# 1. Background and Motivation

For our problem, we chose Maxwell's equations for optical waves in one dimension assuming the medium only varries in the x-direction:

$$\mu\partial_t H_z + \partial_x E_y = 0$$

$$\partial_t D + \partial_x H_z = 0$$

where $H$ is the magnetic field with $H_y$ the y-component of the magnetic field, $E$ is the electric field with $E_z$ the z-component of the electric field, and $D$ is the electrix flux density. We will consider a linear flux. In the linear case:

$$D = \varepsilon E_y + P$$

where $\varepsilon$ is the electrical permitivity constant and $P$ is the polarization current density.

We will consider periodic boundary conditions in the domain $x \in [0, 2\pi]$, $t \in [0, T]$, so that $H_z(0,t) = H_z(2\pi, t)$ and $E_y(0,t) = E_y(2\pi, t)$ for all $t > 0$. We will also consider Neumann boundary conditions and absorbing boundary conditions in the domain $x \in [0, 200]$, $t \in [0, T]$.

In [ ]:

# 2. Description of the Numerical Method

We need to start Yee method(FDTD) from one-dimensional Maxwell's equations In 1-D case, all planes parallel to y-z are zeros, so we can have the following PDE:

$$\partial_t H_x = 0$$

$$\partial_t H_y = \frac{1}{\mu}\partial_x E_z$$

$$\partial_t H_z = -\frac{1}{\mu}\partial_x E_y$$

and

$$\partial_t E_x = 0$$

$$\partial_t E_y = -\frac{1}{\varepsilon}\partial_x H_z$$

$$\partial_t E_z = \frac{1}{\mu}\partial_x H_y$$

Combine them together we have

$$\frac{\partial E_y}{\partial t} = -\frac{1}{\varepsilon}\frac{\partial H_z}{\partial x}$$

$$\frac{\partial H_z}{\partial t} = -\frac{1}{\mu}\frac{\partial E_y}{\partial x}$$

or

$$\frac{\partial E_z}{\partial t} = \frac{1}{\varepsilon}\frac{\partial H_y}{\partial x}$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\mu}\frac{\partial E_z}{\partial x}$$

We choose the first two and by central finite difference method

$$\frac{E_i^{k+1} - E_i^k}{\Delta t} = -\frac{1}{\varepsilon}\frac{H_{i+1/2}^{k+1/2} - H_{i-1/2}^{k+1/2}}{\Delta x}$$

$$\frac{H_{i+1/2}^{k+1/2} - H_{i+1/2}^{k-1/2}}{\Delta t} = -\frac{1}{\mu}\frac{E_{i+1}^k - E_i^k}{\Delta x}$$

Let $E$ :=(is defined to be euqal to) $\sqrt{\frac{\varepsilon}{\mu}}E$. Noting that $c = \frac{1}{\sqrt{\mu\varepsilon}}$, our equation becomes

$$\frac{E_i^{k+1} - E_i^k}{\Delta t} = -c\frac{H_{i+1/2}^{k+1/2} - H_{i-1/2}^{k+1/2}}{\Delta x}$$

$$\frac{H_{i+1/2}^{k+1/2} - H_{i+1/2}^{k-1/2}}{\Delta t} = -c\frac{E_{i+1}^k - E_i^k}{\Delta x}$$

So

$$E_i^{k+1} = E_i^k - \frac{c\Delta t}{\Delta x}(H_{i+1/2}^{k+1/2} - H_{i-1/2}^{k+1/2})$$

$$H_{i+1/2}^{k+1/2} = H_{i+1/2}^{k-1/2} - \frac{c\Delta t}{\Delta x}(E_{i+1}^k - E_i^k)$$

There is the yee method for one-dimensional Maxwell's equations

# 3. Theory

Facts:

The FDTD method belongs in the general class of grid-based differential numerical modeling methods (finite difference methods). The time-dependent Maxwell's equations (in partial differential form) are discretized using central-difference approximations to the space and time partial derivatives.

Analysis:

The Yee algorithm is numerically stable for all $\Delta t < \frac{\Delta x}{c}$. [4]

# Consistency Analysis:

Consider the Yee Algorithm update equations at timestep $n$:

$$\begin{cases} H_{j+1/2}^{n+3/2} &= H_{j+1/2}^{n+1/2} - \frac{\Delta t}{\Delta x}\left(E_{j+1}^{n+1} - E_j^{n+1}\right) \\ E_j^{n+1} &= E_j^n - \frac{\Delta t}{\Delta x}\left(H_{j+1/2}^{n+1/2} - H_{j-1/2}^{n+1/2}\right) \end{cases}$$

Let

$$L_{\Delta t}^H(x+\tfrac{\Delta x}{2}, t+\tfrac{\Delta t}{2}) = \frac{1}{\Delta t}\left(H(x+\tfrac{\Delta x}{2}, t+\tfrac{3\Delta t}{2}) - G_{\Delta t}\left(H(\cdot+\tfrac{\Delta x}{2}, t+\tfrac{\Delta t}{2});x\right)\right)$$

Now, from the update equation we have:

$$L_{\Delta t}^H(x+\tfrac{\Delta x}{2}, t+\tfrac{\Delta t}{2}) = \frac{H(x+\tfrac{\Delta x}{2}, t+\tfrac{3\Delta t}{2}) - H(x+\tfrac{\Delta x}{2}, t+\tfrac{\Delta t}{2})}{\Delta t} + \frac{E(x+\Delta x, t+\Delta t) - E(x, t+\Delta t)}{\Delta x}$$

Now, from Taylor's Theorem,

$$H(x+\tfrac{\Delta x}{2}, t+\tfrac{3\Delta t}{2}) = H(x+\tfrac{\Delta x}{2}, t+\Delta t) + \partial_t H(x+\tfrac{\Delta x}{2}, t+\Delta t)\frac{\Delta t}{2} + \frac{\partial_{tt} H(x+\tfrac{\Delta x}{2}, t+\Delta t)}{2}\frac{\Delta t^2}{4} + \frac{\partial_{ttt} H(x+\tfrac{\Delta x}{2}, \eta_1)}{6}\frac{\Delta t^3}{8}$$

for some $\eta_1 \in (t+\Delta t, t+\frac{3\Delta t}{2})$. Also,

$$H(x+\tfrac{\Delta x}{2}, t+\tfrac{\Delta t}{2}) = H(x+\tfrac{\Delta x}{2}, t+\Delta t) - \partial_t H(x+\tfrac{\Delta x}{2}, t+\Delta t)\frac{\Delta t}{2} + \frac{\partial_{tt} H(x+\tfrac{\Delta x}{2}, t+\Delta t)}{2}\frac{\Delta t^2}{4} - \frac{\partial_{ttt} H(x+\tfrac{\Delta x}{2}, \eta_2)}{6}\frac{\Delta t^3}{8}$$

for some $\eta_2 \in (t+\frac{\Delta t}{2}, t+\Delta t)$. Therefore,

$$\frac{H(x+\tfrac{\Delta x}{2}, t+\tfrac{3\Delta t}{2}) - H(x+\tfrac{\Delta x}{2}, t+\tfrac{\Delta t}{2})}{\Delta t} = \partial_t H(x+\tfrac{\Delta x}{2}, t+\Delta t) + (\partial_{ttt} H(x+\tfrac{\Delta x}{2}, \eta_1) + \partial_{ttt} H(x+\tfrac{\Delta x}{2}, \eta_2))\frac{\Delta t^2}{48}$$

. Now, also from Taylor's Theorem:

$$E(x+\Delta x, t+\Delta t) = E(x+\tfrac{\Delta x}{2}, t+\Delta t) + \partial_x E(x+\tfrac{\Delta x}{2}, t+\Delta t)\frac{\Delta x}{2} + \frac{\partial_{xx} E(x+\tfrac{\Delta x}{2}, t+\Delta t)}{2}\frac{\Delta x^2}{4} + \frac{\partial_{xxx} E(\xi_1, t+\Delta t)}{6}\frac{\Delta x^3}{8}$$

for some $\xi_1 \in (x+\frac{\Delta x}{2}, x+\Delta x)$. Also,

$$E(x, t+\Delta t) = E(x+\tfrac{\Delta x}{2}, t+\Delta t) - \partial_x E(x+\tfrac{\Delta x}{2}, t+\Delta t)\frac{\Delta x}{2} + \frac{\partial_{xx}E(x+\frac{\Delta x}{2}, t+\Delta t)}{2}\frac{\Delta x^2}{4} - \frac{\partial_{xxx}E(\xi_2, t+\Delta t)}{6}\frac{\Delta x^3}{8}$$

for some $\xi_2 \in (x, x+\frac{\Delta x}{2})$. Thus,

$$\frac{E(x+\Delta x, t+\Delta t) - E(x, t+\Delta t)}{\Delta x} = \partial_x E(x+\tfrac{\Delta x}{2}, t+\Delta t) + (\partial_{xxx}E(\xi_1, t+\Delta t) + \partial_{xxx}E(\xi_2, t+\Delta t))\frac{\Delta x^2}{48}$$

