Project Applied Electromagnetism:
Scattering from Dielectric Objects with the FDTD Method

Prof. Dries Vande Ginste, ir. Pieter Decleer, ir. Dries Bosman

October 1, 2019

1 Introduction

The purpose of this project is to numerically simulate the scattering of electromagnetic waves by dielectric objects. To simplify the matter, we will restrict ourselves to two-dimensional (2-D) objects illuminated by transverse magnetic (TM) waves. Although this sounds like a theoretical topic, it has a lot of practical engineering applications. For example, in recent years, many research efforts have been devoted to the detection of breast tumors via the analysis of scattered fields at millimeter-wave frequencies. Similarly, objects buried underground can be detected using so-called ground penetrating radars. In the domain of electrical engineering, adaptations of the technique proposed in this project are employed for the analysis of the Electromagnetic Compatibility behavior of printed circuit boards and other electronic systems.

To model the scattering problem at hand, a well-known numerical technique, i.e.,the Finite-Difference Time-Domain (FDTD) method, will be used. At present, many numerical solvers for Maxwell's equations based on FDTD (in three dimensions) are commercially available, e.g., CST Microwave Studio[1] and the FDTD-solver which is part of Keysight's EMPro[2].

The FDTD method relies on a leapfrog scheme both in space and in time, as further explained in this document. Special attention will also be devoted to the bounding box, the spectrum of the source, the stability of the method, and a simple frequency-domain validation.

2 Basic problem description

The basic geometry under investigation is depicted in Fig. 1. In a free-space background medium (with permittivity ϵ_0 and permeability μ_0), a two-dimensional dielectric object, i.e., invariant along the z-axis, is placed. This object has a relative permittivity ϵ_r. In many practical circumstances, it will be nonmagnetic ($\mu_r = 1$). The object is illuminated by an elementary line source located at (x_s, y_s). Its current density is z-oriented and z-independent, thus

$$\hat{\mathbf{j}}(x, y, t) = \hat{j}_z(x, y, t)\mathbf{u}_z = \hat{J}(t)\delta(x - x_s, y - y_s)\mathbf{u}_z. \tag{1}$$

The time-dependent behavior $\hat{J}(t)$ will be discussed in Section 4. The scattered (or total) fields are observed at one or multiple observation points (x_o, y_o). The simulation domain is delimited by a perfectly electrically conducting (PEC) bounding box, as further discussed in Section 3.5.

For this 2-D situation, it can be shown that Maxwell's curl equations fall apart into two decoupled sets. One set corresponds to z-oriented sources as given in (1), and it is denoted the TM-set, as all magnetic fields are transversal w.r.t. the axis of invariance. The electric field in the TM-set only has a longitudinal component. Hence, in this 2-D TM case, we obtain $\hat{\mathbf{e}}(x, y, t) = \hat{e}_z(x, y, t)\mathbf{u}_z$ and $\hat{\mathbf{h}}(x, y, t) = \hat{h}_x(x, y, t)\mathbf{u}_x + \hat{h}_y(x, y, t)\mathbf{u}_y$. These field components

[1] See www.cst.com

[2] See http://www.keysight.com/en/pc-1297143/empro-3d-em-simulation-software

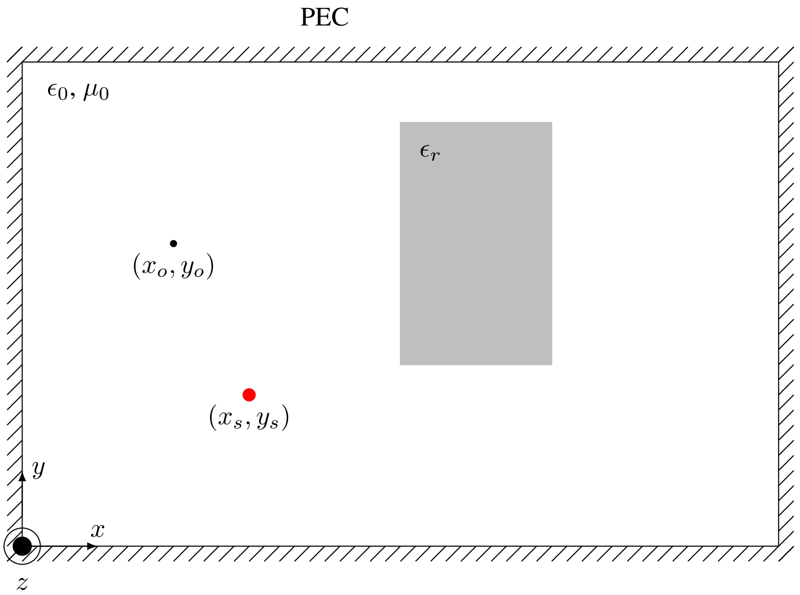


Figure 1: Problem geometry.

