Journal of Computational Physics 313 (2016) 532–548

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Through-the-wall radar detection analysis via numerical modeling of Maxwell's equations ☆

Matthew Charnley [a,*], Aihua Wood [b]

[a] Department of Mathematics, Rutgers University, 110 Frelinghuysen Road, Piscataway, NJ 08854, United States
[b] Department of Mathematics and Statistics, Air Force Institute of Technology, Wright-Patterson AFB, OH 45433, United States

ARTICLE INFO

Article history:
Received 19 October 2015
Received in revised form 22 December 2015
Accepted 15 January 2016
Available online 23 February 2016

Keywords:
FDTD
Through-the-wall imaging
Inverse problems

ABSTRACT

The problem of through-the-wall imaging is considered. A numerical method for Maxwell's Equations is developed and implemented with the goal of generating an approximate solution to this problem. The forward problem is solved using the Yee Scheme, and this solver is used in the inverse problem of detecting and analyzing objects inside a room, with no direct vision of the inside. It is shown how different sizes and shapes of objects have different responses to source waves, and these differences can be used to approximate the object. Numerical results show that this reconstruction procedure gives an accurate approximation to the boundary of the object.

1. Introduction

Inverse problems have many applications to physics and engineering problems, and thus have always been a highly researched area in mathematics. In various types of these problems, one can analyze how to reconstruct material properties from the response to a given input, or try to determine under what conditions there is a unique set of material properties that define a given input response. One particular instance of an inverse problem is through-the-wall imaging. In this case, there is a transmitter and receiver set-up outside of a walled room, with no direct vision of the inside. Using electromagnetic waves, one wants to determine if there is an object inside the room, where the object is located, and potentially what it is.

In most applications of through-the-wall imaging, researchers are interested in human detection and analysis. Many sources have looked at using Doppler-type radar effects to study human motion, either using standard Doppler radar [11] or using noise waveforms, which can have the same levels of resolution, with a smaller chance of being noticed by an external party [7]. There has also been work using so-called micro-Doppler, to analyze smaller scale movements with the through-the-wall method, from arm movement (isolated from torso movement) [6] to the small bodily fluctuations associated with breathing and heart beats [2]. Similar analysis has also been used to analyze human gait, and distinguish between human and animal gait patterns [9,10]. More similar to this paper, Wang et al. used a finite difference time domain method to analyze human movement in a room, but used an incident plane wave for analysis, as opposed to the circular source used here [13].

Some of the most recent work in through-the-wall imaging was performed in the master's thesis [4]. The author there performed a series of experiments involving through-the-wall imaging. With the use of a Support Vector Machine, it was

☆ This research was supported by Oak Ridge Institute for Science and Education, and the second author was supported by AFOSR Grant number: F4FGA04296J003.

* Corresponding author.
E-mail addresses: charnley@math.rutgers.edu (M. Charnley), Aihua.Wood@afit.edu (A. Wood).

http://dx.doi.org/10.1016/j.jcp.2016.01.039
0021-9991/

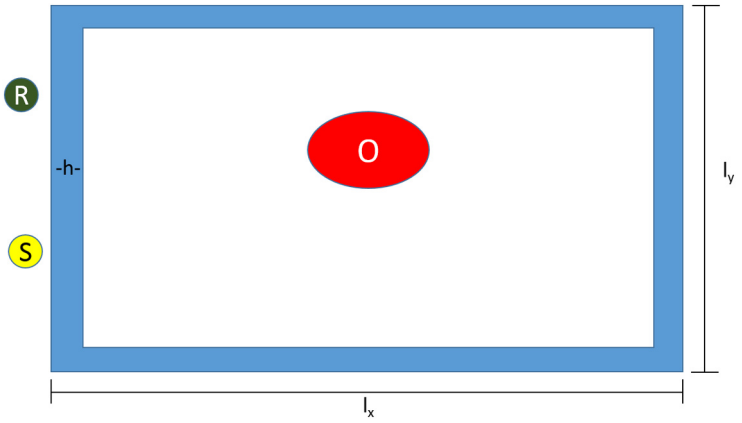


Fig. 1. Model sketch of the problem.

shown that the response data generated from a through-the-wall situation can be used to determine the presence of an object in the room, as well as identify some properties of the object. This last result gives hope that through-the-wall imaging is a well-defined problem, and that numerical and mathematical analysis of this problem could lead to fruitful results.

To that end, in this paper, we develop a numerical method to analyze the through-the-wall imaging problem. We outline the physical system we are trying to analyze in Section 2. In Section 3, we discuss the basic numerical methods involved in approximating Maxwell's Equations in this particular situation. In Section 4, we discuss the adjustments that need to be made to the basic methods in order to accurately approximate the given problem. The facts that material properties are discontinuous and the scattering problem is infinite need to be addressed to ensure that we have the highest order of accuracy possible for our numerical method. In Section 5, we discuss the method for detecting and analyzing the object inside the domain. In Section 6, we show some of the numerical results from these methods, and conclude the paper in Section 7.

2. Model problem

In this paper, we are considering the two-dimensional version of this problem, assuming that all materials are invariant in the z-direction. Our room can be represented by a rectangle of size l_x by l_y. We assume that the walls are of a uniform thickness h, with relative electric permittivity ϵ_w. The object that we are trying to detect is a convex domain $O \subset [0, l_x] \times [0, l_y]$ that has uniform relative electric permittivity ϵ_o. The methods developed here, with a little more work, will function for objects in motion, or objects with varying electric permittivity.

To model the actual physical problem, we will record and use data that could be gathered from a set of antenna transmitters and receivers. Thus, for any given simulation, we will assume that the electromagnetic waves are generated from a source antenna (S) positioned outside of the room and the data is recorded from a (finite) set of receivers (R), also positioned outside the room as shown in Fig. 1.

3. Numerical methods

For this particular version of through-the-wall imaging, we are using electromagnetic radar waves. Therefore, Maxwell's Equations can be used to model this system. Since we are only dealing with electromagnetic waves and have no free charges, the form of Maxwell's Equations we are considering is

$$\begin{cases} \frac{\partial \vec{E}}{\partial t} = \frac{1}{\epsilon} \nabla \times \vec{H} \\ \frac{\partial \vec{H}}{\partial t} = -\frac{1}{\mu} \nabla \times \vec{E} \\ \nabla \cdot \vec{E} = 0 \\ \nabla \cdot \vec{H} = 0 \end{cases}$$

For this problem, we are considering a 2-dimensional problem in the transverse magnetic (TM) mode. In general, most physical situations use transverse electro-magnetic (TEM) waves. Since the TM waves arise naturally from considering this problem, we will study those instead. In order to extend the problem to 3 dimensions and implement it in a physical system, the code will need to be rewritten and implemented for TEM waves. Since we are assuming the problem is invariant in the z direction, only the fields E_z, H_x, and H_y are non-zero. Therefore, we are looking at the 3 equations

$$\begin{cases} \frac{\partial E_z}{\partial t} = \frac{1}{\epsilon}\left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y}\right) \\ \frac{\partial H_x}{\partial t} = -\frac{1}{\mu}\frac{\partial E_z}{\partial y} \\ \frac{\partial H_y}{\partial t} = \frac{1}{\mu}\frac{\partial E_z}{\partial x} \end{cases} \tag{1}$$

These are the equations that will be used in the numerical methods presented in Sections 3.2 and 3.3. However, there is another partial differential equation that the components of Maxwell's Equations satisfy. If we take the curl of the $\frac{\partial \vec{E}}{\partial t}$ equation, we get

$$\frac{\partial(\nabla \times \vec{E})}{\partial t} = \frac{1}{\epsilon}\nabla \times \nabla \times \vec{H}$$

Using the magnetic field equation,

$$\nabla \times \vec{E} = -\mu\frac{\partial \vec{H}}{\partial t}$$

and the vector identity

$$\nabla \times \nabla \times \vec{V} = \nabla(\nabla \cdot \vec{V}) - \Delta\vec{V}$$

we are left with

$$-\mu\frac{\partial^2 \vec{H}}{\partial t^2} = \frac{1}{\epsilon}[\nabla(\nabla \cdot \vec{H}) - \Delta\vec{H}].$$

However, $\nabla \cdot \vec{H} = 0$, so this reduces to

$$\frac{\partial^2 \vec{H}}{\partial t^2} = \frac{1}{\epsilon\mu}\Delta\vec{H} = c^2\Delta\vec{H} \qquad c = \frac{1}{\sqrt{\epsilon\mu}}.$$

Thus, every component of Maxwell's Equations also satisfies a wave equation. These equations will be important when we start to consider the necessary improvements to the model in Section 4.1.

