UNIFICATION OF LEAPFROG AND CRANK-NICOLSON FDTD *

WOUTER TIERENS†

Abstract. In this paper we present a unified theory of explicit (leapfrog) and implicit (Crank-Nicolson) FDTD time-stepping schemes. This enables us to interface leapfrog FDTD with Crank-Nicolson FDTD and we will prove that the result is stable at the Courant limit of the leapfrog part. It also enables us to construct explicit FDTD-like algorithms whose stability condition is less restrictive than that of leapfrog FDTD, and a remarkable explicit 1:2 FDTD refinement scheme which is stable up to the Courant limit of the coarse part.

Key words. FDTD, Maxwell's equations, stability

AMS subject classifications. 35Q61

1. Introduction.

The Finite Difference Time Domain (FDTD) algorithm is an extremely well-known and popular computational method for solving Maxwell's equations [14], for calculating the behaviour of electromagnetic fields in time domain. The FDTD algorithm has a stability condition, the Courant condition[14], which puts an upper bound on the time step. Usually $c\Delta_t < \Delta/\sqrt{d}$ where c is the speed of light, Δ_t is the time step, Δ the discretisation length, and d the number of spatial dimensions.

This stability criterion can be restrictive, as it forces us to choose a small Δ_t and thus to run a very large number of time steps to simulate a given length of time. This is especially so when the phenomena of interest are waves that move much slower than c, but the time step still needs to be chosen based on c [16], when small features, much smaller than the wavelength, need to be resolved [17, 2, 3, 8], or when the simulation region has an extreme aspect ratio [18].

For this reason, many attempts have been made to relax or circumvent the stability condition [1, 20, 12]. One well-known, but not very practical, approach is Crank-Nicolson FDTD [11, 13] a.k.a. FDTD time-stepping with the implicit midpoint rule [9]. Unlike standard explicit "leapfrog" FDTD, Crank-Nicolson FDTD is unconditionally stable. It is much less popular than FDTD because its time-stepping operation is much more complex (it requires the solution of a large sparse set of equations every time step), and because increasing the time step Δ_t still decreases the accuracy.

In this paper, we will show the following

1. Leapfrog FDTD and Crank-Nicolson FDTD are both special cases of a single more general time stepping algorithm, which is itself at least as stable as leapfrog FDTD.
2. Leapfrog FDTD and Crank-Nicolson FDTD can be interfaced in a stable and energy-conserving way, answering a question posed in [9]:"the energy-preserving coupling of the leapfrog method and on the implicit midpoint-rule remains a non-obvious question". This is provably stable at the stability condition of the explicit part.
3. There exist intermediate algorithms: they are stabler than leapfrog FDTD but require only the solution of constant-sized sets of equations.
4. These intermediate algorithms can be interfaced with FDTD in a stable and energy-conserving way, which enables the construction of an explicit FDTD

*Submitted to the editors 12 June 2016.

†Max Planck Institut für Plasmaphysik (wouter.tierens@ipp.mpg.de).

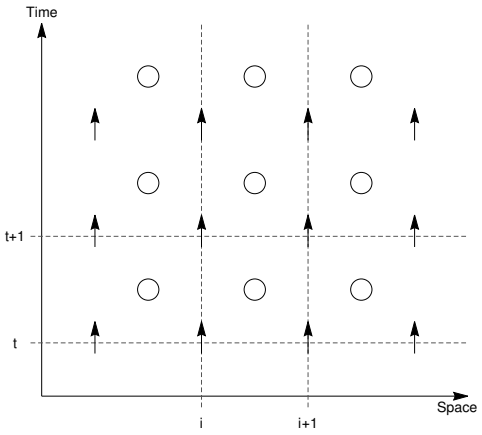

FIG. 1. *Spatial and temporal structure of leapfrog FDTD, as used in eqs. (5,6). Arrows represent E_y, circles represent B_z. This structure is spatially and temporally staggered.*

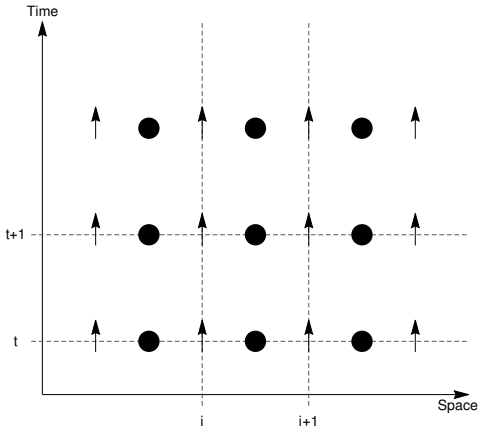

FIG. 2. *Spatial and temporal structure of Crank-Nicolson FDTD, as used in eqs. (7,8). Arrows represent E_y, circles represent B_z. This structure is spatially staggered but temporally collocated.*

refinement scheme that remains stable at the stability condition of the coarse grid, which is long-sought-after [2, 5, 7].

In section 2, we give a brief introduction of leapfrog and Crank-Nicolson FDTD. In section 3, we provide an overview of the general leapfrog time-stepping equation of [19, 2], and the techniques used to prove its stability. Then in section 4, we prove the conditional stability of a general time-stepping operation which is general enough to include both leapfrog and Crank-Nicolson FDTD. Finally in section 5, we give some numerical examples of the various algorithms that fit within our general framework.

2. Maxwell's equations and FDTD. The partial differential equations which FDTD (approximately) solves are Maxwell's equations:

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

$$\nabla \times \left(\mu_r^{-1}\mathbf{B}\right) = \frac{1}{c}\epsilon_r \frac{\partial \mathbf{E}}{\partial t} \tag{2}$$

where $\mathbf{E}$ is the electric field, $\mathbf{B}$ is the magnetic field, c is the speed of light, and μ_r and ϵ_r are the relative permeability and permittivity (assumed real and positive). When we need to refer to specific components of the electric or magnetic field, we will call them $E_x, E_y, E_z, B_x, B_y, B_z$.

For this brief introduction, let us consider the one-dimensional case where E and B depend on x and t, but not on y and z:

$$\frac{\partial E_y}{\partial x} = -\frac{1}{c}\frac{\partial B_z}{\partial t} \tag{3}$$

$$-\frac{\partial B_z}{\partial x} = \frac{\mu_r \epsilon_r}{c}\frac{\partial E_y}{\partial t} \tag{4}$$

2.1. Leapfrog FDTD. In leapfrog FDTD, the field components E_y and B_z are discretized in space and in time as shown in figure 1. We say that E_y is discretized

at integer time points and B_z at half-integer time points, or that they are temporally staggered. (3) and (4) are then discretized as follows:

$$\frac{E^t_{y,i+1} - E^t_{y,i}}{\Delta} = -\frac{1}{c}\frac{B^{t+1/2}_{z,i+1/2} - B^{t-1/2}_{z,i+1/2}}{\Delta_t} \tag{5}$$

$$-\frac{B^{t+1/2}_{z,i+1/2} - B^{t+1/2}_{z,i-1/2}}{\Delta} = \frac{\mu_r \epsilon_r}{c}\frac{E^{t+1}_{y,i} - E^t_{y,i}}{\Delta_t} \tag{6}$$

Repeatedly applying (5) and (6) lets us time step the fields into the future, predicting future values of E_y and B_z given their present values.

Throughout this paper we will colour discretized field points known at integer multiples of Δ_t in black and those known at half-integer multiples of Δ_t in white with black outline, as we did in figure 1.

In general the spatial discretisation grid usually consists of Yee cells [14], although more flexible grid constructions are possible [2]. In this paper we make no assumptions on the precise nature of the spatial grid.

2.2. Crank-Nicolson FDTD. In Crank-Nicolson FDTD, the E_y and B_z points are temporally collocated, but the spatial discretisation remains the same (figure 2). Because of this, the central-difference derivative in the temporal direction must be accompanied by an interpolation:

$$\frac{E^t_{y,i+1} - E^t_{y,i}}{\Delta} + \frac{E^{t+1}_{y,i+1} - E^{t+1}_{y,i}}{\Delta} = -\frac{2}{c}\frac{B^{t+1}_{z,i+1/2} - B^t_{z,i+1/2}}{\Delta_t} \tag{7}$$

$$-\left(\frac{B^t_{z,i+1/2} - B^t_{z,i-1/2}}{\Delta} + \frac{B^{t+1}_{z,i+1/2} - B^{t+1}_{z,i-1/2}}{\Delta}\right) = \frac{2\mu_r \epsilon_r}{c}\frac{E^{t+1}_{y,i} - E^t_{y,i}}{\Delta_t} \tag{8}$$

This algorithm is considerably more complicated than leapfrog FDTD. (7) and (8) must be solved simultaneously (unlike (5) and (6)), requiring the solution of a large sparse system of equations every time step.

3. General leapfrog time-stepping. A general leapfrog time-stepping equation is of the following form [19, 2]:

$$\frac{1}{c}\frac{\begin{bmatrix}\mathbf{e}_+\\ \mathbf{b}_+\end{bmatrix} - \begin{bmatrix}\mathbf{e}_-\\ \mathbf{b}_-\end{bmatrix}}{\Delta_t} = \begin{bmatrix}0 & [\star_\epsilon]^{-1}C^\dagger[\star_{\mu^{-1}}]\\ 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}_-\\ \mathbf{b}_-\end{bmatrix} + \begin{bmatrix}0 & 0\\ -C & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}_+\\ \mathbf{b}_+\end{bmatrix} \tag{9}$$

where $[\mathbf{e}_+, \mathbf{b}_+]^T$ are the discretized electric and magnetic field components at the next step and $[\mathbf{e}_-, \mathbf{b}_-]^T$ are the discretized field components at the current step. C is a discrete curl operator (usually obtained from finite differences). C is typically parametrized by a spatial discretisation length Δ such that $\lim_{\Delta\to 0} C = \nabla\times$.

$[\star_\epsilon]$ and $[\star_{\mu^{-1}}]$ are symmetric and positive definite (typically diagonal) mass matrices representing (position-dependent) electric permittivity ϵ and inverse magnetic permeability μ^{-1}.

The $\dagger$ as in $C^\dagger$ indicates the conjugate transpose. In practice C is usually real, and we will use the transpose C^T when that is the case. The arguments in this section work just as well for complex C, and we will need this in one of our proofs later on, which is why we use the conjugate transpose here.

The time-stepping equation (9) is already a powerful general framework. Classical FDTD, spatially refined FDTD, higher-order FDTD, in uniform or non-uniform

media, can all be brought under the form (9), and they can all be proven to be conditionally stable because (9) is provably conditionally stable. Though this paper focusses on the FDTD context, (9) is not limited to that context, and the results we will obtain are valid for all methods that can be brought under the form (9), including Finite Element and Discontinuous Galerkin methods, provided their mass matrices obey certain commutation relations.

3.1. The time-stepping operator and its eigenvalues. We can rewrite (9) as follows:

$$\begin{bmatrix}\mathbf{e}_+\\ \mathbf{b}_+\end{bmatrix} = \underbrace{\left(\frac{I}{c\Delta_t} - \begin{bmatrix}0 & 0\\ -C & 0\end{bmatrix}\right)^{-1}\left(\frac{I}{c\Delta_t} + \begin{bmatrix}0 & [\star_\epsilon]^{-1}C^\dagger[\star_{\mu^{-1}}]\\ 0 & 0\end{bmatrix}\right)}_{T_{\text{leapfrog}}}\begin{bmatrix}\mathbf{e}_-\\ \mathbf{b}_-\end{bmatrix} \tag{10}$$

The linear operator T_{leapfrog} which maps the past fields onto the future fields is the so-called time-stepping operator associated with (9). The eigenvalues of this operator are key to understanding the long-term behaviour of algorithms based on (9).

- If some eigenvalues λ lie inside the unit circle ($|\lambda| < 1$), the algorithm has unphysical losses. There are decaying modes where the continuous physics is lossless.
- If some eigenvalues λ lie outside the unit circle ($|\lambda| > 1$), there are modes which exhibit unphysical growth. The algorithm is unstable.

