Explicit and provably stable spatiotemporal FDTD refinement

W. Tierens[a,*]

[a] *Max Planck Institut für Plasmaphysik*

Abstract

In this paper we introduce an explicit and provably conditionally stable Finite Difference Time Domain (FDTD) algorithm for Maxwell's equations, with local refinement in both the spatial discretisation length and in the time step (spatiotemporal refinement). This enables local spatial refinement with a locally reduced time step.

Keywords: FDTD, Maxwell's equations, spatial refinement, temporal refinement, multiscale, multirate

1. Introduction

The Finite Difference Time Domain (FDTD) algorithm [16] is a well-known computational method for solving Maxwell's equations. It has many desirable characteristics: it is conceptually simple, easy to implement, easy to parallelize thanks to its explicit nature, and comes with strong mathematical guarantees regarding its stability and discrete conservation laws. One such guarantee is the Courant condition: FDTD is stable if the time step, Δ_t, obeys

$$c\Delta_t < \frac{\Delta}{\sqrt{d}} \tag{1}$$

where Δ is the spatial discretisation length, c the speed of light and d the number of spatial dimensions. (1) is derived under the assumption of a standard Yee grid filled with a uniform medium. More general but not fundamentally different

[*] Corresponding author
Email address: `wouter.tierens@ipp.mpg.de` (W. Tierens)

stability conditions are known in the case of non-Yee grids and non-uniform media, such as (9).

This stability condition, unfortunately, prevents us from resolving small features in the problem domain without choosing a denser grid and thus a (globally) smaller time step. This drawback has been partially overcome: there are ways of constructing spatially refined grids [1] which let us resolve small features locally, but still impose a global reduction of the time step. Features which are small in just one direction (e.g. thin layers of conducting material) can be included using partially implicit approaches, which do not impose global time step reduction nor globally denser meshes [9, 20]. In fact, local spatial refinement with no global time step reduction is always possible provided one is willing to use Crank-Nicolson (implicit) updates in the refined region [19, 20, 21], or by filtering out unstable modes in the refined grid [22], but this reduces both the ease of implementation, and the ease of parallelization.

In this paper we introduce an explicit spatiotemporal FDTD refinement scheme, which lets us update the refined regions with a *locally* smaller time step. We are not the first to attempt this [25, 23, 1, 13, 4, 5, 12, 3, 6], but to our knowledge this scheme is the first that is explicit, provably conditionally stable, and recursively applicable (i.e. the refined region can be further refined). In [19], we constructed an explicit 1:2 refinement scheme that is provably conditionally stable and remains stable at the coarse Courant limit, but is not recursively applicable. The schemes of [4, 5] are provably conditionally stable and stable at the coarse Courant limit, but not explicit: they require, at every step, the solution of a system of equations whose size is proportional to the size of the coarse-fine interface.

This paper is structured as follows: in section 2, we derive a multirate time-stepping operator based on algebraic considerations, which we summarize as a practical algorithm in section 3. In section 4, we give a proof of its conditional stability. In section 5 we combine our multirate time-stepping operator with spatial refinement. Numerical examples are in section 6, and the conclusion is in section 7.

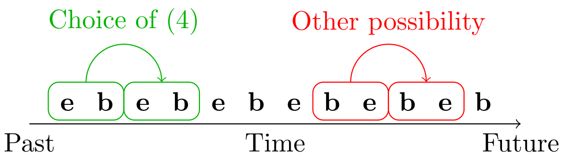

Figure 1: The staggered nature of FDTD time-stepping, and the resulting ambiguity in the definition of the time-stepping operator. The choice of eq. (4) is the one where $[\mathbf{e}_-^T, \mathbf{b}_-^T]^T$ and $[\mathbf{e}_+^T, \mathbf{b}_+^T]^T$ are such that the electric field is discretized half a time step to the past of the magnetic field, traditionally written as $\left[\mathbf{e}^T_{(n-\frac{1}{2})\Delta_t}, \mathbf{b}^T_{n\Delta_t}\right]^T$ and $\left[\mathbf{e}^T_{(n+\frac{1}{2})\Delta_t}, \mathbf{b}^T_{(n+1)\Delta_t}\right]^T$.

2. Multi-rate FDTD time-stepping: an algebraic approach

The FDTD algorithm numerically solves Maxwell's equations

$$\frac{1}{c}\frac{\partial E}{\partial t} = \frac{1}{\varepsilon_r}\nabla \times \mu_r^{-1} B \tag{2}$$

$$\frac{1}{c}\frac{\partial B}{\partial t} = -\nabla \times E \tag{3}$$

where E is the electric field, B the magnetic field, c the speed of light, and μ_r, ε_r the relative magnetic permeability and electric permittivity. We choose in this paper to use Gaussian units, since they make the symmetry of Maxwell's equations more readily apparent: note the factor $1/c$ in both (2) and (3).

FDTD discretizes the curl operator $\nabla\times$ on a discrete grid, with discretisation length Δ, usually by using central differences to approximate the spatial derivatives. The discrete curl operator can then be represented as a matrix C.

FDTD, being a time-domain algorithm, has the task of mapping the discrete electric and magnetic fields known at the current time step, which we will call $[\mathbf{e}_-^T, \mathbf{b}_-^T]^T$, onto those at the next time step, which we will call $[\mathbf{e}_+^T, \mathbf{b}_+^T]^T$ (Due to the staggered nature of “leapfrog” FDTD time-stepping, there is a certain ambiguity in the choice of $[\mathbf{e}_-^T, \mathbf{b}_-^T]^T$ and $[\mathbf{e}_+^T, \mathbf{b}_+^T]^T$, which is resolved in figure 1). This mapping is achieved by solving the following equation [1, 19]

$$\frac{\begin{bmatrix}\mathbf{e}_+\\ \mathbf{b}_+\end{bmatrix} - \begin{bmatrix}\mathbf{e}_-\\ \mathbf{b}_-\end{bmatrix}}{c\Delta_t} = \begin{bmatrix}0 & [\star_\epsilon]^{-1}C^T[\star_\mu^{-1}]\\ 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}_-\\ \mathbf{b}_-\end{bmatrix} + \begin{bmatrix}0 & 0\\ -C & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}_+\\ \mathbf{b}_+\end{bmatrix} \tag{4}$$

where the l.h.s. approximates the time derivatives in Maxwell's equations, and the r.h.s. the curls. The "mass matrices" $[\star_\epsilon]$ and $[\star_\mu^{-1}]$ encode on their diagonal the (possibly inhomogeneous) relative permittivity and (inverse) magnetic permeability.

(4) can be further simplified by a change of basis (from $\mathbf{e}, \mathbf{b}$ to $\boldsymbol{\mathcal{E}}, \boldsymbol{\mathcal{B}}$):

$$\boldsymbol{\mathcal{E}} = [\star_\epsilon]^{1/2}\mathbf{e} \tag{5}$$

$$\boldsymbol{\mathcal{B}} = [\star_\mu^{-1}]^{1/2}\mathbf{b} \tag{6}$$

$$\mathcal{C} = [\star_\mu^{-1}]^{1/2}\ C[\star_\epsilon]^{-1/2} \tag{7}$$

$$\frac{1}{c}\frac{\begin{bmatrix}\boldsymbol{\mathcal{E}}_+\\ \boldsymbol{\mathcal{B}}_+\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mathcal{E}}_-\\ \boldsymbol{\mathcal{B}}_-\end{bmatrix}}{\Delta_t} = \underbrace{\begin{bmatrix}0 & \mathcal{C}^T\\ 0 & 0\end{bmatrix}}_{M_1}\begin{bmatrix}\boldsymbol{\mathcal{E}}_-\\ \boldsymbol{\mathcal{B}}_-\end{bmatrix} + \underbrace{\begin{bmatrix}0 & 0\\ -\mathcal{C} & 0\end{bmatrix}}_{M_2}\begin{bmatrix}\boldsymbol{\mathcal{E}}_+\\ \boldsymbol{\mathcal{B}}_+\end{bmatrix} \tag{8}$$

The anti-symmetry of (8), where $\mathcal{C}^T$ appears in the electric field update and $-\mathcal{C}$ in the magnetic field update, is sometimes called spatial reciprocity [1, 24, 8] and is the key ingredient in proofs that show the stability of (4) and (8). The stability condition associated with (4) and (8) is [1, 19]:

$$c\Delta_t < \frac{2}{\|\mathcal{C}\|} \tag{9}$$

where $\|\cdot\|$ is the induced matrix 2-norm.

We must stress that (8) and (9) do not assume homogeneity: $\mathcal{C}$ depends on the (possibly inhomogeneous) ε_r and μ_r. (8) and (9) are not limited to uniform grids either, a fact of which we will make use to construct the spatial part of our spatiotemporal refinement, in section 5.

Time-stepping equations like (8) have a time-stepping operator associated with them, a linear operator T which directly maps the current fields onto the future fields:

$$\begin{bmatrix}\boldsymbol{\mathcal{E}}_+\\ \boldsymbol{\mathcal{B}}_+\end{bmatrix} = T\begin{bmatrix}\boldsymbol{\mathcal{E}}_-\\ \boldsymbol{\mathcal{B}}_-\end{bmatrix}$$

$$T = \left(\frac{I}{c\Delta_t} - M_2\right)^{-1}\left(\frac{I}{c\Delta_t} + M_1\right) \tag{10}$$

where M_1, M_2 were defined in (8).

Remarkably, T^2, the operator that represents two consecutive FDTD time steps, can be brought under the form (10) as well:

$$T^2 = \left(\frac{I}{c\Delta_t} - M_2\right)^{-1}\left(\frac{I}{c\Delta_t} + M_1\right)\left(\frac{I}{c\Delta_t} - M_2\right)^{-1}\left(\frac{I}{c\Delta_t} + M_1\right) \tag{11}$$

$$= \left(\frac{I}{c\Delta_t} - M_2'\right)^{-1}\left(\frac{I}{c\Delta_t} + M_1'\right) \tag{12}$$

where

$$M_1' = M_2 \begin{bmatrix} I & c\Delta_t\mathcal{C}^T \\ 0 & 0 \end{bmatrix} + M_1 \tag{13}$$

$$M_2' = M_1 \begin{bmatrix} 0 & 0 \\ c\Delta_t\mathcal{C} & I \end{bmatrix} + M_2 \tag{14}$$

The inverses in (11) and (12) are all easily computable using block matrix inversion formulas [10].

