The Finite-Difference Frequency-Domain Method

Hans-Dieter Lang, *Student-Member, IEEE*

***Abstract*—The finite-difference frequency-domain (FDFD) method is introduced and implemented in 1D, 2D and 2.5D for both driven mode and eigenmode analysis. Simulation results and execution time performances are presented and compared to those obtained using the well-known FDTD method.**

***Index Terms*—finite-difference, frequency-domain, FDFD, eigenmode analysis, resonator modes**

I. Introduction

FOR most applications, the finite-difference time-domain (FDTD) method is an effective and versatile technique to solve Maxwell's equations. However, for some cases, the frequency-domain comes more natural and is more useful than the time-domain. Of course, results obtained using the latter can be transformed into the frequency domain using the Fourier transform, but getting the right resolution or being able to pick some special frequency might require a prohibitively large amount of time steps or interpolation. Thus, it can be interesting to solve the models at the frequency of interest right from the beginning. Also, there are a few cases where the simulations are difficult to do or cannot be done (accurately enough) using time-domain methods. One example is simulating the resonance frequencies and quality factors of high-Q resonators [11].

The finite-difference frequency-domain (FDFD) method is a simple yet highly capable frequency-domain technique, based on Yee's cell [8], originally developed for the FDTD method. While often Lui and Cheng [3] are credited with introducing the FDFD method, the idea has been around at least a few years longer and could have been introduced or at least analyzed first by Rappaport and McCartin in 1991 [1].

II. The FDFD Method

A. Derivation in 1-D

To derive the FDFD method, one can start either from Maxwell's (curl) equations in the frequency domain [12]

$$\boldsymbol{\nabla} \times \mathbf{E} = -j\omega\mu\mathbf{H} \tag{1a}$$
$$\boldsymbol{\nabla} \times \mathbf{H} = j\omega\varepsilon\mathbf{E} + \mathbf{J} \tag{1b}$$

or from the wave equations in the frequency domain, e.g.,

$$\left(\boldsymbol{\nabla}^2 + k^2\right)\mathbf{E} = j\omega\mu\,\mathbf{J}\,, \tag{2}$$

where, $k^2 = \omega^2\varepsilon\mu$ is the wavenumber (squared). In this section, Maxwell's curl equations will be used to derive the FDFD in 1-D, whereas the wave equation will be used for the derivation of the 2.5-D eigenmode algorithm in the next section.

Hans-Dieter (H. D.) Lang, ID: 1000182599, is a PhD candidate with the electromagnetics group, at the University of Toronto, Ontario, Canada. Email: hd.lang@mail.utoronto.ca. December 21, 2012, v1.1.

For a y-polarized wave propagating in x-direction, the curl equations result in

$$\partial_x E_y = -j\omega\mu H_z \tag{3a}$$
$$\partial_x H_z = -j\omega\varepsilon E_y - J_y \tag{3b}$$

where the short notation $\partial_x = \partial/\partial x$ is used. These equations can then be discretized using finite differences, as known from the FDTD method:

$$\frac{E^y_{i+1} - E^y_i}{\Delta x} = -j\omega\mu H^z_{i+1/2} \tag{4a}$$
$$\frac{H^z_{i+1/2} - H^z_{i-1/2}}{\Delta x} = -j\omega\varepsilon E^y_i - J^y_i \tag{4b}$$

where, similarly to the FDTD cases, the electrical field is located at integer indices and the magnetic field H in between the E_i-field nodes, at half-integer indices. Since we will have to assemble all the fields in a column vector, it makes sense to change the indices to pure integer index numbers, called n_y for the y-directed electric fields and n_z for the z-directed magnetic fields, as shown in Fig. 1. Note that $n_y = 1 \ldots N_x + 1$ and $n_z = 1 \ldots N_x$, where N_x is the total amount of cells in the x-direction.

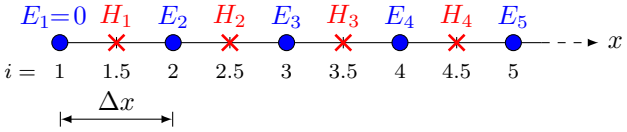


Fig. 1: Discretized space for electric and magnetic fields in 1-D.

As can be seen from Eqs. (4a) and (4b), unlike in the FDTD case[1], these equations cannot be put into an update equation for the electric (magnetic) field depending on the previous values of the electric (magnetic) field and the updated magnetic (electric) fields. Instead, all equations are gathered and put into matrix form[2]:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & \cdots \\ 1/\Delta x & j\omega\mu & -1/\Delta x & 0 & \cdots \\ 0 & 1/\Delta x & j\omega\varepsilon & -1/\Delta x & \\ 0 & 0 & 1/\Delta x & j\omega\mu & \ddots \\ \vdots & \vdots & & \ddots & \ddots \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} E_1 \\ H_1 \\ E_2 \\ H_2 \\ \vdots \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} 0 \\ 0 \\ -J_2 \\ 0 \\ \vdots \end{bmatrix}}_{\mathbf{b}} \tag{5}$$

[1]This is only true for the first-order FDTD case, for higher-order FDTD schemes similar issues arise.

[2]Evidently, (5) is not yet optimized and still contains unnecessary rows in order to show the principle most clearly. Also, it can be more efficient to use a field vector $\mathbf{x} = [\mathbf{E}, \mathbf{H}]^T$ instead of the shown form, with alternating E_i and $H_{i+1/2}$.

where $\mathbf{A}$ is the operator matrix, $\mathbf{x}$ is the unknown field vector and $\mathbf{b}$ is the source vector. This linear system is then solved using either

- Direct inversion

$$\mathbf{Ax} = \mathbf{b} \quad \Rightarrow \quad \mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \tag{6}$$

  (for example using `x=A\b` in Matlab)
- Iterative methods
- Least-square (for badly or ill-conditioned matrixes)

Since the actual method to invert matrices is beyond the scope of this project, we will not consider it any further. Neither will we consider optimizing the operator matrix $\mathbf{A}$ accordingly.