Only if the eigenvalues of T_{leapfrog} all lie on the unit circle (all $|\lambda| = 1$) is it possible for these algorithms to be both stable and energy-conserving. This condition is necessary, although (in very rare cases) it is not sufficient [4]. Understanding the eigenvalues of a class of time-stepping operators which are more general than (10) will be the central focus of this paper.

3.2. Conditional stability.

3.2.1. The Fourier method. The most common way in which FDTD can be shown to be stable is the Fourier method [14], in which the time-stepping equations are written for a uniform unbounded space. Complex exponential solutions with wavevector $\vec{k}$ and frequency ω are inserted into the time-stepping equations. Demanding that a nontrivial solution exists then gives a discrete dispersion relation $f(\vec{k}, \omega) = 0$. If this dispersion relation has real ω for all real $\vec{k}$, the algorithm is stable.

This method can only show stability in uniform media and works best on regular Cartesian grids. It is typically assumed that stability in uniform media implies stability in non-uniform media as well, and exceptions to this rule are very rare. However, there are more powerful proof techniques that can show stability in the non-uniform case as well.

3.2.2. Stability in the non-uniform case. In the general non-uniform case, stability of (9) can be shown using a relation between its eigenvalues and eigenvectors and those of the space-discrete, time-continuous system [19, 2].

THEOREM 1. *The eigenvalues of $T_{leapfrog}$ lie on the unit circle, provided Δ_t is sufficiently small.*

Proof. Eigenvalues λ and eigenvectors $[\mathbf{e},\mathbf{b}]^T$ of T_{leapfrog} obey

$$\begin{aligned}\frac{\lambda-1}{c\Delta_t}\begin{bmatrix}\mathbf{e}\\ \mathbf{b}\end{bmatrix} &= \begin{bmatrix}0 & [\star_\epsilon]^{-1}C^\dagger[\star_{\mu^{-1}}]\\ 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}\\ \mathbf{b}\end{bmatrix} + \lambda\begin{bmatrix}0 & 0\\ -C & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}\\ \mathbf{b}\end{bmatrix}\\ &= \begin{bmatrix}0 & [\star_\epsilon]^{-1}C^\dagger[\star_{\mu^{-1}}]\\ -\lambda C & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}\\ \mathbf{b}\end{bmatrix}\end{aligned} \tag{11}$$

which can be seen by substituting $[\mathbf{e}_-,\mathbf{b}_-]^T=[\mathbf{e},\mathbf{b}]^T$, $[\mathbf{e}_+,\mathbf{b}_+]^T=\lambda[\mathbf{e},\mathbf{b}]^T$ into (9).

Let us consider the time-continuous limit $\Delta_t\to 0$. In this limit $\lambda\to 1$, and $\frac{\lambda-1}{c\Delta_t}\to\frac{i\omega}{c}$ (where ω is to be determined, but it is finite: the limit exists)

$$\frac{i\omega}{c}\begin{bmatrix}\mathbf{e}\\ \mathbf{b}\end{bmatrix} = \begin{bmatrix}0 & [\star_\epsilon]^{-1}C^\dagger[\star_{\mu^{-1}}]\\ -C & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}\\ \mathbf{b}\end{bmatrix} \tag{12}$$

It will be convenient to bring (12) under a form where the matrix in the rhs. becomes anti-hermitian, which can be done by virtue of the symmetricity and positive-definiteness of the mass matrices:

$$\tilde{\mathbf{e}} = [\star_\epsilon]^{1/2}\mathbf{e} \tag{13}$$

$$\tilde{\mathbf{b}} = [\star_{\mu^{-1}}]^{1/2}\mathbf{b} \tag{14}$$

$$\tilde{C} = [\star_{\mu^{-1}}]^{1/2}C[\star_\epsilon]^{-1/2} \tag{15}$$

where the square roots of the symmetric positive-definite mass matrices should be interpreted as Cholesky decomposition. (12) now becomes

$$\frac{i\omega}{c}\begin{bmatrix}\tilde{\mathbf{e}}\\ \tilde{\mathbf{b}}\end{bmatrix} = \begin{bmatrix}0 & \tilde{C}^\dagger\\ -\tilde{C} & 0\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{e}}\\ \tilde{\mathbf{b}}\end{bmatrix} \tag{16}$$

We see that the matrix in the rhs. is anti-hermitian. Its eigenvalues are therefore purely imaginary, and come in complex conjugate pairs. ω is therefore real.

Let us transform (11) in the same way:

$$\frac{\lambda-1}{c\Delta_t}\begin{bmatrix}\tilde{\mathbf{e}}\\ \tilde{\mathbf{b}}\end{bmatrix} = \begin{bmatrix}0 & \tilde{C}^\dagger\\ -\lambda\tilde{C} & 0\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{e}}\\ \tilde{\mathbf{b}}\end{bmatrix} \tag{17}$$

We can now use solutions of (16) to construct solutions of (17). If $\omega_n,[\tilde{\mathbf{e}}_n,\tilde{\mathbf{b}}_n]^T$ is a solution of (16), then $\lambda(\omega_n),[\alpha(\omega_n)\tilde{\mathbf{e}}_n,\tilde{\mathbf{b}}_n]^T$ is a solution of (17):

$$\frac{\lambda(\omega_n)-1}{c\Delta_t}\begin{bmatrix}\alpha(\omega_n)\tilde{\mathbf{e}}_n\\ \tilde{\mathbf{b}}_n\end{bmatrix} = \begin{bmatrix}0 & \tilde{C}^\dagger\\ -\lambda(\omega_n)\tilde{C} & 0\end{bmatrix}\begin{bmatrix}\alpha(\omega_n)\tilde{\mathbf{e}}_n\\ \tilde{\mathbf{b}}_n\end{bmatrix} = \frac{i\omega_n}{c}\begin{bmatrix}\tilde{\mathbf{e}}_n\\ \lambda(\omega_n)\alpha(\omega_n)\tilde{\mathbf{b}}\end{bmatrix} \tag{18}$$

Thus

$$\begin{cases}\frac{\lambda(\omega_n)-1}{c\Delta_t}\alpha(\omega_n) &= \frac{i\omega_n}{c}\\ \frac{\lambda(\omega_n)-1}{c\Delta_t} &= \frac{i\omega_n}{c}\lambda(\omega_n)\alpha(\omega_n)\end{cases} \tag{19}$$

Elliminating $\alpha(\omega_n)$ gives

$$\lambda(\omega_n)^2 + (\Delta_t^2\omega_n^2-2)\lambda(\omega_n)+1=0 \tag{20}$$

This quadratic polynomial can be solved easily for $\lambda(\omega_n)$ (the eigenvalues of T_{leapfrog}), but it is interesting to note that the properties of the roots that we are interested

in (i.e. whether or not they lie on the unit circle) are actually determined by the structure of this polynomial: it is palindromic with real coefficients. This class of polynomials has solutions that are either on the unit circle or on the real axis, and they come in pairs whose product is 1 (if λ_n is a solution, so is $1/\lambda_n$). This is not a coincidence: similar stability analyses for higher-order time-stepping schemes give rise to higher-degree, but still palindromic, polynomials [15].

We seem to find two discrete solutions from a single continuous solution, but this is not so: if we had started from the continuous eigenvalue $-\omega_n$ with corresponding eigenvector $[-\tilde{\mathbf{e}}_n, \tilde{\mathbf{b}}_n]^T$, we would find the same solutions. So a pair of complex conjugate eigenvalues of (16) gives us a pair of eigenvalues of T_{leapfrog}.

The solutions $\lambda_n = \lambda(\omega_n)$ are

$$\lambda_n = \frac{-(\Delta_t^2\omega_n^2 - 2) \pm \Delta_t\omega_n\sqrt{\Delta_t^2\omega_n^2 - 4}}{2} \tag{21}$$

λ_n is a unit phasor if $\Delta_t^2\omega_n^2 \leq 4$. All solutions are stable if λ_n is a unit phasor for all n, i.e. if $\Delta_t \max_n(\omega_n) \leq 2$, or

$$\Delta_t \leq \Delta_{t,\max} = \frac{2}{c\rho\left(\begin{bmatrix} 0 & \tilde{C}^\dagger \\ -\tilde{C} & 0 \end{bmatrix}\right)} = \frac{2}{c\left\|\tilde{C}\right\|} \tag{22}$$

where $\rho(\cdot)$ is the spectral radius and $\|\cdot\|$ is the matrix norm $\|M\| = \max_{|\vec{x}|=1}|M\vec{x}|$.

(22) is the Courant condition, the condition under which the general leapfrog time-stepping equation (9) is stable. ∎

From a pair of eigenvalues of the time-continuous system $(\omega_n, -\omega_n)$ we have obtained a pair of eigenvalues of the leapfrog system $(\lambda_n, 1/\lambda_n)$. It is convenient to have a one-to-one correspondence between ω_n and λ_n, and we make the following arbitrary choice: positive ω_n correspond to λ_n on the unit circle with positive imaginary part, or on the real axis with $-1 < \lambda_n < 0$. Negative ω_n correspond to λ_n on the unit circle with negative imaginary part, or on the real axis with $\lambda_n < -1$. The case $\lambda = -1$ remains undefined. This choice is expressed in the following formulas

$$\lambda_n = \frac{-(\Delta_t^2\omega_n^2 - 2) + \Delta_t\omega_n\sqrt{\Delta_t^2\omega_n^2 - 4}}{2} \tag{23}$$

$$\Delta_t\omega_n = \begin{cases} \sqrt{2 - (\lambda_n^{-1} + \lambda_n)} & \Im(\lambda_n) \geq 0 \vee (-1 < \lambda_n < 0) \\ -\sqrt{2 - (\lambda_n^{-1} + \lambda_n)} & \Im(\lambda_n) < 0 \vee \lambda_n < -1 \end{cases}$$

$$= \mathcal{A}(\lambda_n) \tag{24}$$

We will use the function $\mathcal{A}(\lambda)$ further in this paper. Because $\lambda_n \approx \exp(i\omega_n\Delta_t)$ near $\omega_n\Delta_t = 0$ (this is a second-order accurate approximation, because leapfrog time-stepping is second-order accurate), it follows that $\mathcal{A}(\lambda) \approx \arg(\lambda)$ near $\lambda = 1$. Specifically: $\mathcal{A}(\exp(i\alpha)) = 2\sin(\alpha/2)$ for real α.

4. T-shifting. Figure 3 shows the temporal structure of leapfrog FDTD, with electric fields known at integer multiples of Δ_t, and magnetic fields known at half-integer multiples of Δ_t (see also figure 1), such that the electric fields at $t = n\Delta_t$ can be updated using the magnetic fields at $t = (n + 1/2)\Delta_t$, and the magnetic fields at $t = (n + 1/2)\Delta_t$ can be updated using the electric fields at $t = (n + 1)\Delta_t$, and so on.

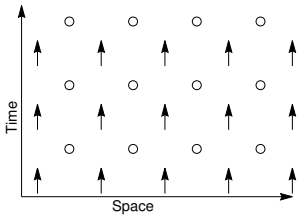

FIG. 3. *Leapfrog discretisation: E_y points (arrows) defined at integer time steps, B_z points (dots) at half-integer time steps.*

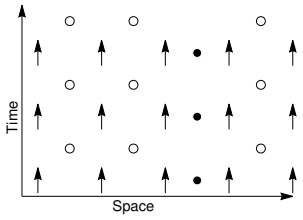

FIG. 4. *Magnetic shift: one magnetic point (B_z, dot) shifted by $\Delta_t/2$ towards the past.*

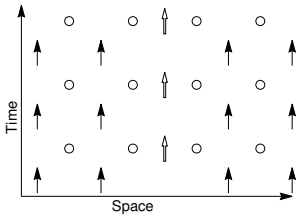

FIG. 5. *Electric shift: one electric point (E_y, arrow) shifted by $\Delta_t/2$ towards the future.*

In figure 4, one magnetic field point has been shifted along the time axis by $-\Delta_t/2$. It must be updated using Crank-Nicolson implicit FDTD. Similarly, in figure 5, one electric field point has been shifted along the time axis by $+\Delta_t/2$. It too must be updated using Crank-Nicolson implicit FDTD.