Let us now consider what a mixed time-stepping operator might look like, one that combines single time steps (T) with double time steps (T^2). To this end, let us introduce diagonal matrices $P(a,b)$ and $Q(a,b)$, which tell us where to use single-step and where to use double-step updates:

$$P(a,b)_{ii} = \begin{cases} a & \text{Electric discretisation point } i \text{ updates in one step} \\ b & \text{Electric discretisation point } i \text{ updates in two steps} \end{cases} \tag{15}$$

$$Q(a,b)_{ii} = \begin{cases} a & \text{Magnetic discretisation point } i \text{ updates in one step} \\ b & \text{Magnetic discretisation point } i \text{ updates in two steps} \end{cases} \tag{16}$$

Let

$$\alpha = \begin{bmatrix} \alpha_E & 0 \\ 0 & \alpha_B \end{bmatrix} \tag{17}$$

$$\alpha_E = P(1,0) \tag{18}$$

$$\alpha_B = Q(1,0) \tag{19}$$

With these, we can construct the following time-stepping equation

$$\frac{1}{c}\frac{\begin{bmatrix}\boldsymbol{\mathcal{E}}_+\\ \boldsymbol{\mathcal{B}}_+\end{bmatrix}-\begin{bmatrix}\boldsymbol{\mathcal{E}}_-\\ \boldsymbol{\mathcal{B}}_-\end{bmatrix}}{\Delta_t}=(\alpha M_1+(I-\alpha)M_1')\begin{bmatrix}\boldsymbol{\mathcal{E}}_-\\ \boldsymbol{\mathcal{B}}_-\end{bmatrix}+(\alpha M_2+(I-\alpha)M_2')\begin{bmatrix}\boldsymbol{\mathcal{E}}_+\\ \boldsymbol{\mathcal{B}}_+\end{bmatrix} \tag{20}$$

i.e. we simply interpolate (at every discretisation point) between M_1, M_2, and M_1', M_2'. But this does not quite suffice: we also want a time-stepping operator to move forward in time by a constant amount of time, say Δ_t. But (20) moves those points where $\alpha_{ii} = 1$ forward by Δ_t, and those where $\alpha_{ii} = 0$ by $2\Delta_t$. We fix this by replacing Δ_t by $(2I-\alpha)^{-1}\Delta_t$, not just in the l.h.s. but also in M_1', M_2'. Then our improved mixed time-stepping equation is

$$\frac{(2I-\alpha)}{c}\frac{\begin{bmatrix}\boldsymbol{\mathcal{E}}_+\\ \boldsymbol{\mathcal{B}}_+\end{bmatrix}-\begin{bmatrix}\boldsymbol{\mathcal{E}}_-\\ \boldsymbol{\mathcal{B}}_-\end{bmatrix}}{\Delta_t}=(\alpha M_1+(I-\alpha)M_1'')\begin{bmatrix}\boldsymbol{\mathcal{E}}_-\\ \boldsymbol{\mathcal{B}}_-\end{bmatrix}+(\alpha M_2+(I-\alpha)M_2'')\begin{bmatrix}\boldsymbol{\mathcal{E}}_+\\ \boldsymbol{\mathcal{B}}_+\end{bmatrix} \tag{21}$$

where

$$M_1''=M_2\begin{bmatrix}I & (2I-\alpha_E)^{-1}c\Delta_t\mathcal{C}^T\\ 0 & 0\end{bmatrix}+M_1 \tag{22}$$

$$M_2''=M_1\begin{bmatrix}0 & 0\\ (2I-\alpha_B)^{-1}c\Delta_t\mathcal{C} & I\end{bmatrix}+M_2 \tag{23}$$

The time-stepping operator defined by (21) can be obtained explicitly. After left-multiplying both sides by $(2I-\alpha)^{-1}$, (21) becomes

$$\begin{aligned}\frac{\begin{bmatrix}\boldsymbol{\mathcal{E}}_+\\ \boldsymbol{\mathcal{B}}_+\end{bmatrix}-\begin{bmatrix}\boldsymbol{\mathcal{E}}_-\\ \boldsymbol{\mathcal{B}}_-\end{bmatrix}}{c\Delta_t} &= \begin{bmatrix}0 & \frac{1}{2}(I+\alpha_E)\mathcal{C}^T\\ -\frac{1}{2}(I-\alpha_B)\mathcal{C} & -\frac{1}{2}(I-\alpha_B)\mathcal{C}(2I-\alpha_E)^{-1}c\Delta_t\mathcal{C}^T\end{bmatrix}\begin{bmatrix}\boldsymbol{\mathcal{E}}_-\\ \boldsymbol{\mathcal{B}}_-\end{bmatrix}\\ &+\begin{bmatrix}\frac{1}{2}(I-\alpha_E)\mathcal{C}^T(2I-\alpha_B)^{-1}c\Delta_t\mathcal{C} & \frac{1}{2}(I-\alpha_E)\mathcal{C}^T\\ -\frac{1}{2}(I+\alpha_B)\mathcal{C} & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{\mathcal{E}}_+\\ \boldsymbol{\mathcal{B}}_+\end{bmatrix}\end{aligned} \tag{24}$$

The corresponding time-stepping operator is

$$\begin{aligned}T_{\text{mix}} &= \begin{bmatrix}2I-\alpha_E-(I-\alpha_E)\mathcal{C}^T(I+\alpha_B)\frac{c^2\Delta_t^2}{2}\mathcal{C} & -c\Delta_t(I-\alpha_E)\mathcal{C}^T\\ c\Delta_t\mathcal{C} & 2I-\alpha_B\end{bmatrix}^{-1}\\ &\cdot\begin{bmatrix}2I-\alpha_E & c\Delta_t\mathcal{C}^T\\ -(I-\alpha_B)c\Delta_t\mathcal{C} & 2I-\alpha_B-(I-\alpha_B)\mathcal{C}(I+\alpha_E)\frac{c^2\Delta_t^2}{2}\mathcal{C}^T\end{bmatrix}\end{aligned} \tag{25}$$

We work out the inverse using block matrix inversion formulas. After further simplifications, noting in particular that $(2I-\alpha)^{-1}(I-\alpha)\equiv\frac{1}{2}(I-\alpha)$ and $(2I-\alpha)^{-1}\equiv\frac{1}{2}(I+\alpha)$, we obtain

$$\begin{aligned}T_{\text{mix}} &= \begin{bmatrix}I & \frac{c\Delta_t}{2}(I-\alpha_E)\mathcal{C}^T\\ -\frac{c\Delta_t}{2}(I+\alpha_B)\mathcal{C} & I-\frac{c^2\Delta_t^2}{4}(I+\alpha_B)\mathcal{C}(I-\alpha_E)\mathcal{C}^T\end{bmatrix}\\ &\cdot\begin{bmatrix}I & \frac{c\Delta_t}{2}(I+\alpha_E)\mathcal{C}^T\\ -\frac{c\Delta_t}{2}(I-\alpha_B)\mathcal{C} & I-\frac{c^2\Delta_t^2}{4}(I-\alpha_B)\mathcal{C}(I+\alpha_E)\mathcal{C}^T\end{bmatrix}\end{aligned} \tag{26}$$

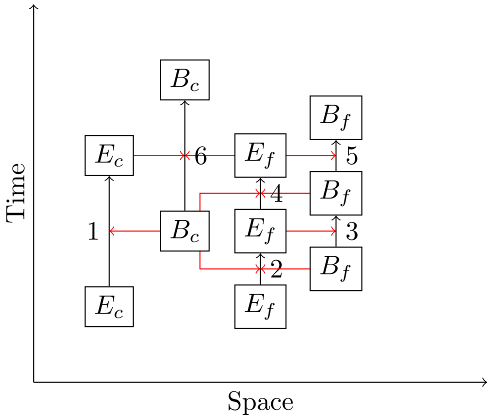

Figure 2: A pictorial representation of the mixed time-stepping procedure. Subscripts indicate 1-step (“coarse”:E_c, B_c) and 2-step (“fine”:E_f, B_f) quantities. Black arrows and accompanying numbers indicate the order of the substeps. Red arrows indicate which quantities are used to calculate the spatial derivatives needed for the curl calculation in the substeps.

3. Practical implementation

T_{mix} can be expanded further:

$$\begin{aligned}
T_{\text{mix}} &= \begin{bmatrix} I & 0 \\ -\frac{c\Delta_t}{2}(I+\alpha_B)\mathcal{C} & I \end{bmatrix} \begin{bmatrix} I & \frac{c\Delta_t}{2}(I-\alpha_E)\mathcal{C}^T \\ 0 & I \end{bmatrix} \\
&\quad \cdot \begin{bmatrix} I & 0 \\ -\frac{c\Delta_t}{2}(I-\alpha_B)\mathcal{C} & I \end{bmatrix} \begin{bmatrix} I & \frac{c\Delta_t}{2}(I+\alpha_E)\mathcal{C}^T \\ 0 & I \end{bmatrix} \\
&= \begin{bmatrix} I & 0 \\ -c\Delta_t\alpha_B\mathcal{C} & I \end{bmatrix} \left(\begin{bmatrix} I & 0 \\ -\frac{c\Delta_t}{2}(I-\alpha_B)\mathcal{C} & I \end{bmatrix} \begin{bmatrix} I & \frac{c\Delta_t}{2}(I-\alpha_E)\mathcal{C}^T \\ 0 & I \end{bmatrix} \right)^2 \\
&\quad \cdot \begin{bmatrix} I & c\Delta_t\alpha_E\mathcal{C}^T \\ 0 & I \end{bmatrix}
\end{aligned} \tag{27}$$

From this, we see that the algorithm proceeds as follows

1. Update the 1-step electric field points (time step Δ_t).
2. Update the 2-step electric field points (time step $\Delta_t/2$).
3. Update the 2-step magnetic field points (time step $\Delta_t/2$).
4. Update the 2-step electric field points (time step $\Delta_t/2$).