3.1. Finite difference approximations

In order to approximate the equations (1), we will use a finite difference scheme. To get the highest accuracy in our system, we would like to use a centered difference approximation to the derivative, namely, for a function $f(x)$,

$$f'(x) \approx \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x}$$

By using Taylor's Theorem, we can show that, as long as f is smooth enough, this is a good approximation to the derivative. It tells us that there exist points η and ξ with $x - \Delta x \le \eta \le x \le \xi \le x + \Delta x$ satisfying

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

as long as f is at least C^3. Subtracting these two equations and rearranging terms gives the desired result that

$$f'(x) = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} + O(\Delta x^2) \tag{2}$$

For general electromagnetism problems, it would be safe to assume that the component functions of $\vec{H}$ and $\vec{E}$ are C^3. However, as will be discussed in Section 4.2, if material properties are discontinuous, then our function will not be more than C^1. Therefore, we need to modify the situation to retain the second order accuracy of this approximation.

3.2. Yee scheme

To create a system where every derivative in (1) can be approximated by a centered difference, we need to use a staggered grid system. The setup we will use is called the "Yee Scheme" after its creator [14]. Even though this scheme is 50 years old, there have been many improvements to the base scheme discussed in [12] that have been implemented here. The specifics of some of these improvements, along with more recent ones, are discussed in section 4. The scheme

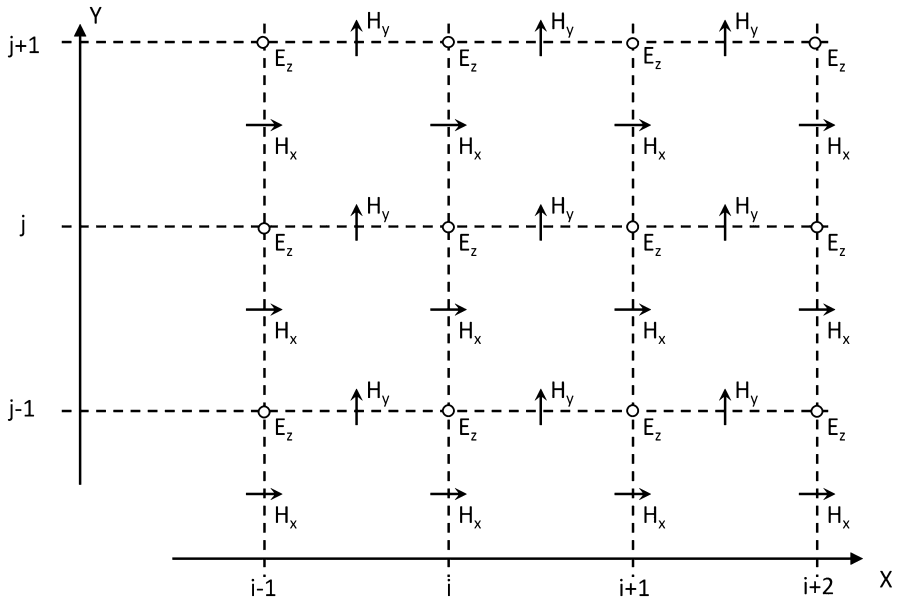


Fig. 2. Diagram of the Yee Scheme [12].

uses a specific staggered grid so that, as desired, every spatial derivative can be computed as a centered difference. However, we also have time derivatives involved in the equations. The setup in the Yee scheme also alternates computing the electric and magnetic fields in half time-steps so that all time derivatives can also be approximated by centered differences.

As can be seen in the Fig. 2, for every point where we calculate E_z, there are points where we calculate H_x directly above and below, and points directly to the left and right where we calculate H_y. Thus, we can approximate $\frac{\partial H_y}{\partial x}$ at the point where we want to calculate E_z by a centered difference. The same goes for $\frac{\partial H_x}{\partial y}$ and all other spatial derivatives in (1). For the time derivatives, we will calculate E_z at integer multiples of Δt, and the magnetic fields at half-integer multiples of Δt. Thus, if we denote the E_z grid by indices (i, j) in the x and y coordinates respectively and k in the t coordinate, we can approximate

$$\frac{E_z^{k+1}(i,j) - E_z^{k}(i,j)}{\Delta t} \approx \frac{\partial E_z}{\partial t}^{k+1/2}(i,j)$$

as a centered difference. Since the H_x and H_y values are given at time value $k + 1/2$, the right hand side of that equation is also specified at the point (i, j) in the x and y-coordinates, and time value $k + 1/2$. Thus, setting these two sides equal makes sense, and gives us the basic scheme for modeling Maxwell's Equations.

3.3. Numerical implementation

Working with the E_z equation in (1), we can replace every derivative by the corresponding centered difference.

$$\frac{E_z^{k+1}(i,j) - E_z^{k}(i,j)}{\Delta t} = \frac{1}{\epsilon}\left[\frac{H_y^{k+1/2}(i+1/2,j) - H_y^{k+1/2}(i-1/2,j)}{\Delta x} - \frac{H_x^{k+1/2}(i,j+1/2) - H_x^{k+1/2}(i,j-1/2)}{\Delta y}\right] \tag{3}$$

For all of the work here, we will assume that we have a square grid, namely, that $\Delta x = \Delta y$. Rearranging 3, we get an expression to update E_z by

$$E_z^{k+1}(i,j) = E_z^{k}(i,j) + \frac{\Delta t}{\epsilon \Delta x}\left[H_y^{k+1/2}(i+1/2,j) - H_y^{k+1/2}(i-1/2,j) - H_x^{k+1/2}(i,j+1/2) + H_x^{k+1/2}(i,j-1/2)\right] \tag{4}$$

and the same process can be carried out for the other two equations. For the implementation of the algorithm, the time-stepping will be implicit, and the array values will update at each time step instead of recording the data from previous

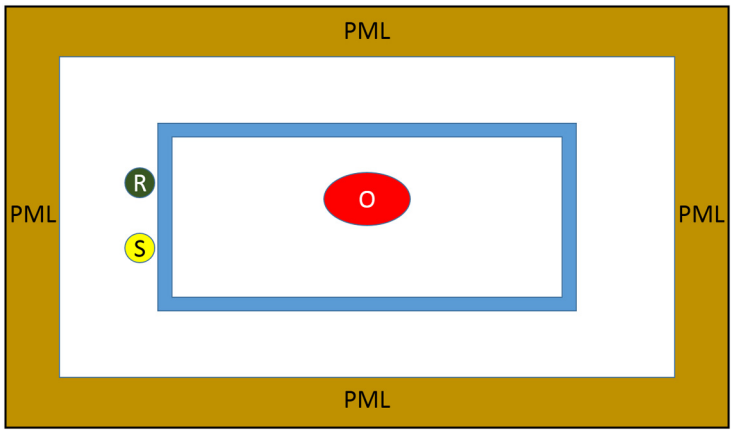


Fig. 3. Model sketch of the Perfectly Matched Layers system and where it is applied.

times. Thus, if we define arrays by

$$\mathbf{E}_z(i,j) = E_z(i,j) \quad \mathbf{H}_x(i,j) = H_x(i,j+1/2) \quad \mathbf{H}_y(i,j) = H_y(i+1/2,j) \tag{5}$$

we get iterative equations for each of these arrays in the form,

$$\mathbf{E}_z(i,j) = \mathbf{E}_z(i,j) + \frac{\Delta t}{\epsilon \Delta x}\left[\mathbf{H}_y(i,j) - \mathbf{H}_y(i-1,j) - \mathbf{H}_x(i,j+1/2) + \mathbf{H}_x(i,j-1)\right]$$
$$\mathbf{H}_x(i,j) = \mathbf{H}_x(i,j) - \frac{\Delta t}{\mu \Delta x}\left[\mathbf{E}_z(i,j) - \mathbf{E}_z(i,j-1)\right]$$
$$\mathbf{H}_y(i,j) = \mathbf{H}_y(i,j) - \frac{\Delta t}{\mu \Delta x}\left[\mathbf{E}_z(i,j) - \mathbf{E}_z(i-1,j)\right]$$

which defines the basic Yee Scheme for modeling Maxwell's Equations.

4. Improvements to the basic algorithm

Now that we have a basic algorithm, we can model the propagation of waves. However, there are a few adjustments that need to be made in order to retain the same level of accuracy for this particular system we are trying to model. These are discoveries that have been developed after the initial definition of the Yee Scheme that help improve its accuracy and ability to model physical situations.

4.1. Perfectly matched layers

The first of these adjustments involves the edges of the computational domain. In general, the scattering problem that is being modeled is infinite; once the waves leave the room, they keep heading off to infinity. Any reflections caused by the boundary of the computational domain would not show up in the physical problem, and so would make the numerical model significantly different from the physical situation. However, the computational domain needs to be truncated in order to carry out the necessary calculations. In order to truncate the computational domain but prevent unnatural reflections, we use the method of Perfectly Matched Layers (PML) [1,5].

In order to formulate the PML, we will use a slightly varied formulation of Maxwell's Equations, where we write the permittivity and permeability as matrices, P_E and P_M so that

$$\begin{cases} P_E \frac{\partial \vec{E}}{\partial t} = \nabla \times \vec{H} \\ P_M \frac{\partial \vec{H}}{\partial t} = -\nabla \times \vec{E} \end{cases}$$

In the case of standard, non-lossy, isotropic material, we have

$$P_E = \epsilon I \qquad P_M = \mu I,$$

so this gives a generalization of the original equations that is easier to understand in the anisotropic case.