In what follows we will discuss time-stepping where an arbitrary subset of electric and magnetic field points has been shifted in this way. To this end we introduce diagonal indicator matrices whose diagonal values indicate whether or not the ith electric or magnetic discretisation point is a shifted point

$$P(x,y)_{i,j} = \begin{cases} 0 & i \neq j \\ x & i\text{th magnetic point is not shifted} \\ y & i\text{th magnetic point is shifted by } -\Delta_t/2 \end{cases} \tag{25}$$

$$Q(x,y)_{i,j} = \begin{cases} 0 & i \neq j \\ x & i\text{th electric point is not shifted} \\ y & i\text{th electric point is shifted by } +\Delta_t/2 \end{cases} \tag{26}$$

Given 2 numbers a, b with $a \neq b$, we have for all x, y:

$$P(x,y) = \frac{P(a,b) - aI}{b-a}(y-x) + xI = P(a,b)\frac{y-x}{b-a} - \frac{a(y-x)}{b-a}I + xI \tag{27}$$

where I is the unit matrix. Thus, if a matrix M commutes with $P(a,b)$, it also commutes with all $P(x,y)$:

$$\begin{aligned} MP(x,y) &= MP(a,b)\frac{y-x}{b-a} - M\frac{a(y-x)}{b-a} + Mx \\ &= P(a,b)\frac{y-x}{b-a}M - \frac{a(y-x)}{b-a}M + xM \\ &= P(x,y)M \end{aligned} \tag{28}$$

Similarly, if M commutes with $Q(a,b)$, it also commutes with all $Q(x,y)$.

4.1. Magnetic T-shifting. The time-stepping equations where some arbitrary subset of magnetic field points has been T-shifted are (29)

$$\begin{aligned}\frac{1}{c}\frac{\begin{bmatrix}\mathbf{e}_+\\ \mathbf{b}_+\end{bmatrix}-\begin{bmatrix}\mathbf{e}_-\\ \mathbf{b}_-\end{bmatrix}}{\Delta_t} &= \begin{bmatrix}0 & [\star_\epsilon]^{-1}C^T[\star_{\mu^{-1}}](I-R_B)\\ -R_BC & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}_-\\ \mathbf{b}_-\end{bmatrix}\\ &+\begin{bmatrix}0 & [\star_\epsilon]^{-1}C^T[\star_{\mu^{-1}}]R_B\\ -(I-R_B)C & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}_+\\ \mathbf{b}_+\end{bmatrix}\end{aligned} \tag{29}$$

where $R_B = P(1, 1/2)$ and thus $I - R_B = P(0, 1/2)$. We will make two more assumptions:

- R_B commutes with $[\star_{\mu^{-1}}]$ (which is trivially true in the common case where $[\star_{\mu^{-1}}]$ is diagonal), or equivalently that all $P(x, y)$ commute with $[\star_{\mu^{-1}}]$
- At least one of the diagonal elements $R_{B,ii}$ is $1/2$. If they are all 1 (i.e. $R_B = I$), then (29) reduces to the purely explicit case.

The time-stepping operator T_{R_B} for (29) is

$$\begin{aligned}T_{R_B} = &\left(\frac{I}{c\Delta_t}-\begin{bmatrix}0 & [\star_\epsilon]^{-1}C^T[\star_{\mu^{-1}}]R_B\\ -(I-R_B)C & 0\end{bmatrix}\right)^{-1}\\ &\cdot\left(\frac{I}{c\Delta_t}+\begin{bmatrix}0 & [\star_\epsilon]^{-1}C^T[\star_{\mu^{-1}}](I-R_B)\\ -R_BC & 0\end{bmatrix}\right)\end{aligned} \tag{30}$$

When $R_B = I$, T_{R_B} reduces to T_{leapfrog}.

4.1.1. Stability. The first step is to show a relation between the unit-norm eigenvalues of T_{R_B} and eigenvalues of a purely explicit time-stepping operator, which we will do in theorem 2.

THEOREM 2. *Let $[\mathbf{e}_n, \mathbf{b}_n]^T$ be an eigenvector of T_{R_B} with corresponding eigenvalue λ_n, i.e.*

$$\lambda_n\begin{bmatrix}\mathbf{e}_n\\ \mathbf{b}_n\end{bmatrix} = T_{R_B}\begin{bmatrix}\mathbf{e}_n\\ \mathbf{b}_n\end{bmatrix} \tag{31}$$

If $|\lambda_n| = 1$ and $\lambda_n \neq -1$, then there exists a leapfrog time-stepping operator T_L of the form (10) and a diagonal matrix κ_B such that

$$\lambda_n\begin{bmatrix}\mathbf{e}_n\\ \kappa_B^{-1}\mathbf{b}_n\end{bmatrix} = T_L\begin{bmatrix}\mathbf{e}_n\\ \kappa_B^{-1}\mathbf{b}_n\end{bmatrix} \tag{32}$$

Proof. Setting $[\mathbf{e}_-, \mathbf{b}_-] = [\mathbf{e}_n, \mathbf{b}_n]$ and $[\mathbf{e}_+, \mathbf{b}_+] = \lambda_n[\mathbf{e}_n, \mathbf{b}_n]$ in (29) gives us an equation for the eigenvalues λ_n and eigenvectors $[\mathbf{e}_n, \mathbf{b}_n]^T$ of T_{R_B}:

$$\begin{aligned}&\frac{1}{c}\frac{\lambda_n-1}{\Delta_t}\begin{bmatrix}\mathbf{e}_n\\ \mathbf{b}_n\end{bmatrix} =\\ &\begin{bmatrix}0 & [\star_\epsilon]^{-1}C^T[\star_{\mu^{-1}}]((\lambda_n-1)R_B+I)\\ (-\lambda_n+(\lambda_n-1)R_B)C & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}_n\\ \mathbf{b}_n\end{bmatrix}\end{aligned} \tag{33}$$

Let us introduce the "hatted" magnetic field

$$\kappa_B\hat{\mathbf{b}}_n = \mathbf{b}_n \tag{34}$$

where the diagonal matrix κ_B will be specified later. In terms of $\hat{\mathbf{b}}_n, \mathbf{e}_n$, (33) becomes

$$\frac{1}{c}\frac{\lambda_n - 1}{\Delta_t}\begin{bmatrix} I & 0 \\ 0 & \kappa_B \end{bmatrix}\begin{bmatrix} \mathbf{e}_n \\ \hat{\mathbf{b}}_n \end{bmatrix} = \tag{35}$$
$$\begin{bmatrix} 0 & [\star_\epsilon]^{-1}C^T[\star_{\mu^{-1}}]((\lambda_n - 1)R_B + I) \\ (-\lambda_n + (\lambda_n - 1)R_B)C & 0 \end{bmatrix}\begin{bmatrix} I & 0 \\ 0 & \kappa_B \end{bmatrix}\begin{bmatrix} \mathbf{e}_n \\ \hat{\mathbf{b}}_n \end{bmatrix}$$

$$\frac{1}{c}\frac{\lambda_n - 1}{\Delta_t}\begin{bmatrix} \mathbf{e}_n \\ \hat{\mathbf{b}}_n \end{bmatrix} = \tag{36}$$
$$\begin{bmatrix} 0 & [\star_\epsilon]^{-1}C^T[\star_{\mu^{-1}}]((\lambda_n - 1)R_B + I)\kappa_B \\ \kappa_B^{-1}(-\lambda_n + (\lambda_n - 1)R_B)C & 0 \end{bmatrix}\begin{bmatrix} \mathbf{e}_n \\ \hat{\mathbf{b}}_n \end{bmatrix}$$

Of course, this procedure does not change the solutions λ_n for any invertible κ_B. What we would like to do is choose κ_B such that (36) becomes an equation for the eigenvalues of a *purely explicit (leapfrog)* time-stepping operator, i.e. we would like to bring (36) under the form (11). What we want is this:

$$\begin{cases} \kappa_B^{-1}(-\lambda_n + (\lambda_n - 1)R_B) = -\lambda_n & (a) \\ [\star_{\mu^{-1}}]((\lambda_n - 1)R_B + I)\kappa_B \text{ is symmetric and positive definite} & (b) \end{cases} \tag{37}$$

(37.a) can be solved easily:

$$\kappa_B = \frac{(1 - \lambda_n)R_B}{\lambda_n} + I \tag{38}$$

(37.b) is trickier: $((\lambda_n - 1)R_B + I)\kappa_B$ is a diagonal matrix which commutes with $[\star_{\mu^{-1}}]$. We must therefore demand that its diagonal elements are strictly positive.

$$\begin{aligned} (((\lambda_n - 1)R_B + I)\kappa_B)_{ii} &= ((\lambda_n - 1)R_{B,ii} + 1)\left(\frac{(1 - \lambda_n)R_{B,ii}}{\lambda_n} + 1\right) \\ &= \begin{cases} 1 & R_{B,ii} = 1 \\ \frac{(1+\lambda_n)^2}{4\lambda_n} & R_{B,ii} = 1/2 \end{cases} \end{aligned} \tag{39}$$

Thus, we must demand

$$\frac{(1 + \lambda_n)^2}{4\lambda_n} > 0 \tag{40}$$

This is possible in two cases: either $\Im(\lambda_n) = 0$ and $\Re(\lambda_n) > 0$, or $|\lambda_n| = 1$ and $\lambda_n \neq -1$.

But, if we can write (36) under the form (11) at all, then $|\lambda_n| = 1$ or λ_n lies on the *negative* real axis (section 3). These conditions are only consistent if $|\lambda_n| = 1$ and $\lambda_n \neq -1$.

Thus, we can use (38) to transform (36) into an equation of the form (11), *if and only if* $|\lambda_n| = 1$ and $\lambda_n \neq -1$. Inserting (38) into (36), we get

$$\frac{1}{c}\frac{\lambda_n - 1}{\Delta_t}\begin{bmatrix} \mathbf{e}_n \\ \hat{\mathbf{b}}_n \end{bmatrix} = \begin{bmatrix} 0 & [\star_\epsilon]^{-1}C^T[\star_{\mu^{-1}}]P\left(1, \cos(\arg(\lambda_n)/2)^2\right) \\ -\lambda_n C & 0 \end{bmatrix}\begin{bmatrix} \mathbf{e}_n \\ \hat{\mathbf{b}}_n \end{bmatrix} \tag{41}$$

keeping in mind that $((\lambda_n - 1)R_B + I)\kappa_B = P\left(1, \cos(\arg(\lambda_n)/2)^2\right)$ is only valid when $|\lambda_n| = 1$. Let us expand this a little

$$\begin{cases} \frac{1}{c}\frac{\lambda_n - 1}{\Delta_t}\begin{bmatrix}\mathbf{e}_n \\ \hat{\mathbf{b}}_n\end{bmatrix} = \begin{bmatrix} 0 & [\star_\epsilon]^{-1}C^T[\star_{\mu^{-1}}]P\left(1, \cos(\alpha_n/2)^2\right) \\ -\lambda_n C & 0 \end{bmatrix}\begin{bmatrix}\mathbf{e}_n \\ \hat{\mathbf{b}}_n\end{bmatrix} & (a) \\ \lambda_n = \exp(i\alpha_n) & (b) \end{cases} \tag{42}$$

where $\alpha \in]-\pi, \pi[$.

(42.a) is now of the form (11), an eigenvalue equation of a leapfrog time-stepping operator (note $[\star_{\mu^{-1}}]P\left(1, \cos(\alpha/2)^2\right)$ in (42.a) plays the role of $[\star_{\mu^{-1}}]$ in (11)). The leapfrog time-stepping operator T_L is

$$T_L = \begin{bmatrix} \frac{I}{c\Delta_t} & 0 \\ C & \frac{I}{c\Delta_t} \end{bmatrix}^{-1}\begin{bmatrix} \frac{I}{c\Delta_t} & [\star_\epsilon]^{-1}C^\dagger[\star_{\mu^{-1}}]P\left(1, \cos\left(\frac{\arg(\lambda_n)}{2}\right)^2\right) \\ 0 & \frac{I}{c\Delta_t} \end{bmatrix} \tag{43}$$

∎

The converse of theorem 2 also holds: should we find a unit-norm solution of (32) (noting that κ_B and T_L depend implicitly on λ_n), then we can transform it into a unit-norm solution of (31). The question is now: how many unit-norm solutions of (32) can we find? If we can find one for all n, then all eigenvalues of (31) have unit norm.