5. Update the 2-step magnetic field points (time step $\Delta_t/2$).
6. Update the 1-step magnetic field points (time step Δ_t).

Every update in this list is a standard FDTD update, with no modifications whatsoever. A pictorial representation of this procedure is shown in figure 2.

(27) is in many ways the simplest and lowest-order temporal refinement scheme imaginable. There is no clever temporal interpolation: we simply do a coarse update, all the fine updates, and another coarse update.

This formulation immediately suggests generalisations. A recursive algorithm for temporal refinement by an arbitrary power of 2 is shown in algorithm 1. For this, we give all points a "refinement level" which is 0 for 1-step updates, 1 for 2-step updates, 2 for 4-step updates, etc. When the maximum refinement level is 1, algorithm 1 reduces to the steps given above.

```
function Update($\Delta_t$,$level$):
    Update the electric field points at refinement level $level$ with time
      step $\Delta_t$, using standard FDTD update equations
    if $level$ is not the maximum level then
        Update($\Delta_t/2$,$level + 1$)
        Update($\Delta_t/2$,$level + 1$)
    end
    Update the magnetic field points at refinement level $level$ with time
      step $\Delta_t$, using standard FDTD update equations
```

Algorithm 1: Recursive updating for temporal refinement by an arbitrary power of 2, where 1-step updates have refinement level 0, 2-step updates have refinement level 1, etc.

4. Stability proof

Lemma 1. *Given a time-stepping equation of the form*

$$\frac{1}{c}\frac{\begin{bmatrix}\boldsymbol{\mathcal{E}}_+\\ \boldsymbol{\mathcal{B}}_+\end{bmatrix} - \begin{bmatrix}\boldsymbol{\mathcal{E}}_-\\ \boldsymbol{\mathcal{B}}_-\end{bmatrix}}{\Delta_t} = H_1\begin{bmatrix}\boldsymbol{\mathcal{E}}_-\\ \boldsymbol{\mathcal{B}}_-\end{bmatrix} + H_2\begin{bmatrix}\boldsymbol{\mathcal{E}}_+\\ \boldsymbol{\mathcal{B}}_+\end{bmatrix} \tag{28}$$

where H_1, H_2 are real square matrices. The eigenvalues of the time-stepping operator associated with (28) all have unit norm if and only if the eigenvalues of $\left(\frac{I}{c\Delta_t} + \frac{H_1 - H_2}{2}\right)^{-1} \frac{H_1+H_2}{2}$ are purely imaginary.

Proof. To find an eigenvector $[\boldsymbol{\mathcal{E}}^T, \boldsymbol{\mathcal{B}}^T]^T$ and eigenvalue λ of the time-stepping operator defined by (28), insert $[\boldsymbol{\mathcal{E}}_+^T, \boldsymbol{\mathcal{B}}_+^T]^T = \lambda[\boldsymbol{\mathcal{E}}^T, \boldsymbol{\mathcal{B}}^T]^T$ and $[\boldsymbol{\mathcal{E}}_-^T, \boldsymbol{\mathcal{B}}_-^T]^T = [\boldsymbol{\mathcal{E}}^T, \boldsymbol{\mathcal{B}}^T]^T$ in (28):

$$\frac{\lambda - 1}{c\Delta_t} \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} = H_1 \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} + H_2 \lambda \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} \tag{29}$$

Let $\mathcal{H}_1 = (H_1 + H_2)/2$, $\mathcal{H}_2 = (H_1 - H_2)/2$, then

$$\frac{\lambda - 1}{c\Delta_t} \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} = (\mathcal{H}_1 + \mathcal{H}_2) \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} + (\mathcal{H}_1 - \mathcal{H}_2)\lambda \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} \tag{30}$$

$$\frac{\lambda}{c\Delta_t} \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} = \frac{I}{c\Delta_t} \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} + (\mathcal{H}_1 + \mathcal{H}_2) \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} + (\mathcal{H}_1 - \mathcal{H}_2)\lambda \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} \tag{31}$$

$$\left(\frac{I}{c\Delta_t} - (\mathcal{H}_1 - \mathcal{H}_2)\right) \lambda \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} = \frac{I}{c\Delta_t} \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} + (\mathcal{H}_1 + \mathcal{H}_2) \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} \tag{32}$$

$$\lambda \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} = \left(\frac{I}{c\Delta_t} - \mathcal{H}_1 + \mathcal{H}_2\right)^{-1} \left(\frac{I}{c\Delta_t} + \mathcal{H}_1 + \mathcal{H}_2\right) \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} \tag{33}$$

$$\lambda \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} = \left(I - \left(\frac{I}{c\Delta_t} + \mathcal{H}_2\right)^{-1} \mathcal{H}_1\right)^{-1} \left(I + \left(\frac{I}{c\Delta_t} + \mathcal{H}_2\right)^{-1} \mathcal{H}_1\right) \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} \tag{34}$$

$$\lambda \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} = f\left(\left(\frac{I}{c\Delta_t} + \mathcal{H}_2\right)^{-1} \mathcal{H}_1\right) \begin{bmatrix} \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{B}} \end{bmatrix} \tag{35}$$

where $f(M) = (I - M)^{-1}(I + M)$.
Thus, $[\boldsymbol{\mathcal{E}}^T, \boldsymbol{\mathcal{B}}^T]^T$ is an eigenvector of $\left(\frac{I}{c\Delta_t} + \mathcal{H}_2\right)^{-1} \mathcal{H}_1$ with eigenvalue κ, and $\lambda = \frac{1+\kappa}{1-\kappa}$. Therefore, $|\lambda| = 1$ if and only if κ is purely imaginary. □

Lemma 2. *If S is a real symmetric positive definite matrix, and A a real anti-symmetric matrix, then the solutions κ of $|A - \kappa S| = 0$ are purely imaginary.*

Proof. Because S is symmetric positive definite, S has a Cholesky decomposition $S = QQ^T$.

$$\begin{aligned} |A - \kappa S| &= 0 \\ |A - \kappa QQ^T| &= 0 \\ |Q^{-1}AQ^{-T} - \kappa I| &= 0 \end{aligned} \tag{36}$$

Thus κ is an eigenvalue of the real anti-symmetric matrix $Q^{-1}AQ^{-T}$, and therefore purely imaginary. □

Theorem 1. *The time-stepping operator T_{mix}, defined equivalently by (24), (26) and (27), which mixes 1-step with 2-step FDTD updates, is conditionally stable, i.e. there is some $\Delta_{t,max} > 0$ such that*

$$0 < \Delta_t < \Delta_{t,max} \implies \begin{cases} \textit{all eigenvalues of } T_{mix} \textit{ have unit norm} & \textit{1.A} \\ T_{mix} \textit{ is diagonalizable} & \textit{1.B} \end{cases} \tag{37}$$

The two implications in (37) will be proven separately, as theorems 1.A and 1.B.

Theorem 1.A. *There is some $\Delta_{t,max} > 0$ such that $0 < \Delta_t < \Delta_{t,max} \implies$ all eigenvalues of T_{mix} have unit norm.*

Proof. The time-stepping equation is of the form (28) (see (24)), so by lemma 1, we need to prove that the eigenvalues of $\left(\frac{I}{c\Delta_t} + \frac{H_1 - H_2}{2}\right)^{-1} \frac{H_1 + H_2}{2}$ are conditionally purely imaginary.

For T_{mix}, $\mathcal{H}_1 = (H_1 + H_2)/2$ and $\mathcal{H}_2 = (H_1 - H_2)/2$ are

$$\mathcal{H}_1 = \frac{1}{2} \begin{bmatrix} \frac{c\Delta_t}{4} P(0,1)\mathcal{C}^T Q(2,1)\mathcal{C} & \mathcal{C}^T \\ -\mathcal{C} & -\frac{c\Delta_t}{4} Q(0,1)\mathcal{C}P(2,1)\mathcal{C}^T \end{bmatrix} \tag{38}$$

$$\mathcal{H}_2 = -\frac{1}{2} \begin{bmatrix} \frac{c\Delta_t}{4} P(0,1)\mathcal{C}^T Q(2,1)\mathcal{C} & -P(1,0)\mathcal{C}^T \\ -Q(1,0)\mathcal{C} & \frac{c\Delta_t}{4} Q(0,1)\mathcal{C}P(2,1)\mathcal{C}^T \end{bmatrix} \tag{39}$$

Without loss of generality, we may assume $P(0,1) = \text{diag}(0, \cdots, 0, 1, \cdots, 1)$ and $Q(0,1) = \text{diag}(0, \cdots, 0, 1, \cdots, 1)$ (we can always permute the order of the discretisation points such that P and Q have this form). Then $\mathcal{C}$ is a block matrix

$$\mathcal{C} = \begin{bmatrix} \mathcal{C}_{11} & \mathcal{C}_{12} \\ \mathcal{C}_{21} & \mathcal{C}_{22} \end{bmatrix} \tag{40}$$

such that

$$\forall a, b : Q(a, b)\mathcal{C} = \begin{bmatrix} a\mathcal{C}_{11} & a\mathcal{C}_{12} \\ b\mathcal{C}_{21} & b\mathcal{C}_{22} \end{bmatrix} \tag{41}$$

$$\forall a, b : P(a, b)\mathcal{C}^T = \begin{bmatrix} a\mathcal{C}_{11}^T & a\mathcal{C}_{21}^T \\ b\mathcal{C}_{12}^T & b\mathcal{C}_{22}^T \end{bmatrix} \tag{42}$$