are governed by Maxwell's 2-D TM curl equations:

$$\frac{\partial \hat{e}_z}{\partial x} = \mu_0 \frac{\partial \hat{h}_y}{\partial t}, \tag{2}$$

$$\frac{\partial \hat{e}_z}{\partial y} = -\mu_0 \frac{\partial \hat{h}_x}{\partial t}, \tag{3}$$

$$\frac{\partial \hat{h}_y}{\partial x} - \frac{\partial \hat{h}_x}{\partial y} = \epsilon_0 \epsilon_r \frac{\partial \hat{e}_z}{\partial t} + \hat{j}_z, \tag{4}$$

where all arguments were omitted to simplify the notation. Note that $\epsilon_r = 1$ in (4) for locations outside the dielectric object(s) and that $\hat{j}_z$ only differs from zero at the location of the line source. The goal is now to solve Maxwell's 2-D TM curl equations (2)–(4).

3 Finite-Difference Time-Domain (FDTD) method

3.1 Discretization in space and time

The electric and magnetic field components in the simulation domain vary continuously as a function of the space coordinates x and y and as a function of the time t. To numerically compute the fields, the continuous problem is discretized in space and in time:

- The first discretization is performed in space, as shown in Fig. 2. The simulation domain is discretized into a rectilinear grid, with uniform space steps Δ_x and Δ_y along the x- and y-direction, respectively. The electric field component $\hat{e}_z(x, y, t)$ is computed at the $N_x \times N_y$ discrete locations $(x = i\Delta_x, y = j\Delta_y)$, $i = 0, \ldots, N_x - 1$, $j = 0, \ldots, N_y - 1$, indicated with green dots on Fig. 2. The x-oriented magnetic field component $\hat{h}_x(x, y, t)$ is calculated at the $N_x \times (N_y - 1)$ discrete locations $(x = i\Delta_x, y = (j + \frac{1}{2})\Delta_y)$, $i = 0, \ldots, N_x - 1$, $j = 0, \ldots, N_y - 2$, indicated with dark blue arrows on Fig. 2. The y-oriented magnetic field component $\hat{h}_y(x, y, t)$ is calculated at the $(N_x - 1) \times N_y$ discrete locations $(x = (i + \frac{1}{2})\Delta_x, y = j\Delta_y)$, $i = 0, \ldots, N_x - 2$,

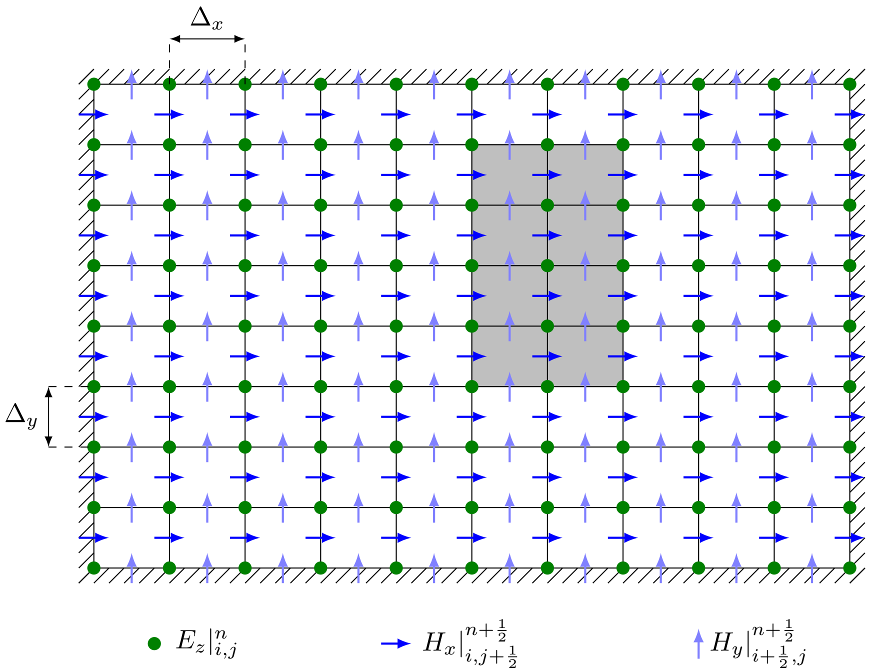

Δ_x
Δ_y
$E_z|_{i,j}^{n}$
$H_x|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}}$
$H_y|_{i+\frac{1}{2},j}^{n+\frac{1}{2}}$

Figure 2: Discretization in space and time of the simulation domain.