A unit-norm solution of (32) can be written as follows:

$$\exp(i\alpha_n)\begin{bmatrix}\mathbf{e}_n \\ \kappa_B^{-1}\mathbf{b}_n\end{bmatrix} = T_L\begin{bmatrix}\mathbf{e}_n \\ \kappa_B^{-1}\mathbf{b}_n\end{bmatrix} \tag{44}$$

with $-\pi < \alpha_n < \pi$, $\lambda_n = \exp(i\alpha_n)$.

We know the eigenvalues of T_L from (23), so (44) becomes:

$$\exp(i\alpha_n) = \frac{-(\Delta_t^2\omega_n(\alpha_n)^2 - 2) + \Delta_t\omega_n(\alpha_n)\sqrt{\Delta_t^2\omega_n(\alpha_n)^2 - 4}}{2} \tag{45}$$

where $i\omega_n(\alpha_n)/c$ is an eigenvalue of

$$\begin{bmatrix} 0 & [\star_\epsilon]^{-1}C^T[\star_{\mu^{-1}}]P\left(1, \cos(\alpha_n/2)^2\right) \\ -C & 0 \end{bmatrix} \tag{46}$$

Applying the function $\mathcal{A}$ (see (24)) to both sides of (45) gives

$$2\sin\left(\frac{\alpha_n}{2}\right) = \Delta_t\omega_n(\alpha_n) \tag{47}$$

The maximum Δ_t for which (47) has a solution for all n, which is the maximum Δ_t for which all eigenvalues of T_{R_B} have unit norm, is given by theorem 3.

THEOREM 3. *(47) has a solution α_n for all n, with $-\pi < \alpha_n < \pi$, if*

$$\Delta_t < \frac{2}{\max_n |\omega_n(\pi)|} \tag{48}$$

Proof. (48) implies that for all n

$$\omega_n(\pi)\Delta_t < 2\sin\left(\frac{\pi}{2}\right) \tag{49}$$

$$\omega_n(-\pi)\Delta_t > 2\sin\left(\frac{-\pi}{2}\right) \tag{50}$$

Thus, by the intermediate value theorem, $\omega_n(\alpha_n) - 2\sin\left(\frac{\alpha_n}{2}\right) = 0$ must have a root with $-\pi < \alpha_n < \pi$. ∎

Note that the condition (48) can be interpreted as the Courant condition of the explicit part: it is the Courant condition for a configuration in which μ_r is effectively ∞ in the implicit part, no waves propagate there, so only the explicit part matters.

4.1.2. Energy conservation. In order to rigorously prove energy conservation, one ingredient is still missing. In addition to Δ_t being small enough that all eigenvalues of the time-stepping operator lie on the unit circle, the time-stepping operator itself must be diagonalizable.

If the time-stepping operator T_{R_B} is diagonalizable, then there exists a matrix S and a diagonal matrix $D = \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$ such that

$$SDS^{-1} = T_{R_B} \tag{51}$$

If the time step is small enough for the algorithm to be stable (i.e. all $|\lambda_i| = 1$), then for any vector v we have

$$|Dv|^2 = |v|^2 \tag{52}$$

The time-stepping operator maps v_t onto v_{t+1}:

$$v_{t+1} = T_{R_B} v_t = SDS^{-1} v_t \tag{53}$$

$$S^{-1} v_{t+1} = DS^{-1} v_t \tag{54}$$

$$|S^{-1} v_{t+1}|^2 = |S^{-1} v_t|^2 \tag{55}$$

Thus, the quantity $|S^{-1}v|^2$ is exactly conserved by the discrete time-stepping. It is a discrete analogon of the continuous conserved energy.

THEOREM 4. *T_{R_B} is diagonalizable.*

Proof. The diagonalization theorem states that a matrix is diagonalizable if and only if the algebraic multiplicity of every eigenvalue equals its geometric multiplicity.

Consider an eigenvalue λ_n of T_{R_B}. Suppose it has algebraic multiplicity k. Then

$$\lambda_n \begin{bmatrix} \mathbf{e}_{n,j} \\ \mathbf{b}_{n,j} \end{bmatrix} = T_{R_B} \begin{bmatrix} \mathbf{e}_{n,j} \\ \mathbf{b}_{n,j} \end{bmatrix} \tag{56}$$

with $j = 1, \cdots, k$. The geometric multiplicity is

$$\dim\left(\mathrm{span}\left(\begin{bmatrix} \mathbf{e}_{n,1} \\ \mathbf{b}_{n,1} \end{bmatrix}, \cdots, \begin{bmatrix} \mathbf{e}_{n,k} \\ \mathbf{b}_{n,k} \end{bmatrix}\right)\right) \tag{57}$$

Recall theorem 2, in particular (32):

$$\lambda_n \begin{bmatrix} \mathbf{e}_{n,j} \\ \kappa_B^{-1}\mathbf{b}_{n,j} \end{bmatrix} = T_L \begin{bmatrix} \mathbf{e}_{n,j} \\ \kappa_B^{-1}\mathbf{b}_{n,j} \end{bmatrix} \tag{58}$$

Being eigenvectors of a leapfrog time-stepping operator T_L, the $[\mathbf{e}_{n,j}, \kappa_B^{-1}\mathbf{b}_{n,j}]^T$ are known to be linearly independent for Δ_t strictly below the Courant limit (this is not proven in this paper, but see e.g. [2]). Because κ_B is invertible, the $[\mathbf{e}_{n,j}, \mathbf{b}_{n,j}]^T$ are also linearly independent. Therefore,

$$\dim\left(\mathrm{span}\left(\begin{bmatrix} \mathbf{e}_{n,1} \\ \mathbf{b}_{n,1} \end{bmatrix}, \cdots, \begin{bmatrix} \mathbf{e}_{n,k} \\ \mathbf{b}_{n,k} \end{bmatrix}\right)\right) = k \tag{59}$$

The geometric multiplicity of λ_n equals its algebraic multiplicity. T_{R_B} is diagonalizable. ∎

4.2. Electric T-shifting. An entirely analogous line of reasoning works for electric T-shifting, so we will not go into detail. The fully discrete system with hybrid explicit/implicit updating is

$$\begin{aligned}\frac{1}{c}\frac{\begin{bmatrix}\mathbf{e}_+\\ \mathbf{b}_+\end{bmatrix}-\begin{bmatrix}\mathbf{e}_-\\ \mathbf{b}_-\end{bmatrix}}{\Delta_t} &= \begin{bmatrix}0 & R_E[\star_\epsilon]^{-1}C^T[\star_{\mu^{-1}}]\\ -C(I-R_E) & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}_-\\ \mathbf{b}_-\end{bmatrix}\\ &+\begin{bmatrix}0 & (I-R_E)[\star_\epsilon]^{-1}C^T[\star_{\mu^{-1}}]\\ -CR_E & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}_+\\ \mathbf{b}_+\end{bmatrix}\end{aligned} \tag{60}$$

where $R_E = Q(1,1/2)$ and thus $I-R_E = Q(0,1/2)$, and $[\star_\epsilon]$ must commute with R_E.

4.3. General T-shifting. What do the time-stepping equations look like if we allow both electric and magnetic T-shifted points, given by R_E and R_B respectively? We seek an equation with the following properties:

- If $R_E = I$, it should reduce to (29).
- If $R_B = I$, it should reduce to (60).
- If $R_E = I$ and $R_B = I$, it should reduce to (9).
- If $R_E = I/2$ and $R_B = I/2$, it should reduce to (61).

$$\frac{1}{c}\frac{\begin{bmatrix}\mathbf{e}_+\\ \mathbf{b}_+\end{bmatrix}-\begin{bmatrix}\mathbf{e}_-\\ \mathbf{b}_-\end{bmatrix}}{\Delta_t} = \begin{bmatrix}0 & 0\\ -C & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}_-\\ \mathbf{b}_-\end{bmatrix}+\begin{bmatrix}0 & [\star_\epsilon]^{-1}C^T[\star_{\mu^{-1}}]\\ 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}_+\\ \mathbf{b}_+\end{bmatrix} \tag{61}$$

The simplest such equation is

$$\begin{aligned}&\frac{1}{c}\frac{\begin{bmatrix}\mathbf{e}_+\\ \mathbf{b}_+\end{bmatrix}-\begin{bmatrix}\mathbf{e}_-\\ \mathbf{b}_-\end{bmatrix}}{\Delta_t} =\\ &\begin{bmatrix}0 & \left(R_E-\frac{I}{2}\right)\widehat{C}^T+\widehat{C}^T\left(R_B-\frac{I}{2}\right)\\ -(I-R_B)C-C(I-R_E) & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}_-\\ \mathbf{b}_-\end{bmatrix}\\ &+\begin{bmatrix}0 & (I-R_E)\widehat{C}^T+\widehat{C}^T(I-R_B)\\ -\left(R_B-\frac{I}{2}\right)C-C\left(R_E-\frac{I}{2}\right) & 0\end{bmatrix}\begin{bmatrix}\mathbf{e}_+\\ \mathbf{b}_+\end{bmatrix}\end{aligned} \tag{62}$$

where for brevity we used $\widehat{C}^T = [\star_\epsilon]^{-1}C^T[\star_{\mu^{-1}}]$. Again $[\star_\epsilon]$ must commute with R_E, and $[\star_{\mu^{-1}}]$ must commute with R_B.

4.3.1. Stability. The time-stepping operator for (62) is T_{R_E,R_B}:

$$\begin{aligned}T_{R_E,R_B} = &\left(\frac{I}{c\Delta_t}-\begin{bmatrix}0 & (I-R_E)\widehat{C}^T+\widehat{C}^T(I-R_B)\\ -\left(R_B-\frac{I}{2}\right)C-C\left(R_E-\frac{I}{2}\right) & 0\end{bmatrix}\right)^{-1}\\ &\cdot\left(\frac{I}{c\Delta_t}+\begin{bmatrix}0 & \left(R_E-\frac{I}{2}\right)\widehat{C}^T+\widehat{C}^T\left(R_B-\frac{I}{2}\right)\\ -(I-R_B)C-C(I-R_E) & 0\end{bmatrix}\right)\end{aligned} \tag{63}$$

THEOREM 5. *Let $[\mathbf{e}_n,\mathbf{b}_n]^T$ be an eigenvector of T_{R_E,R_B} with corresponding eigenvalue λ_n, i.e.*

$$\lambda_n\begin{bmatrix}\mathbf{e}_n\\ \mathbf{b}_n\end{bmatrix} = T_{R_E,R_B}\begin{bmatrix}\mathbf{e}_n\\ \mathbf{b}_n\end{bmatrix} \tag{64}$$

If $|\lambda_n| = 1$, then there exists a leapfrog time-stepping operator T_L such that

$$\lambda_n \begin{bmatrix} \tilde{\mathbf{e}}_n \\ \tilde{\mathbf{b}}_n \end{bmatrix} = T_L \begin{bmatrix} \tilde{\mathbf{e}}_n \\ \tilde{\mathbf{b}}_n \end{bmatrix} \tag{65}$$

where $\tilde{\mathbf{e}}_n = [\star_\epsilon]^{1/2}\mathbf{e}_n$, $\tilde{\mathbf{b}}_n = [\star_{\mu^{-1}}]^{1/2}\mathbf{b}_n$ as in (13-14).

Proof. It can be shown by induction that if a diagonal matrix A commutes with a symmetric positive-definite matrix B, then A also commutes with the Cholesky factors of B. In particular, from our assumption that $[\star_\epsilon]$ commutes with R_E, we get that R_E also commutes with $[\star_\epsilon]^{1/2}$, and the same for R_B and $[\star_{\mu^{-1}}]$.