We can now write $\mathcal{H}_1, \mathcal{H}_2$ more explicitly:

$$\mathcal{H}_1 = \begin{bmatrix} 0 & 0 & \frac{1}{2}\mathcal{C}_{11}^T & \frac{1}{2}\mathcal{C}_{21}^T \\ \frac{c\Delta_t}{8}\left(\mathcal{C}_{12}^T 2\mathcal{C}_{11} + \mathcal{C}_{22}^T\mathcal{C}_{21}\right) & \frac{c\Delta_t}{8}\left(\mathcal{C}_{12}^T 2\mathcal{C}_{12} + \mathcal{C}_{22}^T\mathcal{C}_{22}\right) & \frac{1}{2}\mathcal{C}_{12}^T & \frac{1}{2}\mathcal{C}_{22}^T \\ \frac{-1}{2}\mathcal{C}_{11} & \frac{-1}{2}\mathcal{C}_{12} & 0 & 0 \\ \frac{-1}{2}\mathcal{C}_{21} & \frac{-1}{2}\mathcal{C}_{22} & \frac{-c\Delta_t}{8}\left(\mathcal{C}_{21} 2\mathcal{C}_{11}^T + \mathcal{C}_{22}\mathcal{C}_{12}^T\right) & \frac{-c\Delta_t}{8}\left(\mathcal{C}_{21} 2\mathcal{C}_{21}^T + \mathcal{C}_{22}\mathcal{C}_{22}^T\right) \end{bmatrix} \tag{43}$$

$$\mathcal{H}_2 = -\begin{bmatrix} 0 & 0 & \frac{-1}{2}\mathcal{C}_{11}^T & \frac{-1}{2}\mathcal{C}_{21}^T \\ \frac{c\Delta_t}{8}\left(\mathcal{C}_{12}^T 2\mathcal{C}_{11} + \mathcal{C}_{22}^T\mathcal{C}_{21}\right) & \frac{c\Delta_t}{8}\left(\mathcal{C}_{12}^T 2\mathcal{C}_{12} + \mathcal{C}_{22}^T\mathcal{C}_{22}\right) & 0 & 0 \\ \frac{-1}{2}\mathcal{C}_{11} & \frac{-1}{2}\mathcal{C}_{12} & 0 & 0 \\ 0 & 0 & \frac{c\Delta_t}{8}\left(\mathcal{C}_{21} 2\mathcal{C}_{11}^T + \mathcal{C}_{22}\mathcal{C}_{12}^T\right) & \frac{c\Delta_t}{8}\left(\mathcal{C}_{21} 2\mathcal{C}_{21}^T + \mathcal{C}_{22}\mathcal{C}_{22}^T\right) \end{bmatrix} \tag{44}$$

Let us try to find matrices K_L, K_R such that $K_L\mathcal{H}_1K_R$ is anti-symmetric and $K_L\left(\frac{I}{c\Delta_t} + \mathcal{H}_2\right)K_R$ is symmetric.

$$K_R = \begin{bmatrix} I & 0 & 0 & -\frac{1}{2}c\Delta_t\mathcal{C}_{21}^T \\ 0 & I & 0 & -\frac{1}{4}c\Delta_t\mathcal{C}_{22}^T \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \tag{45}$$

$$K_L = \begin{bmatrix} I & 0 & 0 & 0 \\ q\frac{c^2\Delta_t^2}{8}\mathcal{C}_{22}^T\mathcal{C}_{21} & q & \frac{c\Delta_t}{2}q\mathcal{C}_{12}^T & \frac{c\Delta_t}{4}q\mathcal{C}_{22}^T \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \tag{46}$$

$$q = \left(I - \frac{c^2\Delta_t^2}{16}\mathcal{C}_{22}^T\mathcal{C}_{22}\right)^{-1} \tag{47}$$

Then $K_L\mathcal{H}_1K_R$ is indeed anti-symmetric:

$$K_L\mathcal{H}_1K_R = \begin{bmatrix} 0 & 0 & \frac{1}{2}\mathcal{C}_{11}^T & \frac{1}{2}\mathcal{C}_{21}^T \\ 0 & 0 & \frac{1}{2}\mathcal{C}_{12}^T & \frac{1}{2}\mathcal{C}_{22}^T \\ \frac{-1}{2}\mathcal{C}_{11} & \frac{-1}{2}\mathcal{C}_{12} & 0 & \frac{c\Delta_t}{8}\left(\mathcal{C}_{11}2\mathcal{C}_{21}^T+\mathcal{C}_{12}\mathcal{C}_{22}^T\right) \\ \frac{-1}{2}\mathcal{C}_{21} & \frac{-1}{2}\mathcal{C}_{22} & \frac{-c\Delta_t}{8}\left(\mathcal{C}_{21}2\mathcal{C}_{11}^T+\mathcal{C}_{22}\mathcal{C}_{12}^T\right) & 0 \end{bmatrix} \tag{48}$$

And $K_L\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)K_R$ is indeed symmetric (note q commutes with $\mathcal{C}_{22}^T\mathcal{C}_{22}$):

$$K_L\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)K_R =$$

$$\begin{bmatrix} \frac{I}{c\Delta_t} & 0 & \frac{1}{2}\mathcal{C}_{11}^T & 0 \\ 0 & q\left(\frac{I}{c\Delta_t}-\frac{c\Delta_t}{8}\mathcal{C}_{22}^T\mathcal{C}_{22}\right) & \frac{1}{2}\mathcal{C}_{12}^T & 0 \\ \frac{1}{2}\mathcal{C}_{11} & \frac{1}{2}\mathcal{C}_{12} & \frac{I}{c\Delta_t} & \frac{-c\Delta_t}{8}\left(\mathcal{C}_{11}2\mathcal{C}_{21}^T+\mathcal{C}_{12}\mathcal{C}_{22}^T\right) \\ 0 & 0 & \frac{-c\Delta_t}{8}\left(\mathcal{C}_{21}2\mathcal{C}_{11}^T+\mathcal{C}_{22}\mathcal{C}_{12}^T\right) & \frac{I}{c\Delta_t}-\frac{c\Delta_t}{8}\left(\mathcal{C}_{21}2\mathcal{C}_{21}^T+\mathcal{C}_{22}\mathcal{C}_{22}^T\right) \end{bmatrix} \tag{49}$$

Recall that we are interested in the eigenvalues κ of $\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)^{-1}\mathcal{H}_1$, and want them to be purely imaginary, since that implies the time-stepping operator (26) has unit-norm eigenvalues. These eigenvalues obey

$$\left|\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)^{-1}\mathcal{H}_1-\kappa I\right| = 0 \tag{50}$$

$$\left|\mathcal{H}_1-\kappa\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)\right| = 0 \tag{51}$$

K_L and K_R are invertible: their determinant is nonzero. We can thus left- and right-multiply by K_L and K_R without changing the solutions:

$$\left|\underbrace{K_L\mathcal{H}_1K_R}_{\text{anti-symmetric}}-\kappa\underbrace{K_L\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)K_R}_{\text{symmetric}}\right| = 0 \tag{52}$$

By lemma 2, the eigenvalues κ are purely imaginary if $K_L\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)K_R$ is positive definite.

We will now show that it is indeed possible to choose Δ_t small enough such that $K_L\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)K_R$ is positive definite. First, consider element (2,2) in (49). Under weak assumptions ($c\Delta_t < \frac{4}{\|\mathcal{C}_{22}\|}$, which is the stability condition of the two-step part, see (9)) we can use a series expansion:

$$q\left(\frac{I}{c\Delta_t}-\frac{c\Delta_t}{8}\mathcal{C}_{22}^T\mathcal{C}_{22}\right) = \frac{I}{c\Delta_t}-\sum_{k=1}^{\infty}\frac{(c\Delta_t)^{2k-1}}{16^k}(\mathcal{C}_{22}^T\mathcal{C}_{22})^k \tag{53}$$

So we can write $K_L\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)K_R$ as

$$K_L\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)K_R = \frac{I}{c\Delta_t}+T_0+T_1c\Delta_t - \sum_{k=2}^{\infty}\frac{(c\Delta_t)^{2k-1}}{16^k}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \mathcal{C}_{22}^T\mathcal{C}_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}^k \tag{54}$$

where T_0 is the term in $(c\Delta_t)^0$ and T_1 the term in $(c\Delta_t)^1$. $K_L\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)K_R$ is positive definite if, for all nonzero vectors $\mathbf{x}$

$$\forall \mathbf{x}\neq 0 : \mathbf{x}^T K_L\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)K_R\mathbf{x} > 0 \tag{55}$$

Using (54)

$$\frac{|\mathbf{x}|^2}{c\Delta_t}+\mathbf{x}^T T_0\mathbf{x}+\mathbf{x}^T T_1 c\Delta_t\mathbf{x} - \sum_{k=2}^{\infty}\frac{(c\Delta_t)^{2k-1}}{16^k}\mathbf{x}^T\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \mathcal{C}_{22}^T\mathcal{C}_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}^k\mathbf{x} > 0 \tag{56}$$

Since T_0, T_1 and $\mathcal{C}_{22}^T\mathcal{C}_{22}$ are Hermitian, we get the following lower bounds

$$\mathbf{x}^T T_0\mathbf{x} \geq \lambda_{\min}(T_0)|\mathbf{x}|^2 \tag{57}$$

$$\mathbf{x}^T T_1\mathbf{x} \geq \lambda_{\min}(T_1)|\mathbf{x}|^2 \tag{58}$$

and upper bound

$$\mathbf{x}^T\mathcal{C}_{22}^T\mathcal{C}_{22}\mathbf{x} \leq \lambda_{\max}(\mathcal{C}_{22}^T\mathcal{C}_{22})|\mathbf{x}|^2 \tag{59}$$

So the following condition is sufficient

$$\frac{1}{c\Delta_t}+\lambda_{\min}(T_0)+\lambda_{\min}(T_1)c\Delta_t > \sum_{k=2}^{\infty}\frac{(c\Delta_t)^{2k-1}}{16^k}\lambda_{\max}(\mathcal{C}_{22}^T\mathcal{C}_{22})^k \tag{60}$$

The l.h.s. increases without bound as $\Delta_t \to 0$, while the r.h.s. decreases to 0 in the same limit, so it is clearly possible to choose Δ_t small enough yet strictly positive. □

Theorem 1.B. *In theorem 1.A, we chose Δ_t small enough that $K_L\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)K_R$ is positive definite. This choice also guarantees that T_{mix} is diagonalizable.*