B. PML for FDFD

The derivation of the PML for FDFD is straightforward and similar to the PML for FDTD. The fields are absorbed in a lossy material; the electric fields in a lossy dielectric and the magnetic fields in a lossy magnetic material. Both are achieved by adding a loss term (non-zero conductivity) to the permittivity and permeability, respectively:

$$\begin{matrix} j\omega\varepsilon E^y_{i+1} \\ j\omega\mu H^z_{i+1/2} \end{matrix} \xrightarrow{\text{inside PML}} \begin{matrix} \left(j\omega + \frac{\sigma_{i+1}}{\varepsilon}\right)\varepsilon E^y_{i+1} \\ \left(j\omega + \frac{\sigma_{i+1/2}}{\varepsilon}\right)\mu H^y_{i+1/2} \end{matrix} \tag{7}$$

where σ is the electric conductivity and $\sigma/\varepsilon = \sigma^*/\mu$ (μ^* is the magnetic conductivity) was used (similar to FDTD), to minimize reflections.

To minimize numerical artifacts, the conductivity σ inside the PML region is gradually varied using a polynomial of order p. Similar to the FDTD, the maximum conductivity $\sigma_{\max}$ along with the order of the polynomial p are chosen according to empirical rules. The rules for optimizing PML for FDTD can be applied and result in good overall performances, but other rules [2], [4], [5] have been shown to be even better.

For more dimensional problems anisotropic PML are used, just as in FDTD.

C. Results

1) PML Verification: From the steady-state solution, the reflection coefficient can be easily calculated, for example using

$$|\Gamma| = \frac{\mathrm{VSWR} - 1}{\mathrm{VSWR} + 1} \tag{8}$$

where VWSR is the *voltage standing wave ratio*. Here instead of the voltages the electric fields are used directly. The phase could be found by comparing the destinations of the minima (or zeros) and maxima compared to the distance to the reflective interface. To see if the PML is working, several one-dimensional simulations were made, an example is shown in Fig. 4. It shows the electric and magnetic fields along a transmission line, once with shorted ends and once with the PML absorber in place. Since in that case no reflection occurs, the result is the same as terminating the line with its characteristic impedance. Note that this termination has an impact all along the line, which can be seen by the envelopes (absolute values).

2) Dispersive media: As is well known, modeling dispersive media in FDTD can become quite involved. The frequency-dependences are most commonly modeled using time derivatives of additionally introduced variables, resulting in auxiliary differential equations. The resulting computational complexity is high and instability problems can arise, for some special values for the dispersion models, which might require a finer spatial resolution and therefore even more increase in computational demands.

In the frequency domain the situation is much simpler: since the frequency of interest ω_0 is chosen and put into Maxwell's equations, the same can be done with media parameter model which then results in a new single value for $\varepsilon(\omega)|_{\omega=\omega_0}$ and $\mu(\omega)|_{\omega=\omega_0}$.

Reflection at an interface of air and a well-known Lorentz media with the frequency-dependent permittivity modeled by

$$\varepsilon(\omega) = \varepsilon_\infty + (\varepsilon_s - \varepsilon_\infty)\,\frac{\omega_0^2}{\omega_0^2 + 2j\omega\delta - \omega^2} \tag{9}$$

(using $\varepsilon_\infty = \varepsilon_0$, $\varepsilon_s = 2.25\,\varepsilon_0$, $\delta = 0.28 \times 10^{16}\,\mathrm{s}^{-1}$ and $\omega_0 = 4.0 \times 10^{16}$ rad/sec) was simulated both using FDTD and FDFD and the results are shown in Fig. 2. As can be seen,

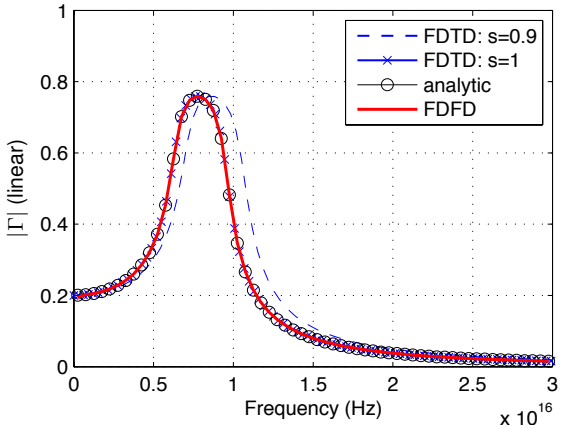
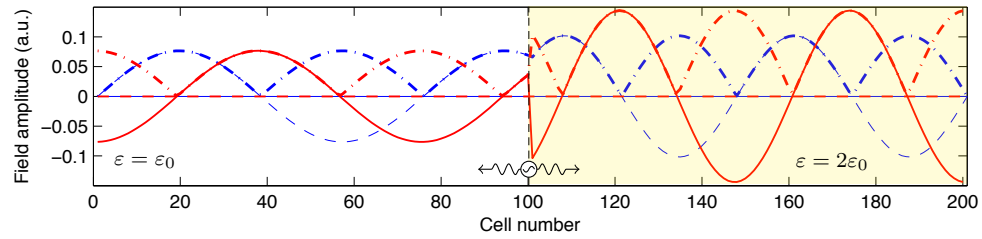


Fig. 2: Reflection coefficient vs. frequency of FDFD (40 values, simulation time $< 3\,\mathrm{s}$), FDTD (8192 values, approx. $10\,\mathrm{s}$) and analytic methods.

the agreement is very good, even though for the FDFD line only 40 frequency steps were simulated (compared to 8192 for the FDTD). Of course, with FDTD being a broadband simulation method, the reflection factor is obtained for a large frequency span (depending on the time discretization and the total number of time steps), whereas with the FDFD only a value for a single frequency results for each simulation. However, when it is looked for the reflection factor at a special frequency, a high number of time steps (or interpolation) might be required for FDTD, whereas for FDFD just one simulation at that frequency has to be carried out.