Knowing this, we can start from (62), use the transformation (13-14) and write an equation which the eigenvalues of T_{R_E,R_B} obey:

$$\frac{1}{c}\frac{\lambda_n - 1}{\Delta_t} \begin{bmatrix} \tilde{\mathbf{e}}_n \\ \tilde{\mathbf{b}}_n \end{bmatrix} = \begin{bmatrix} 0 & F_E(\lambda_n)\tilde{C}^T + \tilde{C}^T F_B(\lambda_n) \\ G_B(\lambda_n)\tilde{C} + \tilde{C}G_E(\lambda_n) & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{e}}_n \\ \tilde{\mathbf{b}}_n \end{bmatrix} \tag{66}$$

$$F_E(\lambda_n) = R_E(1-\lambda_n) - \frac{I}{2} + \lambda_n I = Q\left(\frac{1}{2}, \frac{\lambda_n}{2}\right) \tag{67}$$

$$F_B(\lambda_n) = R_B(1-\lambda_n) - \frac{I}{2} + \lambda_n I = P\left(\frac{1}{2}, \frac{\lambda_n}{2}\right) \tag{68}$$

$$G_B(\lambda_n) = R_B(1-\lambda_n) + \frac{\lambda_n I}{2} - I = -P\left(\frac{\lambda_n}{2}, \frac{1}{2}\right) \tag{69}$$

$$G_E(\lambda_n) = R_E(1-\lambda_n) + \frac{\lambda_n I}{2} - I = -Q\left(\frac{\lambda_n}{2}, \frac{1}{2}\right) \tag{70}$$

If $|\lambda_n| = 1$, we can show with straightforward arithmetic (using $\lambda_n^\dagger = \overline{\lambda_n} = \lambda_n^{-1}$) that

$$G_B(\lambda_n)\tilde{C} + \tilde{C}G_E(\lambda_n) = -\lambda_n \left(F_E(\lambda_n)\tilde{C}^T + \tilde{C}^T F_B(\lambda_n)\right)^\dagger \tag{71}$$

Let us introduce the notation $\left(F_E(\lambda_n)\tilde{C}^T + \tilde{C}^T F_B(\lambda_n)\right)^\dagger = \Psi(\lambda_n)$ for brevity.

We can now relate unit-norm eigenvalues of T_{R_E,R_B} to unit-norm eigenvalues of a leapfrog time-stepping operator, just as we did in section 4.1:

$$\begin{cases} \frac{\lambda_n - 1}{c\Delta_t} \begin{bmatrix} \tilde{\mathbf{e}}_n \\ \tilde{\mathbf{b}}_n \end{bmatrix} = \begin{bmatrix} 0 & \Psi(\exp(i\alpha_n))^\dagger \\ -\lambda_n \Psi(\exp(i\alpha_n)) & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{e}}_n \\ \tilde{\mathbf{b}}_n \end{bmatrix} & (a) \\ \lambda_n = \exp(i\alpha_n) & (b) \end{cases} \tag{72}$$

(72.a) is of the form (17): it is an equation for the eigenvalues of a leapfrog time-stepping operator (note $\Psi(\exp(i\alpha_n))$ in (72.a) plays the role of $\tilde{C}$ in (17)). This expression is the reason why we allowed the possibility of complex C in section 3.

The leapfrog time-stepping operator T_L is

$$T_L = \begin{bmatrix} \frac{I}{c\Delta_t} & 0 \\ \Psi(\lambda_n) & \frac{I}{c\Delta_t} \end{bmatrix}^{-1} \begin{bmatrix} \frac{I}{c\Delta_t} & \Psi(\lambda_n) \\ 0 & \frac{I}{c\Delta_t} \end{bmatrix} \tag{73}$$

∎

The argument now proceeds entirely as it did in section 4.1. We know the solutions of (72.a)

$$\begin{cases} \lambda_n = \frac{-(\Delta_t^2\omega_n(\alpha_n)^2 - 2) + \Delta_t\omega_n(\alpha_n)\sqrt{\Delta_t^2\omega_n(\alpha_n)^2 - 4}}{2} & (a) \\ \lambda_n = \exp(i\alpha_n) & (b) \end{cases} \tag{74}$$

where $i\omega_n(\alpha_n)/c$ is an eigenvalue of the anti-Hermitian matrix

$$\begin{bmatrix} 0 & \Psi(\exp(i\alpha_n))^\dagger \\ -\Psi(\exp(i\alpha_n)) & 0 \end{bmatrix} \tag{75}$$

Simplifying (74) (in the same manner as (45)→(47)) gives

$$\omega_n(\alpha_n)\Delta_t = 2\sin(\alpha_n/2) \tag{76}$$

As in section 4.1, finding the unit-norm eigenvalues becomes a matter of finding intersections between these purely real continuous functions, and if an intersection exists for all n then all eigenvalues have unit norm. These intersections are guaranteed to exist for all n if

$$\Delta_t < \frac{2}{\max_n |\omega_n(\pi)|} \tag{77}$$

What remains to be shown is that the stability condition (77) is no more restrictive than for the leapfrog case.

$$\begin{aligned} \frac{2}{\max_n |\omega_n(\pi)|} &= \frac{2}{c\|F_E(\exp(i\pi))\tilde{C}^T + \tilde{C}^T F_B(\exp(i\pi))\|} \\ &= \frac{2}{c\left\|Q\left(\frac{1}{2}, \frac{\exp(i\pi)}{2}\right)\tilde{C}^T + \tilde{C}^T P\left(\frac{1}{2}, \frac{\exp(i\pi)}{2}\right)\right\|} \\ &= \frac{2}{c\|\tilde{C}\| n(\pi)} \end{aligned} \tag{78}$$

where $n(\alpha) = \frac{\|Q(\frac{1}{2}, \frac{\exp(i\alpha)}{2})\tilde{C}^T + \tilde{C}^T P(\frac{1}{2}, \frac{\exp(i\alpha)}{2})\|}{\|\tilde{C}\|}$. It follows immediately from the properties of a submultiplicative matrix norm that $n(\pi) \leq 1$: the stability condition (77) is no more restrictive than that of leapfrog FDTD.

4.3.2. Energy conservation. As in section 4.1.2, T_{R_E,R_B} should be diagonalizable in order to guarantee the existence of a discrete conserved energy. That T_{R_E,R_B} is indeed diagonalizable can be shown by an argument which is entirely analogous to that in the proof of theorem 4.

4.4. Discussion. (62) is the main result of this paper. It tells us how to time step in FDTD-like discretisation grids where the temporal locations of the discretized electric and magnetic field components (i.e. at integer or half-integer time steps) are chosen arbitrarily. In what follows, we will give examples of how, by carefully choosing the temporal locations of the discretized electric and magnetic field components, we can construct explicit time-stepping algorithms that are stabler than FDTD.

Consider figure 6. This shows the 2^{12} possible $n(\alpha)$ curves (78) for a periodic 2D configuration with just 12 degrees of freedom (4 E_x points, 4 E_y points, and 4 B_z points, so 2^8 possible choices for R_E and 2^4 possible choices for R_B). Two curves have $n(\alpha) = 1$, these are the cases $R_E = I, R_B = I$ or $R_E = I/2, R_B = I/2$ where (62) reduces to leapfrog FDTD. Two curves have $n(\alpha) = \cos(\alpha/2)$, these are the cases $R_E = I, R_B = I/2$ or $R_E = I/2, R_B = I$ where (62) reduces to Crank-Nicolson FDTD. 4092 remaining curves are intermediate methods, which are stabler than leapfrog FDTD but not as computationally intensive as Crank-Nicolson FDTD. Of particular interest are algorithms that are twice stabler than FDTD, and algorithms that are $2\sqrt{2}$ times stabler than FDTD. We will discuss these in the next section.

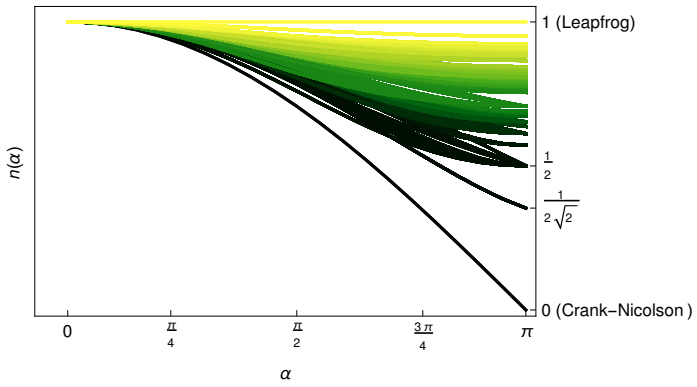

FIG. 6. *All* 2^{12} *possible* $n(\alpha)$ *functions for a periodic 2D FDTD configuration with* 2×2 *cells, i.e. 4* E_x *points, 4* E_y *points, and 4* B_z *points. The maximum stable time step* $\Delta_{t,max} \propto n(\pi)^{-1}$*: the smaller* $n(\pi)$*, the stabler the associated algorithm. The gap between leapfrog and Crank-Nicolson is bridgeable.*

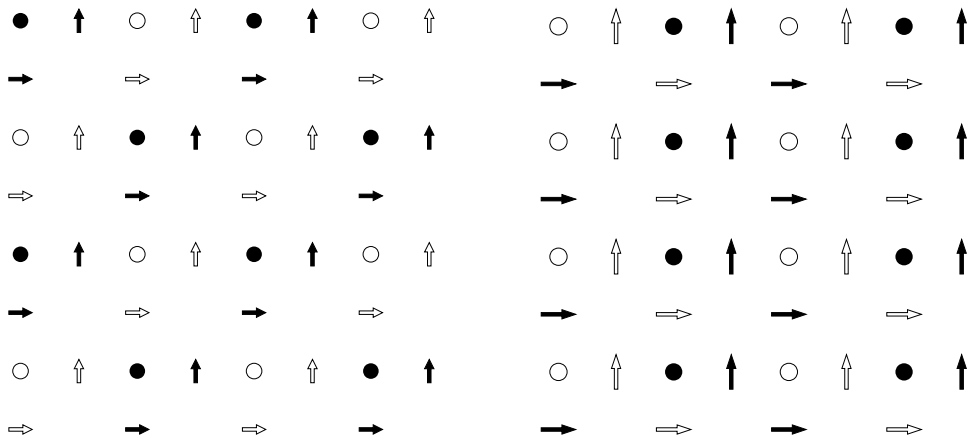

FIG. 7. *Standard 2D Yee-cell discretisation for Maxwell's equations, with nonstandard temporal discretisation. Arrows are locations of* E_x, E_y *discretisation points, circles are locations of* B_z *discretisation points. Color indicates temporal location (solid black=integer step, white with black outline=half-integer step). The grid on the left has Courant condition* $c\Delta_t < \sqrt{2}\Delta$*, or 2 times better than FDTD. The grid on the right has Courant condition* $c\Delta_t < 2\Delta$*, or* $2\sqrt{2}$ *times better than FDTD.*

5. Examples.

5.1. Two explicit FDTD-like methods that are stabler than Leapfrog FDTD. The configurations of figure 7 have stability conditions $c\Delta_t < \sqrt{2}\Delta$ and $c\Delta_t < 2\Delta$, or respectively 2 and $2\sqrt{2}$ times stabler than FDTD.

Both these configurations can be time-stepped explicitly using a leapfrog approach: update first the integer-time-step (black) field quantities using values from the neighbouring half-integer-time-step (white with black outline) field quantities, and then update the half-integer-time-step field quantities using the neighbouring integer-time-step field quantities.

We suspect that, within the framework of (62), with the standard second-order accurate curl operator of 2D FDTD, it is not possible to have a Courant limit larger

than that of figure 7 (right), except for the unconditionally stable fully implicit case. Figure 6 provides some evidence for this conjecture: if stabler configurations exist, they are not representable within a periodic 2×2 grid.