The idea of PML is to add a layer of lossy material near the edge of the computational domain, as shown in Fig. 3, so that any waves heading towards the boundary decay away before they get there. Since the waves have decayed, we can then put a hard boundary at the edge of the computational domain ($E_z = 0$), causing a negligible reflection, but the reflection is

significantly smaller than the same system without the PML. However, we also need to make this boundary of this region with the interior reflectionless so as to not significantly change the value of the wave in the internal region. This lossy region does not exist in the physical problem, and so should not affect the numerical solution inside the room.

The portion of a wave incident on an interface that is reflected at the boundary is determined by the impedance of the material

$$\eta = \sqrt{\frac{\mu^*}{\epsilon^*}}.$$

If the impedance of the material on both sides of an interface are equal, there will be no reflection at that interface, and the entire wave will be transmitted. We know the impedance on the plain material side is $\sqrt{\frac{\mu}{\epsilon}}$. In order to make the media reflectionless, we define new permeability and permittivity matrices $\tilde{P}_E$ and $\tilde{P}_M$ with

$$\begin{cases} \tilde{P}_E^* = \begin{bmatrix} \epsilon\epsilon_{F_x}^*(x,y,z) & 0 & 0 \\ 0 & \epsilon\epsilon_{F_y}^*(x,y,z) & 0 \\ 0 & 0 & \epsilon\epsilon_{F_z}^*(x,y,z) \end{bmatrix} \\ \tilde{P}_M^* = \begin{bmatrix} \mu\mu_{F_x}^*(x,y,z) & 0 & 0 \\ 0 & \mu\mu_{F_y}^*(x,y,z) & 0 \\ 0 & 0 & \mu\mu_{F_z}^*(x,y,z) \end{bmatrix} \end{cases} \tag{6}$$

where the new factors ($\epsilon_{F_x}^*$ etc.) are the fake coefficients needed to form the PML. For the reflectionless condition to be satisfied, we need that, for $m = x, y, z$,

$$\epsilon_{F_m}^*(x,y,z) = \mu_{F_m}^*(x,y,z). \quad \text{and} \quad \epsilon_{F_m}^*(x,y,z) = 1 \text{ outside the PML region.} \tag{7}$$

From this construction, we also see that the $\tilde{P}_E$ and $\tilde{P}_M$ matrices coincide with the free space ϵ and μ coefficients outside the PML region. In this particular situation, the PML equations for the transverse magnetic 2-dimensional problem reduce to

$$\begin{cases} i\omega\epsilon\epsilon_{F_z}^*(x,y)E_z = \frac{\partial H_x}{\partial y} - \frac{\partial H_y}{\partial x} \\ i\omega\mu\mu_{F_x}^*(x,y)H_x = -\frac{\partial E_z}{\partial y} \\ i\omega\mu\mu_{F_y}^*(x,y)H_y = \frac{\partial E_z}{\partial x} \end{cases} \tag{8}$$

Making some simplifying assumptions as outlined in [12] and following the implementation of the PML as a lossy material, these equations become

$$\begin{cases} i\omega\epsilon\left(1+\frac{\sigma(x)}{i\omega\epsilon}\right)\left(1+\frac{\sigma(y)}{i\omega\epsilon}\right)E_z = \frac{\partial H_x}{\partial y} - \frac{\partial H_y}{\partial x} \\ i\omega\mu\left(1+\frac{\sigma(x)}{i\omega\epsilon}\right)^{-1}\left(1+\frac{\sigma(y)}{i\omega\epsilon}\right)H_x = -\frac{\partial E_z}{\partial y} \\ i\omega\mu\left(1+\frac{\sigma(x)}{i\omega\epsilon}\right)\left(1+\frac{\sigma(y)}{i\omega\epsilon}\right)^{-1}H_y = \frac{\partial E_z}{\partial x} \end{cases} \tag{9}$$

The H_x and H_y equations are symmetric. Starting with the H_x equation, we can move the inverted term over to the other side, giving

$$i\omega\mu\left(1+\frac{\sigma(y)}{i\omega\epsilon}\right)H_x = -\frac{\partial E_z}{\partial y} - \frac{\sigma(x)}{i\omega\epsilon}\frac{\partial E_z}{\partial y}.$$

Distributing everything, we get

$$i\omega\mu H_x + \frac{\sigma(y)\mu}{\epsilon}H_x = -\frac{\partial E_z}{\partial y} - \frac{\sigma(x)}{i\omega\epsilon}\frac{\partial E_z}{\partial y}.$$

The first term is a time derivative once brought back to the time domain, the next two are standard terms, and the last one represents an integral. Thus, the time domain version of this equation is

$$\mu\frac{\partial H_x}{\partial t} + \frac{\mu\sigma(y)}{\epsilon}H_x = -\frac{\partial E_z}{\partial y} - \frac{\sigma(x)}{\epsilon}\int_0^t \frac{\partial E_z}{\partial y}(s)\,ds$$

When we try to discretize this equation, two terms cause issues. The last term, the integral, is going to need to be approximated, but we can use the earlier values of E_z (in particular $\frac{\partial E_z}{\partial y}$) to continually update this integral as a Riemann Sum. The other problematic term is the H_x on the left hand side. Every other term in this equation is evaluated at the kth time step, but H_x is evaluated at $(k+1/2)$ time steps. This allowed the easy set up of the Yee Scheme and the staggered evaluation of the magnetic and electric fields. In order to get a centered difference approximation, we need the value of H_x at the kth time step. We can approximate this value by the average of the values of H_x at the $(k+1/2)$ and $(k-1/2)$ time steps. Carrying out the discretization of this equation and using this approximation for the H_x term, we get

$$\begin{aligned}\frac{\mu}{\Delta t}(H_x^{k+1/2}(i,j+1/2)-H_x^{k-1/2}(i,j+1/2))+\frac{\sigma(y)\mu}{2\epsilon}(H_x^{k+1/2}(i,j+1/2)+H_x^{k-1/2}(i,j+1/2))\\ =-\frac{\Delta t}{\mu\Delta x}(E_z^k(i,j+1)-E_z^k(i,j)+\frac{\sigma(x)\Delta t}{\epsilon}I_{curlx}(k))\end{aligned}$$

with

$$I_{curlx}(k)=I_{curlx}(k-1)+E_z^k(i,j+1)-E_z^k(i,j).$$

The other two equations can be handled similarly to get discretized equations. The E_z one is a slightly different, because the integral shows up in a different term, but the techniques are the same. This process gives us modified equation that we can implement in the computer for modeling this problem.

$$\begin{cases} I^k_{curlx}(i,j) & = I^{k-1}_{curlx}(i,j)+E^k_z(i,j+1)-E^k_z(i,j)\\ I^k_{curly}(i,j) & = I^{k-1}_{curly}(i,j)+E^k_z(i+1,j)-E^k_z(i,j)\\ I^k_{Ez}(i,j) & = I^{k-1}_{Ez}(i,j)+E^k_z(i,j)\\ H_x^{k+1/2}(i,j+1/2) & = \left(\frac{1-\frac{\sigma(y)\Delta t}{2\epsilon}}{1+\frac{\sigma(y)\Delta t}{2\epsilon}}\right)H_x^{k-1/2}(i,j+1/2)-\\ & \quad \left(\frac{1}{1+\frac{\sigma(y)\Delta t}{2\epsilon}}\right)\left[\frac{\Delta t}{\mu\Delta x}(E_z^k(i,j+1)-E_z^k(i,j)+\frac{\sigma(x)\Delta t}{\epsilon}I^k_{curlx}(i,j))\right]\\ H_y^{k+1/2}(i+1/2,j) & = \left(\frac{1-\frac{\sigma(x)\Delta t}{2\epsilon}}{1+\frac{\sigma(x)\Delta t}{2\epsilon}}\right)H_y^{k-1/2}(i+1/2,j)+\\ & \quad \left(\frac{1}{1+\frac{\sigma(x)\Delta t}{2\epsilon}}\right)\left[\frac{\Delta t}{\mu\Delta y}(E_z^k(i+1,j)-E_z^k(i,j)+\frac{\sigma(y)\Delta t}{\epsilon}I^k_{curly}(i,j))\right]\\ E_z^{k+1}(i,j) & = \frac{1}{1+\frac{\sigma(x)\Delta t}{2\epsilon}+\frac{\sigma(y)\Delta t}{2\epsilon}+\frac{\sigma(x)\sigma(y)(\Delta t)^2}{\epsilon^2}}\Bigg[\left[1-\frac{\sigma(x)\Delta t}{2\epsilon}-\frac{\sigma(y)\Delta t}{2\epsilon}\right]E_z^k(i,j)\\ & \quad -\frac{\sigma(x)\sigma(y)(\Delta t)^2}{\epsilon^2}I^k_{Ez}(i,j)+\frac{\Delta t}{\epsilon\Delta x}\Big(H_y^{k+1/2}(i+1/2,j)-H_y^{k+1/2}(i-1/2,j))\\ & \quad -\frac{\Delta t}{\epsilon\Delta y}(H_x^{k+1/2}(i,j+1/2)-H_x^{k+1/2}(i,j-1/2))\Big)\Bigg]\end{cases} \tag{10}$$