. Altogether, we have:

$$L^H_{\Delta t}(x+\tfrac{\Delta x}{2}, t+\tfrac{\Delta t}{2}) = \partial_t H(x+\tfrac{\Delta x}{2}, t+\Delta t) + (\partial_{ttt}H(x+\tfrac{\Delta x}{2}, \eta_1) + \partial_{ttt}H(x+\tfrac{\Delta x}{2}, \eta_2))\frac{\Delta t^2}{48} + \partial_x E(x+\tfrac{\Delta x}{2}, t+\Delta t) + (\partial_{xxx}E(\xi_1, t+\Delta t)$$

Now, from the PDE, we know $\partial_t H + \partial_x E = 0$ at any point in the domain. Thus,

$$L^H_{\Delta t}(x+\tfrac{\Delta x}{2}, t+\tfrac{\Delta t}{2}) = (\partial_{ttt}H(x+\tfrac{\Delta x}{2}, \eta_1) + \partial_{ttt}H(x+\tfrac{\Delta x}{2}, \eta_2))\frac{\Delta t^2}{48} + (\partial_{xxx}E(\xi_1, t+\Delta t) + \partial_{xxx}E(\xi_2, t+\Delta t))\frac{\Delta x^2}{48}$$

Now, assuming a constant mesh ratio so that $\frac{\Delta t}{\Delta x} = \lambda \in \mathbb{R}$, we have:

$$L^H_{\Delta t}(x+\tfrac{\Delta x}{2}, t+\tfrac{\Delta t}{2}) = (\partial_{ttt}H(x+\tfrac{\Delta x}{2}, \eta_1) + \partial_{ttt}H(x+\tfrac{\Delta x}{2}, \eta_2))\frac{\Delta t^2}{48} + (\partial_{xxx}E(\xi_1, t+\Delta t) + \partial_{xxx}E(\xi_2, t+\Delta t))\frac{\lambda^2\Delta t^2}{48}$$

Finally, assuming $\partial_{ttt}H$ and $\partial_{xxx}E$ are bounded in our domain by $C_1$ and $C_2$ respectively, we have:

$$L^H_{\Delta t}(x+\tfrac{\Delta x}{2}, t+\tfrac{\Delta t}{2}) \leq \left(\frac{C_1 + C_2\lambda^2}{24}\right)\Delta t^2$$

Therefore, the local trunction error in the update of $H$ is order $\Delta t^2$. Now, we consider $E$. Let

$$L^E_{\Delta t}(x, t) = \frac{1}{\Delta t}(E(x, t+\Delta t) - G_{\Delta t}(E(\cdot, t); x))$$

From the update equation, we have:

$$L^E_{\Delta t}(x,t) = \frac{E(x,t+\Delta t)-E(x,t)}{\Delta t} + \frac{H(x+\frac{\Delta x}{2},t+\frac{\Delta t}{2})-H(x-\frac{\Delta x}{2},t+\frac{\Delta t}{2})}{\Delta x}$$

From Taylor's Theorem, we have:

$$E(x,t+\Delta t) = E(x,t+\tfrac{\Delta t}{2}) + \partial_t E(x,t+\tfrac{\Delta t}{2})\frac{\Delta t}{2} + \frac{\partial_{tt}E(x,t+\frac{\Delta t}{2})}{2}\frac{\Delta t^2}{4} + \frac{\partial_{ttt}E(x,\eta_3)}{6}\frac{\Delta t}{8}$$

for some $\eta_3 \in (t+\frac{\Delta t}{2}, t+\Delta t)$. Also,

$$E(x,t) = E(x,t+\tfrac{\Delta t}{2}) - \partial_t E(x,t+\tfrac{\Delta t}{2})\frac{\Delta t}{2} + \frac{\partial_{tt}E(x,t+\frac{\Delta t}{2})}{2}\frac{\Delta t^2}{4} - \frac{\partial_{ttt}E(x,\eta_4)}{6}\frac{\Delta t}{8}$$

for some $\eta_4 \in (t, t+\frac{\Delta t}{2})$. Thus,

$$\frac{E(x,t+\Delta t)-E(x,t)}{\Delta t} = \partial_t E(x,t+\tfrac{\Delta t}{2}) + (\partial_{ttt}E(x,\eta_3)+\partial_{ttt}E(x,\eta_4))\frac{\Delta t^2}{48}$$

. Also, by Taylor's Theorem:

$$H(x+\tfrac{\Delta x}{2},t+\tfrac{\Delta t}{2}) = H(x,t+\tfrac{\Delta t}{2}) + \partial_x H(x,t+\tfrac{\Delta t}{2})\frac{\Delta x}{2} + \frac{\partial_{xx}H(x,t+\frac{\Delta t}{2})}{2}\frac{\Delta x^2}{4} + \frac{\partial_{xxx}H(\xi_3,t+\frac{\Delta t}{2})}{6}\frac{\Delta x^3}{8}$$

for some $\xi_3 \in (x, x+\frac{\Delta x}{2})$. And,

$$H(x-\tfrac{\Delta x}{2},t+\tfrac{\Delta t}{2}) = H(x,t+\tfrac{\Delta t}{2}) - \partial_x H(x,t+\tfrac{\Delta t}{2})\frac{\Delta x}{2} + \frac{\partial_{xx}H(x,t+\frac{\Delta t}{2})}{2}\frac{\Delta x^2}{4} - \frac{\partial_{xxx}H(\xi_4,t+\frac{\Delta t}{2})}{6}\frac{\Delta x^3}{8}$$

for some $\xi_4 \in (x-\frac{\Delta x}{2}, x)$. Therefore,

$$\frac{H(x+\frac{\Delta x}{2},t+\frac{\Delta t}{2})-H(x-\frac{\Delta x}{2},t+\frac{\Delta t}{2})}{\Delta x} = \partial_x H(x,t+\tfrac{\Delta t}{2}) + (\partial_{xxx}H(\xi_3,t+\tfrac{\Delta t}{2})+\partial_{xxx}H(\xi_4,t+\tfrac{\Delta t}{2}))\frac{\Delta x^2}{48}$$

. Ergo, we have:

$$L^E_{\Delta t}(x,t) = \partial_t E(x, t+\tfrac{\Delta t}{2}) + (\partial_{ttt}E(x,\eta_3) + \partial_{ttt}E(x,\eta_4))\frac{\Delta t^2}{48} + \partial_x H(x, t+\tfrac{\Delta t}{2}) + (\partial_{xxx}H(\xi_3, t+\tfrac{\Delta t}{2}) + \partial_{xxx}H(\xi_4, t+\tfrac{\Delta t}{2}))\frac{\Delta x^2}{48}$$

. Since from the PDE $\partial_t E + \partial_x H = 0$ at any point in the domain,

$$L^E_{\Delta t}(x,t) = (\partial_{ttt}E(x,\eta_3) + \partial_{ttt}E(x,\eta_4))\frac{\Delta t^2}{48} + (\partial_{xxx}H(\xi_3, t+\tfrac{\Delta t}{2}) + \partial_{xxx}H(\xi_4, t+\tfrac{\Delta t}{2}))\frac{\Delta x^2}{48}$$

. Assuming $\partial_{ttt}E$ and $\partial_{xxx}H$ are bounded on the domain by $C_3$ and $C_4$ respectively, and using the same constant mesh ratio, we find:

$$L^E_{\Delta t}(x,t) \leq \left(\frac{C_3 + C_4\lambda^2}{24}\right)\Delta t^2$$

. Thus, the local trunction error for the update of $E$ is also order $\Delta t^2$. Thus, if we consider the vector

$$L_{\Delta t} = \begin{bmatrix} L^H_{\Delta t} \\ L^E_{\Delta t} \end{bmatrix},$$

we find that $||L_{\Delta t}||_\infty \to 0$ as $\Delta t \to 0$, since each component is order $\Delta t^2$ and approches 0. Thus, the scheme is consistent, and we say the method is order $\Delta t^2$.

# Stability Analysis:

Consider the Yee Algorithm update equations at timestep $n$:

$$H^{n+3/2}_{j+1/2} = H^{n+1/2}_{j+1/2} - \frac{\Delta t}{\Delta x}\left(E^{n+1}_{j+1} - E^{n+1}_j\right)$$

$$E^{n+1}_j = E^n_j - \frac{\Delta t}{\Delta x}\left(H^{n+1/2}_{j+1/2} - H^{n+1/2}_{j-1/2}\right)$$

We can replace $E^{n+1}_j$ and $E^{n+1}_{j+1}$ in the first equation:

$$H^{n+3/2}_{j+1/2} = H^{n+1/2}_{j+1/2} - \frac{\Delta t}{\Delta x}\left(\left[E^n_{j+1} - \frac{\Delta t}{\Delta x}\left(H^{n+1/2}_{j+3/2} - H^{n+1/2}_{j+1/2}\right)\right] - \left[E^n_j - \frac{\Delta t}{\Delta x}\left(H^{n+1/2}_{j+1/2} - H^{n+1/2}_{j-1/2}\right)\right]\right)$$

$$H_{j+1/2}^{n+3/2} = H_{j+1/2}^{n+1/2} - \frac{\Delta t}{\Delta x}\left(E_{j+1}^{n} - E_{j}^{n}\right) + \frac{\Delta t^2}{\Delta x^2}\left(H_{j+3/2}^{n+1/2} - 2H_{j+1/2}^{n+1/2} + H_{j-1/2}^{n+1/2}\right)$$