We will analyse the configuration of figure 7 (left) in some detail. The 2D Maxwell's equations are

$$\frac{\partial E_x}{\partial y} - \frac{\partial E_y}{\partial x} = \frac{1}{c}\frac{\partial B_z}{\partial t} \tag{79}$$

$$\frac{\partial B_z}{\partial y} = \frac{\epsilon_r \mu_r}{c}\frac{\partial E_x}{\partial t} \tag{80}$$

$$-\frac{\partial B_z}{\partial x} = \frac{\epsilon_r \mu_r}{c}\frac{\partial E_y}{\partial t} \tag{81}$$

In the discretisation of figure 7 (left), discrete time-stepping can be done by repeatedly updating single "cells" (neighbouring E_x, E_y, B_z of the same colour i.e. at the same temporal location, figure 8). A "cell update" requires only the solution of the constant 3×3 set of equations (82)

$$\frac{\begin{bmatrix} B_{z1}^{t+\Delta_t} \\ E_{x2}^{t+\Delta_t} \\ E_{y2}^{t+\Delta_t} \end{bmatrix} - \begin{bmatrix} B_{z1}^{t} \\ E_{x2}^{t} \\ E_{y2}^{t} \end{bmatrix}}{c\Delta_t} = \frac{1}{\Delta} \begin{bmatrix} 0 & \frac{1}{\mu_r \epsilon_r} & \frac{1}{\mu\epsilon} \\ 0 & 0 & \frac{-1}{\mu_r \epsilon_r} \\ 0 & \frac{-1}{\mu_r \epsilon_r} & 0 \\ 1 & 0 & 0 \\ -1 & 0 & 0 \\ 1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}^T \begin{bmatrix} (B_{z1}^{t+\Delta_t} + B_{z1}^{t})/2 \\ B_{z2}^{t+\Delta_t/2} \\ B_{z3}^{t+\Delta_t/2} \\ E_{x1}^{t+\Delta_t/2} \\ (E_{x2}^{t+\Delta_t} + E_{x2}^{t})/2 \\ E_{y1}^{t+\Delta_t/2} \\ (E_{y2}^{t+\Delta_t} + E_{y2}^{t})/2 \end{bmatrix} \tag{82}$$

This set of equations can be solved in advance, so this method is explicit (In the case of figure 7 (right), updates are even simpler, requiring only the solution of a 2×2 set of equations to update B_z and E_y, and E_x updates are identical to those of leapfrog FDTD.).

The discrete dispersion relation (83) can be obtained using Fourier analysis. Assuming $\mu_r = 1, \epsilon_r = 1$, inserting complex exponential solutions and writing the condition for the existence of nontrivial solutions gives

$$\cos\left(\frac{\Delta_t \omega}{2}\right) = \frac{c^2\Delta_t^2 y \pm \sqrt{c^4\Delta_t^4\left(y^2-4\right) + 16\Delta^4}}{2c^2\Delta_t^2 + 4\Delta^2} \tag{83}$$

where $y = \cos(k_x\Delta) + \cos(k_y\Delta)$. (83) could be unstable in two cases:

- when $c^4\Delta_t^4\left(y^2-4\right) + 16\Delta^4 < 0$ for any real y, which can be avoided if $-4c^4\Delta_t^4 + 16\Delta^4 < 0$ or $\sqrt{2}\Delta < c\Delta_t$.
- when the rhs. is < -1 or > 1. To find the maxima and minima of the rhs,

$$\frac{\partial}{\partial y}\frac{c^2\Delta_t^2 y \pm \sqrt{c^4\Delta_t^4\left(y^2-4\right) + 16\Delta^4}}{2c^2\Delta_t^2 + 4\Delta^2} = 0 \tag{84}$$

$$\mp c^2\Delta_t^2 = \frac{c^4\Delta_t^4 y}{\sqrt{c^4\Delta_t^4\left(y^2-4\right) + 16\Delta^4}} \tag{85}$$

$$\mp\sqrt{c^4\Delta_t^4\left(y^2-4\right) + 16\Delta^4} = c^2\Delta_t^2 y \tag{86}$$

$$c^4\Delta_t^4\left(y^2-4\right) + 16\Delta^4 = c^4\Delta_t^4 y^2 \tag{87}$$

$$-4c^4\Delta_t^4 + 16\Delta^4 = 0 \tag{88}$$

Below the Courant limit, (88) is false, so the derivative has no zeroes and the rhs. of (83) is either an increasing or a decreasing function of y, it has no minima or maxima. The extrema thus occur at the extremal values of y, i.e. at $y = \pm 2$. At $y = 2$, the value is $\frac{2c^2\Delta_t^2 \pm 4\Delta^2}{2c^2\Delta_t^2 + 4\Delta^2} \in [-1, 1]$. At $y = -2$, the value is $\frac{-2c^2\Delta_t^2 \pm 4\Delta^2}{2c^2\Delta_t^2 + 4\Delta^2} \in [-1, 1]$. So, the rhs. of (83) is never < -1 or > 1.

In (83), only one of the roots corresponds to solutions near $\omega = 0$,$k_x = k_y = 0$, where we expect (83) to approximate the continuous dispersion relation. This is the root with the + sign. A Taylor series expansion of (83) with the + sign in Δ_t and Δ gives

$$1 - \frac{\omega^2\Delta_t^2}{8} = 1 + \frac{c^2\Delta_t^2}{4\Delta^2}\left(-\frac{\Delta^2}{2}(k_x^2 + k_y^2) + O(\Delta^4)\right) + O(\Delta_t^4) \tag{89}$$

$$\frac{\omega^2}{8} = \frac{c^2}{4}\left(\frac{1}{2}(k_x^2 + k_y^2) + O(\Delta^2)\right) + O(\Delta_t^2) \tag{90}$$

$$\omega^2 = c^2(k_x^2 + k_y^2) + O(\Delta^2) + O(\Delta_t^2) \tag{91}$$

So (83) is indeed a second-order accurate approximation of the continuous dispersion relation $\omega^2 = c^2k^2$.

At the Courant limit $\sqrt{2}\Delta = c\Delta_t$, (83) becomes

$$\cos\left(\frac{\sqrt{2}\Delta\omega}{2c}\right) = \frac{2\Delta^2 y \pm \sqrt{4\Delta^4\left(y^2 - 4\right) + 16\Delta^4}}{4\Delta^2 + 4\Delta^2} = \frac{y \pm |y|}{4} \tag{92}$$

thus there are two solutions $\cos\left(\frac{\sqrt{2}\Delta\omega}{2c}\right) = \frac{y}{2}$ or $\cos\left(\frac{\sqrt{2}\Delta\omega}{2c}\right) = 0$. The first is exactly the dispersion relation of standard leapfrog FDTD at its Courant limit. The second are unphysical solutions which exist only at (not below) the Courant limit. It is therefore probably best to run (82) with a time step slightly below its Courant limit.

To illustrate all this, let us look at figure 9. The exact dispersion relation $\omega^2 = (k_x^2 + k_y^2)c^2$ cannot be represented in $(y, \cos(\omega\Delta_t/2))$ coordinates, because $(k_x^2 + k_y^2)$ cannot be recovered from $y = \cos(k_x\Delta) + \cos(k_y\Delta)$. Nonetheless, only a limited subset of $(y, \cos(\omega\Delta_t/2))$ can possibly obey the exact dispersion. This subset is shaded in blue. The dispersion relation (83) is shown in red (95% of Courant limit) and yellow (at the Courant limit). The solutions (92) are clearly visible in the yellow lines. The exact and discrete dispersion relations match in the well-resolved limit, i.e. the upper right corner of the plot.

It is true that the update equations (82) are more complex than those of FDTD, thus calculating a time step takes more time, which counterbalances the fact that the time step can be larger. However, there is another point that needs to be made: suppose we are running the configuration of figure 7 (left) in a parallelized fashion. Suppose the vertical line in figure 8 is a boundary between processors: everything to the left of it is updated by processor A, everything to the right of it by processor B. Both in FDTD, and in this configuration, we need to communicate E_{y1} from A to B and B_{z1} from B to A once per time step. Thus, by using the configuration of figure 7 (left), we can effectively halve the amount of interprocessor communication per unit of time ellapsed in the simulation, which is useful given that interprocessor communication is a bottleneck for many parallelized algorithms.

5.2. Convergence. In figure 10, we numerically investigate the rate of convergence of (62). In order to do this, we constructed a 2D rectangular cavity with PEC

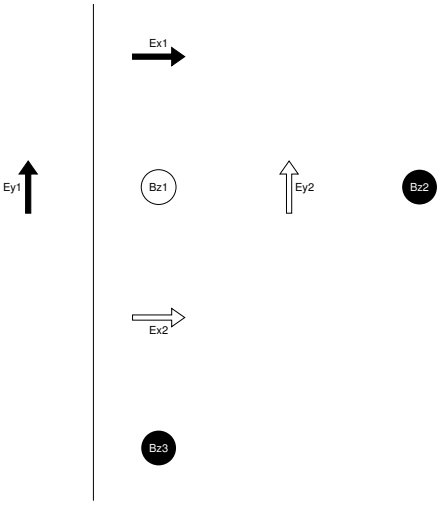

FIG. 8. *Location of the quantities used in (82), which describes how to update a single "cell", i.e. the three outlined (half-integer time step) points B_{z1}, E_{x2}, E_{y2}, using only neighbouring black (integer time step) points.*

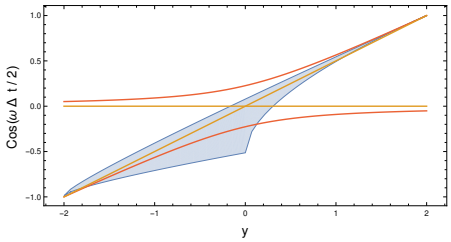

FIG. 9. *Plot of the dispersion relation (83) at 95% of the Courant limit (red) and at the Courant limit (yellow). The exact dispersion relation (not exactly representable in these coordinates) is shaded in blue. The well-resolved limit is at $y = 2, \cos(\omega\Delta_t/2) = 1$, where the discrete and exact dispersion relations match.*

boundary conditions, consisting of 30×30 cells. We excited modes of this cavity with wavenumbers $k_x = 2k_y$ (the discretisation error depends on the direction of propagation, so we need to keep k_x/k_y constant). We compared the numerically obtained electric and magnetic fields in this cavity with the known exact solution, being careful to evaluate the exact solution at the correct temporal locations. The error shown in figure 10 is the mean absolute difference between the exact and the numerical solution. $k\Delta$ on the x-axis is a measure of the well-resolvedness of the wave: it should be small in order for a finite-difference algorithm to give accurate results. When the error $\propto (k\Delta)^n$, we say that the algorithm is nth-order accurate. On the log-log plot in figure 10, we can find n from the slope of the lines.

We see that not all choices of R_E, R_B give rise to second-order accurate algorithms. In particular, algorithm (82) is only first-order accurate. Leapfrog and Crank-Nicholson FDTD are second-order accurate. Three randomly chosen intermediate algorithms (with diagonal values of R_E, R_B in (62) chosen at random) also appear to be second-order accurate, from which we conclude that second-order accuracy is likely to be the rule, rather than the exception, among intermediate algorithms of the from (62).

We do not currently have a general answer to the question of which choices of R_E, R_B make (62) second-order accurate, and which do not. We also did not investigate higher-order discrete curl operators[6] or Lobatto cells[2], both of which are perfectly compatible with (62), and which may be expected to improve the order of accuracy.

5.3. Numerical investigation of long-term stability and energy conservation. Leapfrog FDTD conserves the following energy exactly[2]

$$e = \mathbf{b}_-^T[\star_{\mu^{-1}}]\mathbf{b}_- + \mathbf{e}_+^T[\star_\epsilon]\mathbf{e}_- \tag{93}$$

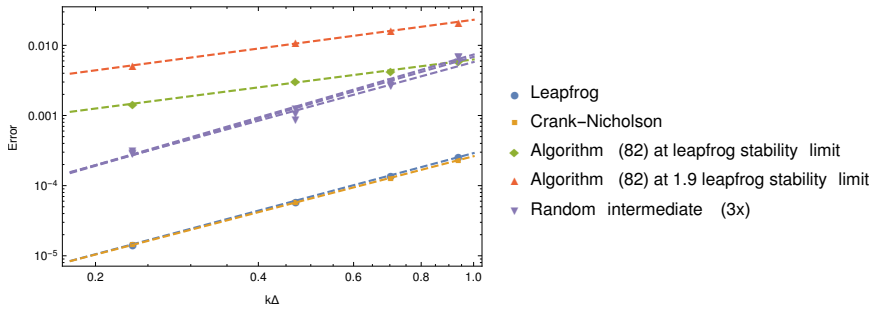

FIG. 10. *Point sampled error for eigenmodes of a square cavity, simulated with various algorithms, vs. $k\Delta$. The slope of the lines gives the order of accuracy: Leapfrog and Crank-Nicholson FDTD (nearly overlapping lines at the bottom) are second-order accurate, while algorithm (82) is first-order accurate. Three randomly-chosen intermediate algorithms also appear to be second-order accurate.*

Crank-Nicolson FDTD conserves the following energy exactly

$$e = \mathbf{b}_-^T[\star_{\mu^{-1}}]\mathbf{b}_- + \mathbf{e}_-^T[\star_\epsilon]\mathbf{e}_- \tag{94}$$

Although we know how to construct an exact conserved energy for (62) (by diagonalizing the time-stepping operator, sections 4.1.2 and 4.3.2), we will plot (94) instead, as is common [10, 2, 15]. For energy-conserving algorithms, this quantity oscillates around a constant value, and the amplitude of this oscillation increases with Δ_t.