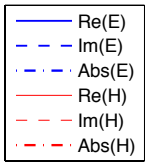
Fig. 3 shows snapshots from both simulation techniques:

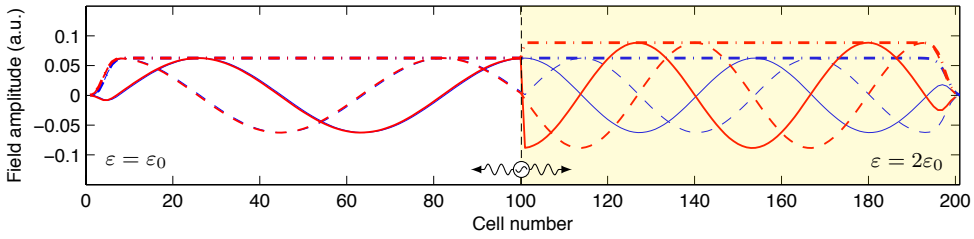
- In the time domain (FDTD, subfigure a) the wave (Gaussian pulse) is launched at the blue dashed line (traveling both to the left and right) and measured at the black dashed line. First the incident wave is measured and then the reflection from the interface (red line) is measured



(a) Transmission line with shorted ends (PEC boundary)

- No PML: shorted TL
- VSWR $\to \infty$
- E and H (V and I) out of phase $\to$ no real power flow

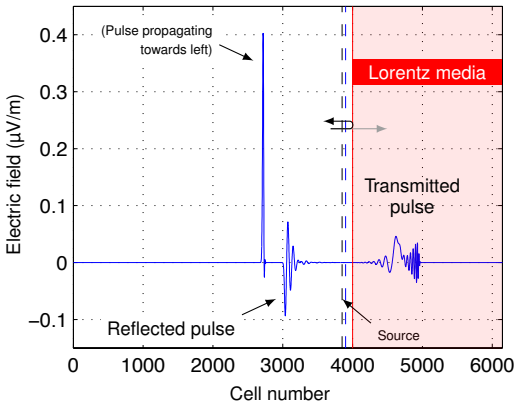




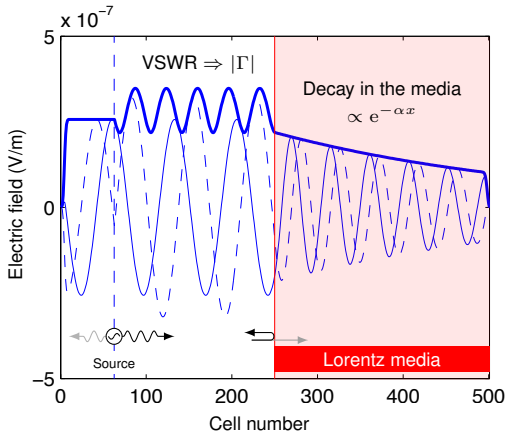
(b) Transmission line with matched ends (PML boundary)

- With PML: matched TL
- VSWR $\to 1$
- E and H (V and I) out in phase $\to$ real power flow towards the ends (loads)

Fig. 4: Evidence of a working PML in a 1-D FDFD transmission line simulation using the standing wave ratio of the envelopes. (Source at center, different permittivities/wave impedances: left $\varepsilon = \varepsilon_0$, $Z = \eta$; right $\varepsilon = 2\varepsilon_0$, $Z = \eta/\sqrt{2}$)



(a) FDTD



(b) FDFD

Fig. 3: Lorentz media modeled in FDTD and FDFD.

and the two are compared after transforming them to the frequency domain.

- Using FDFD (subfigure b) the wave is launched at the blue dashed line again and also travels to the left and right side. Since this is a steady-state simulation we cannot get away with moving the left boundary far away enough, so that no reflection thereof is recorded. Instead, an absorbing boundary (PML) has to be used. The same is true for the right end (inside the media), where the PML just uses the $\varepsilon_{\text{Lorentz}}(\omega)$ at the simulated frequency $\omega = \omega_0$ and no problems as in dispersive PML for FDTD arise. The magnitude of the reflection coefficient can be calculated using the VSWR between the interface and the source. Also, the exponential decay in the Lorentz media, due to the inherit loss is visual.

Evidently both methods lead to the same results here, but the FDFD implementation is certainly much simpler and straightforward than the one in FDTD. As discussed, absorbing boundaries (PML) can be considered more important for the FDTD method, since in the resulting steady-case we cannot get away with moving unwanted problems far enough away from the probe and therefore the domain always has to be terminated by the respective boundary conditions.

III. FDFD Eigenfrequency Analysis

There are two kinds of eigenfrequency or eigenmode analyses that can be carried out:

- eigenfrequency analysis, to find the resonance frequencies ω_n of some structure (e.g. a resonator)
- generalized eigenfrequency analysis (or eigenmode analysis), to find some propagation constant β according at some frequency ω

For both the corresponding eigenvectors (field distributions) can also be found[3]. The first is more general and was used to find resonance modes in resonators.

The 2.5-D FDFD eigenmode analysis algorithm is based on a similar idea to the one of the 2.5-D FDTD method for guided wave structures introduced by Xiao and Vahldieck 1992/93 [9], [10], based on Yee's mesh [8].