Or, using the array implementation discussed in (5) and noting that we can calculate I_{curlx} and I_{curly} from I_{Ez}, these can be written as

$$\begin{cases} I_{curlx}(i,j) & = I_{Ez}(i,j+1)-I_{Ez}(i,j)\\ I_{curly}(i,j) & = I_{Ez}(i+1,j)-I_{Ez}(i,j)\\ \mathbf{H}_x(i,j) & = \left(\frac{1-\frac{\sigma(y)\Delta t}{2\epsilon}}{1+\frac{\sigma(y)\Delta t}{2\epsilon}}\right)\mathbf{H}_x(i,j)-\left(\frac{1}{1+\frac{\sigma(y)\Delta t}{2\epsilon}}\right)\left[\frac{\Delta t}{\mu\Delta x}(\mathbf{E}_z(i,j+1)-\mathbf{E}_z(i,j)+\frac{\sigma(x)\Delta t}{\epsilon}I_{curlx}(i,j))\right]\\ \mathbf{H}_y(i,j) & = \left(\frac{1-\frac{\sigma(x)\Delta t}{2\epsilon}}{1+\frac{\sigma(x)\Delta t}{2\epsilon}}\right)\mathbf{H}_y(i,j)+\left(\frac{1}{1+\frac{\sigma(x)\Delta t}{2\epsilon}}\right)\left[\frac{\Delta t}{\mu\Delta y}(\mathbf{E}_z(i+1,j)-\mathbf{E}_z(i,j)+\frac{\sigma(y)\Delta t}{\epsilon}I_{curly}(i,j))\right]\\ \mathbf{E}_z(i,j) & = \frac{1}{1+\frac{\sigma(x)\Delta t}{2\epsilon}+\frac{\sigma(y)\Delta t}{2\epsilon}+\frac{\sigma(x)\sigma(y)(\Delta t)^2}{\epsilon^2}}\Bigg[\left[1-\frac{\sigma(x)\Delta t}{2\epsilon}-\frac{\sigma(y)\Delta t}{2\epsilon}\right]\mathbf{E}_z(i,j)-\frac{\sigma(x)\sigma(y)(\Delta t)^2}{\epsilon^2}I_{Ez}(i,j)\\ & \quad +\frac{\Delta t}{\epsilon\Delta x}(\mathbf{H}_y(i,j)-\mathbf{H}_y(i-1,j))-\frac{\Delta t}{\epsilon\Delta y}(\mathbf{H}_x(i,j)-\mathbf{H}_x(i,j))\Bigg]\\ I_{Ez}(i,j) & = I_{Ez}(i,j)+\mathbf{E}_z(i,j)\end{cases} \tag{11}$$

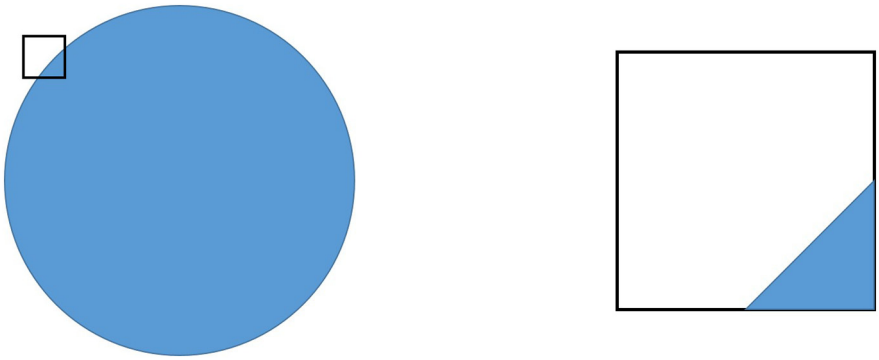


Fig. 4. Diagram of the subpixel smoothing procedure.

Finally, we need to choose what σ is going to be. Following the work in [12], we choose σ so that

$$\bar{\sigma} = \frac{\sigma \Delta t}{2\epsilon_0}$$

is zero in the interior region and has a cubic turn-on rate as it approaches the boundary. Substituting this in for σ in (11) gives us a method to model these equations with PML on the boundary. Numerical results for this formulation can be seen in Section 6.

4.2. Subpixel smoothing

Another issue that needs to be dealt with in simulating this problem is the fact that the material properties we are looking at are discontinuous [3,8]. The fact that the finite difference approximations are "close" to the actual derivatives comes from Taylor's Theorem as in (2). However, if ϵ is not continuous, then the perpendicular component of the electric field at the interface will also be discontinuous, and so we can not apply Taylor's Theorem with remainder to know that the error term is order $(\Delta x)^2$. Therefore, we need a way to make these properties continuous in order to ensure the accuracy of our method. We do this by an averaging method.

Assume we want to find the electric permittivity at some point (x, y). We can calculate the value at exactly that point (it is either in the object, or it is not), but as (x, y) changes, this function will not vary continuously. In order to make a version of the permittivity that is continuous, we construct a square S centered at (x, y) with side length equal to our grid spacing in the Yee Scheme. We then define our smoothed permittivity by the average value of the permittivity on S,

$$\tilde{\epsilon}(x, y) = \frac{1}{|S|} \int_S \epsilon(r, t)\, dr\, dt.$$

If S is entirely contained inside the object, or entirely contained in the free space portion of the room, the value of $\tilde{\epsilon}$ will be the same as $\epsilon(x, y)$, which is what we want. However, if the square spans the boundary of the two regions as shown in Fig. 4, the value of $\tilde{\epsilon}$ will be somewhere between that of the object and that of free space. This function $\tilde{\epsilon}$ is now a continuous function of position, giving us the smoothing that we need to apply Taylor's Formula to the finite difference scheme.

The key assumption in carrying out these calculations is that we are trying to do object detection for things that are orders of magnitude larger than our grid size. Thus, we can approximate the boundary of the object by a straight line cutting through the square. We want to assign the permittivity value at the center of the square to be the average of the permittivity over the entire square. Since we are dealing with uniform space, and a uniform object, this just amounts to finding the area of the region of the square contained in the object. A simple geometric calculation gives a fairly straightforward way to do this in most cases, assuming that we know what the normal vector to the object is at each point. There are a few cases where this method fails, but that can only occur when one region takes up a very small corner of the square. In this case, we just set the value of the permittivity to that of the center of the square, ignoring the small error this causes.

5. Object reconstruction

In the previous sections, we have been discussing the forward problem; that is, knowing where the objects and sources are, and calculate the value of E_z at every point on the grid. The more interesting problem, specifically for through-the-wall imaging, is the inverse problem. For this case, we assume that we know the size and shape of the room, but nothing about the object inside. From the scattering data, we want to locate and characterize the object in the room. To make this simulation more applicable to a physical scenario, we will also assume that we can only gather scattering data at a finite number of points on the outside of the room: a finite number of receivers placed in appropriate locations.

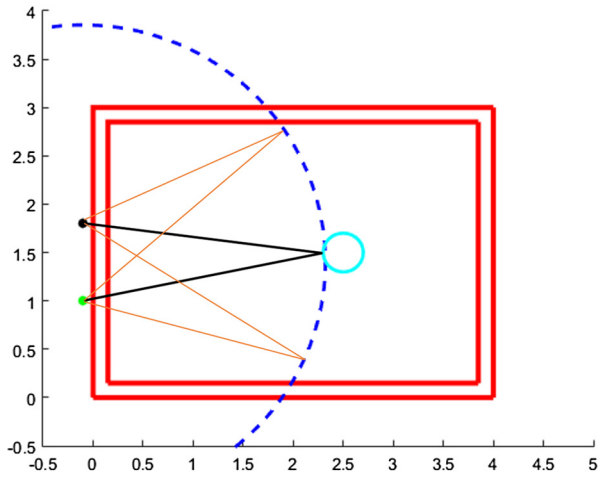


Fig. 5. Illustration of the ellipse procedure.

5.1. Object location

Since we know the shape and size of the room, we can use our calculations from the forward problem to calculate what the wave propagation would look like in an empty room. The only difference between the empty room and the data gathered by the receivers is the presence of the object. Therefore, the difference between the empty room data and the data on the receivers is due to the object inside the room. Since Maxwell's Equations (and wave equations in general) have finite speed of propagation, we can use the first moment in time that the receiver data differs from the empty room calculations to determine how far the object is away from the source and receiver, assuming that the wave front that hits the object and returns to the receiver travels in straight lines as shown in Fig. 5. The black line shows the actual path the wave traveled, and the thinner orange lines show other possible paths that have the same length.

