITD and FDTD simulation of slowly varying electromagnetic fields, Int. J. Numer. Modelling: Electron. Networks, Devices Fields (NumMod) 12 (2) (1999) 3–22.
[6] J.E. Dendy, Black box multigrid, J. Comput. Phys. 48 (1982) 366–386.
[7] M. Edwards, Elimination of adaptive grid interface errors in the discrete cell centered pressure equation, J. Comput. Phys. 126 (1996) 356–372.
[8] R.E. Ewing, R.D. Lazarov, P.S. Vassilevski, Local refinement techniques for elliptic problems on cell-centered grids i, error analysis, Math. Comp. 56 (1991) 437461.
[9] N. Geng, C.E. Baum, L. Carin, On the low-frequency natural response of conducting and permeable targets, IEEE Trans. Geosci. Remote Sensing 37 (1999) 347–359.
[10] P.M. Gresho, R.L. Sani, On pressure boundary conditions for the incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 7 (1987) 1111–1145.
[11] E. Haber, U. Ascher, Fast finite volume simulation of 3D electromagnetic problems with highly discontinuous coefficients, SIAM J. Sci. Comput. 22 (2001) 1943–1961.
[12] E. Haber, U. Ascher, D. Aruliah, D. Oldenburg, Fast simulation of 3D electromagnetic using potentials, J. Comput. Phys. 163 (2000) 150–171.
[13] E. Haber, U. Ascher, D. Oldenburg, On optimization techniques for solving nonlinear inverse problems, Inverse Problems 16 (2000) 1263–1280.
[14] R. Hiptmair, Multigrid method for Maxwell's equations, SIAM J. Numer. Anal. 36 (1998) 204–225.
[15] G.R. Hjaltason, H. Samet, Speeding up construction of quadtrees for spatial indexing, VLDB J. 11 (2002) 109–137.
[16] J. Hu, R. Tuminaro, P. Bochev, C. Garasi, A. Robinson, Towards an *h*-independent algebraic multigrid method for maxwell's equation, SIAM J. Sci. Comput. 27 (5) (2006) 1669–1708.
[17] J.M. Hyman, M. Shashkov, Mimetic discretizations for Maxwell's equations, J. Comput. Phys. 151 (1999) 881–909.
[18] Sanjay Kumar Khattri, Numerical tools for multicomponent, multiphase, reactive processes: flow of CO_2 in porous medium, Ph.D. Dissertation, University of Bergen, Norway, 2006.
[19] S. Knapek, Matrix-dependent multigrid homogenization for diffusion problems, SIAM J. Sci. Comp. 20 (2) (1999) 515–533.
[20] K. Lipnikov, J. Morel, M. Shashkov, Mimetic finite difference methods for diffusion equations on non-orthogonal amr meshes, J. Comput. Phys. 199 (2004) 589–597.
[21] F. Losasso, R. Fedkiw, S. Osher, Spatially adaptive techniques for level set methods and incompressible flow, Comput. Fluids 35 (2006) 457–462.
[22] F. Losasso, F. Gibou, R. Fedkiw, Simulating water and smoke with an octree data structure, SIGGRAPH 23 (2004) 457–462.

[23] J. Hu, M. Sala, R.S. Tuminaro, Ml 3.1 smoothed aggregation user's guide, Technical Report SAND2004-4819, Sandia National Laboratories, 2004.
[24] P. Monk, Finite Element Methods for Maxwell's Equations, Oxford University Press, Oxford, 2003.
[25] R. Reitzinger, J. Schoberl, An algebraic multigrid method for finite element discretizations with edge elements, Numer. Linear Algebra Appl. 9 (2002) 223–238.
[26] D. Sidilkover, U. Ascher, A multigrid solver for the steady state Navier–Stokes equations using the pressure-poisson formulation, Comput. Appl. Math. (SBMAC) 14 (1995) 21–35.
[27] U. Trottenberg, C. Oosterlee, A. Schuller, Multigrid, Academic Press, New York, 2001.
[28] P. Vanek, J. Mandel, M. Brezina, Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems, Technical Report UCD-CCM-036, 1995, p. 1.
[29] J.W.L. Wan, Interface preserving coarsening for elliptic problems with highly discontinuous coefficients, Numer. Linear Alg. Appl. 7 (2000) 727–741.
[30] Z.J. Wang, A.J. Przekwasb, Y. Liuc, A fv-td electromagnetic solver using adaptive cartesian grids, Comput. Phys. Commun. 148 (2002) 17–29.
[31] S.H. Ward, G.W. Hohmann, Electromagnetic theory for geophysical applications, Electromag. Meth. Appl. Geophys. 1 (1988) 131–311, Soc. Expl. Geophys.
[32] K.S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, IEEE Trans. Antennas Propag. 14 (1966) 302–307.