We use the discrete Fourier transform:

$$\sum_{k=0}^{N-1} \hat{H}_k^{n+3/2} e^{i(j+1/2)\Delta xk} = \sum_{k=0}^{N-1} \hat{H}_k^{n+1/2} e^{i(j+1/2)\Delta xk} - \frac{\Delta t}{\Delta x}\left(\sum_{k=0}^{N-1} \hat{E}_k^{n} e^{i(j+1)\Delta xk} - \sum_{k=0}^{N-1} \hat{E}_k^{n} e^{ij\Delta xk}\right) + \frac{\Delta t^2}{\Delta x^2}\Big(\sum_{k=0}^{N-1} \hat{H}_k^{n+1/2} e^{i(j+3/2)\Delta xk} - 2\sum_{k=0}^{N-1} \hat{H}_k^{n+1/2} e^{i(j}$$

$$\sum_{k=0}^{N-1} \hat{H}_k^{n+3/2} e^{i(j+1/2)\Delta xk} = \sum_{k=0}^{N-1} \left[\hat{H}_k^{n+1/2} - \frac{\Delta t}{\Delta x}\hat{E}_k^{n}\left(e^{\frac{1}{2}i\Delta xk} - e^{-\frac{1}{2}i\Delta xk}\right) + \frac{\Delta t^2}{\Delta x^2}\hat{H}_k^{n+1/2}\left(e^{i\Delta xk} - 2 + e^{-i\Delta xk}\right)\right] e^{i(j+1/2)\Delta xk}$$

Then for each $k$ we have:

$$\hat{H}_k^{n+3/2} = \left(1 + \frac{\Delta t^2}{\Delta x^2}\left(e^{i\Delta xk} - 2 + e^{-i\Delta xk}\right)\right)\hat{H}_k^{n+1/2} - \frac{\Delta t}{\Delta x}\left(e^{\frac{1}{2}i\Delta xk} - e^{-\frac{1}{2}i\Delta xk}\right)\hat{E}_k^{n}$$

$$\hat{H}_k^{n+3/2} = \left(1 + \frac{\Delta t^2}{\Delta x^2}\left(2\cos(\Delta xk) - 2\right)\right)\hat{H}_k^{n+1/2} - \frac{\Delta t}{\Delta x}\left(2i\sin\left(\frac{\Delta xk}{2}\right)\right)\hat{E}_k^{n}$$

Now, since $\sin^2 x = \frac{1-\cos 2x}{2}$, we know $\cos 2x = 1 - 2\sin^2 x$. We use this identity and find:

$$\hat{H}_k^{n+3/2} = \left(1 - \frac{4\Delta t^2}{\Delta x^2}\sin^2\left(\frac{\Delta xk}{2}\right)\right)\hat{H}_k^{n+1/2} - \frac{\Delta t}{\Delta x}\left(2i\sin\left(\frac{\Delta xk}{2}\right)\right)\hat{E}_k^{n}$$

Now, for our update equation for the electric field $E$ we consider the discrete Fourier transform:

$$\sum_{k=0}^{N-1} \hat{E}_k^{n+1} e^{ij\Delta xk} = \sum_{k=0}^{N-1} \hat{E}_k^{n} e^{ij\Delta xk} - \frac{\Delta t}{\Delta x}\left(\sum_{k=0}^{N-1} \hat{H}_k^{n+1/2} e^{i(j+1/2)\Delta xk} - \sum_{k=0}^{N-1} \hat{H}_k^{n+1/2} e^{i(j-1/2)\Delta xk}\right)$$

$$\sum_{k=0}^{N-1} \hat{E}_k^{n+1} e^{ij\Delta xk} = \sum_{k=0}^{N-1} \left[\hat{E}_k^{n} - \frac{\Delta t}{\Delta x}\hat{H}_k^{n+1/2}\left(e^{\frac{1}{2}i\Delta xk} - e^{-\frac{1}{2}i\Delta xk}\right)\right] e^{ij\Delta xk}$$

Then for each $k$ we have:

$$\hat{E}_k^{n+1} = \hat{E}_k^n - \frac{\Delta t}{\Delta x}\hat{H}_k^{n+1/2}(e^{\frac{1}{2}i\Delta xk} - e^{-\frac{1}{2}i\Delta xk})$$

$$\hat{E}_k^{n+1} = \hat{E}_k^n - \frac{\Delta t}{\Delta x}(2i\sin(\frac{\Delta xk}{2}))\hat{H}_k^{n+1/2}$$

Now we can put the amplification factors for $H$ and $E$ into a matrix form:

$$\begin{bmatrix} \hat{H}_k^{n+3/2} \\ \hat{E}_k^{n+1} \end{bmatrix} = \begin{bmatrix} 1 - 4\lambda^2\sin^2(\xi) & -2i\lambda\sin(\xi) \\ -2i\lambda\sin(\xi) & 1 \end{bmatrix} \begin{bmatrix} \hat{H}_k^{n+1/2} \\ \hat{E}_k^n \end{bmatrix}$$

Where $\lambda = \frac{\Delta t}{\Delta x}$ and $\xi = \frac{\Delta xk}{2}$. We can further simplify this amplification matrix, called $\hat{Q}$ as

$$\hat{Q} = \begin{bmatrix} 1 - 4\alpha^2 & -2i\alpha \\ -2i\alpha & 1 \end{bmatrix}$$

Where $\alpha = \lambda\sin(\xi)$. Now we consider the eigenvalues $\mu$ of this matrix:

$$\begin{vmatrix} 1 - 4\alpha^2 - \mu & -2i\alpha \\ -2i\alpha & 1 - \mu \end{vmatrix} = 0$$

$$(1 - 4\alpha^2 - \mu)(1 - \mu) - (-2i\alpha)(-2i\alpha) = 0$$

$$1 - 4\alpha^2 - \mu - \mu + 4\alpha^2\mu + \mu^2 - (-4\alpha^2) = 0$$

$$1 + (4\alpha^2 - 2)\mu + \mu^2 = 0$$

Now we can find the eigenvalues by solving this quadratic:

$$\mu = \frac{2 - 4\alpha^2 \pm \sqrt{16\alpha^4 - 16\alpha^2 + 4 - 4}}{2}$$

$$\mu = \frac{2 - 4\alpha^2 \pm 4\sqrt{\alpha^4 - \alpha^2}}{2}$$

$$\mu = 1 - 2\alpha^2 \pm 2\sqrt{\alpha^4 - \alpha^2},$$

or

$$\mu_1 = 1 - 2\alpha^2 + 2\sqrt{\alpha^4 - \alpha^2}, \mu_2 = 1 - 2\alpha^2 - 2\sqrt{\alpha^4 - \alpha^2}$$

For von Neumann stability, we require that each eigenvalue is less than or equal to $1$ in magnitude. First let's consider if $\frac{\Delta t}{\Delta x} > 1$. Recalling that $\alpha = \frac{\Delta t}{\Delta x}\sin(\xi)$, we have:

$$\mu = 1 - 2\frac{\Delta t^2}{\Delta x^2}\sin^2(\xi) \pm 2\sqrt{\left(\frac{\Delta t^2}{\Delta x^2}\sin^2(\xi)\right)^2 - \frac{\Delta t^2}{\Delta x^2}\sin^2(\xi)}$$

Recall that $\Delta x = \frac{2\pi}{N}$. Therefore, $\xi = \frac{\Delta x k}{2} = \frac{2\pi k}{2N} = \frac{\pi k}{N}$ Now, if we consider the Fourier mode when $k = \frac{N}{2}$ we have $\xi = \frac{\pi}{2}$. Thus for this mode, $\sin(\xi) = 1$. Therefore, considering the magnitude of $\mu_2$ we have:

$$|\mu_2| = \left|1 - 2\frac{\Delta t^2}{\Delta x^2} - 2\sqrt{\left(\frac{\Delta t^2}{\Delta x^2}\right)^2 - \frac{\Delta t^2}{\Delta x^2}}\right|$$

Now, since $\frac{\Delta t}{\Delta x} > 1$, $\frac{\Delta t^2}{\Delta x^2} > \frac{\Delta t}{\Delta x}$, and so $\frac{\Delta t^2}{\Delta x^2} - \frac{\Delta t}{\Delta x} > 0$. Therefore,

$$|\mu_2| = \left|1 - 2\frac{\Delta t^2}{\Delta x^2} - 2\sqrt{\left(\frac{\Delta t^2}{\Delta x^2}\right)^2 - \frac{\Delta t^2}{\Delta x^2}}\right| > \left|1 - 2 - 2\sqrt{\varepsilon}\right| = \left|-1 - 2\sqrt{\varepsilon}\right| > 1,$$

where $\varepsilon = \frac{\Delta t^2}{\Delta x^2} - \frac{\Delta t}{\Delta x}$. Therefore, the Yee Scheme is not von Neumann stable for $\frac{\Delta t}{\Delta x} > 1$.