If we find the first difference was observed at a time T, then the nominal distance the wave traveled is given by cT, where $c = \frac{1}{\sqrt{\epsilon_0 \mu_0}}$ is the speed of light in the room. However, since the wave also passes through a wall with a relative electric permittivity ϵ_w, the speed of light is slower in that region. Therefore, this time needs to be adjusted. Since the speed of light goes as $\frac{1}{\sqrt{\epsilon_r}}$, we know that the speed the wave traveled inside the wall is given by $c_0/\sqrt{\epsilon_w}$. Thus, the total distance traveled by the wave can be calculated as

$$d = t_{out} \cdot c_0 + t_{in} \cdot \frac{c_0}{\sqrt{\epsilon_w}}$$

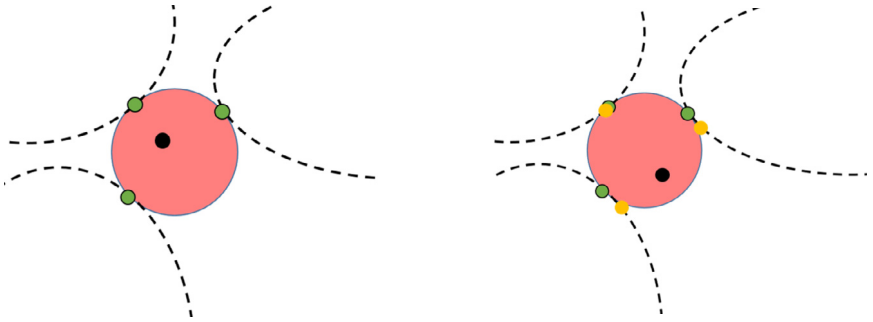
where t_{out} denotes the time spent outside of the wall (in free space), and t_{in} is the time spent within the wall. However, we know $t_{out} + t_{in} = T$, and $t_{in} \approx 2 * h/(c_0/\sqrt{\epsilon_w})$ where h is the thickness of the wall, since the wave passes through the wall twice, on the way out and returning to the receiver. This approximation ignores the fact that the wave is most likely going to pass through the wall at an angle, but since we don't know where the object is, we have no way to adjust for this. Plugging this in and simplifying gives that

$$d = T \cdot c_0 + 2 \cdot h \cdot (1 - \sqrt{\epsilon_w}).$$

This distance d lets us construct an ellipse with foci at the source and receiver to which the object must be externally tangent. Creating these ellipses for each pair of source/receiver data gives an approximate outline as to where the object is. However, the accuracy of our reconstruction is limited by the range resolution of our radar scheme. In this case, standard calculations give that the resolution is around 5 cm, since we use a 600 MHz wave with a pulse length of 2 cycles. Depending on the physical situation being modeled, this error may be insignificant, particularly if we are looking to identify objects on a meter scale. From the data at one receiver, it is not possible to determine where on the ellipse the wave bounced back from, but by combining the data from multiple ellipses, we can get an idea of where the object is, as we will see in Section 6.

5.2. Object characterization

Once these ellipses have been drawn, it is possible to get a rough idea of the shape and size of the object. To do this, we look for the 'center' of the object, and try to find the point on each ellipse that is closest to that center. In general, this could be the center of mass of the object, but we are basically looking for some point contained inside the domain of the object. The basic idea of this process is similar to a trilateration procedure. In the standard method, knowing the



(a) Average of points of tangency is inside O.

(b) Closest points are near points of tangency.

Fig. 6. Illustrations for the iterative method for finding the center of an object. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

distance of each receiver to the object allows circles or spheres to be constructed. The object must be on all of the spheres, so their intersection points give the possible options for the location of the object. In the particular case we have here, the source and receiver are in two distinct places, which means that instead of circles, we generate ellipses. Since the object is significantly larger than the wavelength and we calculated distance to the outside edge of the object, the distance data we obtain does not all refer to the same point. Therefore, we need to carry out this sort of averaging procedure to find an approximation to a single reference point that we can use to pick the desired points on each ellipse. In order to find this point, we use an iterative method.

1. Make a guess for the center point. The initial guess is the center of the room.
2. Calculate the closest point on each ellipse to the current center point guess.
3. Average the x and y coordinates of these closest points to get a new guess for the center point.
4. Repeat this procedure from step 2 until the center points converge to within a desired accuracy.

This method is fairly successful at getting close to the center point. Assume our object O is a convex domain in the room. If, in the ideal case, our selected point on each ellipse is the point of tangency with the boundary of O, our new guess for the center point, defined as the average of the x and y coordinates of these selected points, will be inside of O by convexity. Since the ellipses are also convex, the two domains can only touch in at most one point, so the 'closest' point on the ellipse to this new center will be fairly close to the actual point of tangency. Since the points on the ellipses didn't move much, the new center point can not move very far either, as illustrated in Fig. 6. In this figure, the green points are the points of tangency with the ellipses. In the first, the black point is the average of the darker green points. In the second, the black is a previous guess for the center point, and the lighter yellow points are the closest points on the ellipses to it. As can be seen here, the yellow points are fairly close to the green points, and this holds for any guessed center point within O. Thus, once the center point gets inside the object O, it remains fairly constant.

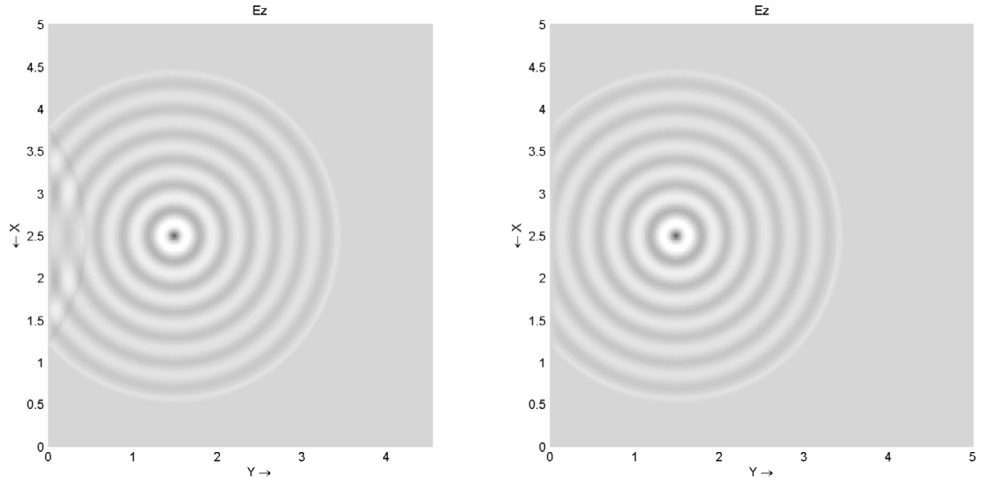
On the other hand, assume we start with a center point far away from O. This point will most likely be contained in several of the ellipses, so they will have 'closest' points in random positions around the curve. However, there are also other ellipses that are very far away from this point, because they made contact with the far side of O, and do not extend to this side. The point on this ellipse may or may not be close to the point of tangency with O, but it lies on the opposite side of O. Therefore, the average will be pulled towards O, assuming that the set of sources and receivers is fairly well spaced around the object.

Thus, the iterative process defined above has the potential to converge to a point inside of O, where the "closest points" on each ellipse are fairly close to the tangency points with O. More interestingly, these "closest points" give a reasonable outline of the object, allowing us to see what the object looks like from this data. Whether or not we find the actual center of the object and how close we get to the boundary of O depends on the set of sources and receivers. For example, if there is no data from the top side of the object, this method will have no reason to move the center point closer to the top side, because there is nothing pulling it that way. This may result in skewed results, but the system works well if there is data coming from all sides.

As shown in the numerical results, this procedure gets us fairly close to the actual center point of the object, as well as give a decent outline of the object, providing an approximation to its shape.