Proof. It follows from the proof of lemma 1 that the time-stepping operator is

$$T_{\text{mix}} = f\left(\left(\frac{I}{c\Delta_t} + \mathcal{H}_2\right)^{-1} \mathcal{H}_1\right) \tag{61}$$

where $f(M) = (I - M)^{-1}(I + M)$ and $\mathcal{H}_1, \mathcal{H}_2$ are given in (38) and (39). We make use of K_L, K_R defined by (46),(45)

$$\begin{aligned} T_{\text{mix}} &= f\left(\left(\frac{I}{c\Delta_t} + \mathcal{H}_2\right)^{-1} \mathcal{H}_1\right) \\ &= f\left(\left(\frac{I}{c\Delta_t} + \mathcal{H}_2\right)^{-1} K_L^{-1} K_L \mathcal{H}_1\right) \\ &= f\left(\left(K_L\left(\frac{I}{c\Delta_t} + \mathcal{H}_2\right)\right)^{-1} K_L \mathcal{H}_1\right) \\ &= f\left(K_R\left(K_L\left(\frac{I}{c\Delta_t} + \mathcal{H}_2\right) K_R\right)^{-1} K_L \mathcal{H}_1\right) \end{aligned} \tag{62}$$

Since $K_L\left(\frac{I}{c\Delta_t} + \mathcal{H}_2\right) K_R$ is positive definite by assumption, so is its inverse and the Cholesky decomposition $\left(K_L\left(\frac{I}{c\Delta_t} + \mathcal{H}_2\right) K_R\right)^{-1} = QQ^T$ exists.

$$\begin{aligned} T_{\text{mix}} &= f\left(K_R Q Q^T K_L \mathcal{H}_1 K_R Q Q^{-1} K_R^{-1}\right) \\ &= f\left((K_R Q) \quad \underbrace{Q^T K_L \mathcal{H}_1 K_R Q}_{\substack{\text{anti-symmetric matrix} \\ \text{therefore diagonalizable}}} \quad (K_R Q)^{-1}\right) \\ &= f\left(\underbrace{(K_R Q) Q^T K_L \mathcal{H}_1 K_R Q (K_R Q)^{-1}}_{\substack{\text{Similar to diagonalizable matrix} \\ \text{therefore diagonalizable}}}\right) \end{aligned} \tag{63}$$

Let $(K_R Q) Q^T K_L \mathcal{H}_1 K_R Q (K_R Q)^{-1} = PDP^{-1}$ with D diagonal,

$$\begin{aligned} T_{\text{mix}} &= f\left(PDP^{-1}\right) \\ &= (I - PDP^{-1})^{-1}(I + PDP^{-1}) \\ &= (P(I - D)P^{-1})^{-1}(P(I + D)P^{-1}) \\ &= (P(I - D)^{-1}P^{-1})(P(I + D)P^{-1}) \\ &= P \underbrace{(I - D)^{-1}(I + D)}_{\text{diagonal}} P^{-1} \end{aligned} \tag{64}$$

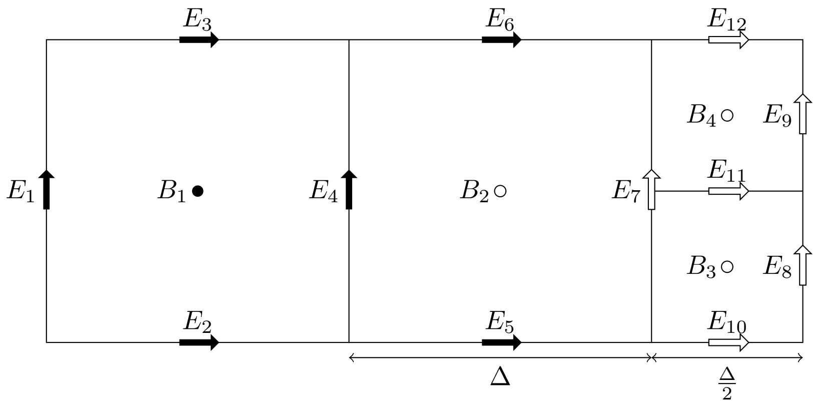

Figure 3: 1:2 spatial refinement in 2D, with coarse discretisation length Δ and fine discretisation length $\Delta/2$. Field discretisation points in solid black get 1-step updates, those in white with black outline get 2-step updates (including B_2).

(64) is the diagonalisation of T_{mix}. □

4.1. Discussion

Theorem 1.A guarantees that no exponentially growing modes can exist when $K_L\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)K_R$ is positive definite. Theorems 1.A and 1.B together guarantee the existence of a discrete conserved energy, and forbid the existence of polynomially growing modes.

We cannot stress enough that the condition (60) is sufficient for stability, but not necessary. Even the positive-definiteness of $K_L\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)K_R$ is sufficient rather than necessary. In practice, one can use Cholesky decomposition [14] or Gershgorin circles [7] to efficiently find a Δ_t at which $K_L\left(\frac{I}{c\Delta_t}+\mathcal{H}_2\right)K_R$ is positive definite, but this quantity may be only of theoretical interest, since it underestimates the true stability condition.

5. spatiotemporal refinement

The theory developed in the preceding sections does not depend on the specific form of the discrete curl operator $\mathcal{C}$. The time-stepping equation from which we started, (4), is general enough to handle spatially refined grids. All we have to do is construct $C, [\star_\epsilon]$ and $[\star_\mu^{-1}]$ for the refined grid. To do this, we use the theory of [1, 2], which is based on constructing Finite Element-like basis-functions on Yee cells of different sizes, and then assembling them using assembly operators that maintain the appropriate physical continuity conditions at the cell boundaries. We will not discuss this in detail here, instead we will just give the result for the 1 : 2 case, as in figure 3.

$$C = \begin{bmatrix} -\frac{1}{\Delta} & \frac{1}{\Delta} & -\frac{1}{\Delta} & \frac{1}{\Delta} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{\Delta} & \frac{1}{\Delta} & -\frac{1}{\Delta} & \frac{1}{\Delta} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{2}{\Delta} & \frac{2}{\Delta} & 0 & \frac{2}{\Delta} & -\frac{2}{\Delta} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{2}{\Delta} & 0 & \frac{2}{\Delta} & 0 & \frac{2}{\Delta} & -\frac{2}{\Delta} \end{bmatrix} \tag{65}$$

$$[\star_\epsilon] = \operatorname{diag}\left(\frac{\Delta^2}{2}, \frac{\Delta^2}{2}, \frac{\Delta^2}{2}, \Delta^2, \frac{\Delta^2}{2}, \frac{\Delta^2}{2}, \frac{3\Delta^2}{4}, \frac{\Delta^2}{8}, \frac{\Delta^2}{8}, \frac{\Delta^2}{8}, \frac{\Delta^2}{4}, \frac{\Delta^2}{8}\right) \tag{66}$$

$$[\star_\mu^{-1}] = \operatorname{diag}\left(\Delta^2, \Delta^2, \frac{\Delta^2}{4}, \frac{\Delta^2}{4}\right) \tag{67}$$

Since the mass matrices $[\star_\epsilon], [\star_\mu^{-1}]$ commute with $P(a,b), Q(a,b)$, it is not strictly necessary to transform them away using the change of basis (5-7): we can apply multi-rate time-stepping to (4) directly. We also need

$$\alpha_E = P(1,0) = \operatorname{diag}(1,1,1,1,1,1,0,0,0,0,0,0) \tag{68}$$

$$\alpha_B = Q(1,0) = \operatorname{diag}(1,0,0,0) \tag{69}$$

Note that the coarse B_z point next to the interface (B_2) has a 2-step update. We could also give it a 1-step update, but that decreases the maximum stable time step, as we shall see shortly. Coarse B_z points further from the interface, like B_1, should have 1-step updates.

We now have all we need to write the spatiotemporally refined time-stepping operator, according to the recipe of section 3.

1. Start from the fields $\mathbf{e_0}, \mathbf{b_0}$. Update the 1-step electric field points (time step Δ_t).

   $$\mathbf{e_1} = \mathbf{e_0} + c\Delta_t \alpha_E [\star_\epsilon]^{-1} C^T [\star_\mu^{-1}] \mathbf{b_0}$$

   Update the 2-step electric field points (time step $\Delta_t/2$).

   $$\mathbf{e_2} = \mathbf{e_1} + \frac{c\Delta_t}{2}(I - \alpha_E)[\star_\epsilon]^{-1} C^T [\star_\mu^{-1}] \mathbf{b_0}$$

2. Update the 2-step magnetic field points (time step $\Delta_t/2$).

$$\mathbf{b_1} = \mathbf{b_0} - \frac{c\Delta_t}{2}(I - \alpha_B)C\mathbf{e_2}$$

3. Update the 2-step electric field points (time step $\Delta_t/2$).

$$\mathbf{e_3} = \mathbf{e_2} + \frac{c\Delta_t}{2}(I - \alpha_E)[\star_\epsilon]^{-1}C^T[\star_\mu^{-1}]\mathbf{b_1}$$

4. Update the 2-step magnetic field points (time step $\Delta_t/2$).

$$\mathbf{b_2} = \mathbf{b_1} - \frac{c\Delta_t}{2}(I - \alpha_B)C\mathbf{e_3}$$

   Update the 1-step magnetic field points (time step Δ_t).

$$\mathbf{b_3} = \mathbf{b_2} - c\Delta_t\alpha_B C\mathbf{e_3}$$

   The updated fields are $\mathbf{e_3}, \mathbf{b_3}$.

5.1. Stability analysis

Since our proof of section 4 provides only a sufficient condition for stability, we must investigate the actual stability condition numerically. We will first investigate the stability of small configurations like figure 3, before moving to larger problems and numerical investigations of the long-term stability in the numerical examples.

For the configuration of figure 3, whether or not B_2 is given a 2-step update, the sufficient stability condition given in our proof, i.e. the value of $c\Delta_t/\Delta$ below which $K_L\left(\frac{I}{c\Delta_t} + \mathcal{H}_2\right)K_R$ becomes positive definite, is $c\Delta_t < \Delta/\sqrt{3}$, about 1.6 times higher than the stability condition for purely spatial refinement, $c\Delta_t < \Delta/(2\sqrt{2})$.