A. *FDFD Resonator Eigenfrequency Analysis*

1) Dipole Resonances in 1-D: The eigenvalue problem in one dimension (e.g. eigenfrequency analysis of a simple transmission line resonator) can be obtained by discretizing the one-dimensional wave equation for the electric field in eigenvalue problem formulation:

$$\left(\partial_x^2 + k^2\right) E_y = 0 \quad \xrightarrow{\text{Eigenvalue}} \quad \partial_x^2 E_y = \underbrace{-\omega^2 \varepsilon \mu}_{\lambda} E_y \tag{10}$$

(Note that in this section λ stands for the eigenvalue, not wavelength!). The second-order derivative is approximated using the second-order finite difference

$$\partial_x^2 E_y \approx \frac{E_{i+1}^y - 2E_i^y + E_{i-1}^y}{\Delta x^2} \qquad i = 1 \ldots N_x \,, \tag{11}$$

where $E_0^y = E_{N_x+1}^y = 0$, for now (implicit short or PEC ends). The full[4] eigenvalue problem in matrix form becomes

$$\frac{1}{\Delta x^2} \underbrace{\begin{bmatrix} 2 & -1 & 0 & \ldots \\ -1 & 2 & -1 & \ddots \\ 0 & -1 & 2 & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}}_{\mathbf{A}} \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ \vdots \end{bmatrix} = \lambda \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ \vdots \end{bmatrix} . \tag{12}$$

In this case the matrix $\mathbf{A}$ is not just sparse but a symmetric tridiagonal band matrix. Also, the eigenvalue can be replaced by $\lambda' = \lambda \Delta x^2$, which makes the whole problem independent of the actual discretization; the only remaining dependence is on the actual number of nodes N_x.

For the open ends, instead of the Dirichlet boundary conditions of implicit ends $E_0^x = E_{N_x+1}^x = 0$ we use Neumann boundary conditions, i.e.,

$$\partial_x E^y \approx \frac{1}{\Delta x}(E_1^y - E_0^y) = 0 \quad \Rightarrow \quad E_0^y = E_1^y \tag{13}$$

and similarly for $E_{N_x+1}^y$. This changes the finite difference at these positions from (11) to, e.g.,

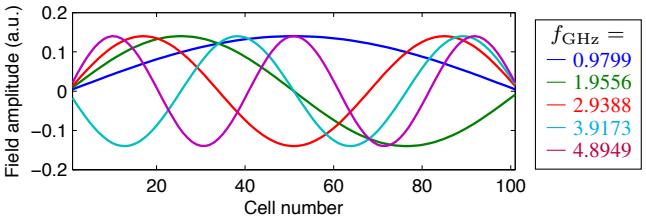
$$\partial_x^2 E_y\big|_{x=0} \approx \left.\frac{E_{i+1}^y - 2E_i^y + E_{i-1}^y}{\Delta x^2}\right|_{i=1} = \frac{E_2^y - E_1^y}{\Delta x^2} \tag{14}$$

and, thus, in the first and last entries of the matrix $\mathbf{A}$ the twos are replaced by ones.

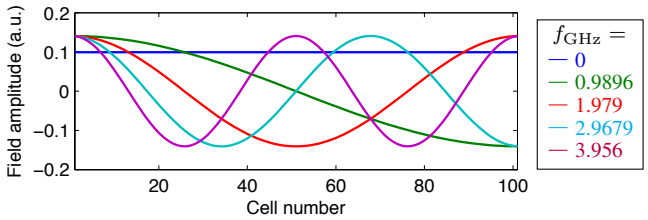
[3]In Matlab the function `eig()` is used to solve the eigenvalue problems. If the function is called using two variables (instead of just one for the eigenvalues), i.e. `[eigenvectors eigenvalues]=eig(A);`, both the eigenvalues and the corresponding eigenvectors (in the modal matrix) can be found using a single command.

[4]Note that, this example the matrix $\mathbf{A}$ is already optimized, meaning implicitly reduced by zero rows and columns since these would be the entries for E_0^x and $E_{N_x+1}^y$, which are not contained in the matrix.

Fig. 5 shows the eigenmodes of a 1-D transmission line resonator of length $l = 150\,\text{mm}$, with shorted (a) and open (b) ends. These results can be interpreted as the current and electric field distributions along a dipole antenna (half-wavelength at around 1 GHz) at various frequencies. Note that, although the computation in this case runs very fast[5], i.e. $t_{\text{sim}} < 0.01\,\text{s}$, because of the nice properties of the matrix, $\mathbf{A}$ is already of dimensions $(N_x + 1) \times (N_x + 1)$.



(a) Shorted ends, similar to current distributions of a dipole antenna.



(b) Open ends, similar to electric field amplitudes along a dipole antenna.

Fig. 5: Dipole resonance frequencies and field distributions.

2) 2-D Cavity Resonator Modes: The 2-D resonator eigenfrequency solver is obtained using the wave equation of the electric fields in two dimensions (for TE modes) and following steps very similar to the ones presented for the one-dimensional case. For simplicity only PEC boundaries (zero tangential electric fields) are considered.

Figs. 6 and 7 show some results obtained using the FDFD eigenmode method in two dimensions. The total nodes of the electric field E_z, thus total number of equations, are $N_{\text{equations}} = (N_x + 1) \times (N_y + 1)$, where N_x and N_y are the number of cells in the respective dimension. The operator matrix $\mathbf{A}$ is then of size $N_{\text{equations}} \times N_{\text{equations}} > (N_x N_y)^2$. Thus, the matrix contains more entries than the square of the total number of cells. While the matrix can be optimized[6] considerably (because of zero rows and columns, due to the boundary conditions), it becomes clear why the FDFD method is known to require considerably more RAM than the FDTD method and why iterative matrix inversion algorithms are often more practical than direct inversion. Since $\mathbf{A}$ is very sparse, especially for very large matrices the RAM consumption can be greatly reduced by storing just the non-zero values, along

[5]Reference System: Intel Core i7 (Dual-Core) 2 GHz, 8 GB 1600 MHz DDR3 RAM, SSD

[6]In matlab this can be realized using:

```
A(~any(A,2),:)=[]; % delete zero rows
A(:,~any(A,1))=[]; % delete zero columns
```

with their positions[7].