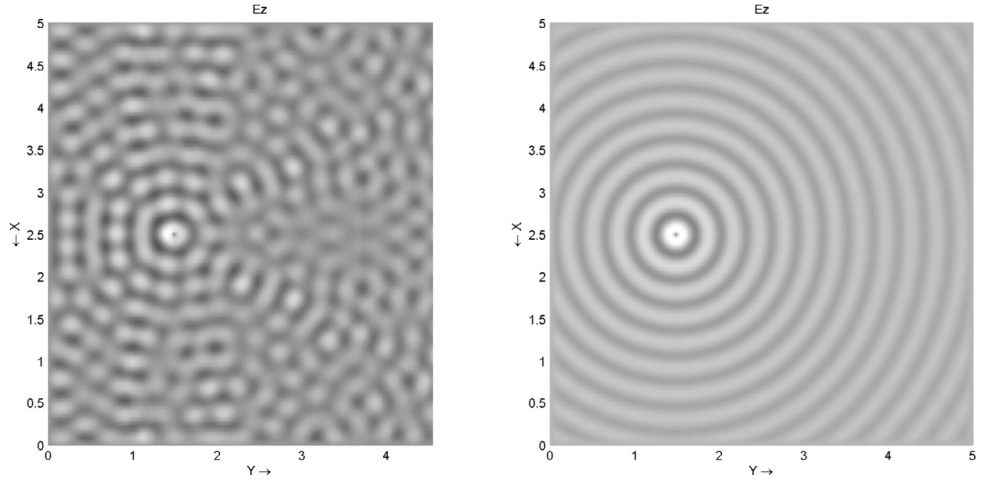
6. Numerical results

For these numerical calculations, we will take our room to be 4 m $\times$ 3 m, surrounded by concrete walls of thickness 0.15 m, with relative permittivity $\epsilon_w = 5$. The object in the room will be of varied sizes and shapes, with $\epsilon_o = 80$. The



(a) Wave propagation, T = $200\Delta t$

(b) Wave propagation with PML, T = $200\Delta t$



(c) Wave propagation, T = $2000\Delta t$

(d) Wave propagation with PML, T = $2000\Delta t$

Fig. 7. Perfectly Matched Layers on the boundary prevent the edges of the computation domain from causing extra reflections.

source will be monochromatic with $\omega = 600$ MHz, and will be a pulse wave, emitting two full cycles of the wave before turning off.

6.1. Forward problem calculations

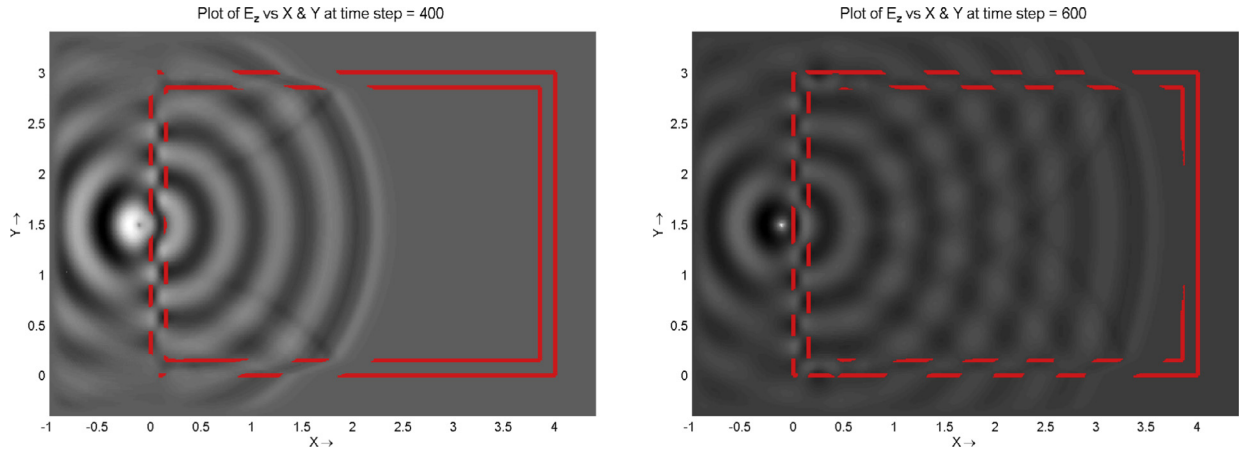
First, we demonstrate the forward problem solver. Fig. 7 shows a toy problem with and without the implementation of the Perfectly Matched Layers scheme derived in Section 4.1. In an ideal case, the waves would be perfectly spherical, because they radiate off to infinity. This is attained in the images with PML, but far from the case when PML is not used.

Figs. 8 and 9 show the forward problem wave propagation once we have set up the walls of the room. Fig. 8 shows a room with nothing in it, while Fig. 9 has a circular object in the room. The walls of the room are the two rectangles outlined in red, and the circular object (if present) is shown with a cyan outline.

6.2. Inverse problem calculations

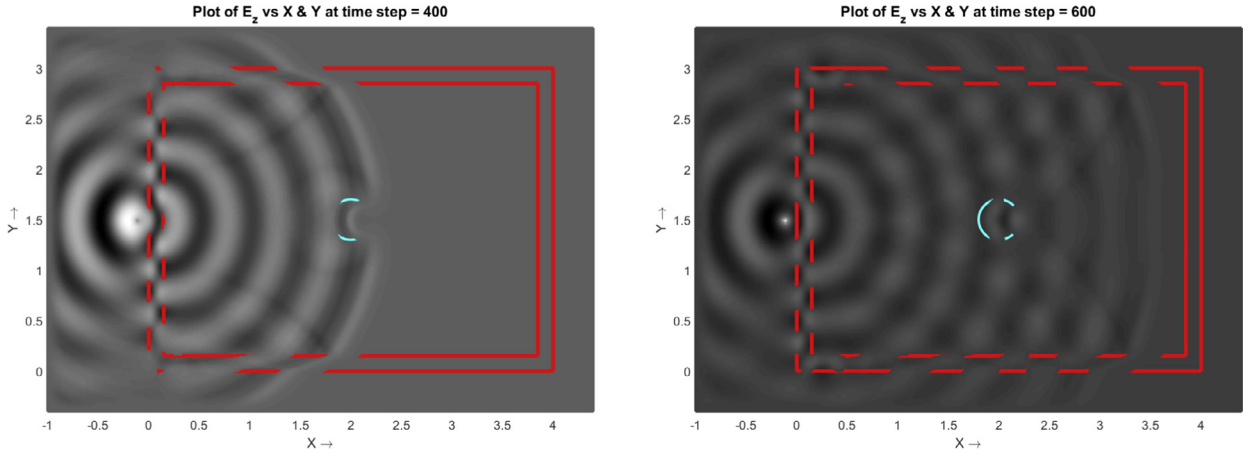
With the forward solver verified, we now look to solve the inverse problem. Using the method from Section 5, Fig. 10 shows the ellipse generated from a single source/receiver pair.

In Fig. 10 and the rest of the figures in this section, the walls of the room are outlined in red and the object, if drawn, is in cyan. Ellipses from the reconstruction method are shown in a dashed blue line. Sources will be drawn in green dots, and receivers will be black. From this single picture alone, we know that the object should be externally tangent to this ellipse. However, at this point, we do not know where on the ellipse the object is. By adding in more sources and receivers, we



(a) Time $= 400\Delta t$ (b) Time $= 600\Delta t$

Fig. 8. Wave propagation in an empty room.



(a) Time $= 400\Delta t$ (b) Time $= 600\Delta t$

Fig. 9. Wave propagation in a room with a circular object. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

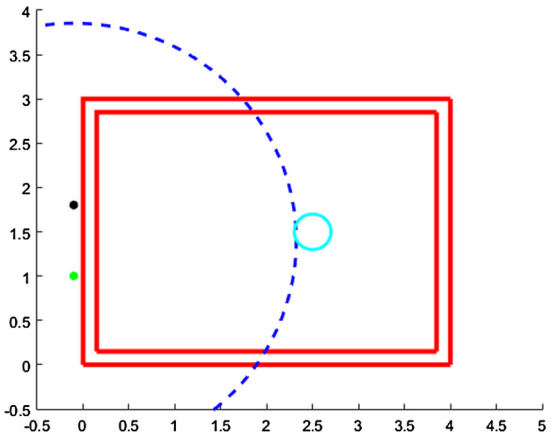
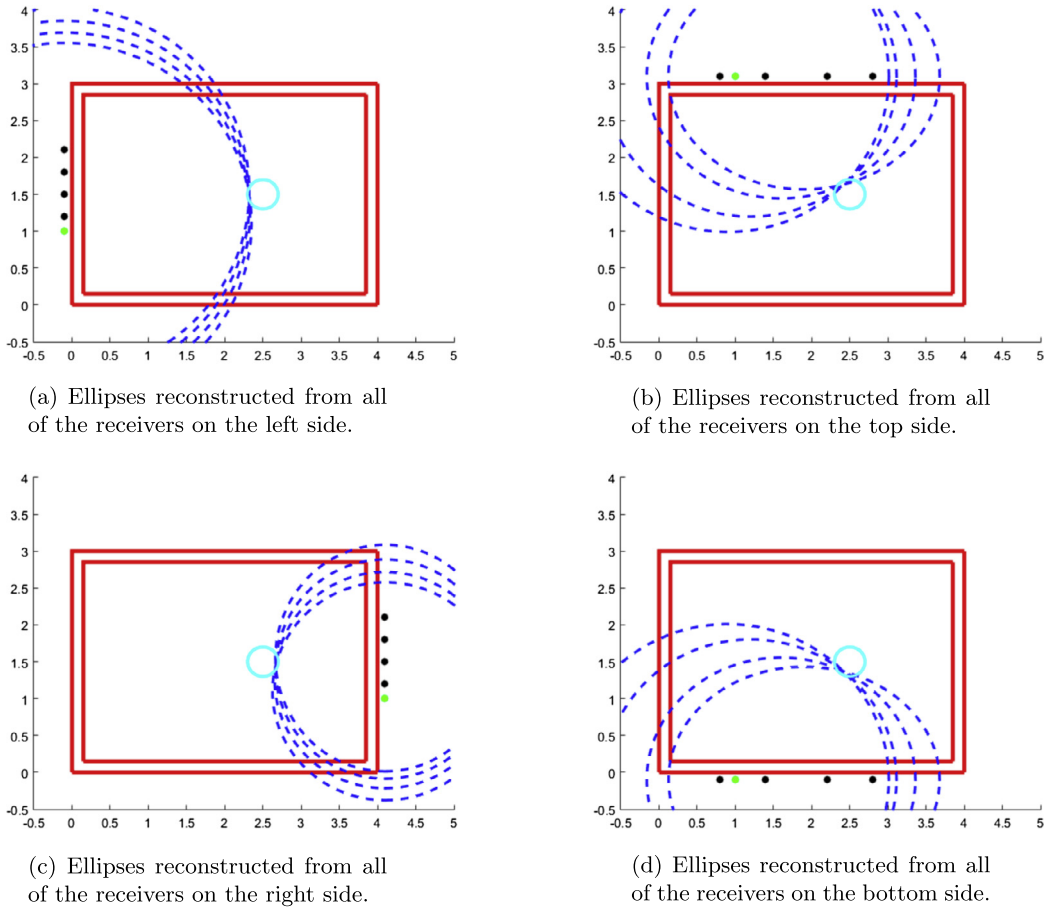