Now, consider the other case that $0 < \frac{\Delta t}{\Delta x} \leq 1$. We know $0 \leq \sin^2(\xi) \leq 1$ for all $\xi$, and since $0 \leq \frac{\Delta t}{\Delta x} \leq 1, 0 \leq \frac{\Delta t^2}{\Delta x^2} \leq 1$. Therefore, we know $0 \leq \frac{\Delta t^2}{\Delta x^2}\sin^2(\xi) \leq 1$. With this we consider the term under the square root for $\mu$. If $\frac{\Delta t^2}{\Delta x^2}\sin^2(\xi) = 0$ or $\frac{\Delta t^2}{\Delta x^2}\sin^2(\xi) = 1$, then

$$\left(\frac{\Delta t^2}{\Delta x^2}\sin^2(\xi)\right)^2 - \frac{\Delta t^2}{\Delta x^2}\sin^2(\xi) = 0$$

Otherwise, if $0 < \frac{\Delta t^2}{\Delta x^2}\sin^2(\xi) < 1$ then $\left(\frac{\Delta t^2}{\Delta x^2}\sin^2(\xi)\right)^2 < \frac{\Delta t^2}{\Delta x^2}\sin^2(\xi)$, and so:

$$(\frac{\Delta t^2}{\Delta x^2}\sin^2(\xi))^2 - \frac{\Delta t^2}{\Delta x^2}\sin^2(\xi) < 0$$

Therefore we will have $\mu_1$ and $\mu_2$ complex in this case. Then to find the magnitude of both $\mu$ we use the complex modulus:

$$|\mu| = \left(\left[1-2\alpha^2\right]^2 + \left[2\sqrt{\alpha^4-\alpha^2}\right]^2\right)^{1/2}$$

$$|\mu| = 1 - 4\alpha^2 + 4\alpha^4 + 4(\alpha^4 - \alpha^2) = 1$$

Thus, in the case that $0 < \frac{\Delta t}{\Delta x} \leq 1$ we have the von Neumann condition satisfied.

However, the Neumann condition is only a sufficient condition for stability in the case that the amplification matrix is uniformly diagonalizable. First, let us consider the case that $0 < \frac{\Delta t}{\Delta x} < 1$. First, we find the eigenvectors $\mathbf{v}_{1,2}$ of the amplification matrix associated with each eigenvalue:

$$(\hat{Q} - I\mu_1)\mathbf{v}_1 = \begin{bmatrix} 1-4\alpha^2-(1-2\alpha^2+2\sqrt{\alpha^4-\alpha^2}) & -2i\alpha \\ -2i\alpha & 1-(1-2\alpha^2+2\sqrt{\alpha^4-\alpha^2}) \end{bmatrix} \begin{bmatrix} v_1^{(1)} \\ v_1^{(2)} \end{bmatrix} = 0$$

$$\begin{bmatrix} -2\alpha^2-2\sqrt{\alpha^4-\alpha^2} & -2i\alpha \\ -2i\alpha & 2\alpha^2-2\sqrt{\alpha^4-\alpha^2} \end{bmatrix} \begin{bmatrix} v_1^{(1)} \\ v_1^{(2)} \end{bmatrix} = 0$$

This yields the system of equations:

$$\begin{cases} (-2\alpha^2-2\sqrt{\alpha^4-\alpha^2})v_1^{(1)} - 2i\alpha v_1^{(2)} & = 0 \\ -2i\alpha v_1^{(1)} + (2\alpha^2-2\sqrt{\alpha^4-\alpha^2})v_1^{(2)} & = 0 \end{cases}$$

Solving the second equation we find:

$$v_1^{(1)} = \frac{-2\alpha^2+2\sqrt{\alpha^4-\alpha^2}}{-2i\alpha}v_1^{(2)}$$

$$v_1^{(1)} = \left(-i\alpha + i\frac{\sqrt{\alpha^4-\alpha^2}}{\alpha}\right)v_1^{(2)}$$

Since we assume $0 < \frac{\Delta t}{\Delta x} < 1$, we know $\alpha^4 < \alpha^2$ as before. Therefore, we can rewrite $\sqrt{\alpha^4 - \alpha^2} = i\sqrt{\alpha^2 - \alpha^4}$ where $\sqrt{\alpha^2 - \alpha^4} \in \mathbb{R}$. So, we have the first eigenvector:

$$\mathbf{v}_1 = \begin{bmatrix} -\frac{\sqrt{\alpha^2-\alpha^4}}{\alpha} - i\alpha \\ 1 \end{bmatrix}$$

Similarly, for $\mathbf{v}_2$:

$$\begin{bmatrix} 1 - 4\alpha^2 - (1 - 2\alpha^2 - 2\sqrt{\alpha^4 - \alpha^2}) & -2i\alpha \\ -2i\alpha & 1 - (1 - 2\alpha^2 - 2\sqrt{\alpha^4 - \alpha^2}) \end{bmatrix} \begin{bmatrix} v_2^{(1)} \\ v_2^{(2)} \end{bmatrix} = 0$$

$$\begin{bmatrix} -2\alpha^2 + 2\sqrt{\alpha^4 - \alpha^2} & -2i\alpha \\ -2i\alpha & 2\alpha^2 + 2\sqrt{\alpha^4 - \alpha^2} \end{bmatrix} \begin{bmatrix} v_2^{(1)} \\ v_2^{(2)} \end{bmatrix} = 0$$

Which yields this system:

$$\begin{cases} (-2\alpha^2 + 2\sqrt{\alpha^4 - \alpha^2})v_2^{(1)} - 2i\alpha v_2^{(2)} & = 0 \\ -2i\alpha v_2^{(1)} + (2\alpha^2 + 2\sqrt{\alpha^4 - \alpha^2})v_2^{(2)} & = 0 \end{cases}$$

Again solving the second equation, we find:

$$v_2^{(1)} = \frac{2\alpha^2 + 2\sqrt{\alpha^4 - \alpha^2}}{-2i\alpha} v_2^{(2)}$$

$$v_2^{(1)} = \left(-i\alpha + \frac{\sqrt{\alpha^2 - \alpha^4}}{\alpha}\right) v_2^{(2)}$$

Thus, the second eigenvector is:

$$\mathbf{v}_2 = \begin{bmatrix} \frac{\sqrt{\alpha^2-\alpha^4}}{\alpha} - i\alpha \\ 1 \end{bmatrix}$$

Let $\beta = \frac{\sqrt{\alpha^2-\alpha^4}}{\alpha}$, and we form the matrix $S = (\mathbf{v}_1, \mathbf{v}_2)$, so

$$S = \begin{bmatrix} -\beta - i\alpha & \beta - i\alpha \\ 1 & 1 \end{bmatrix}$$

Then,

$$S^{-1} = \frac{1}{-2\beta} \begin{bmatrix} 1 & -\beta + i\alpha \\ -1 & -\beta - i\alpha \end{bmatrix}$$

We now wish to bound the norms of $S$ and $S^{-1}$. We know $||S||_2^2 = \rho(S^*S)$, where $S^*$ is the conjugate transpose of $S$. Let us calculate the eigenvalues of $S^*S$:

$$S^* = \begin{bmatrix} -\beta + i\alpha & 1 \\ \beta + i\alpha & 1 \end{bmatrix}$$

Then,

$$S^*S = \begin{bmatrix} -\beta + i\alpha & 1 \\ \beta + i\alpha & 1 \end{bmatrix} \begin{bmatrix} -\beta - i\alpha & \beta - i\alpha \\ 1 & 1 \end{bmatrix}$$

$$S^*S = \begin{bmatrix} \beta^2 + \alpha^2 + 1 & -\beta^2 + 2i\alpha\beta + \alpha^2 + 1 \\ -\beta^2 - 2i\alpha\beta + \alpha^2 + 1 & \beta^2 + \alpha^2 + 1 \end{bmatrix}$$

Now, since $\beta = \frac{\sqrt{\alpha^2-\alpha^4}}{\alpha}$,

$$\beta^2 = \frac{\alpha^2 - \alpha^4}{\alpha^2} = 1 - \alpha^2$$

So, we have:

$$S^*S = \begin{bmatrix} 2 & 2\alpha^2 + 2i\alpha\beta \\ 2\alpha^2 - 2i\alpha\beta & 2 \end{bmatrix}$$

Now, to find the eigenvalues of $S^*S$:

$$\det(S^*S - I\lambda) = \begin{vmatrix} 2-\lambda & 2\alpha^2 + 2i\alpha\beta \\ 2\alpha^2 - 2i\alpha\beta & 2-\lambda \end{vmatrix} = 0$$

$$(2-\lambda)(2-\lambda) - (2\alpha^2 + 2i\alpha\beta)(2\alpha^2 - 2i\alpha\beta) = 0$$

$$\lambda^2 - 4\lambda + 4 - 4\alpha^4 - 4\alpha^2\beta^2 = 0$$

$$\lambda^2 - 4\lambda + 4 - 4\alpha^4 - 4\alpha^2(1-\alpha^2) = 0$$

$$\lambda^2 - 4\lambda + 4 - 4\alpha^2 = 0$$

$$\lambda = \frac{4 \pm \sqrt{16 - 16 + 16\alpha^2}}{2}$$

$$\lambda = 2 \pm 2\sqrt{\alpha^2}$$

$$\lambda = 2 \pm 2|\alpha|$$

Now, since $0 \leq \alpha < 1$, $|\lambda| < 4$, and so therefore we have $||S||_2^2 = \rho(S^*S) < 4$. Thus, the norm of $S$ is bounded. Now, we consider $||S^{-1}||$. Since $||S^{-1}||_2^2 = \rho(S^{-1^*}S^{-1})$, we now seek the eigenvalues of $S^{-1^*}S^{-1}$:

$$S^{-1} = \frac{1}{-2\beta}\begin{bmatrix} 1 & -\beta + i\alpha \\ -1 & -\beta - i\alpha \end{bmatrix}$$