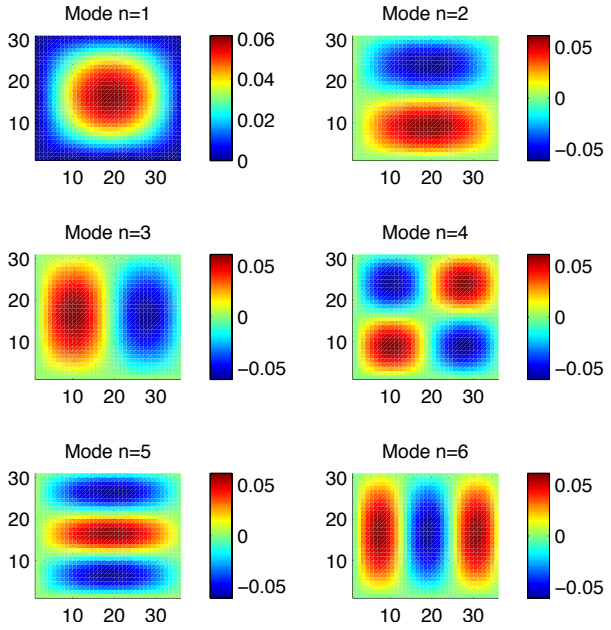


Fig. 6: The first six TE resonator modes of a rectangular cavity (35×30 cells, $\Delta x = \Delta y = 1\,\text{mm}$).

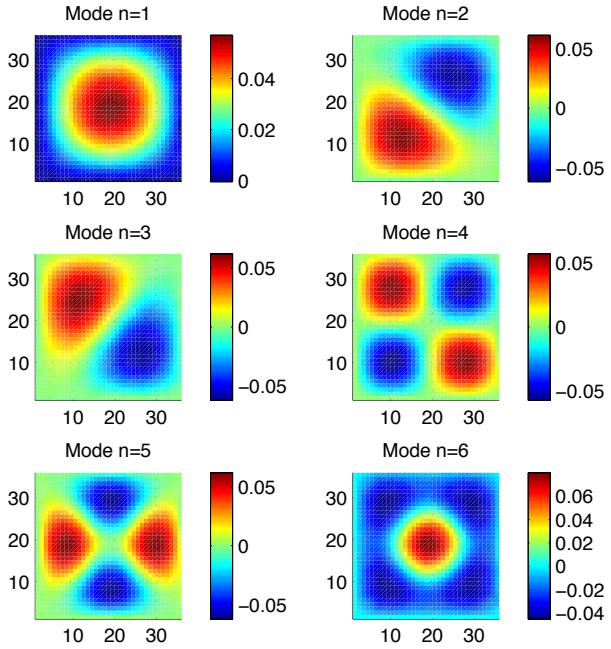


Fig. 7: The first six TE resonator modes of a square cavity (35×35 cells, $\Delta x = \Delta y = 1\,\text{mm}$).

At first glance, the diagonal modes in Fig. 7 might come somewhat unexpected, but these field distributions are the result of superposition of the second and third modes of the rectangular cavity when the widths in x-direction and y-direction become equal (and thus both modes are present at the same (eigen-)frequency). The same is true for modes 5 and 6, where the fields cancel (mode 5) or add (mode 6) along both diagonals for the square cavity.

3) Comparison to FDTD: Compared to the eigenfrequency analysis using FDTD, the FDFD is again more straightforward — and will find the modes with certainty. Using FDTD, the field in the cavity would be excited by a broadband pulse at some position inside, and be probed at some other point(s). To "ensure" that all modes are picked up, multiple field probes could be distributed randomly. An other way could be to locate the probes at points of irrational factors of the aspect ratio of the resonator (e.g. $1/\sqrt{2}$), since the zeros always occur at rational fractions of the width and length of the resonator. Since the grid is discretized, there are no real irrational points available and, thus, both methods cannot ensure to pick up all the modes reliably all the time (which should generally not be a problem, however, if enough probes are used). Also, to reduce "noise" and make the modes separable enough, FDTD mode analysis can require a high number of time steps, especially for higher-order modes.

With the FDFD eigenmode analysis, depending on the actual eigenvalue decomposition algorithm, all eigenfrequencies (that can be resolved by the discretized spatial grid, according to the Nyquist-limit) or those close to a chosen value, are found directly. With combined eigenvalue/eigenvector solvers also the field distributions can be found directly. Thus, also in this case, the frequency-domain methods come more naturally and straightforward than the time-domain counterparts.

[7]Matlab provides the command `sparse()` for this purpose (which is efficient for sparse matrices with densities of about 2/3 or lower, as given in the help manual). However, some functions such as `eig()` cannot deal with sparse matrices (if they do not have special properties, e.g. symmetry) and the command `full()` has to be used first, to again receive a common full matrix, thus reducing the effect for unoptimized applications.

B. 2.5-D FDFD Simulation of Propagation Constants of a Guided Wave Structure

1) Derivation: If we assume that the guided wave structure is uniform along the z-axis and the wave propagates in the positive z-direction, the guided fields can be expressed as: [6]

$$\mathbf{E}(x,y,z) = \left[\hat{\mathbf{x}}E^x(x,y) + \hat{\mathbf{y}}E^y(x,y) + \hat{\mathbf{z}}E^z(x,y)\right] \mathrm{e}^{-j\beta z}\,, \quad (15a)$$

$$\mathbf{H}(x,y,z) = \left[\hat{\mathbf{x}}H^x(x,y) + \hat{\mathbf{y}}H^y(x,y) + \hat{\mathbf{z}}H^z(x,y)\right] \mathrm{e}^{-j\beta z}\,, \quad (15b)$$

where, as usual, the harmonic time variation $\mathrm{e}^{j\omega t}$ is suppressed. Further, derivatives with respect to z can be taken analytically and result in

$$\partial_z G^v = -j\beta G^v \quad (16)$$

for any field quantity G in direction v. Thus follows for Maxwell's curl equations

$$\boldsymbol{\nabla} \times \mathbf{E} = \begin{bmatrix} \partial_y E^z + j\beta E^y \\ -j\beta E^x - \partial_x E^z \\ \partial_x E^y - \partial_y E^x \end{bmatrix} = -j\omega\mu \begin{bmatrix} H^x \\ H^y \\ H^z \end{bmatrix} \quad (17a)$$

$$\Downarrow$$

$$\begin{bmatrix} j\omega\mu H^y - \partial_x E^z \\ -\partial_y E^z - j\omega\mu H^x \\ \partial_x E^y - \partial_y E^x + j\omega\mu H^z \end{bmatrix} = j\beta \begin{bmatrix} E^x \\ E^y \\ 0 \end{bmatrix} \quad (17b)$$