In figure 11, we consider the four cases of leapfrog FDTD, Crank-Nicolson FDTD, (62) with randomly chosen R_E, R_B, and algorithm (82). We plot the energy (94) for all four cases. The configuration consists of 10×10 cells with perfectly conducting boundary conditions. The fields were initialized randomly at the first step, the same for each algorithm. All four cases ran at the leapfrog FDTD courant limit.

Clearly, all four cases are stable and energy-conserving, and in fact the intermediate methods have a lower energy oscillation than leapfrog FDTD at the same Δ_t.

5.4. Resonant frequencies of a square cavity. In figure 12, we show the Fourier spectra for fields inside a $10m \times 10m$ square cavity with perfectly conducting boundary conditions. These spectra have peaks near the resonant frequencies of this cavity, the first few of which are shown as vertical dashed lines.

Lower frequencies are better resolved and are thus predicted more accurately (e.g. the peaks at $1.5 \cdot 10^7$ and $2 \cdot 10^7$ Hz). Higher frequencies like the peaks near $3 \cdot 10^7$ and $3.5 \cdot 10^7$ Hz are less well-resolved and are predicted less accurately. In any case, the accuracy of algorithm (82) is comparable to that of leapfrog and Crank-Nicolson FDTD, both when ran at the leapfrog Courant limit ($c\Delta_t = \Delta/\sqrt{2}$) and near its own Courant limit ($c\Delta_t = 1.9\Delta/\sqrt{2}$).

5.5. Transmission through a material in which the local propagation speed is higher. Consider a rectangular waveguide filled with a material with $\epsilon = 4$, i.e. a material in which the local speed of light is half of that in vacuum. Naturally, we can simulate this with leapfrog FDTD, and we can choose the time step based on this reduced local light speed: the time step can be twice larger than it would be if the waveguide was filled with vacuum.

As soon as even a small portion of the waveguide is filled with vacuum (or air),

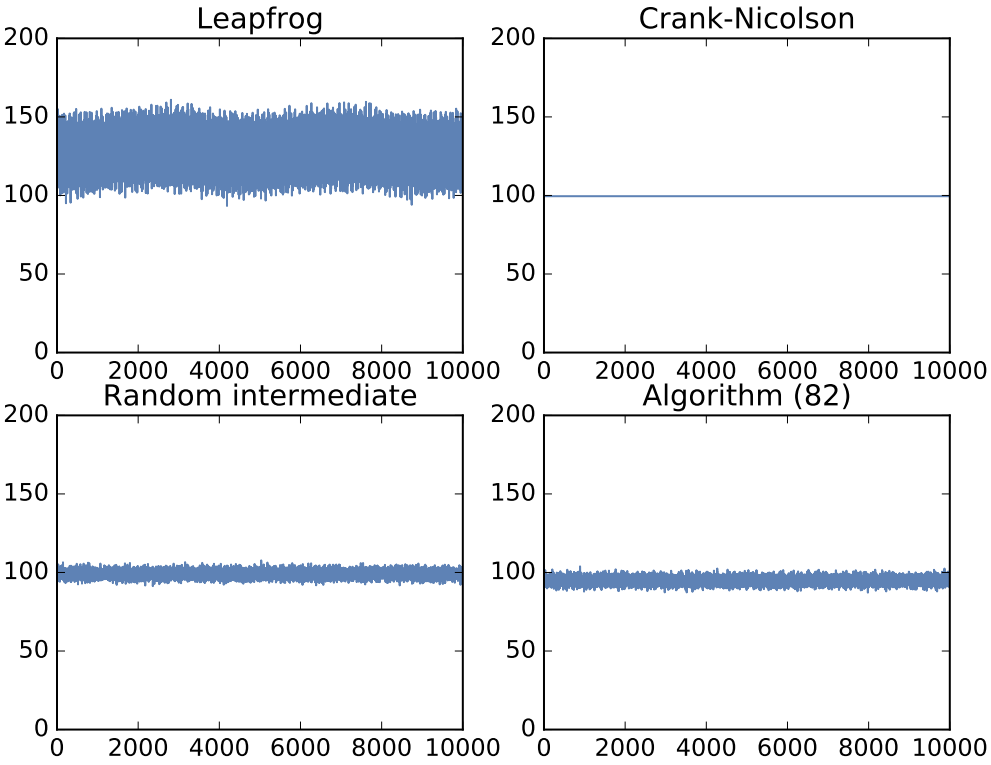

FIG. 11. *Energy vs. time step for leapfrog FDTD, Crank-Nicolson FDTD, and two intermediate algorithms. The energy plotted here, eq. (94), is only exactly conserved by Crank-Nicolson FDTD.*

this ceases to be possible: the FDTD time step must be chosen based on the highest local propagation speed.

In figure 13, we consider exactly this situation: waves propagate from the left, mostly through an $\epsilon = 4$ material, and then they pass through a layer (40 cells wide, located between $x = 40$ and $x = 80$) of $\epsilon = 1$ material. We simulate this configuration in three ways

- Pure leapfrog FDTD: this requires 800 steps, shown in the top figure.
- Leapfrog FDTD in the $\epsilon = 4$ material combined with algorithm (82) in the $\epsilon = 1$ material. We now require only 400 time steps, and only need to repeatedly solve small 3×3 systems of equations.
- Leapfrog FDTD in the $\epsilon = 4$ material combined with Crank-Nicolson FDTD in the $\epsilon = 1$ material. We now require only 400 time steps, but we need to repeatedly solve large $(3 \cdot 40^2) \times (3 \cdot 40^2)$ sparse systems of equations.

Our configuration is 2D, 300 cells wide and 40 cells high, with perfectly conducting (PEC) boundary conditions. There is no absorbing boundary on the right: the waves simply have not had enough time to fully propagate to the right edge. Waves are excited from the left, and have a wavelength in the vertical direction equal to the height of the waveguide.

The difference between the figures is barely visible to the eye, although very close inspection of the vacuum layer in the central figure reveals a "blocky" pattern which is not present in the top and bottom figures, and which is due to the non-constant

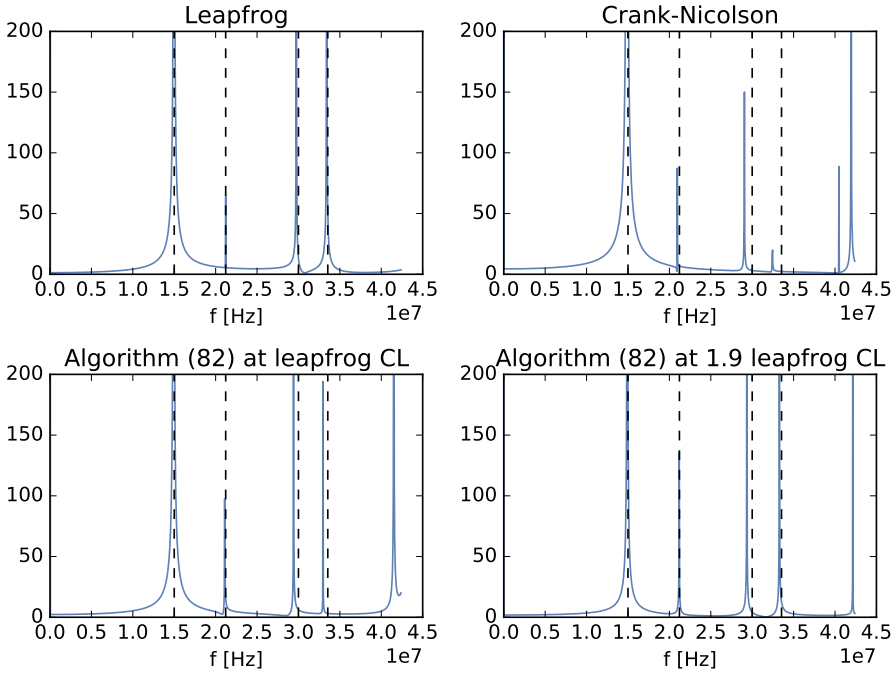

FIG. 12. *Fourier spectra of the fields in a square cavity, as simulated with 3 algorithms. Vertical dashed lines indicate analytical resonant frequencies.*

Algorithm	Transmitted B_z amplitude
Leapfrog	1.141
Leapfrog + algorithm (82)	1.149
Leapfrog + Crank-Nicolson	1.145

TABLE 1
Transmitted wave amplitudes for the cases of figure 13.

temporal locations of the B_z discretisation points.

The numerically obtained amplitude of the transmitted wave is shown in table 1 for the three cases. The difference is on the order of 1%.

5.6. Explicit subgridding. Given that algorithm (82) is twice as stable as FDTD, it is natural to ask if we can combine it with explicit 1:2 FDTD refinement (as in [2]) to obtain an explicit 1:2 FDTD refinement scheme that remains stable up to the Courant limit of the coarse grid (usually, explicit subgridding schemes are stable up to the Courant limit of the fine grid [2], which limits their usefulness). The discretisation grid for such a thing is shown in figure 14. To do this, we use the refinement scheme of [2], which lets us construct a curl operator C on the refined grid, and then use eq. (62) for the time stepping. In the coarse grid, this leads to the standard leapfrog FDTD update equations. In the fine grid, this gives the update equation (82). On the interface, we get two distinct update equations, one involving a 3×3 system of equations to update points at integer multiples of Δ_t (black) and one involving a 4×4 system of equations to update points at half-integer multiples of Δ_t (white with black outline). These equations are (see figure 16, Δ is the coarse

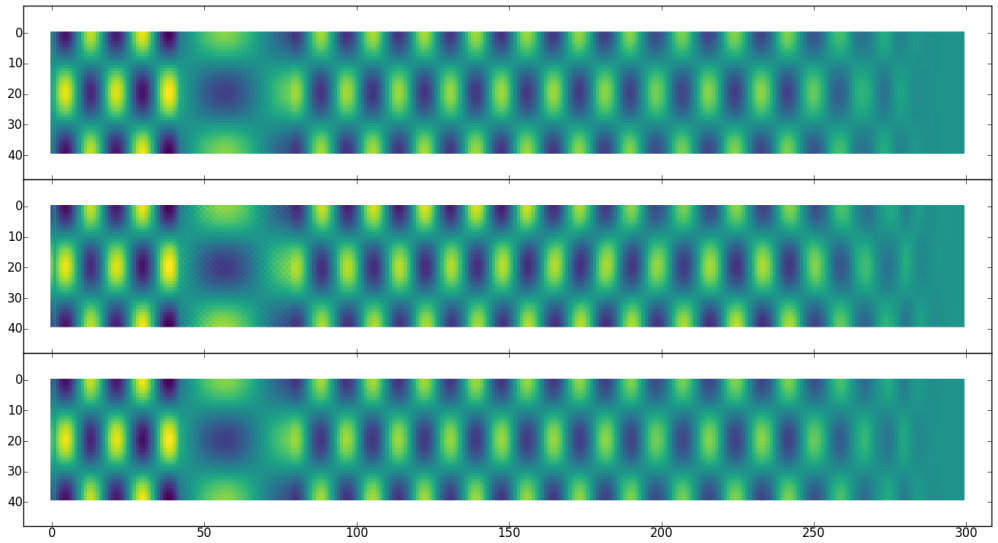

FIG. 13. *B_z for waves in a waveguide with $\epsilon = 4$ being transmitted through a layer of air or vacuum ($\epsilon = 1$, between $x = 40$ and $x = 80$), calculated with leapfrog FDTD everywhere (top), leapfrog FDTD in the $\epsilon = 4$ material and algorithm (82) in the vacuum layer (center), and leapfrog FDTD in the $\epsilon = 4$ material and Crank-Nicolson FDTD in the vacuum layer (bottom).*