Then,

$$S^{-1^*} = \frac{1}{-2\beta}\begin{bmatrix} 1 & -1 \\ -\beta - i\alpha & -\beta + i\alpha \end{bmatrix}$$

And so,

$$S^{-1^*}S^{-1} = \frac{1}{4\beta^2}\begin{bmatrix} 1 & -1 \\ -\beta - i\alpha & -\beta + i\alpha \end{bmatrix}\begin{bmatrix} 1 & -\beta + i\alpha \\ -1 & -\beta - i\alpha \end{bmatrix}$$

$$S^{-1^*}S^{-1} = \frac{1}{4\beta^2}\begin{bmatrix} 2 & 2i\alpha \\ -2i\alpha & 2\beta^2 + 2\alpha^2 \end{bmatrix}$$

$$S^{-1^*}S^{-1} = \frac{1}{4\beta^2}\begin{bmatrix} 2 & 2i\alpha \\ -2i\alpha & 2 \end{bmatrix}$$

$$S^{-1^*}S^{-1} = \begin{bmatrix} \frac{1}{2\beta^2} & \frac{i\alpha}{2\beta^2} \\ -\frac{i\alpha}{2\beta^2} & \frac{1}{2\beta^2} \end{bmatrix}$$

Now, we find the eigenvalues:

$$\det(S^{-1^*}S^{-1} - \lambda I) = \begin{vmatrix} \frac{1}{2\beta^2} - \lambda & \frac{i\alpha}{2\beta^2} \\ -\frac{i\alpha}{2\beta^2} & \frac{1}{2\beta^2} - \lambda \end{vmatrix} = 0$$

$$\left(\tfrac{1}{2\beta^2} - \lambda\right)\left(\tfrac{1}{2\beta^2} - \lambda\right) - \left(\tfrac{i\alpha}{2\beta^2}\right)\left(-\tfrac{i\alpha}{2\beta^2}\right) = 0$$

$$\lambda^2 - \tfrac{\lambda}{\beta^2} + \tfrac{1}{4\beta^2} - \tfrac{\alpha^2}{4\beta^2} = 0$$

$$\lambda^2 - \tfrac{\lambda}{\alpha^2} + \tfrac{1-\alpha^2}{4\beta^4} = 0$$

$$\lambda^2 - \tfrac{\lambda}{\alpha^2} + \tfrac{\beta^2}{4\beta^4} = 0$$

$$\tfrac{1}{\beta^2}\left(\beta^2\lambda^2 - \lambda + \tfrac{1}{4}\right) = 0$$

Recalling that $\beta^2 = 1 - \alpha^2$ and since we assume $0 \leq \alpha^2 < 1$, we know $0 < \beta^2 \leq 1$. Thus, we have:

$$\lambda = \frac{1 \pm \sqrt{1 - \beta^2}}{2\beta^2}$$

$$\lambda = \frac{1 \pm \sqrt{\alpha^2}}{2(1 - \alpha^2)}$$

$$\lambda = \frac{1 \pm |\alpha|}{2(1 - \alpha^2)}$$

Then, since $\alpha^2 < 1$, the denominator is non-zero, and so the eigenvalues are bounded.

Now, we can verify the matrix $S^{-1}\hat{Q}S$:

$$S^{-1}\hat{Q}S = \frac{1}{-2\beta}\begin{bmatrix} 1 & -\beta + i\alpha \\ -1 & -\beta - i\alpha \end{bmatrix}\begin{bmatrix} 1-4\alpha^2 & -2i\alpha \\ -2i\alpha & 1 \end{bmatrix}\begin{bmatrix} -\beta - i\alpha & \beta - i\alpha \\ 1 & 1 \end{bmatrix}$$

$$S^{-1}\hat{Q}S = \frac{1}{-2\beta}\begin{bmatrix} 1 & -\beta + i\alpha \\ -1 & -\beta - i\alpha \end{bmatrix}\begin{bmatrix} -\beta + 4\alpha^2\beta - 3i\alpha + 4i\alpha^2 & \beta - 4\alpha^2\beta - 3i\alpha + 4i\alpha^2 \\ 1 + 2i\alpha\beta - 2\alpha^2 & 1 - 2i\alpha\beta - 2\alpha^2 \end{bmatrix}$$

$$S^{-1}\hat{Q}S = \frac{1}{-2\beta}\begin{bmatrix} -2\beta + 4\alpha^2\beta + 2i\alpha^3 - 2i\alpha - 2i\alpha\beta^2 & 2i\alpha^3 - 2i\alpha + 2i\alpha\beta^2 \\ -2i\alpha^3 + 2i\alpha - 2i\alpha\beta^2 & -2\beta + 4\alpha^2\beta - 2i\alpha^3 + 2i\alpha + 2i\alpha\beta^2 \end{bmatrix}$$

$$S^{-1}\hat{Q}S = \frac{1}{-2\beta}\begin{bmatrix} -2\beta + 4\alpha^2\beta + 2i\alpha^3 - 2i\alpha - 2i\alpha(1-\alpha^2) & 2i\alpha^3 - 2i\alpha + 2i\alpha(1-\alpha^2) \\ -2i\alpha^3 + 2i\alpha - 2i\alpha(1-\alpha^2) & -2\beta + 4\alpha^2\beta - 2i\alpha^3 + 2i\alpha + 2i\alpha(1-\alpha^2) \end{bmatrix}$$

$$S^{-1}\hat{Q}S = \frac{1}{-2\beta}\begin{bmatrix} -2\beta + 4\alpha^2\beta + 4i\alpha^3 - 4i\alpha & 0 \\ 0 & -2\beta + 4\alpha^2\beta - 4i\alpha^3 + 4i\alpha \end{bmatrix}$$

$$S^{-1}\hat{Q}S = \begin{bmatrix} 1 - 2\alpha^2 - 2\frac{\alpha - \alpha^3}{\beta}i & 0 \\ 0 & 1 - 2\alpha^2 + 2\frac{\alpha - \alpha^3}{\beta}i \end{bmatrix}$$

$$S^{-1}\hat{Q}S = \begin{bmatrix} 1 - 2\alpha^2 - 2\frac{\alpha^2 - \alpha^4}{\sqrt{\alpha^2 - \alpha^4}}i & 0 \\ 0 & 1 - 2\alpha^2 + 2\frac{\alpha^2 - \alpha^4}{\sqrt{\alpha^2 - \alpha^4}}i \end{bmatrix}$$

$$S^{-1}\hat{Q}S = \begin{bmatrix} 1-2\alpha^2 - 2\sqrt{\alpha^2-\alpha^4}i & 0 \\ 0 & 1-2\alpha^2 + 2\sqrt{\alpha^2-\alpha^4}i \end{bmatrix}$$

So, we can see that $S$ and $S^{-1}$ are the diagonalizers of $\hat{Q}$, where $S^{-1}\hat{Q}S = \Lambda = \text{diag}(\mu_1, \mu_2)$. Therefore,

$$||\hat{Q}||_2 = ||S\Lambda S^{-1}||_2 \leq ||S||_2||\Lambda||_2||S^{-1}||_2 \leq C$$

for some constant C, as we have previously bounded these norms, in the case that $0 < \frac{\Delta t}{\Delta x} < 1$.

Now, consider the case that $\frac{\Delta t}{\Delta x} = 1$. In this case, consider the Fourier mode when $k = \frac{N}{2}$, so that $\xi = \frac{\pi}{2}$, and thus $\alpha = \frac{\Delta t}{\Delta x}\sin(\xi) = 1$. Now, in this case we have:

$$\hat{Q} = \begin{bmatrix} -3 & -2i \\ -2i & 1 \end{bmatrix} = -I + \begin{bmatrix} -2 & -2i \\ -2i & 2 \end{bmatrix} = -I + B$$

Now, consider $B^2$:

$$B^2 = \begin{bmatrix} -2 & -2i \\ -2i & 2 \end{bmatrix}\begin{bmatrix} -2 & -2i \\ -2i & 2 \end{bmatrix} = \begin{bmatrix} 4-4 & 4i-4i \\ 4i-4i & -4+4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Therefore, $B^n = 0$ for all $n \in \mathbb{N}, n \geq 2$. However, we can now use the binomal theorem:

$$\hat{Q}^n = (-I+B)^n = \sum_{k=0}^{n} \binom{n}{k} B^k(-I)^{n-k}$$

$$\hat{Q}^n = \binom{n}{0} B^0(-I)^n + \binom{n}{1} B(-I)^{n-1}$$

$$\hat{Q}^n = (-1)^n I + nB(-1)^{n-1}I$$

Now, since $(-1)^2 = 1$,

$$\hat{Q}^n = (-1)^n\left(I - nB\right) = (-1)^n \begin{bmatrix} 1+2n & 2ni \\ 2ni & 1-2n \end{bmatrix}$$

We see that the entries in $\hat{Q}^n$ grow linearly in $n$, and so this matrix cannot be power bounded. Therefore, the method is unstable for $\frac{\Delta t}{\Delta x} = 1$.