$j = 0, \ldots, N_y - 1$, indicated with light blue arrows on Fig. 2.
Note that the discrete locations for the field components are *staggered*.

- Also the time is uniformily discretized with step Δ_t, and this again in a staggered way. The electric field component is evaluated at N_t time instants $t = n\Delta_t$, $n = 0, \ldots, N_t - 1$; both magnetic field components at N_t time instants $t = (n + \frac{1}{2})\Delta_t$, $n = 0, \ldots, N_t - 1$.

The choice for the space steps Δ_x and Δ_y and time step Δ_t will be discussed in Section 3.6.
→ *In any case, in this project, for simplicity, we will always choose the dielectric object's boundary to coincide with the grid lines.*
Now we introduce the following compact notation for the discrete field components for all i, j and n:

$$\hat{e}_z(i\Delta_x, j\Delta_y, n\Delta_t) = E_z|_{i,j}^{n} \ , \tag{5}$$

$$\hat{h}_x(i\Delta_x, (j+\frac{1}{2})\Delta_y, (n+\frac{1}{2})\Delta_t) = H_x|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} \ , \tag{6}$$

$$\hat{h}_y((i+\frac{1}{2})\Delta_x, j\Delta_y, (n+\frac{1}{2})\Delta_t) = H_y|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} \ . \tag{7}$$

3.2 Central finite differences

To derive a discretized version of the first curl equation (2), we calculate the spatial derivative of the electric field at position X between two nodes i and $i+1$, as schematically depicted in Fig. 3. The Taylor expansions for $E_z|_{i,j}^{n}$ and

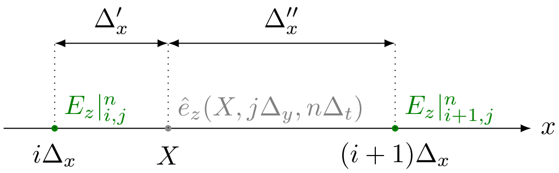


Figure 3: Relevant to the derivation of the central finite difference scheme.

$E_z|_{i+1,j}^n$, centered about $x = X$, are given by

$$E_z|_{i,j}^n = \hat{e}_z(X, j\Delta_y, n\Delta_t) + \left.\frac{\partial \hat{e}_z(x, j\Delta_y, n\Delta_t)}{\partial x}\right|_{x=X} (-\Delta_x') + \frac{1}{2}\left.\frac{\partial^2 \hat{e}_z(x, j\Delta_y, n\Delta_t)}{\partial x^2}\right|_{x=X} (-\Delta_x')^2 + \mathcal{O}\left((\Delta_x')^3\right), \tag{8}$$

$$E_z|_{i+1,j}^n = \hat{e}_z(X, j\Delta_y, n\Delta_t) + \left.\frac{\partial \hat{e}_z(x, j\Delta_y, n\Delta_t)}{\partial x}\right|_{x=X} \Delta_x'' + \frac{1}{2}\left.\frac{\partial^2 \hat{e}_z(x, j\Delta_y, n\Delta_t)}{\partial x^2}\right|_{x=X} (\Delta_x'')^2 + \mathcal{O}\left((\Delta_x'')^3\right), \tag{9}$$

where $\Delta_x' = X - i\Delta_x$ and $\Delta_x'' = (i+1)\Delta_x - X$. For X chosen in the center of the ith segment, and thus $\Delta_x' = \Delta_x'' = \Delta_x/2$, the Taylor expansions read

$$\begin{aligned} E_z|_{i,j}^n = E_z|_{i+\frac{1}{2},j}^n &+ \left.\frac{\partial \hat{e}_z(x, j\Delta_y, n\Delta_t)}{\partial x}\right|_{x=(i+\frac{1}{2})\Delta_x} \left(\frac{-\Delta_x}{2}\right) \\ &+ \frac{1}{2}\left.\frac{\partial^2 \hat{e}_z(x, j\Delta_y, n\Delta_t)}{\partial x^2}\right|_{x=(i+\frac{1}{2})\Delta_x} \left(\frac{-\Delta_x}{2}\right)^2 + \mathcal{O}\left((\Delta_x)^3\right), \end{aligned} \tag{10}$$

$$\begin{aligned} E_z|_{i+1,j}^n = E_z|_{i+\frac{1}{2},j}^n &+ \left.\frac{\partial \hat{e}_z(x, j\Delta_y, n\Delta_t)}{\partial x}\right|_{x=(i+\frac{1}{2})\Delta_x} \left(\frac{\Delta_x}{2}\right) \\ &+ \frac{1}{2}\left.\frac{\partial^2 \hat{e}_z(x, j\Delta_y, n\Delta_t)}{\partial x^2}\right|_{x=(i+\frac{1}{2})\Delta_x} \left(\frac{\Delta_x}{2}\right)^2 + \mathcal{O}\left((\Delta_x)^3\right). \end{aligned} \tag{11}$$