Fig. 10. A single ellipse drawn with the reconstruction algorithm. The ellipse is almost tangent to the object, within a small amount of error.



(a) Ellipses reconstructed from all of the receivers on the left side.

(b) Ellipses reconstructed from all of the receivers on the top side.

(c) Ellipses reconstructed from all of the receivers on the right side.

(d) Ellipses reconstructed from all of the bottom side.

Fig. 11. Object reconstruction algorithm. The ellipses are all close to being externally tangent to the object, giving an idea of where the object has to be.

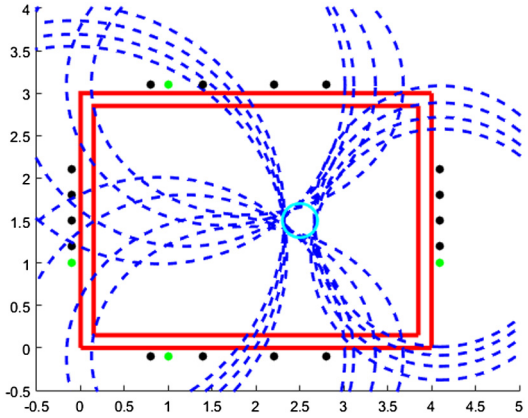
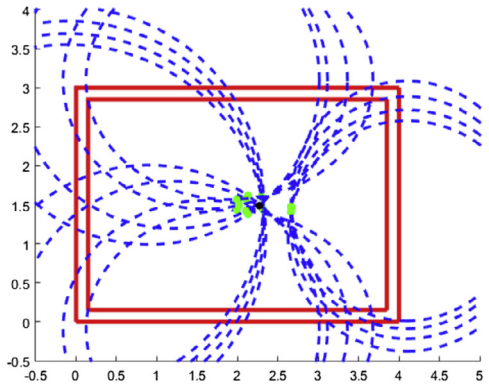


Fig. 12. All of the ellipses from Fig. 11 in one image. The set of ellipses encircles the object, so we have a good idea of where it is.

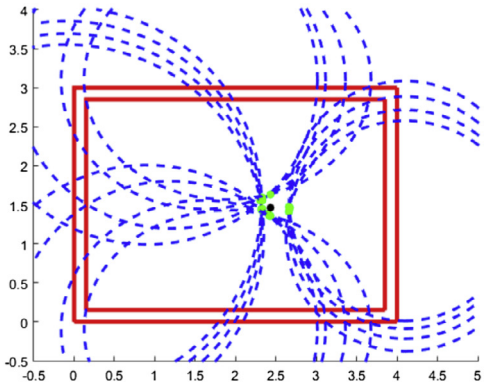
can get a better idea of this. Fig. 11 shows the ellipses constructed from a set of four receivers on each side of the room. Combining all of these together gives an idea of where the object is in the room.

Fig. 12 combines the pictures from Fig. 11 to show the ellipses surrounding the object, telling us almost exactly where the object is.

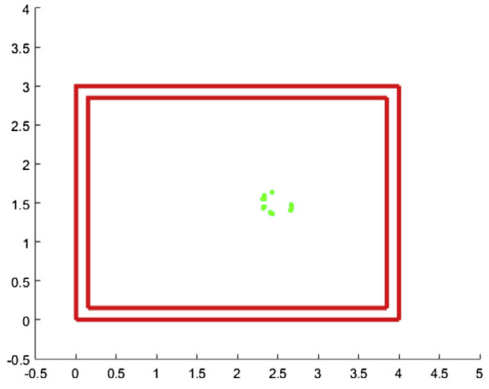
Although it is easy for us to see where the object is given this pictures, it still remains to have the program figure out where it is. Applying the methods in Section 5.2 allow us to do this. Fig. 13 outlines this process, starting from the initial guess in the center of the room and the closest points on each ellipse to it. After the iteration procedure, it can be seen that the “closest points” found by the algorithm line up fairly well with the circular object that was used for the simulation.



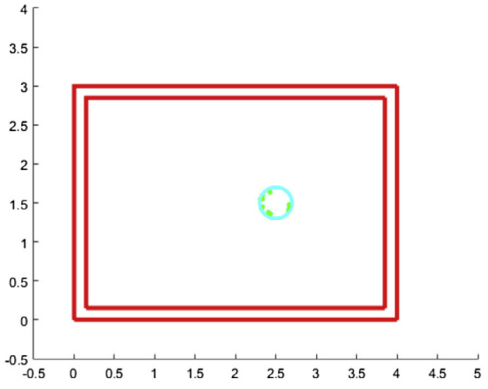
(a) Initial guess as to the center of the object and the closest points on the ellipses.



(b) Final guess for the center of the object and the corresponding closest points.

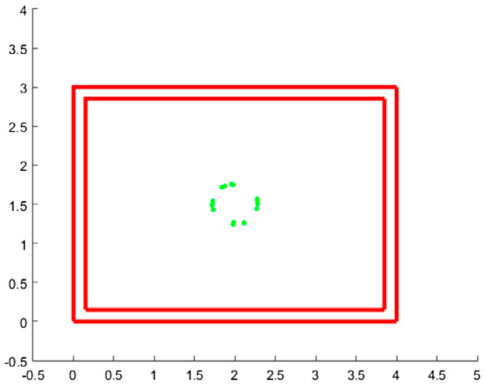


(c) Isolated points that the method finds are on the boundary of the object.

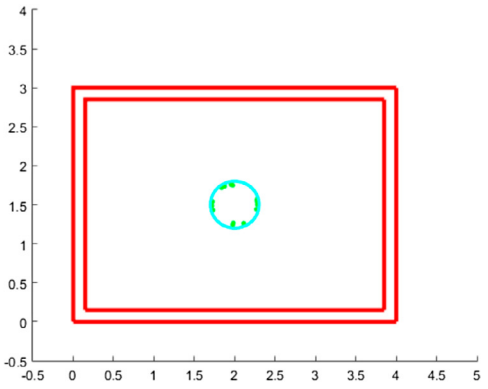


(d) Points from the method along with the object.

Fig. 13. Object reconstruction procedure for a circular object.



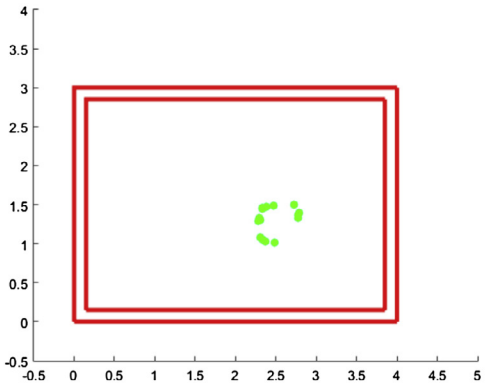
(a) Final guess of points on the boundary of the object.



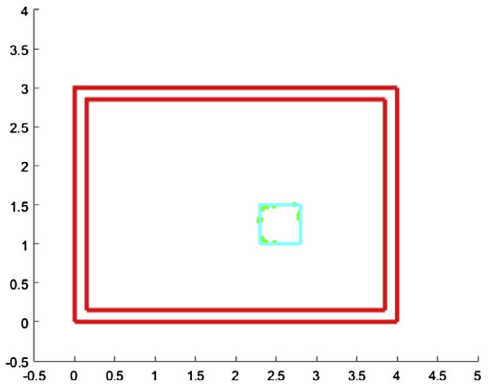
(b) Points from the method and the object.