discretisation length, $\epsilon_r = \mu_r = 1$):

$$\frac{\begin{bmatrix} B_3^{t+\Delta_t/2} \\ E_4^{t+\Delta_t/2} \\ E_6^{t+\Delta_t/2} \\ E_9^{t+\Delta_t/2} \end{bmatrix} - \begin{bmatrix} B_3^{t-\Delta_t/2} \\ E_4^{t-\Delta_t/2} \\ E_6^{t-\Delta_t/2} \\ E_9^{t-\Delta_t/2} \end{bmatrix}}{c\Delta_t} = \frac{1}{\Delta} \begin{bmatrix} 0 & \frac{-2}{3} & 2 & 2 \\ 2 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ 0 & \frac{4}{3} & 0 & 0 \\ 0 & \frac{-2}{3} & -2 & 0 \\ 0 & 0 & 0 & -2 \\ 2 & 0 & 0 & 0 \end{bmatrix}^T \begin{bmatrix} \frac{B_3^{t-\Delta_t/2} + B_3^{t+\Delta_t/2}}{2} \\ \frac{E_4^{t-\Delta_t/2} + E_4^{t+\Delta_t/2}}{2} \\ \frac{E_6^{t-\Delta_t/2} + E_6^{t+\Delta_t/2}}{2} \\ \frac{E_9^{t-\Delta_t/2} + E_9^{t+\Delta_t/2}}{2} \\ B_1^t \\ B_2^t \\ B_4^t \\ E_7^t \end{bmatrix} \tag{95}$$

$$\frac{\begin{bmatrix} B_2^{t+\Delta_t} \\ E_5^{t+\Delta_t} \\ E_8^{t+\Delta_t} \end{bmatrix} - \begin{bmatrix} B_2^t \\ E_5^t \\ E_8^t \end{bmatrix}}{c\Delta_t} = \frac{1}{\Delta} \begin{bmatrix} -2 & 2 & 2 \\ -2 & 0 & 0 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & -2 \\ 0 & -2 & 0 \end{bmatrix}^T \begin{bmatrix} \frac{B_2^{t+\Delta_t} + B_2^t}{2} \\ \frac{E_5^{t+\Delta_t} + E_5^t}{2} \\ \frac{E_8^{t+\Delta_t} + E_8^t}{2} \\ E_4^{t+\Delta_t/2} \\ E_6^{t+\Delta_t/2} \\ B_5^{t+\Delta_t/2} \\ B_6^{t+\Delta_t/2} \end{bmatrix} \tag{96}$$

Time-stepping the configuration of figure 14 remains just a special case of (62), and is therefore guaranteed to be conditionally stable. Because the grid is non-uniform, it is not possible to determine the stability condition through Fourier analysis, and the analytical calculation of $n(\pi)$ is impractical. However, numerical results (figure 15) indicate that it is indeed stable up to the Courant limit of the coarse grid, as expected. A plot of waves propagating from the coarse FDTD grid into the fine

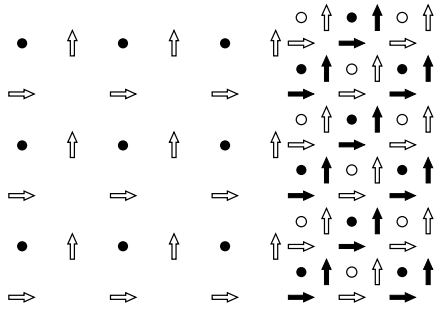

FIG. 14. *Spatial discretisation grid for 1:2 subgridding where the coarse grid is standard leapfrog FDTD and the fine grid is T-shifted as in figure 7 (left).*

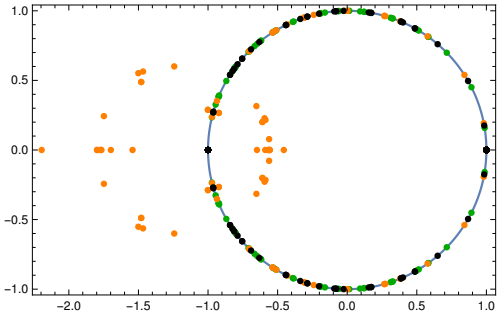

FIG. 15. *Numerically determined eigenvalues of the time-stepping operator for this 1:2 refinement scheme, at the Courant limit of the coarse grid (black), below that Courant limit (green), and above it (unstable, orange).*

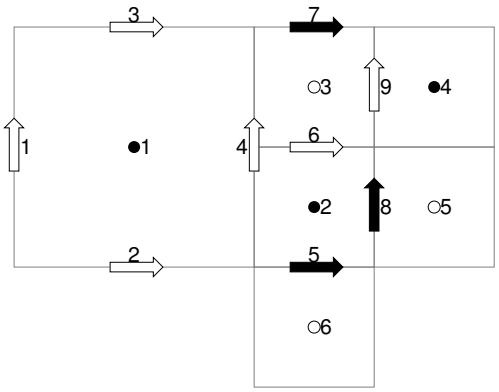

FIG. 16. *A piece of the coarse-fine interface in the configuration of figure 14, with fields labeled for use in (95)-(96). Arrows represent E_x, E_y, circles represent B_z. Solid black indicates discretisation at integer time steps, white with black outline indicates discretisation at half-integer time steps.*

grid is shown in figure 17, and figure 18 confirms that this subgridding scheme is energy-conserving.

6. Conclusion. In this paper we proved that in FDTD and several other FDTD-like methods, we can "T-shift" (change discretisation location along the time axis, replacing explicit updates by implicit ones or vice versa, see figures 3,4,5) an arbitrary set of discretisation points without adversely affecting the stability. Neither spurious gains nor spurious losses are introduced by T-shifting: in the absence of disipative materials, the eigenvalues of the time-stepping operator remain on the unit circle and a discrete energy remains conserved. The order of accuracy of the algorithm, on the other hand, is not guaranteed to remain the same after T-shifting.

Implicit Crank-Nicolson FDTD can be obtained by T-shifting explicit leapfrog FDTD. Both algorithms can be written as special cases of (62). In this sense, we have unified leapfrog and Crank-Nicolson FDTD.

With these results, we have constructed an explicit time-domain algorithm for Maxwell's equations (algorithm (82)) which is twice stabler than leapfrog FDTD but

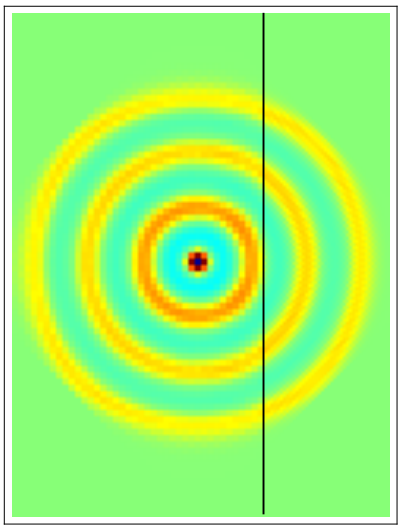

FIG. 17. *Plot of B_z for waves propagating from an FDTD grid (left) into a finer grid (right) as in figure 14. This is stable up to the Courant limit of the coarse grid.*

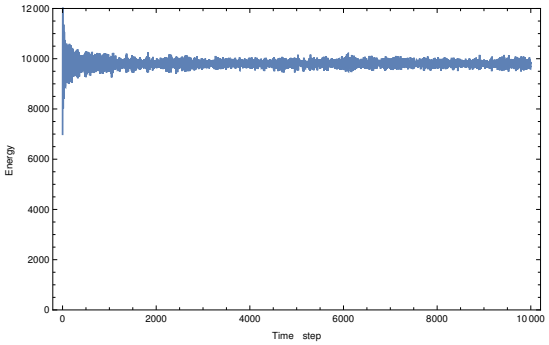

FIG. 18. *Energy for this 1:2 refinement scheme, run at 99% of the coarse Courant limit. See also figure 11.*

first-order accurate. We have also constructed a remarkable explicit 1:2 FDTD refinement scheme that remains stable up to the Courant limit of the coarse grid.

REFERENCES

[1] C. CHANG AND C. D. SARRIS, *A spatially filtered Finite-Difference Time-Domain scheme with controllable stability beyond the CFL limit: Theory and applications*, IEEE Transactions on Microwave Theory and Techniques, 61 (2013), pp. 351–359, doi:10.1109/TMTT.2012.2224670.

[2] 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.

[3] R. A. CHILTON AND R. LEE, *Conservative and provably stable FDTD subgridding*, IEEE Transactions on Antennas and Propagation, 55 (2007), pp. 2537–2549.

[4] R. A. CHILTON AND R. LEE, *The discrete origin of FETD-Newmark late time instability, and a correction scheme*, Journal of Computational Physics, 224 (2007), pp. 1293 – 1306, doi:http://dx.doi.org/10.1016/j.jcp.2006.11.021, //www.sciencedirect.com/science/article/pii/S0021999106005870.

[5] J. DIAZ AND M. GROTE, *Energy conserving explicit local time-stepping for second-order wave equations.*, SIAM Journal on Scientific Computing, (2009), pp. 1985–2014.

[6] S. V. GEORGAKOPOULOS, C. R. BIRTCHER, C. A. BALANIS, AND R. A. RENAUT, *Higher-order finite-difference schemes for electromagnetic radiation, scattering, and penetration. 1. theory*, IEEE Antennas and Propagation Magazine, 44 (2002), pp. 134–142.

[7] M. GROTE AND T. MITKOVA, *Explicit local time-stepping methods for Maxwells equations.*, Journal of Computational and Applied Mathematics, (2010), pp. 3283–3302.

[8] A. V. LONDERSELE, D. D. ZUTTER, AND D. V. GINSTE, *A new hybrid implicit-explicit FDTD method for local subgridding in multiscale 2D-TE scattering problems*, IEEE Transactions on Antennas and Propagation, PP (2016), pp. 1–1, doi:10.1109/TAP.2016.2576477.

[9] S. PIPERNO, *Symplectic local time-stepping in non-dissipative DGTD methods applied to wave propagation problems*, ESAIM: Mathematical Modelling and Numerical Analysis, 40 (2006), pp. 815–841.

[10] R. Rieben, D. White, and G. Rodrigue, *High-order symplectic integration methods for finite element solutions to time dependent Maxwell equations*, IEEE Transactions on Antennas and Propagation, 52 (2004), pp. 2190–2195.

[11] H. Rouf, F. Costen, and S. Garcia, *3D Crank-Nicolson finite difference time domain method for dispersive media*, Electronics letters, 45 (2009), pp. 961–962.

[12] J. Shibayama, M. Muraki, J. Yamauchi, and H. Nakano, *Efficient implicit FDTD algorithm based on locally one-dimensional scheme*, Electronics Letters, 41 (2005), pp. 1046–1047, doi:10.1049/el:20052381.

[13] G. Sun and C. Trueman, *Efficient implementations of the Crank-Nicolson scheme for the finite-difference time-domain method*, Microwave Theory and Techniques, IEEE Transactions on, 54 (2006), pp. 2275–2284.

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

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

[16] W. Tierens and D. D. Zutter, *An unconditionally stable time-domain discretization on Cartesian meshes for the simulation of nonuniform magnetized cold plasma*, Journal of Computational Physics, (2012), pp. 5144–5156.

[17] W. Tierens and D. D. Zutter, *Implicit local refinement for evanescent layers combined with classical FDTD*, IEEE Microwave and Wireless Components Letters, 23 (2013), pp. 225–227.

[18] J. Wang, B. Zhou, L. Shi, C. Gao, and B. Chen, *A novel 3-D HIE-FDTD method with one-step leapfrog scheme*, IEEE Transactions on Microwave Theory and Techniques, 62 (2014), pp. 1275–1283.

[19] R. Weiland and T. Schuhmann, *Conservation of discrete energy and related laws in the finite integration technique*, Progress In Electromagnetics Research, 32 (2001), pp. 301–316.

[20] F. Zheng, Z. Chen, and J. Zhang, *A finite-difference time-domain method without the courant stability conditions*, IEEE Microwave and Guided Wave Letters, 9 (1999), pp. 441–443, doi:10.1109/75.808026.