Subtracting (10) from (11) yields

$$E_z|_{i+1,j}^n - E_z|_{i,j}^n = \left.\frac{\partial \hat{e}_z(x, j\Delta_y, n\Delta_t)}{\partial x}\right|_{x=(i+\frac{1}{2})\Delta_x} \Delta_x + \mathcal{O}\left((\Delta_x)^3\right). \tag{12}$$

Division of this result by Δ_x leads to an expression for the sought-after spatial derivative, as follows:

$$\left.\frac{\partial \hat{e}_z(x, j\Delta_y, n\Delta_t)}{\partial x}\right|_{x=(i+\frac{1}{2})\Delta_x} = \frac{E_z|_{i+1,j}^n - E_z|_{i,j}^n}{\Delta_x} + \mathcal{O}\left((\Delta_x)^2\right). \tag{13}$$

It is observed that this derivative, which is approximated by a *central* finite difference, remains *second-order* accurate. Consequently, for small Δ_x, we can replace the l.h.s. of (2) by (13) and maintain very good accuracy, as follows:

$$\frac{E_z|_{i+1,j}^n - E_z|_{i,j}^n}{\Delta_x} = \mu_0 \left.\frac{\partial \hat{h}_y((i+\frac{1}{2})\Delta_x, j\Delta_y, t)}{\partial t}\right|_{t=n\Delta t}. \tag{14}$$

The reader may verify that also applying a central finite difference scheme for the time derivative of the current in the r.h.s. of (14) results in

$$\frac{E_z|_{i+1,j}^n - E_z|_{i,j}^n}{\Delta_x} = \mu_0 \frac{H_y|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - H_y|_{i+\frac{1}{2},j}^{n-\frac{1}{2}}}{\Delta_t}. \tag{15}$$

Similarly, the second curl equation (3) is approximated as

$$\frac{E_z|_{i,j+1}^{n} - E_z|_{i,j}^{n}}{\Delta_y} = -\mu_0 \frac{H_x|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j+\frac{1}{2}}^{n-\frac{1}{2}}}{\Delta_t}. \tag{16}$$

Lastly, we consider the *sourceless* third curl equation (4), that is for all locations (x, y) where $\hat{j}_z(x, y) = 0$. Adopting the central finite difference approach, (4) is cast as

$$\frac{H_y|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - H_y|_{i-\frac{1}{2},j}^{n+\frac{1}{2}}}{\Delta_x} - \frac{H_x|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta_y} = \epsilon_0 \epsilon_r \frac{E_z|_{i,j}^{n+1} - E_z|_{i,j}^{n}}{\Delta_t}. \tag{17}$$

So, from the discrete versions (15), (16) and (17) of the curl equations, it is also observed that the proposed finite difference scheme indeed leads to evaluations of field components at nodes that are staggered in space (integer i and j multiples of Δ_x and Δ_y, respectively, for $\hat{e}_z$; half-integer $i + \frac{1}{2}$ multiples of Δ_x and integer j multiples of Δ_y for $\hat{h}_y$; and integer i multiples of Δ_x and half-integer $j + \frac{1}{2}$ multiples of Δ_y for $\hat{h}_x$) and in time (integer n multiples of Δ_t for $\hat{e}_z$ and half-integer $n + \frac{1}{2}$ multiples of Δ_t for both $\hat{h}_x$ and $\hat{h}_y$).