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import display, clear_output
import time
```

# 4. Implementation of standing mode

In [2]:
```
def yee_mesh(xl, xr, N): #creates the meshnodes for periodic boundary conditions
    #inputs:
    # xl: the left endpoint of the interval
    # xr: the right endpoint of the interval
    # N: the number of mesh nodes for the electric field grid
    ##########################################################
    #outputs:
    # h: the spacing between meshnodes
    # x_E: the meshnodes for the electric field
    # x_H: the meshnodes for the magnetic field

    h = (xr-xl)/(N-1)
    x_E = np.linspace(xl, xr, N)
    x_H = np.linspace(xl+h/2, xr+h/2, N)
    return h, x_E, x_H
```

In [3]:
```
def yee1D(x_E, x_H, h, E0, H0, t0, T, cfl): #calculate the electric and magnetic fields for a
                                            #non-disapative, lossless, isotropic material
    #inputs:
    # x_E: the meshnodes for the electric field
    # x_H: the meshnodes for the magnetic field
    # h: the spacing between meshnodes
    # E0: the initial condition for the electric field
    # H0: the initial condition for the magnetic field
    # t0: the initial time
    # T: the final time to compute the electric field
    # cfl: the mesh ratio
```

```python
    E = np.copy(E0)
    H = np.copy(H0)
    t = t0
    while t < T:
        H_bdy = H[0] - cfl*(E[1] - E[0])
        H[1:-1] = H[1:-1] - cfl*(E[2:] - E[1:-1])
        H[0] = H_bdy
        H[-1] = H_bdy
        E_bdy = E[0] - cfl*(H[0] - H[-2])
        E[1:-1] = E[1:-1] - cfl*(H[1:-1] - H[:-2])
        E[0] = E_bdy
        E[-1] = E_bdy
        t += cfl*h
    return H, E, t
```

# 5. Verification

In [4]:

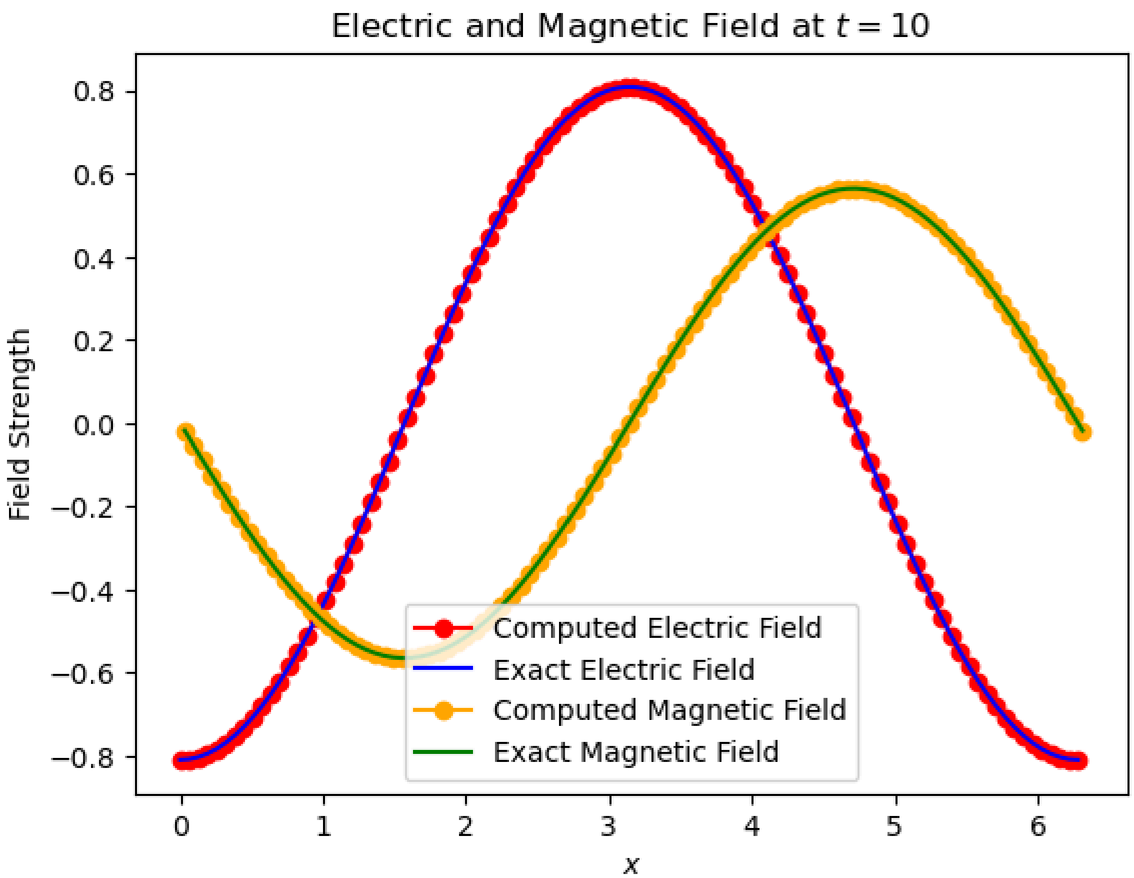
```python
cfl = 0.9
E0 = lambda x: np.cos(x)  ##it is the same thing as def E0(x): return np.cos(x)
H0 = lambda x, cfl, h: np.sin(x)*np.sin(-cfl*h/2)
E_ex = lambda x, t: np.cos(x)*np.cos(t)
H_ex = lambda x, t: np.sin(x)*np.sin(t)
h, x_E, x_H = yee_mesh(0, 2*np.pi, 100)
H, E, t = yee1D(x_E, x_H, h, E0(x_E), H0(x_H,cfl,h), 0, 10, cfl)
```

In [5]:

```python
plt.plot(x_E, E, color='red', marker='o', label='Computed Electric Field')
plt.plot(x_E, E_ex(x_E, t), color='blue', label='Exact Electric Field')
plt.plot(x_H, H, color='orange', marker='o', label='Computed Magnetic Field')
plt.plot(x_H, H_ex(x_H, t-cfl*h/2), color='green', label='Exact Magnetic Field')
plt.xlabel('$x$')
plt.ylabel('Field Strength')
plt.legend()
plt.title('Electric and Magnetic Field at $t=10$')
```

Out[5]:

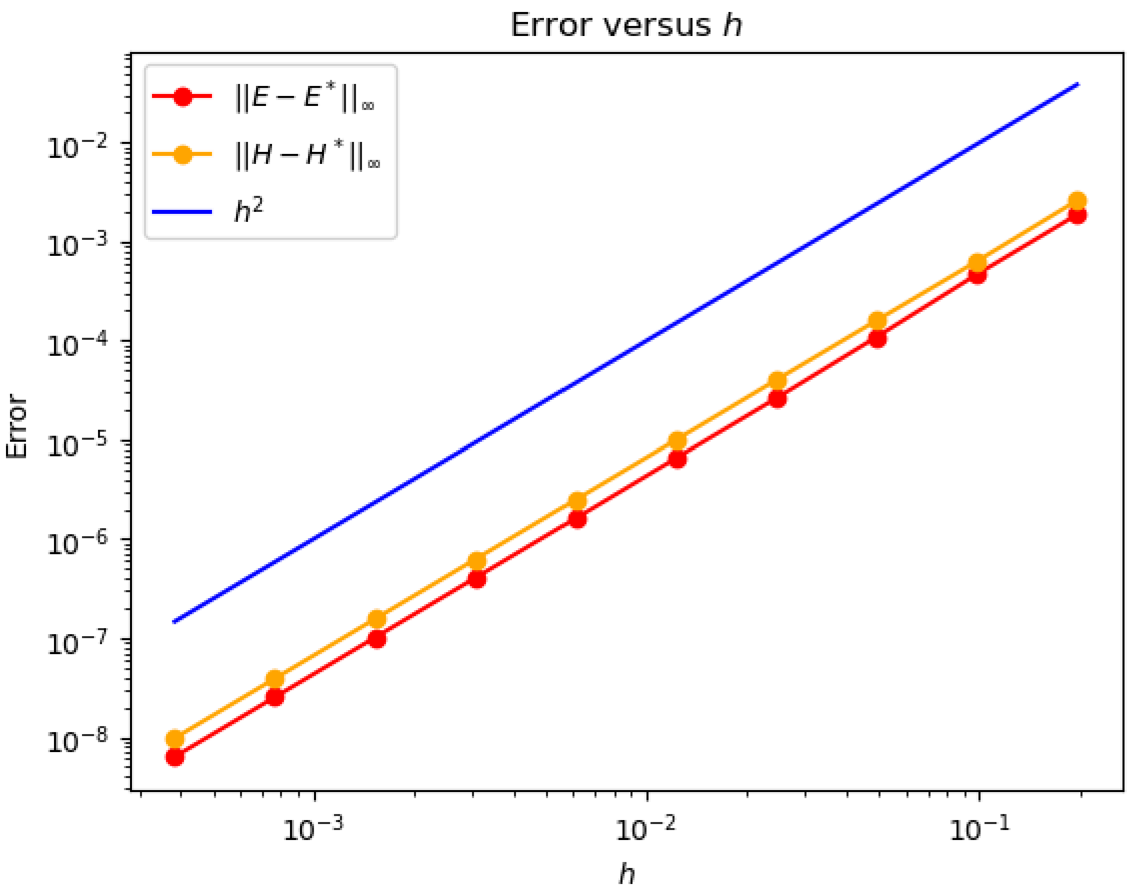
```
Text(0.5, 1.0, 'Electric and Magnetic Field at $t=10$')
```




In [6]:
```
N = [2**k+1 for k in range(5, 15)]
N = np.array(N)
h_vals = 2*np.pi/(N-1)
cfl = 0.9
e = np.zeros((len(N),2))
E_ex = lambda x, t: np.cos(x)*np.cos(t)
H_ex = lambda x, t: np.sin(x)*np.sin(t)
for i in range(10):
    h, x_E, x_H = yee_mesh(0, 2*np.pi, N[i])
    E0 = lambda x: np.cos(x)
    H0 = lambda x, cfl, h: np.sin(x)*np.sin(-cfl*h/2)
    H, E, t = yee1D(x_E, x_H, h, E0(x_E), H0(x_H,cfl,h), 0, 10, cfl)
    E_exact = E_ex(x_E, t)
```