The characteristic polynomial of the time-stepping operator for figure 3 is a palindromic polynomial with real coefficients, of degree 16. This may be difficult to show in general, but is easy to check for the case of figure 3. Palindromicity is common among characteristic polynomials of time-reversible systems [11]: it guarantees that if $\exp(i\omega\Delta_t)$ is an eigenvalue, so is $\exp(-i\omega\Delta_t)$.

Let this characteristic polynomial be $f(\lambda)$ and have degree $d = 16$, so the eigenvalues λ are the points where $f(\lambda) = 0$. Since $f(\lambda)$ is palindromic of even degree, it can be written as $f(\lambda) = \lambda^{d/2} g(\lambda + \lambda^{-1})$, where g is a polynomial of degree $d/2 = 8$. Roots of f on the unit circle correspond to real roots of g in $[-2, 2]$. Sturm's theorem [17] then allows us to count the number of roots of g in $[-2, 2]$, and thereby the number of eigenvalues on the unit circle. All this can be done with exact arithmetic. The number of unit-circle eigenvalues vs. $\frac{c\sqrt{2}\Delta_t}{\Delta}$ is plotted in figure 4. We see that the time-stepping operator becomes unstable before $\frac{c\sqrt{2}\Delta_t}{\Delta} = 1$, the coarse stability condition. The case where B_2 gets 2-step updates has a slightly higher maximum stable Δ_t than the case where B_2 gets 1-step updates, and becomes unstable only above $\frac{c\sqrt{2}\Delta_t}{\Delta} = 0.9$.

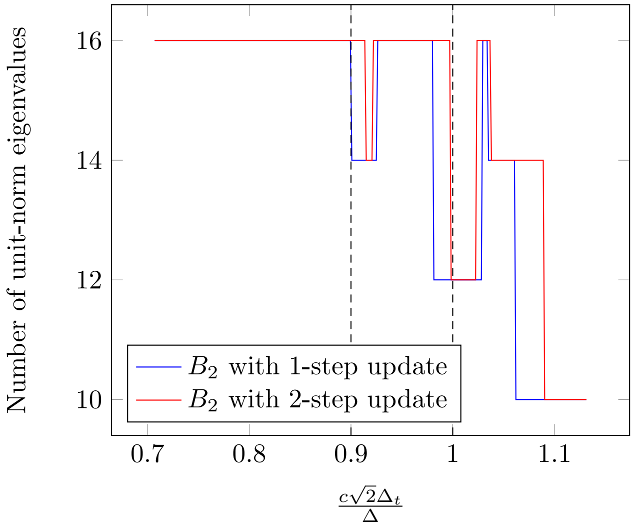

Figure 4: Number of eigenvalues of the time-stepping operator that lie on the unit circle vs. $c\sqrt{2}\Delta_t/\Delta$ for the configuration of figure 3. If the number of eigenvalues on the unit circle is less than 16 (the number of degrees of freedom for this configuration), it is unstable. The vertical black dashed lines indicate the coarse stability limit $c\sqrt{2}\Delta_t/\Delta = 1$ and the reduced stability limit $c\sqrt{2}\Delta_t/\Delta = 0.9$

6. Numerical examples

6.1. Reflection at the spatiotemporal refinement interface

In figure 5, we show the numerical reflection at a 1:2 spatiotemporal refinement interface. We let cylindrical wavefronts propagate through a spatiotemporal refinement interface. The reflection is then defined as

$$\frac{\int |B_{z,\text{with refinement}} - B_{z,\text{without refinement}}| dV}{\int |B_{z,\text{without refinement}}| dV} \tag{70}$$

and is plotted in figure 5. The reflection decreases roughly as $(\lambda/\Delta)^{-1.4}$ (where λ is the wavelength).

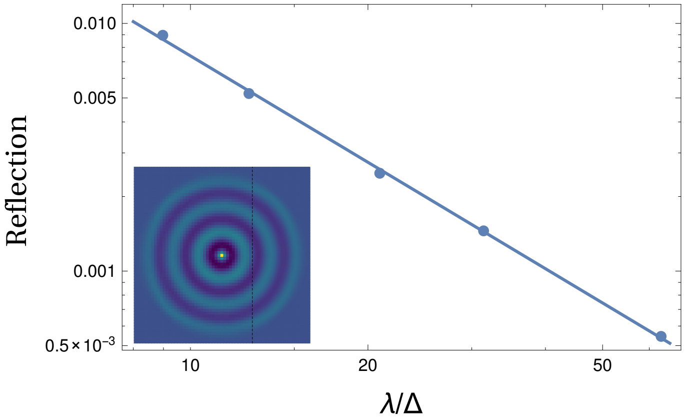

Figure 5: Reflection at a 1:2 spatiotemporal refinement interface. The reflection plotted is $\frac{\int |B_{z,\text{with refinement}} - B_{z,\text{without refinement}}| dV}{\int |B_{z,\text{without refinement}}| dV}$. It is small for well-resolved waves, and decreases as $(\lambda/\Delta)^{-1.4}$.

6.2. Recursive refinement: resolving a small cylinder

6.2.1. A perfect conductor

As stated in section 3, our refinement scheme can be used recursively to obtain spatiotemporal refinement by an arbitrary power of 2. To illustrate this, we consider wave scattering around a small cylinder. The source frequency is chosen such that the wavelength $\lambda = 10\Delta$, the cylinder radius is 3Δ. A 128-fold recursively refined grid is shown in figure 6.

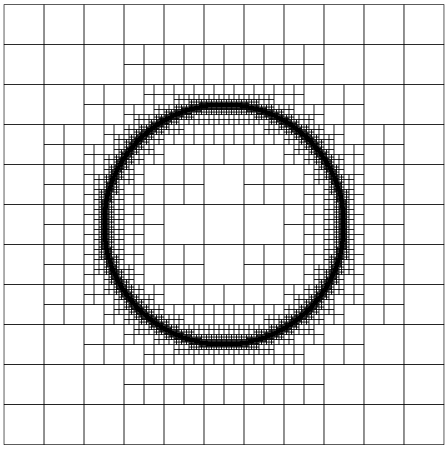

Figure 6: 128-fold recursive refinement around a cylinder.

We first consider a perfectly conducting cylinder, shown in figure 7. Waves are excited by an $E_y = \sin(\omega t)$ boundary condition on the left, and have not had time to propagate all the way to the right. In the non-refined case, the cylinder is very ill-resolved, which leads to so-called "staircasing error" [15]. On the refined grid, the cylinder is represented much more accurately.

6.2.2. Long-term stability

The stability condition for 2^n-fold *spatial* refinement in 2D, constructed according to the method of [2, 1], is

$$c\Delta_t < \left(\frac{1}{2}\right)^n \frac{\Delta_{\text{coarse}}}{\sqrt{2}} \tag{71}$$

For spatiotemporal refinement, we expect it to be

$$c\Delta_t < \frac{\chi^n \Delta_{\text{coarse}}}{\sqrt{2}} \tag{72}$$

If $\chi = 1$, the refinement does not influence the time step at all. As long as $\chi > 1/2$, the spatiotemporal refinement allows larger time steps than the spatial refinement. If $\chi \leq 1/2$, the spatiotemporal refinement is useless.

In numerical experiments, we have found $\chi \approx 0.9$ which agrees with the results of section 5.1. We initialized the fields around the PEC cylinder with

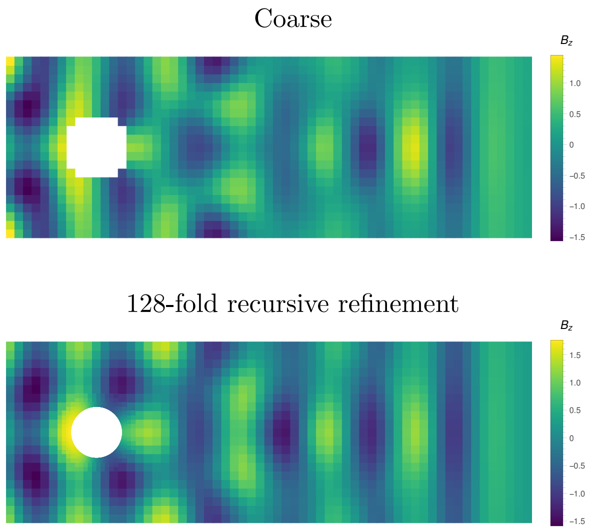

Figure 7: B_z field around an ill-resolved perfectly conducting cylinder. Local $2^7 = 128$-fold spatiotemporal refinement allows us to properly resolve the cylinder. Both figures were generated with $c\Delta_t = \frac{0.9^7 \Delta_{\text{coarse}}}{\sqrt{2}}$, and 180 time steps.

random values at $t = 0$, and ran the 2^7-fold refined configuration for $\approx 10^5$ coarse steps, or $\approx 10^7$ fine steps. The total energy is shown in figure 8.

$$\text{energy} = \frac{1}{8\pi} \int_{\text{volume}} (E^2 + B^2) dV \tag{73}$$

This energy is not exactly conserved by FDTD [1, 19, 18] (not even by non-refined FDTD), but it does oscillate around a constant value when the discretisation (and the underlying physics) is conservative. This is what we see in figure 8. No instability is observed.

Suppose we want a local spatial refinement by a factor $1/S$ (S is a power of 2). To do this in a stable way, we need to reduce the global time step by a factor $1/T$. For purely spatial refinement, $S = T$. For the spatio-temporal refinement considered here, $T = 0.9^{-\log_2(S)} \approx S^{0.15}$, so T increases sublinearly with S.