3.3 Iterative solution via the leapfrog scheme

From (15)–(17) we derive the following update equations:

$$H_y|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} = H_y|_{i+\frac{1}{2},j}^{n-\frac{1}{2}} + \frac{\Delta_t}{\mu_0 \Delta_x} \left(E_z|_{i+1,j}^{n} - E_z|_{i,j}^{n} \right), \tag{18}$$

$$H_x|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} = H_x|_{i,j+\frac{1}{2}}^{n-\frac{1}{2}} - \frac{\Delta_t}{\mu_0 \Delta_y} \left(E_z|_{i,j+1}^{n} - E_z|_{i,j}^{n} \right), \tag{19}$$

$$E_z|_{i,j}^{n+1} = E_z|_{i,j}^{n} + \frac{\Delta_t}{\epsilon_0 \epsilon_r \Delta_x} \left(H_y|_{i+\frac{1}{2},j}^{n+\frac{1}{2}} - H_y|_{i-\frac{1}{2},j}^{n+\frac{1}{2}} \right) - \frac{\Delta_t}{\epsilon_0 \epsilon_r \Delta_y} \left(H_x|_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} - H_x|_{i,j-\frac{1}{2}}^{n+\frac{1}{2}} \right). \tag{20}$$

With these update equations, new field values can be computed from the knowledge of those at a previous time step. They form the core of the so-called *leapfrog scheme*. The basic FDTD algorithm, to be implemented in this project, is thus simply as follows:

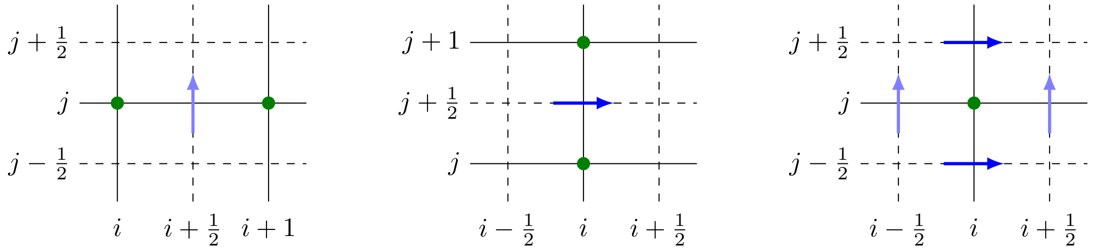
Step 1 Initialize all field components with a zero value.

Step 2 Update the y-component of the magnetic field in all positions at time step $n + \frac{1}{2}$ using (18), viz., by relying on the knowledge of the y-component of the magnetic field at the previous time step $n - \frac{1}{2}$ and on the values of the electric field in two adjacent nodes half a time step earlier, i.e., at time step n. This is sketched in Fig. 4a.

Step 3 Update the x-component of the magnetic field in all positions at time step $n + \frac{1}{2}$ using (19), viz., by relying on the knowledge of the x-component of the magnetic field at the previous time step $n - \frac{1}{2}$ and on the values of the electric field in two adjacent nodes half a time step earlier, i.e., at time step n. This is sketched in Fig. 4b.

Step 4 Update the z-component of the electric field in all positions at time step $n + 1$ using (20), viz., by relying on the knowledge of the z-component of the electric field at the previous time step n and on the values of the magnetic field in four adjacent nodes half a time step earlier, i.e., at time step $n + \frac{1}{2}$. This is sketched in Fig. 4c.

Step 5 Iterate for all n (i.e., repeat Steps 2, 3 and 4).



(a) New H_y updated from old H_y and two old E_z-neighbors.

(b) New H_x updated from old H_x and two old E_z-neighbors.

(c) New E_z updated from old E_z and from old H_x and H_y fields surrounding that E_z field.

Figure 4: Illustration of the update equations (18)–(20).

3.4 Inclusion of sources

When starting with zero fields (see Step 1 of the above algorithm) and iteratively updating according to (18)–(20), all field values would remain zero for all times n. That is because we have not taken the source term into account yet.
To include the source term $\hat{j}_z(x, y, t)$ into our update equations, we revisit the curl equation (4). This equation is obviously the well-known Ampère law, written in its general form as

$$\boldsymbol{\nabla} \times \hat{\mathbf{h}} = \epsilon_0 \epsilon_r \frac{\partial \hat{\mathbf{e}}}{\partial t} + \hat{\mathbf{j}}, \tag{21}$$

which in integral representation becomes

$$\oint_c \hat{\mathbf{h}} \cdot \mathbf{dl} = \epsilon_0 \epsilon_r \iint_S \frac{\partial \hat{\mathbf{e}}}{\partial t} \cdot \mathbf{u}_n \,\mathrm{d}S + \hat{I}, \tag{22}$$

with

$$\hat{I} = \iint_S \hat{\mathbf{j}} \cdot \mathbf{u}_n \,\mathrm{d}S. \tag{23}$$