for the electric field (Faraday's law) and from the curl of the magnetic field (Ampère's law)

$$\boldsymbol{\nabla}\times\mathbf{H}=\begin{bmatrix}\partial_y H^z+j\beta H^y\\ -j\beta H^x-\partial_x H^z\\ \partial_x H^y-\partial_y H^x\end{bmatrix}=j\omega\mu\begin{bmatrix}E^x\\E^y\\E^z\end{bmatrix} \tag{18a}$$

$$\Downarrow$$

$$\begin{bmatrix}-\partial_x H^z-j\omega\varepsilon E^y\\ j\omega\varepsilon E^x-\partial_y H^z\\ \partial_x H^y-\partial_y H^x-j\omega\varepsilon E^z\end{bmatrix}=j\beta\begin{bmatrix}H^x\\H^y\\0\end{bmatrix} \tag{18b}$$

These equations could already be implemented as

$$\begin{bmatrix}\mathbf{0}&\mathbf{0}&-\boldsymbol{\partial}_x&\mathbf{0}&j\omega\boldsymbol{\mu}&\mathbf{0}\\ \mathbf{0}&\mathbf{0}&-\boldsymbol{\partial}_x&-j\omega\boldsymbol{\mu}&\mathbf{0}&\mathbf{0}\\ -\boldsymbol{\partial}_y&\boldsymbol{\partial}_x&\mathbf{0}&\mathbf{0}&\mathbf{0}&j\omega\boldsymbol{\mu}\\ \mathbf{0}&-j\omega\boldsymbol{\varepsilon}&\mathbf{0}&\mathbf{0}&\mathbf{0}&-\boldsymbol{\partial}_x\\ j\omega\boldsymbol{\varepsilon}&\mathbf{0}&\mathbf{0}&\mathbf{0}&\mathbf{0}&-\boldsymbol{\partial}_y\\ \mathbf{0}&\mathbf{0}&-j\omega\boldsymbol{\varepsilon}&-\boldsymbol{\partial}_y&\boldsymbol{\partial}_x&\mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{E}^x\\\mathbf{E}^y\\\mathbf{E}^z\\\mathbf{H}^x\\\mathbf{H}^y\\\mathbf{H}^z\end{bmatrix}=j\beta\begin{bmatrix}\mathbf{E}^x\\\mathbf{E}^y\\\mathbf{0}\\\mathbf{H}^x\\\mathbf{H}^y\\\mathbf{0}\end{bmatrix} \tag{19}$$

but since two (matrix) equations are independent of the eigenvalue β, the system can be optimized further. By solving the respective rows for the z-components and filling the results into the respective spots, we get

$$\begin{bmatrix}\mathbf{0}&\mathbf{0}&\frac{\partial_x\partial_y}{j\omega\varepsilon}&j\omega\mu-\frac{\partial_x^2}{j\omega\varepsilon}\\ \mathbf{0}&\mathbf{0}&\frac{\partial_y^2}{j\omega\varepsilon}-j\omega\mu&-\frac{\partial_y\partial_x}{j\omega\varepsilon}\\ -\frac{\partial_x\partial_y}{j\omega\mu}&\frac{\partial_x^2}{j\omega\mu}-j\omega\varepsilon&\mathbf{0}&\mathbf{0}\\ j\omega\varepsilon-\frac{\partial_y^2}{j\omega\mu}&\frac{\partial_y\partial_x}{j\omega\mu}&\mathbf{0}&\mathbf{0}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{E}^x\\\mathbf{E}^y\\\mathbf{H}^x\\\mathbf{H}^y\end{bmatrix}=j\beta\begin{bmatrix}\mathbf{E}^x\\\mathbf{E}^y\\\mathbf{H}^x\\\mathbf{H}^y\end{bmatrix} \tag{20}$$

where we once again see the coupled nature of Maxwell's equations: the electric field depends only on the magnetic field and vice versa. The equation can be further simplified by solving for the magnetic fields and filling them into the equations for the electric fields, resulting in the well-known wave equations

$$\left(\omega^2\varepsilon\mu+\partial_x^2+\partial_y^2\right)\begin{bmatrix}\mathbf{E}^x\\\mathbf{E}^y\end{bmatrix}=\beta^2\begin{bmatrix}\mathbf{E}^x\\\mathbf{E}^y\end{bmatrix}\ . \tag{21}$$

which can be discretized straightforwardly using finite differences for the second-order derivatives as discussed previously.

Note that this simplification assumed uniform μ and ε, which is not true for most problems. In general, the respective $\varepsilon_{p,q}$ and $\mu_{p,q}$ (where $x=p\cdot\Delta x$ and $y=q\cdot\ \Delta y$ are the discretized positions) have to be carried all the way through the derivation, leading to coupled formulations for the two electric field components of the form [7]

$$\begin{aligned}a_1^x E^x_{p+1,q}+a_2^x E^x_{p-1,q}+a_3^x E^x_{p,q-1}+a_4^x E^x_{p,q+1}\\ +a_5^x E^y_{p+1,q-1}+a_6^x E^y_{p+1,q}+a_7^x E^y_{p,q-1}+a_8^x E^y_{p,q}\\ +a_9^x E^x_{p,q}=\beta^2 E^x_{p,q}\end{aligned} \tag{22}$$

where a_i^x $(i=1\ldots 9)$ are coefficients, related to an average permittivity around the node, e.g. [7]

$$a_1^x=-\frac{2}{\Delta x^2}\cdot\frac{\varepsilon_{p,q}+\varepsilon_{p,q+1}}{\varepsilon_{p,q}+\varepsilon_{p,q-1}+\varepsilon_{p+1,q}+\varepsilon_{p+1,q-1}}\ . \tag{23}$$