6.2.3. Finite conductivity

We can do the same with a cylinder of finite conductivity. Strictly speaking, the theory of the preceding sections does not apply to lossy materials. We will

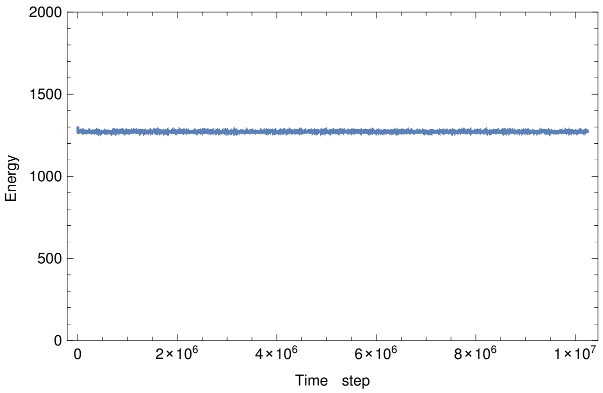

Figure 8: Energy vs. fine time step for the 2^7-fold refined configuration shown in figure 7. The fields around the PEC cylinder were initialized with random values at $t = 0$. With $c\Delta_t = \frac{0.9^7\Delta_{\text{coarse}}}{\sqrt{2}}$, the energy in the simulation region is conserved, no signs of instability are observed.

consider finite conductivity nonetheless, since introducing losses is unlikely to reduce the maximum stable time step, and resolving the skin depth in conductors is one of the interesting applications of spatiotemporal FDTD refinement. To include conductivity, we need to modify the electric update step. Ampère's law with conductivity σ (so $J = \sigma E$) is

$$\frac{1}{c}\left(4\pi\sigma + \frac{\partial}{\partial t}\right)E = \nabla \times B \tag{74}$$

To discretize this, let ς be a diagonal matrix such that $\varsigma_{ii} = \sigma$ at electric field discretisation point i, and let $\varepsilon_r = 1, \mu_r = 1$:

$$\frac{1}{c}4\pi\varsigma\frac{\mathbf{e}_+ + \mathbf{e}_-}{2} + \frac{1}{c}\frac{\mathbf{e}_+ - \mathbf{e}_-}{\Delta_t} = C^T\mathbf{b}_- \tag{75}$$

$$\left(\frac{2\pi}{c}\varsigma + \frac{I}{c\Delta_t}\right)\mathbf{e}_+ = C^T\mathbf{b}_- - \frac{1}{c}2\pi\varsigma\mathbf{e}_- + \frac{1}{c}\frac{\mathbf{e}_-}{\Delta_t} \tag{76}$$

so

$$\mathbf{e}_+ = \left(\frac{2\pi}{c}\varsigma + \frac{I}{c\Delta_t}\right)^{-1}\left(C^T\mathbf{b}_- + \left(\frac{I}{c\Delta_t} - \frac{2\pi}{c}\varsigma\right)\mathbf{e}_-\right) \tag{77}$$

These updates can be used with the algorithm of section 3.

The skin depth δ is

$$\delta = \frac{c}{\sqrt{2\pi\sigma\omega\mu}} \tag{78}$$

We choose the conductivity of the cylinder such that $\delta = \frac{\Delta}{10}$. The 128-fold local spatiotemporal refinement then allows us to resolve not only the cylinder itself, but also its skin depth. This is shown in figures 9 and 10.

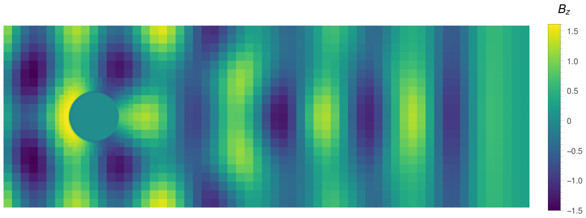

Figure 9: B_z field around cylinder with finite conductivity. 128-fold local spatiotemporal refinement allows us to properly resolve the cylinder.

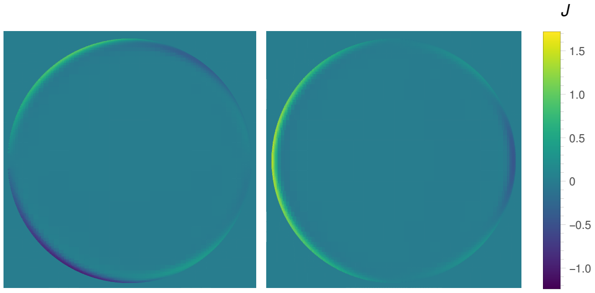

Figure 10: Current components J_x (left) and J_y (right) in the cylinder of figure 9. The currents are nonzero only in the neighbourhood of the surface, and decay on the length scale of the skin depth δ. The local refinement allows us to resolve even this length scale.

6.3. Runtime

Using spatiotemporal refinement, the amount of time spent calculating coarse updates decreases, since we can use larger time steps ($\frac{0.9^n \Delta_{\text{coarse}}}{c\sqrt{2}}$ vs. $\frac{0.5^n \Delta_{\text{coarse}}}{c\sqrt{2}}$). The amount of time spent calculating fine updates increases, since we have to use smaller time steps ($\frac{0.9^n 0.5^n \Delta_{\text{coarse}}}{c\sqrt{2}}$ vs. $\frac{0.5^n \Delta_{\text{coarse}}}{c\sqrt{2}}$). Thus, an appreciable decrease in runtime occurs when the runtime is dominated by the coarse grid (i.e. there are many more coarse points than fine points). On a grid containing a PEC cylinder with 2^5-fold local spatiotemporal refinement (figure 11), with 30552 discretisation points at the coarsest level and 5352 discretisation points at the finest level, a calculation with spatiotemporal refinement took 5s, one with purely spatial refinement took 12s.

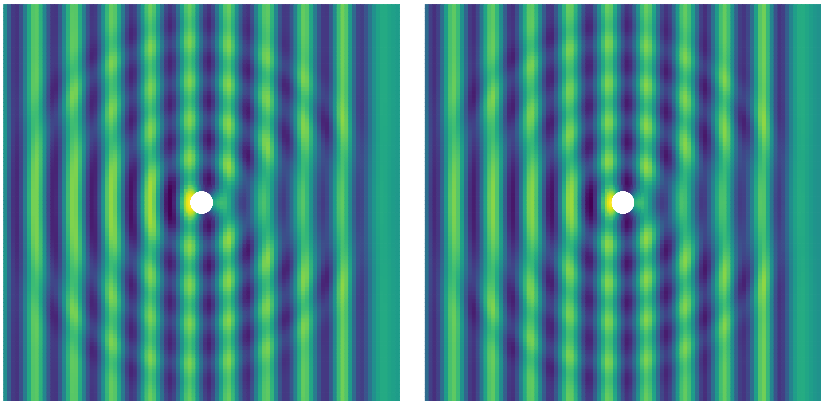

Figure 11: B_z for a plane wave scattering at a cylinder, calculated with purely spatial refinement (left, runtime 12s) and spatiotemporal refinement (right, runtime 5s). 2^5-fold local refinement is used to resolve the cylinder.

6.4. Comparison with semi-analytical calculations

Consider a PEC cylinder of radius R_0, centered at $(0,0)$. It is surrounded by a square cavity of width $W > 2R_0$, as in figure 12.

B_z in frequency domain around a PEC cylinder in empty space is of the form

$$B_z = \sum_{n=-\infty}^{\infty} c_n \left(J_n\left(\frac{r\omega}{c}\right) - \frac{Y_n\left(\frac{r\omega}{c}\right)\left(J_{n-1}\left(\frac{R_0\omega}{c}\right) - J_{n+1}\left(\frac{R_0\omega}{c}\right)\right)}{Y_{n-1}\left(\frac{R_0\omega}{c}\right) - Y_{n+1}\left(\frac{R_0\omega}{c}\right)} \right) \exp(in\theta) \tag{79}$$

where the c_n are unknown coefficients, r, θ are coordinates in a cylindrical coordinate system, and J_n, Y_n are Bessel functions.

In order to obtain eigenmodes of the square cavity, we want to choose the

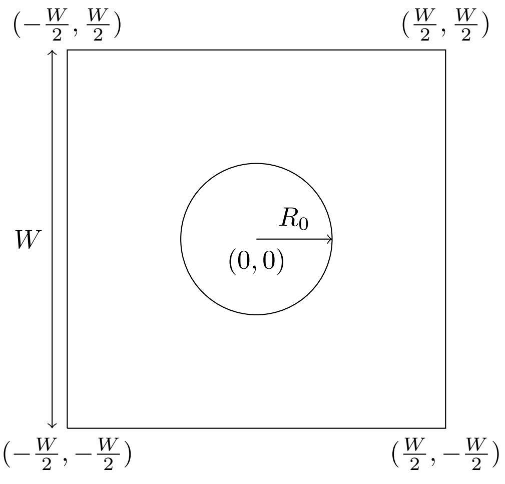
$(-\frac{W}{2},\frac{W}{2})$
$(\frac{W}{2},\frac{W}{2})$
W
R_0
$(0,0)$
$(-\frac{W}{2},-\frac{W}{2})$
$(\frac{W}{2},-\frac{W}{2})$

Figure 12: A square cavity that contains a PEC cylinder at its center.

c_n such that B_z obeys the PEC boundary condition at the square cavity walls

$$\frac{\partial B_z}{\partial x} = 0 \qquad \text{at the vertical walls } x = \pm W/2$$
$$\frac{\partial B_z}{\partial y} = 0 \qquad \text{at the horizontal walls } y = \pm W/2$$

Naturally we cannot solve this analytically. But we can pick a finite number of modes (so $-n_{\max} \le n \le n_{\max}$) and a finite number of points on the square cavity walls at which we impose the PEC boundary condition (say n_p points, $n_p \gg 2n_{\max}$). We can solve the resulting overdetermined $n_p \times (2n_{\max}+1)$ linear system of equations, in a least-squares sense, for the coefficients c_n. If the error on the least-squares solution is large, ω is not an eigenfrequency. If it is small, ω is approximately an eigenfrequency. A comparison of the thus-calculated eigenfrequencies and those calculated using spatiotemporally refined FDTD is shown in figure 13.