Consider now the case where the surface S is an elementary cell of the FDTD-grid, i.e., the yellow rectangle ABCD indicated in Fig. 5, with the node (i, j) at its center. From (23), with $\mathbf{u}_n = \mathbf{u}_z$, $\hat{I}(t)$ is the total current flowing through that cell. In case a line source (1) is present in that cell, we simply get $\hat{I} = \hat{J}(t)$ (and zero in case of no source). We now make simple approximations for the integrals in (22): (i) a surface integral is equal to the mean value of the integrand times the surface area; (ii) a line integral is equal to the mean value of the integrand times the length of the interval. This leads to the following approximation for (22):

$$H_y\big|^{n+\frac{1}{2}}_{i+\frac{1}{2},j} \Delta_y - H_x\big|^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} \Delta_x - H_y\big|^{n+\frac{1}{2}}_{i-\frac{1}{2},j} \Delta_y + H_x\big|^{n+\frac{1}{2}}_{i,j-\frac{1}{2}} \Delta_x = \epsilon_0 \epsilon_r \frac{E_z\big|^{n+1}_{i,j} - E_z\big|^{n}_{i,j}}{\Delta_t} \Delta_x \Delta_y + I\big|^{n+\frac{1}{2}}_{i,j}, \tag{24}$$

where a self-explanatory notation for the discrete current was introduced. Note that (24) is consistent with (17). Consequently, the update equation (20) is now readily adapted to include the source term for usage in the leapfrog scheme presented in Section 3.3.
→ *This simple task is left to the reader.*

3.5 Boundary conditions

Obviously, we cannot simulate infinitely large structures or domains using the FDTD method[3]. In this project, the simulation domain is bounded by PEC walls.

[3] At least not without employing some clever techniques that can emulate an infinite domain. Such techniques are described, e.g., in the course "Computational Solutions of Wave Problems".

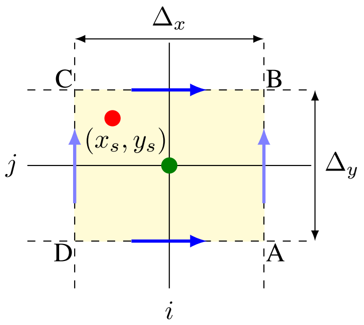


Figure 5: Illustration of the inclusion of a current source in update equation (20).

The main advantage of using PEC walls, is that it leads to a very simple implementation of the corresponding boundary conditions. It is well known that tangential electric fields vanish at PECs. Therefore, in our implementation, we make sure – by proper choice of Δ_x and Δ_y – that the PEC boundaries run through the nodes (i, j) where the z-component of the electric field is evaluated (see also Fig. 2). To implement the PECs, it then suffices to demand that these field values are always zero, i.e.,to hard-code that $E_z|_{i,j}^n = 0$ for all n and for all locations (i, j) on the boundaries[4].
The drawback of this approach is that waves impinging upon the PEC boundaries will get reflected, whereas in reality the fields scattered from a dielectric object would propagate "till infinity" and vanish. Here, the reflected waves will pollute our solutions of the field values in the simulated domain.
$\rightarrow$ *Task: make sure that the simulation and recording of scattered or total fields at well-chosen locations* (x_o, y_o) *stop before a reflected wave arrives!*

3.6 Courant limit

The space steps Δ_x and Δ_y should be small enough to resolve all wave phenomena. In a 2-D (or 3-D) FDTD implementation, as a rule of thumb, one typically chooses the discretization steps within the range[5]

$$\frac{\lambda_{\text{min}}}{30} \leq \Delta_\alpha \leq \frac{\lambda_{\text{min}}}{20}, \quad \alpha = x \text{ or } y, \tag{25}$$

where λ_{min} is the minimal wavelength that is present in the simulation domain. This minimal wavelength stems from the highest frequency that is introduced by the source (see Section 4). One can check that the solution should not change drastically when the grid is refined, i.e., the solution should converge with decreasing spatial step.
Once the space step is selected, an upper limit for the time step Δ_t is given by

$$v\Delta_t \leq \frac{1}{\sqrt{\frac{1}{(\Delta_x)^2} + \frac{1}{(\Delta_y)^2}}}, \tag{26}$$

where v in the l.h.s. of (26) is the maximum occurring phase velocity, which is typically — and also in this project — the free-space speed of light c. This condition limits the maximum allowed time step and it is called the Courant-Friedrichs-Lewy (CFL) condition. It can be shown that a higher time step renders the FDTD simulations unstable and noncausal. The proof follows from an observation of the dispersion relation where the wave propagation in the discretized scheme should not exceed the speed of light.