Fig. 14. Object reconstruction algorithm for a larger, circular object.

Seeing the success of this method, we extend to several different situations. Fig. 14 does the reconstruction procedure on a larger circular object. Fig. 15 does the same process on a square object, and Figs. 16 and 17 use triangular objects for the reconstruction. For these figures, we only show the set of "closest points" found at the end of the iteration procedure and how closely they line up with the actual object being imaged.

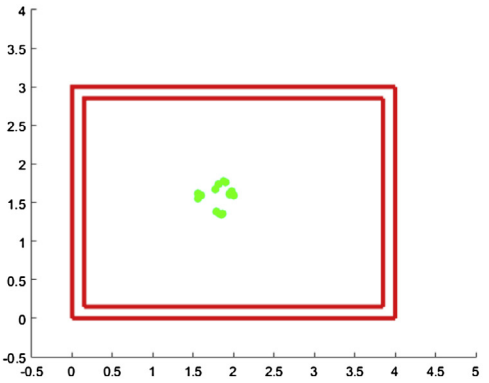


(a) Final guess of points on the boundary of the object.

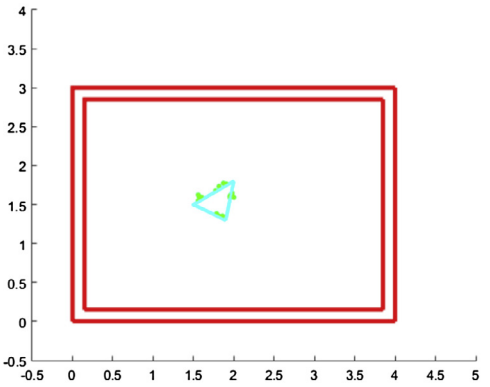


(b) Points from the method and the object.

Fig. 15. Object reconstruction algorithm for a square object.

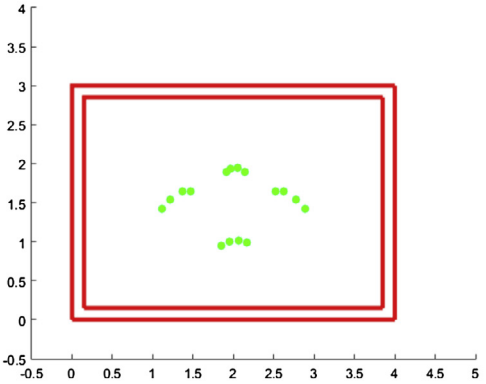


(a) Final guess of points on the boundary of the object.

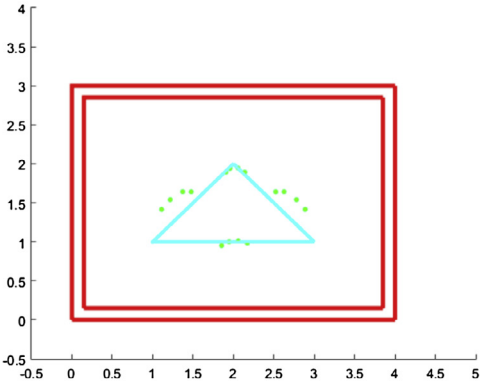


(b) Points from the method and the object.

Fig. 16. Object reconstruction algorithm for a small triangular object.



(a) Final guess of points on the boundary of the object.



(b) Points from the method and the object.

Fig. 17. Object reconstruction algorithm for a large triangular object.

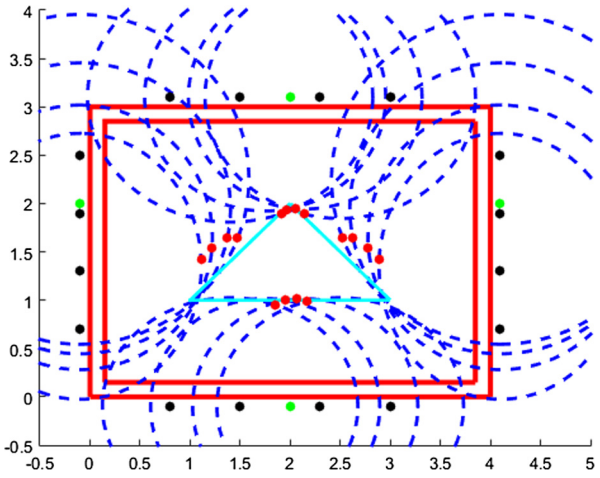


Fig. 18. All ellipses for the large triangle reconstruction, with closest points marked.

As can be seen in these images, the circles and squares are reconstructed fairly accurately, but the triangular region has some issues. This can be attributed to the sharp corners on the triangle and the length of the line segments in the large triangle. Fig. 18 shows the final set of ellipses constructed for the triangle algorithm, along with the "closest points" to the approximated center. As can be seen, the ellipses that are tangent to the triangle at either of the bottom corners have different points, marked in red, that are closer in absolute distance to the center point than the corresponding corner of the triangle. A different method would be needed to find the boundary of the triangle, and the same method could probably be used to improve the reconstruction of the square as well.

7. Conclusion

Inverse problems are an important area of research in physics and engineering, and this paper presents a numerical reconstruction method to approximate a solution to one of these problems. We first construct a numerical algorithm to solve the forward problem of electromagnetic scattering in a 2-dimensional room. Implementing Perfectly Matched Layers and a subpixel smoothing method allows the truncation of the computation domain and the presence of discontinuities in the permittivity to have little to no effect on the accuracy of the numerical simulation. Using this forward problem solver, we construct the inverse problem solution using the difference from collected radar data and the solution to the forward problem in an empty room. In using this method, we have seen that objects of different shapes and sizes show up differently in this method, and the results are fairly consistent with the objects being analyzed. Therefore, we have developed a reliable method of using radar data to discover and analyze objects of several different shapes in a through-the-wall setting for TM waves. In order to extend this to the full 3-dimensional case, this code will need to be modified, both to extend to a 3-dimensional problem, but also to implement TEM waves. For other shapes, a new method will be needed to reconstruct points on the boundary of the object. Of particular interest moving forward would be the problem of characterizing the permittivity of the object from collected radar data. This analysis would allow for a full characterization of a single object in a through-the-wall system.

References

[1] Jean-Pierre Berenger, A perfectly matched layer for the absorption of electromagnetic waves, J. Comput. Phys. 114 (1994) 185–200.
[2] A.S. Bugaev, V.V. Chapursky, S.I. Ivashov, V.V. Razevig, A.P. Sheyko, I.A. Vasilyev, Through wall sensing of human breathing and heart beating by monochromatic radar, in: Proceedings of the Tenth International Conference on Ground Penetrating Radar, 2004, vol. 1, 2004, pp. 291–294.
[3] A. Farjadpour, D. Roundy, A. Rodriguez, M. Ibanescu, P. Bermel, J.D. Joannopoulos, S.G. Johnson, Improving accuracy by subpixel smoothing in the finite-difference time domain, Opt. Lett. 31 (2006).
[4] Stephanie R. Keith, Discrimination between child and adult forms using radar frequency signature analysis, Air Force Institute of Technology, AFIT-ENP-13-M-20, 2013.
[5] Matti Lassas, Erkki Somersalo, On the existence and convergence of the solution of PML equations, Computing 60 (1998) 229–241.
[6] R.M. Narayanan, M.C. Shastry, P.H. Chen, M. Levi, Through-the-wall detection of stationary human targets using Doppler radar, Prog. Electromagn. Res. B 20 (2010) 147–166.
[7] Ram M. Narayanan, Through-wall radar imaging using UWB noise waveforms, J. Franklin Inst. 345 (2008) 659–678.
[8] Ardavan Oskooi, Steven G. Johnson, Accurate FDTD simulation of discontinuous materials by subpixel smoothing, in: Allen Taflove (Ed.), Advances in FDTD Computational Electrodynamics: Photonics and Nanotechnology, Artech House, 2013.
[9] M. Otero, Application of continuous wave radar for human gait recognition, Proc. SPIE 5809 (2005) 538–548.
[10] S.S. Ram, Hao Ling, Microdoppler signature simulation of computer animated human and animal motions, in: Antennas and Propagation Society International Symposium, 2008. AP-S 2008. IEEE, 2008, p. 104.
[11] Shobha Sundar Ram, Yang Li, Adrian Lin, Hao Ling, Doppler-based detection and tracking of humans in indoor environments, J. Franklin Inst. 345 (2008) 679–699.

[12] Dennis M. Sullivan, Electromagnetic Simulation Using the FDTD Method, IEEE, 2000.
[13] Hong Wang, R.M. Narayanan, Zheng Ou Zhou, Through-wall imaging of moving targets using UWB random noise radar, IEEE Antennas Wirel. Propag. Lett. 8 (2009) 802–805.
[14] Kane S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, IEEE Trans. Antennas Propag. 14 (1966) 302–307.