We choose $R_0 = 6\Delta_{\text{coarse}}$, $W = 101\Delta_{\text{coarse}}$, 2^5-fold refinement for the cylinder, and $c\Delta_t = \frac{0.9^5\Delta_{\text{coarse}}}{\sqrt{2}}$. After running spatiotemporally refined FDTD for $2\cdot10^4$ coarse steps, we Fourier transform B_z to find the eigenmodes numerically. We compare this with semi-analytical eigenmodes determined by the procedure described above, and find excellent agreement in figure 14.

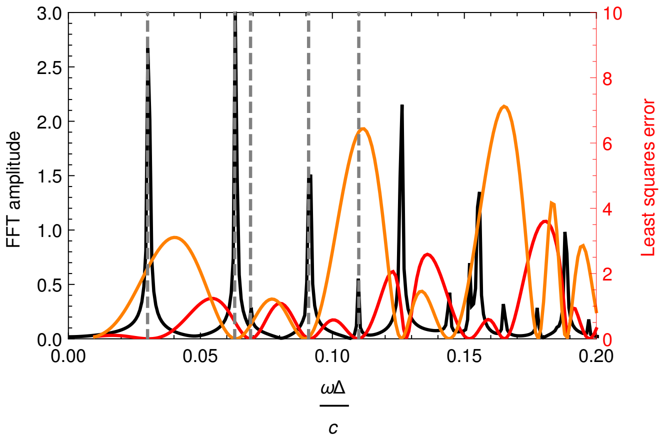

Figure 13: Eigenfrequencies of the square cavity with a cylinder from figure 12. Black: eigenfrequencies correspond to the *peaks* of the Fourier spectrum of the fields computed with spatiotemporally refined FDTD. Red, orange: eigenfrequencies correspond to *minima* of the least-squares error for the solution of the overconstrained system of equations that must be solved to determine c_n in the semi-analytical solution. (the red curve corresponds to eigenmodes that are symmetric w.r.t. the vertical axis, the orange curve to eigenmodes that are anti-symmetric w.r.t. the vertical axis). Vertical gray dashed lines show that the peaks and minima indeed occur at the same frequencies.

7. Conclusion

In this paper we developed an explicit and provably conditionally stable spatiotemporal FDTD refinement scheme. This enables truly local FDTD refinement, with a local discretisation length and a local time step. To our knowledge, this is the first such scheme that is provably conditionally stable. The (sufficient, provable) stability condition for 1:2 spatiotemporal refinement is (see (49))

$$K_L \left(\frac{I}{c\Delta_t} + \mathcal{H}_2 \right) K_R \text{ is positive definite} \tag{80}$$

The stability condition that seems to be valid in practice for 1:2^n refinement in 2D is

$$c\Delta_t < \frac{0.9^n \Delta_{\text{coarse}}}{\sqrt{2}} \tag{81}$$

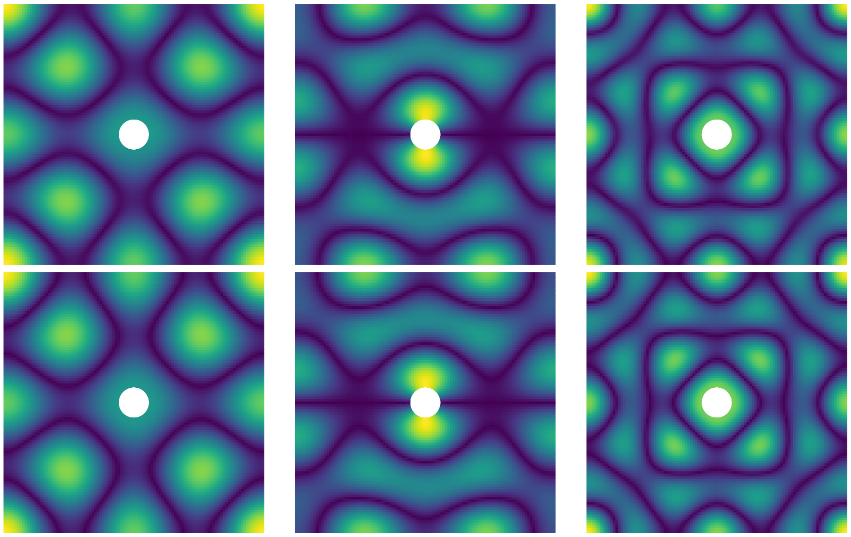

Figure 14: $|B_z|$ for three eigenmodes of a square cavity with a PEC cylinder at its center. Top row: calculated with a semi-analytical approach. Bottom row: calculated by Fourier transforming the fields calculated by spatiotemporally refined FDTD. (note: to make this plot we sampled the semi-analytical solution at the same coordinates as the FDTD solution, so the sampling is denser around the cylinder in both cases)

We have given 2-dimensional examples, but our stability proof makes no assumptions about the dimensionality: it remains valid in 3D.

[1] R. A. Chilton. *H-, P- and T-refinement strategies for the finite-difference-time-domain (FDTD) method developed via finite-element (FE) principles.* PhD thesis, Ohio State University, 2008.

[2] R. A. Chilton and R. Lee. Conservative and provably stable FDTD subgridding. *IEEE Transactions on Antennas and Propagation*, 55(9):2537–2549, Sep 2007.

[3] P. Chow, Tetsuyuki Kubota, and Takefumi Namiki. A stable FDTD subgridding method for both spatial and temporal spaces. In *2008 IEEE Antennas and Propagation Society International Symposium*, pages 1–4, July 2008.

[4] Francis Collino, Thierry Fouquet, and Patrick Joly. Conservative space-time mesh refinement methods for the FDTD solution of Maxwells equations. *Journal of Computational Physics*, 211(1):9–35, 2006.

[5] Abdelaziz Ezziani and Patrick Joly. Local time stepping and discontinuous Galerkin methods for symmetric first order hyperbolic systems. *Journal of Computational and Applied Mathematics*, 234(6):1886 – 1895, 2010. Eighth International Conference on Mathematical and Numerical Aspects of Waves (Waves 2007).

[6] C. Fumeaux, D. Baumann, P. Leuchtmann, and R. Vahldieck. A generalized local time-step scheme for efficient FVTD simulations in strongly inhomogeneous meshes. *IEEE Transactions on Microwave Theory and Techniques*, 52(3):1067–1076, 2004.

[7] S. Gershgorin. Über die Abgrenzung der Eigenwerte einer Matrix. *Izv. Akad. Nauk. USSR Otd. Fiz.-Mat. Nauk*, 1931.

[8] Lukasz Kulas and Michal Mrozowski. Reciprocity principle for stable subgridding in the finite difference time domain method. In *EUROCON, 2007. The International Conference on" Computer as a Tool"*, pages 106–111. IEEE, 2007.

[9] A. Van Londersele, D. De Zutter, and D. Vande Ginste. A new hybrid implicit-explicit FDTD method for local subgridding in multiscale 2D-TE scattering problems. *IEEE Transactions on Antennas and Propagation*, PP(99):1–1, 2016.

[10] Tzon-Tzer Lu and Sheng-Hua Shiou. Inverses of 2 × 2 block matrices. *Computers and Mathematics with Applications*, 43(1):119 – 129, 2002.

[11] I. Markovsky and S. Rao. Palindromic polynomials, time-reversible systems, and conserved quantities. *16th Mediterranean Conference on Control and Automation*, 2008.

[12] Emmanuel Montseny, Sébastien Pernet, Xavier Ferrières, and Gary Cohen. Dissipative terms and local time-stepping improvements in a spatial high order discontinuous Galerkin scheme for the time-domain Maxwell's equations. *Journal of Computational Physics*, 227(14):6795–6820, 2008.

[13] Y. Nakazono and H. Asai. Multi-rate FDTD method for fast electromagnetic simulation. In *2007 IEEE Electrical Performance of Electronic Packaging*, pages 283–286, Oct 2007.

[14] Siegfried M Rump. Verification of positive definiteness. *BIT Numerical Mathematics*, 46(2):433–452, 2006.

[15] T. Rylander and A. Bondeson. Stability of explicit–implicit hybrid time-stepping schemes for Maxwell's equations. *Journal of Computational Physics*, 179(2):426–438, 2002.

[16] A. Taflove and S. Hagness. *Computational Electrodynamics: The Finite-Difference Time-Domain Method.* Artech House, 2005.

[17] J. M. Thomas. Sturm's theorem for multiple roots. *National Mathematics Magazine*, 15(8):391–394, 1941.

[18] W. Tierens. Higher-order hybrid implicit/explicit FDTD time-stepping. *Journal of Computational Physics*, 327:643–652, 2016.

[19] W. Tierens. Unification of leapfrog and Crank-Nicolson Finite Difference Time Domain methods. *SIAM journal on Scientific Computing*, 2018.

[20] Arne Van Londersele, Daniël De Zutter, and Dries Van de Ginste. An in-depth stability analysis of nonuniform FDTD combined with novel local implicitization techniques. *Journal of Computational Physics*, 342:177–193, 2017.

[21] Arne Van Londersele, Daniël De Zutter, and Dries Van de Ginste. Provably stable local application of crank-nicolson time integration to the FDTD

method with nonuniform gridding and subgridding. *2018 International Applied Computational Electromagnetics Society Symposium*, 2018.

[22] Xiao-Kun Wei, Nectaria Diamanti, Xingqi Zhang, A Peter Annan, and Costas D Sarris. Spatially-filtered FDTD subgridding for ground penetrating radar numerical modeling. In *Advanced Ground Penetrating Radar (IWAGPR), 2017 9th International Workshop on*, pages 1–4. IEEE, 2017.

[23] Kai Xiao, David J Pommerenke, and James L Drewniak. A three-dimensional FDTD subgridding algorithm with separated temporal and spatial interfaces and related stability analysis. *IEEE Transactions on Antennas and Propagation*, 55(7):1981–1990, 2007.

[24] Dan Yan, Jin; Jiao. An unsymmetric FDTD subgridding algorithm with unconditional stability. *IEEE Transactions on Antennas and Propagation*, 2018.

[25] S. S. Zivanovic, K. S. Yee, and K. K. Mei. A subgridding method for the time-domain finite-difference method to solve Maxwell's equations. *IEEE Transactions on Microwave Theory and Techniques*, 39(3):471–479, Mar 1991.