[4] There exist more elegant methods, which do not require the storage of zeros on the boundary, but these are less straightforward to implement.
[5] Note that, in other computational methods, ten divisions per wavelength often suffice. Here, however, such a coarse discretization would lead to too much grid dispersion.

4 Source current, spectral contect, and frequency-domain analysis

A typical choice for the time-dependent behavior of the current source (1) is as follows:

$$\hat{J}(t) = J_0\, e^{-\frac{(t-t_c)^2}{2\sigma^2}}\,, \tag{27}$$

with J_0 the source amplitude, t_c the central time, and σ the pulse width. This Gaussian pulse has a limited bandwidth, as the magnitude of its spectrum — obtained via the Fourier transform — is given by

$$|J(\omega)| = \left|\int_{-\infty}^{+\infty} \hat{J}(t)e^{-j\omega t}\,\mathrm{d}t\right| = J_0\sqrt{2\pi}\,\sigma\, e^{-\frac{\sigma^2\omega^2}{2}}\,, \tag{28}$$

where ω denotes the angular frequency. So, for $\omega = 3/\sigma$, $|J(\omega)|$ has dropped to a value which is only 1% of its DC value $|J(0)|$. In good approximation, we may assume that the spectral content of the source is limited to this maximum value $\omega_{\max} = 3/\sigma$. Then, the minimum wavelength in the simulation domain is given by $\lambda_{\min} = 2\pi\frac{v_{\min}}{\omega_{\max}}$, where $v_{\min} = c/\sqrt{\epsilon_r}$ is now the slowest wave velocity in the simulation domain, i.e., the velocity in the medium with the highest permittivity. This value of $\lambda_{\min}$ needs to be used to determine the discretizations in space and time, taking (25) and (26) into account. More accurate results can be obtained when choosing, e.g., $\omega_{\max} = 5/\sigma$, ultimately leading to smaller discretization steps. However, this will slow down the simulations.
$\rightarrow$ ***It is left to the students to explore the typical trade-off between accuracy and simulation speed!***[6]
A remark needs to be made on the choice of the central time t_c. The source is only switched on at time $t = 0$, or differently put, $\hat{J}(t) = 0$ for $t < 0$. This can be achieved by multiplying $\hat{J}(t)$ with the Heaviside step function. However, when t_c is too small compared to σ, the above frequency-domain analysis will then no longer be valid, as there will be a jump discontinuity at time $t = 0$, and the source signal will no longer be a symmetrical Gaussian pulse. To avoid this, it is recommended to, e.g., choose $t_c \geq 3\sigma$.
Other choices for $\hat{J}(t)$ can be made as well. To mimick a mobile network's signal, one can multiply (27) by a sine function:

$$\hat{J}(t) = J_0\, e^{-\frac{(t-t_c)^2}{2\sigma^2}}\, \sin(\omega_C t)\,. \tag{29}$$

This is called a Gaussian-modulated sinusoidal radio-frequency (RF) pulse, and this signal has the same bandwidth as (27), but now the frequency band is centered around the central frequency $f_C = \omega_C/(2\pi)$ and $-f_C$.
Often, it is interesting to analyze the scattering behavior (of the dielectric object) as a function of frequency. Thereto, one may be tempted to try the following current profile:

$$\hat{J}(t) = J_0 \sin(\omega_C t)\; H(t), \tag{30}$$

with $H(t)$ the Heaviside step function and where simulations are performed for several values of ω_C. Note, however, that the Fourier transform of (30) is *not* the hoped-for function $\frac{1}{2j}\left(\delta(\omega-\omega_C) - \delta(\omega+\omega_C)\right)$, which you would expect in case of a monocromatic wave. This is of course due to the switching on of the source at time $t = 0$, which introduces a transient behavior. Only after a sufficiently long time, all transients will have died out and we obtain a monochromatic regime situation, with only sinusoidally varying currents and fields. Unfortunately, in many situations, the PEC boundary walls will already have caused reflected signals in the simulation domain before the regime sets in. Instead, to obtain frequency-domain results, it is better to:

1. perform a time-domain simulation using (27) or (29) (depending on the desired frequency band that needs to be investigated);
2. record the fields at one or multiple observation points (x_o, y_o) *(but be careful for reflections from the PEC boundaries!)*;

[6] Efficient programming leads to shorter simulation times. For example, for-loops are generally slow in Python; in contrast, vector and matrix manipulations are fast.