```python
    H_exact = H_ex(x_H, t-cfl*h/2)
    e[i,0] = np.linalg.norm(E-E_exact, np.inf)
    e[i,1] = np.linalg.norm(H-H_exact, np.inf)
plt.loglog(h_vals, e[:,0], color='red', marker='o', label=r'$||E-E^*||_{\infty}$')
plt.loglog(h_vals, e[:,1], color='orange', marker='o', label=r'$||H-H^*||_{\infty}$')
plt.loglog(h_vals, h_vals**2, color='blue', label='$h^2$')
plt.legend()
plt.title("Error versus $h$")
plt.xlabel('$h$')
plt.ylabel('Error')
```

Out[6]: Text(0, 0.5, 'Error')



# 6. Application

In [7]:

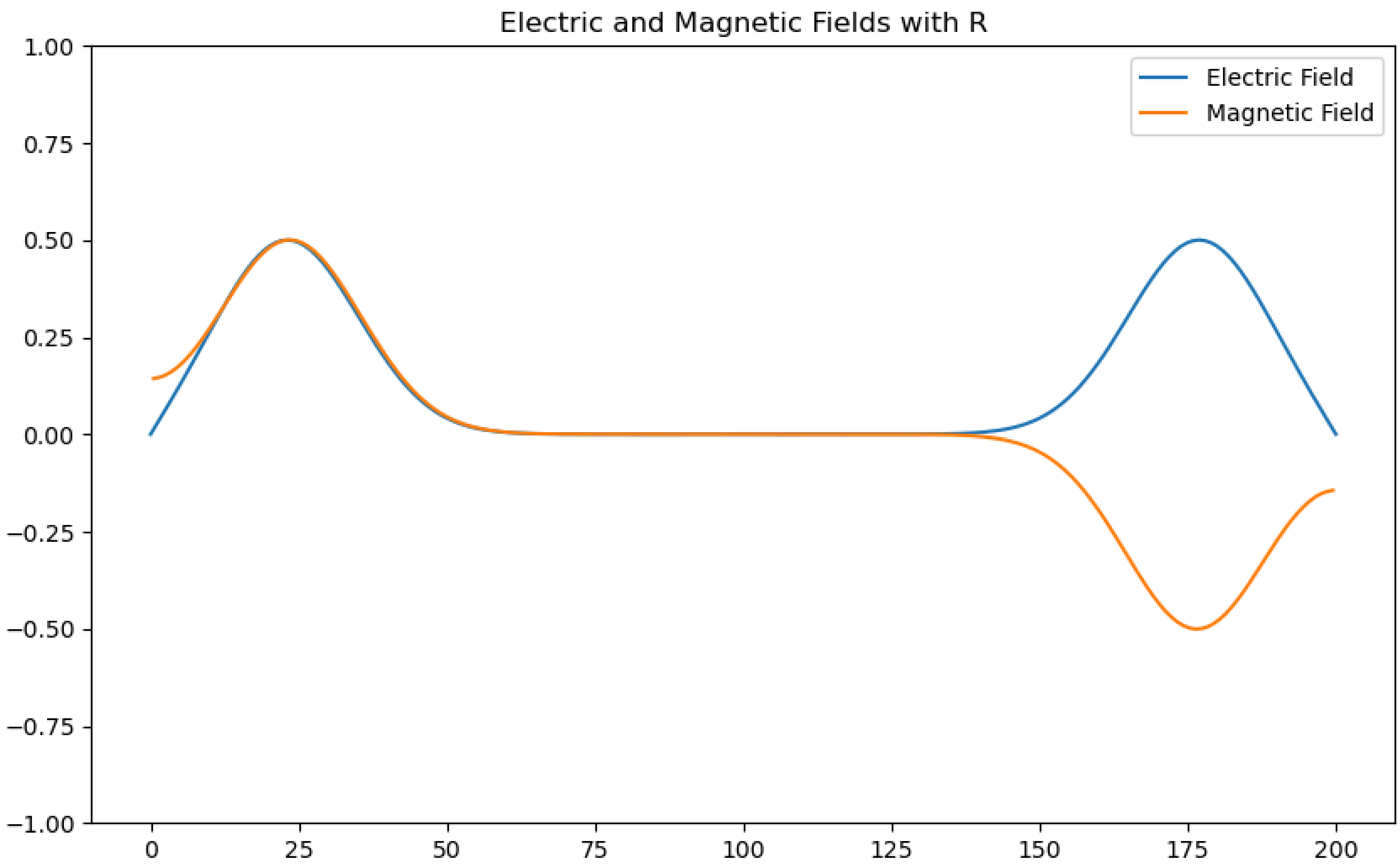
```
def yee_mesh_np(xl, xr, N): #creates the yee mesh for non-periodic boundary conditions
    #inputs:
    # xl: the left endpoint of the interval
    # xr: the right endpoint of the interval
    # N: the number of mesh nodes for the electric field grid
    ##########################################################
    #outputs:
    # h: the spacing between meshnodes
    # x_E: the meshnodes for the electric field
    # x_H: the meshnodes for the magnetic field

    h = (xr-xl)/ (N-1)
    x_E = np.linspace(xl, xr, N)
    x_H = np.linspace(xl+h/2, xr-h/2, N-1)
    return h, x_E, x_H
```

## Neumann Boundary Condition

In [8]:

```
L = 200
N_steps = 800
h, x_E, x_H = yee_mesh_np(0, L, 200)
ey = np.zeros((len(x_E), N_steps + 1), dtype=np.float64)
hz = np.zeros((len(x_H), N_steps + 1), dtype=np.float64)
d = L / 2
tau = 12
cfl = 0.9
a = 1

fig = plt.figure(figsize=(10, 6))

ey[:, 0] = a * np.exp(-0.5 * ((x_E - d) / tau) ** 2)
# a is the height
# d is the position of the center of the peak (the mean of the Gaussian distribution)
#tau is the standard deviation of the Gaussian distribution, controlling the width of the curve.
```