Eqs. (21) was implemented to simulate the propagation constant β of a common waveguide (PEC walls). Using finite differences for the derivatives follows

$$\begin{aligned}\frac{E^t_{p+1,q}-2E^t_{p,q}+E^t_{p-1,q}}{\Delta x^2}+\frac{E^t_{p,q+1}-2E^t_{p,q}+E^t_{p,q-1}}{\Delta y^2}\\ +\omega^2\varepsilon\mu E^t_{p,q}=\beta^2 E^t_{p,q}\end{aligned} \tag{24}$$

To state this in a linear system consisting of a matrix $\mathbf{A}$ acting on the electric fields in form of a vector $\mathbf{x}=[\mathbf{E}^x,\mathbf{E}^y]^T$, the coordinate indices p and q are mapped for the x and y components, respectively, according to

$$E^x_{p_x,q_x}:\ (p_x,q_x)\ \longmapsto\ n=p_x+(q_x-1)N_x \tag{25a}$$

$$E^y_{p_y,q_y}:\ (p_y,q_y)\ \longmapsto\ m=p_y+(q_y-1)(N_x+1)\ . \tag{25b}$$

where

$$\begin{aligned}p_x&=1,\ \ldots,N_x & q_x&=1,\ \ldots,N_y+1\\ p_y&=1,\ \ldots,N_x+1 & q_y&=1,\ \ldots,N_y\end{aligned}$$

Note that both mappings contain the number of cells in x-direction N_x, since that is the dimension along which the vectors are numbered first. Thus, we get, for example for $N_x=2$ and $N_y=3$ a situation as shown in Fig. 8.

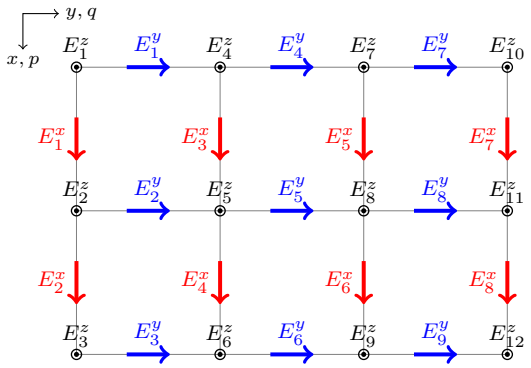


Fig. 8: Simplified Yee's cells in 2.5D using wave equations (E fields only, E^z is just shown for completeness, it is only implicitly contained in the algorithm), for $N_x=2$ and $N_y=3$.

Using these mappings, the equations become

$$\begin{aligned}\frac{E^x_{n+1}-2E^x_n+E^x_{n-1}}{\Delta x^2}+\frac{E^x_{n+N_x}-2E^x_n+E^x_{n-N_x}}{\Delta y^2}\\ +\omega^2\varepsilon\mu E^x_n=\beta^2 E^x_n\qquad n\in 1,2,\ \ldots,\ N_x(N_y+1)\end{aligned} \tag{26}$$

for the x-directed and

$$\begin{aligned}\frac{E^y_{m+1}-2E^y_n+E^y_{m-1}}{\Delta x^2}+\frac{E^y_{m+(N_x+1)}-2E^y_m+E^y_{m-(N_x+1)}}{\Delta y^2}\\ +\omega^2\varepsilon\mu E^y_m=\beta^2 E^y_m\qquad m\in 1,2,\ \ldots,\ (N_x+1)N_y\end{aligned} \tag{27}$$

for the y-directed fields.

2) Boundary conditions: The problem area (grid) has to be of finite extent and at the boundaries we have to define some physical relations for the fields. The simplest and most common boundary is a perfect electric conductor (PEC), which states that tangential fields are zero at the boundary and normal components can be dealt with using image theory. For E^x the boundaries are

- $E^x_{p_x,q_x} = 0$ for $q_x = 1, \forall\, p_x$
  $\rightarrow E^x_n = 0$ for $n = 1,\ ...,\ N_x$
- $E^x_{p_x,q_x} = 0$ for $q_x = N_y + 1, \forall\, p_x$
  $\rightarrow E^x_n = 0$ for $n = N_x\, N_y + 1,\ ...\ , N_x(N_y + 1)$
- $E^x_{p_x-1,q_x} = E^x_{p_x,q_x}$ (image) for $p_x = 1, \forall\, q_x$
  $\rightarrow E^x_{n-1} = E^x_n$ for $n = 1 + (q_x - 1)N_x$
- $E^x_{p_x+1,q_x} = E^x_{p_x,q_x}$ (image) for $p_x = N_x, \forall\, q_x$
  $\rightarrow E^x_{n+1} = E^x_n$ for $n = q_x N_x$

and similarly for E^y. The zeros dominate (are applied after the images) in case of boundaries on both p and q.

3) Algorithm: The following steps are taken in the algorithm to assemble the matrix:

a) Initialization: preassign zero square matrices of dimensions $N_x(N_y + 1)$ by $(N_x + 1)N_y$ for the E^x and E^y, respectively.
b) Put $\omega^2\varepsilon\mu - 2/\Delta x^2 - 2/\Delta y^2$ on the main diagonal of the matrix and $1/\Delta x^2$ along the diagonals one to the left and right of the main diagonal as well as $1/\Delta y^2$ to the diagonals at a $\pm N_x$ distance. Do the same for the E^y matrix, but use a distance of $N_x + 1$ for the distance to the diagonal in the later case.
c) Apply the correct boundaries: first apply the images by changing the respective positions on the main diagonals and then set the right cells to zero to cancel the tangential field parts.
d) At this point other boundary conditions (inside the problem domain) could be applied.

4) Results: The propagation characteristics β and α of the first three TE_{n0} modes were simulated using a finite-difference implementation of the wave equation of the electric field and the results compared to analytical solutions, as shown in Fig. 9. The dimensions of the waveguide are $19.05\,\mathrm{mm} \times 9.525\,\mathrm{mm}$ (to be able to compare the results to [7]), which was discretized into 24×12 cells. The matrix dimension becomes 300×300, but the simulation time is still shorter than one second. As can be seen, the agreement is very good.