3. perform a Fourier transform of these observed fields. One can simply use the Fast Fourier Transform (FFT) routines provided by Python's `numpy.fft`. FFT is in essence a Discrete Fourier Transform (DFT). Be careful, however, as there are many ways to define the DFT, varying in the sign of the exponent, normalization, etc. So, carefully check the specific implementation of `numpy.fft`;
4. divide the result of step 3 by the spectral content of the source to get the sought-after frequency response[7].

The above procedure immediately provides a simple way of validating the FDTD code. Consider a z-oriented elementary line source in the frequency domain, i.e., with a purely sinusoidally varying current density:

$$j_z(x, y, \omega) = J_0\, \delta(x - x_s, y - y_s), \tag{31}$$

where, as usual, the $e^{j\omega t}$ dependence is omitted. This line source is placed in free space. It is known from electromagnetic theory, that the electric field emitted by this source in free space is given by

$$e_z(x, y, \omega) = -J_0 \frac{\omega \mu_0}{4} H_0^{(2)} \left(k_0 \sqrt{(x - x_s)^2 + (y - y_s)^2} \right), \tag{32}$$

with wavenumber $k_0 = \omega/c$. The function $H_0^{(2)}(Z)$ is the zeroth-order Hankel function of the second kind (and with argument Z). It is also sometimes called a Bessel function of the third kind. For the interested reader, detailed mathematical information is provided on `https://dlmf.nist.gov/10`. In essence, this Hankel function simply represents outgoing 2-D cylindrical waves and it can be implemented in Python via `scipy.special.hankel2(0,z)`. Applying the aforementioned four-step procedure to obtain frequency-domain results in any observation point (x_o, y_o), should — in case of an elementary line source in free space — lead to the analytical result (32).

5 Project description

At least the followings tasks should be performed:

1. Write a program in Python3 [8] that implements the 2-D TM FDTD technique for the scattering at one or multiple dielectric rectangular object(s). Provide a (simple) dialog to feed the program with the necessary data, namely:
   (a) the size of the simulation domain;
   (b) the size and position of the object(s);
   (c) the material parameters (ϵ_r) of the object(s);
   (d) the position (x_s, y_s) of the line source;
   (e) the current profile $\hat{J}(t)$ of the line source;
   (f) the discretization steps Δ_x, Δ_y, and Δ_t;
   (g) the location of one (or multiple) position(s) (x_o, y_o) where the field values are recorded.
2. Validate your code by comparison with the Hankel function, as described in Section 4. *(Can you think of other ways to further validate your code?)*
3. Include one or multiple dielectric block(s) and run your FDTD code for several interesting values of the input parameters. *(Why do the time-domain waveforms look the way they do?)*
4. Investigate the accuracy vs. efficiency trade-off and, in particular, explore the effect of the Courant limit. For time steps that are smaller than required by the Courant limit, the results should remain valid. What happens if Δ_t becomes too large?

[7] When using FFTs, be careful to restrict this response to the bandwith of the source. Otherwise, you will divide by numbers that are (nearly) zero, leading to nonsensical results.

[8] Do NOT use Python2, as it will no longer be maintained past 2020 (see `https://pythonclock.org/`)!

5. Investigate the influence of the PEC boundaries.

6. Write a report that contains (at least) the following information:

   (a) a very short explanation about the used method;

   (b) a clear description of how to use the program;

   (c) numerical results (i.e., output of your program for several interesting choices of parameters), presented as nice and clear graphs;

   (d) a thorough discussion of the obtained results and observations!!

   (e) a brief but clear indication of the contribution of each group member.

   The report should *not* exceed 15 pages.

This assignment is performed in groups of maximum three people. **Deadline 1**: Subscribe to a group of your choice via Ufora before Friday, Nov. 8, 2019. Missing the deadline means losing the free choice of group. **Deadline 2**: Upload your written report (PDF-file) + the actual Python code in a *single* ZIP-file named GroupX.zip, X being your group number, via "Ufora-tools: Assignments" before Monday, Jan. 6, 2020. Missing the deadline means losing marks.

Notes:

- The following (non-standard) Python functions may be useful:
  - `numpy.diff`
  - `numpy.fft.fftfreq`
  - `numpy.fft.fft`
  - `numpy.fft.fftshift`
  - `scipy.special.hankel2`
  - `matplotlib.pyplot.imshow`
  - `matplotlib.pyplot.pcolormesh`
- Make sure that your code is self-contained and that it can be easily tested by us.
- Your report should be a self-contained PDF document.
- Self-efficacy and engineering spirit are encouraged, with extra marks for creative ideas that focus on the electromagnetic aspects of the problem.
- The project (theory + implementation + interpretation of results) can also be a topic on the exam.