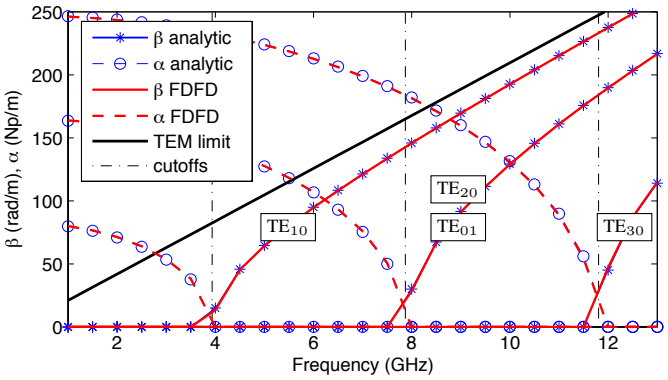


Fig. 9: The propagation constants β and attenuation constants α for the TE_{10}, TE_{20}, TE_{01} and TE_{30} modes of a rectangular waveguide. Dimensions $19.05\,\mathrm{mm} \times 9.525\,\mathrm{mm}$, according to [7].

IV. Conclusions & Outlook

A. Summary

The FDFD method was introduced and applied to several well-known problems, where the results and execution time performances were compared to those obtained with the well-known FDTD method. As shown, the FDFD method is a straightforward application of finite differences to the spatial derivatives (curls) of Maxwell's equations in the frequency domain (phasor form).

Solving Maxwell's equations in the frequency domain can be advantageous over the time domain for some applications, especially:

- Simulations using dispersive media
- Eigenmode analysis
- Simulations of high-Q resonators and their properties

As shown, FDFD methods are quite fast when results have to be obtained at a certain frequency, where with FDTD a large number of time steps (or interpolation or other enhancements) have to be simulated, to get the same resolution. Also, unlike the FDTD method, FDFD is inherently stable, since it is essentially inverting a matrix to solve a linear system. On the other hand, this matrix, containing all the spatial derivatives of the fields in the model, grows very fast with increasing model sizes and complexities. That is why FDFD is known to require (much) more RAM compared to FDTD, since this matrix needs to be stored at once.

B. Personal Notes

On a personal note, doing simulations in FDFD (and verifying and debugging an FDFD code) has proved to be very different than FDTD. First of all, since FDFD simulates the steady-state, we can no longer get away with certain issues by "moving them far enough away" so that their effects do not compromise the simulation results. Thus, a good PML boundary is much more important than in FDTD. Also, using FDFD there is no way to start a simulation and then for example observe what happens to the propagating wave to see where a problem is located — a property of the FDTD method which the author grew to like quite a bit over many simulations.

Although the FDFD method is commonly used in Photonics (since there simulations of dispersive media are very common) and other areas, literature on this topic is quite sparse. Most reports focus on special cases and enhancements for specific applications.

C. Outlook

For further investigations, for example simulation of dielectric resonators in multiple dimensions a working (anisotropic) PML suited for eigenmode analysis is required. Also, especially for circular (disk) resonators, it could be interesting

to have a conformal mesh, e.g. like the polar coordinate system, where the FDFD could be implemented by discretizing Maxwell's equations in cylindrical coordinates.

ACKNOWLEDGMENT

The author would like to thank Prof. Costas D. Sarris, Dr. G. Kevin Zhu and Neeraj Sood for their helpful hints and comments.

REFERENCES

[1] C. M. Rappaport, B. J. McCartin
FDFD Analysis of Electromagnetic Scattering in Anisotropic Media Using Unconstrained Triangular Meshes
IEEE Transactions on Antennas and Propagation, Vol. 39, No. 3, March 1991

[2] C. M. Rappaport
Perfectly Matched Absorbing Boundary Conditions Based on Anisotropic Lossy Mapping of Space
IEEE Microwave and Guided Wave Letters, Vol. 5, No. 3, March 1995

[3] M.-L. Lui, Z. Cheng
A direct computation of propagation constant using compact 2-D full-wave eigen-based finite-difference frequency-domain technique
Proceedings of the 1999 International Conference on Computational Electromagnetics and Its Applications (ICCEA '99), p. 78-81, 1999

[4] E. A. Marengo, C. M. Rappaport, E. L. Miller
Optimum PML ABC Conductivity Profile in FDFD
IEEE Transactions on Magnetics, Vol. 35, No. 3, May 1999

[5] C. M. Rappaport, M. Kilmer, E. L. Miller
Accuracy considerations in using the PML ABC with FDFD Helmholtz equation computation
International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, Vol. 13, No. 5, October 2000

[6] Y.-J. Zhao, K.-L. Wu, K.-K. M. Cheng
A Compact 2-D Full-Wave Finite-Difference Frequency-Domain Method for General Guided Wave Structures
IEEE Transactions on Microwave Theory and Techniques, Vol. 50, No. 7, July 2002

[7] L.-Y. Li, J.-F. Mao
An Improved Compact 2-D Finite-Difference Frequency-Domain Method for Guided Wave structures
IEEE Microwave and Wireless Components Letters, Vol. 13, No. 12, December 2003

[8] K. S. Yee
Numerical solution of initial boundary value problems involving Maxwells equations in isotropic media
IEEE Transactions on Antennas and Propagation, Vol. 14, May 1966

[9] S. Xiao, R. Vahldieck, H. Jin
A fast 2-D FDTD full wave analyzer for arbitrary guided wave structures
IEEE Microwave Guided Wave Letters, Vol. 2, No. 5, May 1992

[10] S. Xiao, R. Vahldieck
An Efficient 2-D FDTD Algorithm Using Real Variables
IEEE Microwave and Guided Wave Letters, Vol. 3, No. 5, May 1993

[11] Aliaksandra Ivinskaya
Finite-Difference Frequency-Domain Method in Nanophotonics
PhD Thesis, Technical University of Denmark, Lyngby, 2011

[12] Umran S. Inan, Robert A. Marshall
Numerical Electromagnetics – The FDTD Method
Cambridge University Press 2011