```python
for n in range(N_steps):
    hz[:, n] = hz[:, n - 1]  - cfl * (ey[1:, n] - ey[:-1, n])
    ey[1:-1, n + 1] = ey[1:-1, n] - cfl * (hz[1:, n] - hz[:-1, n])
    plt.clf()
    plt.plot(x_E, ey[:, n], label='Electric Field')
    plt.plot(x_H, hz[:, n], label='Magnetic Field')
    plt.ylim([-1, 1])
    plt.title('Electric and Magnetic Fields with R')
    plt.legend()
    display(fig)
    clear_output(wait=True)
    time.sleep(0.01)
plt.close()
```




## Absorbing Boundary Condition (ABC)

ABC condition means at the edge of the grid, the fields should only be traveling to the edge. We use left boundary as example and it will satisfied the equation $\partial x E_z - \sqrt{\mu\varepsilon}\partial t E_z = 0$.

In left boundary, stable ABC will result if the equation is expanded about the space-time point $(\frac{\Delta x}{2}, (k+\frac{1}{2})\Delta t)$.

To obtain an approximation of $E_z^{k+1}(\frac{1}{2})$, we use the average, $\frac{E_z^{k+1}(0)+E_z^{k+1}(1)}{2}$.

Similar, an approximation of $E_z^{k}(\frac{1}{2})$, we use the average, $\frac{E_z^{k}(0)+E_z^{k}(1)}{2}$.

Therefore,

$$\partial x E_z = \frac{\frac{E_z^{k+1}(1)+E_z^{k}(1)}{2} - \frac{E_z^{k+1}(0)+E_z^{k}(0)}{2}}{\Delta x}$$

$$\sqrt{\mu\varepsilon}\partial t E_z = \sqrt{\mu\varepsilon}\frac{\frac{E_z^{k+1}(0)+E_z^{k+1}(1)}{2} - \frac{E_z^{k}(0)+E_z^{k}(1)}{2}}{\Delta t}$$

Combine them and we will have

$$\frac{\frac{E_z^{k+1}(1)+E_z^{k}(1)}{2} - \frac{E_z^{k+1}(0)+E_z^{k}(0)}{2}}{\Delta x} = \sqrt{\mu\varepsilon}\frac{\frac{E_z^{k+1}(0)+E_z^{k+1}(1)}{2} - \frac{E_z^{k}(0)+E_z^{k}(1)}{2}}{\Delta t}$$

let $\mathrm{cfl} = \frac{\Delta t}{\sqrt{\mu\varepsilon}\Delta x}$ we have

$$\mathrm{cfl}(E_z^{k+1}(1) + E_z^{k}(1) - E_z^{k+1}(0) - E_z^{k}(0)) = E_z^{k+1}(0) + E_z^{k+1}(1) - E_z^{k}(0) - E_z^{k}(1)$$

$$E_z^{k+1}(0) - E_z^{k}(1) + \mathrm{cfl}(E_z^{k+1}(0) - E_z^{k}(1)) = \mathrm{cfl}(E_z^{k+1}(1) - E_z^{k}(0)) - (E_z^{k+1}(1) - E_z^{k}(0))$$

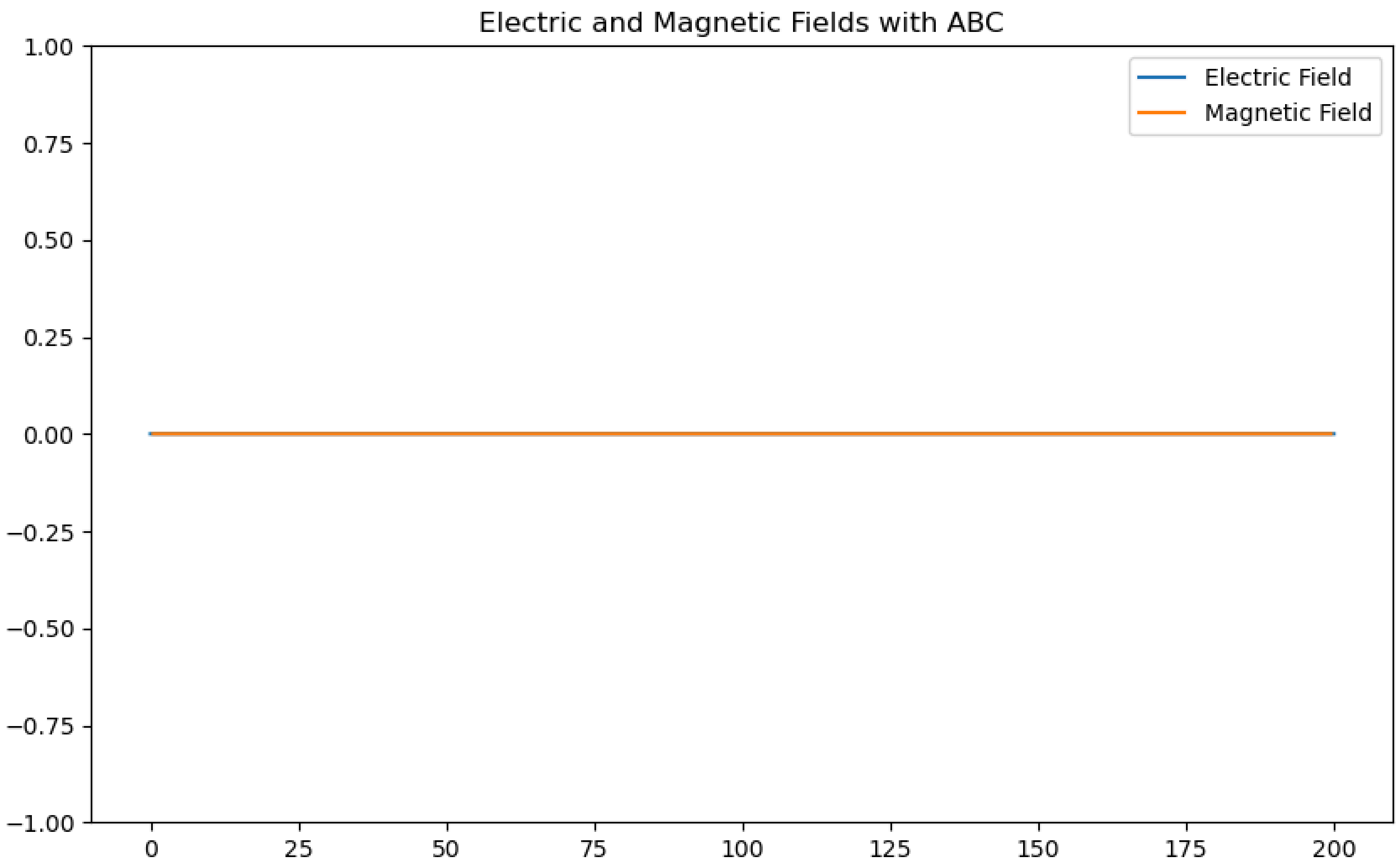
Then we have

$$E_z^{k+1}(0) = E_z^{k}(1) + \frac{\mathrm{cfl}-1}{\mathrm{cfl}+1}(E_z^{k+1}(1) - E_z^{k}(0))$$

Similar, on right boundary we have

$$E_z^{k+1}(-1) = E_z^{k}(-2) + \frac{\mathrm{cfl}-1}{\mathrm{cfl}+1}(E_z^{k+1}(-2) - E_z^{k}(-1))$$

In [9]:
```
L = 200
N_steps = 800
h, x_E, x_H = yee_mesh_np(0, L, 201)
ey = np.zeros((len(x_E), N_steps + 1), dtype=np.float64)
hz = np.zeros((len(x_H), N_steps + 1), dtype=np.float64)
d = L / 2
tau = 12
cfl = 0.9

fig = plt.figure(figsize=(10, 6))

ey[:, 0] = np.exp(-0.5 * ((x_E - d) / tau) ** 2)

for n in range(N_steps):
    hz[:, n] = hz[:, n - 1]  - cfl * (ey[1:, n] - ey[:-1, n])
    ey[1:-1, n + 1] = ey[1:-1, n] - cfl * (hz[1:, n] - hz[:-1, n])
    ey[-1, n + 1] = ey[-2, n] + ((cfl - 1)/(cfl + 1)) * (ey[-2, n + 1] - ey[-1, n])
    plt.clf()
    plt.plot(x_E, ey[:, n], label='Electric Field')
    plt.plot(x_H, hz[:, n], label='Magnetic Field')
    plt.ylim([-1, 1])
    plt.title('Electric and Magnetic Fields with ABC')
    plt.legend()
    display(fig)
    clear_output(wait=True)
    time.sleep(0.01)
plt.close()
```




## Lossy Media in One Dimension

In this case $\sigma$ may be non-zero for some points in the simulation, our maxwell equation leads to:

$$D = \varepsilon E_z + P$$

$$H_z = \frac{B}{\mu}$$

$$\frac{\partial E_y}{\partial t} = -\frac{1}{\varepsilon}(\frac{\partial H_z}{\partial x} - \frac{\sigma}{\varepsilon}E_y)$$

$$\frac{\partial H_z}{\partial t} = -\frac{1}{\mu}\frac{\partial E_y}{\partial x}$$

which P = $\frac{\sigma}{\varepsilon}E_y$ represents the linear polarization field.

$$\frac{E_i^{k+1} - E_i^k}{\Delta t} = -\frac{1}{\varepsilon}\frac{H_{i+1/2}^{k+1/2} - H_{i-1/2}^{k+1/2}}{\Delta x} - \frac{\sigma}{\varepsilon}E_i^{k+\frac{1}{2}}$$

$$\frac{H_{i+1/2}^{k+1/2} - H_{i+1/2}^{k-1/2}}{\Delta t} = -\frac{1}{\mu}\frac{E_{i+1}^k - E_i^k}{\Delta x}$$

an approximation of $E_i^{k+\frac{1}{2}}$ is $\frac{E_i^{k+1}+E_i^k}{2}$.

So we have

$$\frac{E_i^{k+1} - E_i^k}{\Delta t} = -\frac{1}{\varepsilon}\frac{H_{i+1/2}^{k+1/2} - H_{i-1/2}^{k+1/2}}{\Delta x} - \frac{\sigma}{\varepsilon}\frac{E_i^{k+1} + E_i^k}{2}$$

$$\frac{E_i^{k+1} - E_i^k}{\Delta t} + \frac{\sigma}{\varepsilon}\frac{E_i^{k+1} + E_i^k}{2} = \frac{1}{\varepsilon}\frac{H_{i+1/2}^{k+1/2} - H_{i-1/2}^{k+1/2}}{\Delta x} - \frac{\sigma}{\varepsilon}\frac{E_i^{k+1} + E_i^k}{2}$$

$$\frac{2\varepsilon(E_i^{k+1} - E_i^k)}{2\varepsilon\Delta t} + \frac{\Delta t\sigma}{\varepsilon}\frac{E_i^{k+1} + E_i^k}{2\Delta t} = \frac{1}{\varepsilon}\frac{H_{i+1/2}^{k+1/2} - H_{i-1/2}^{k+1/2}}{\Delta x}$$

We will have

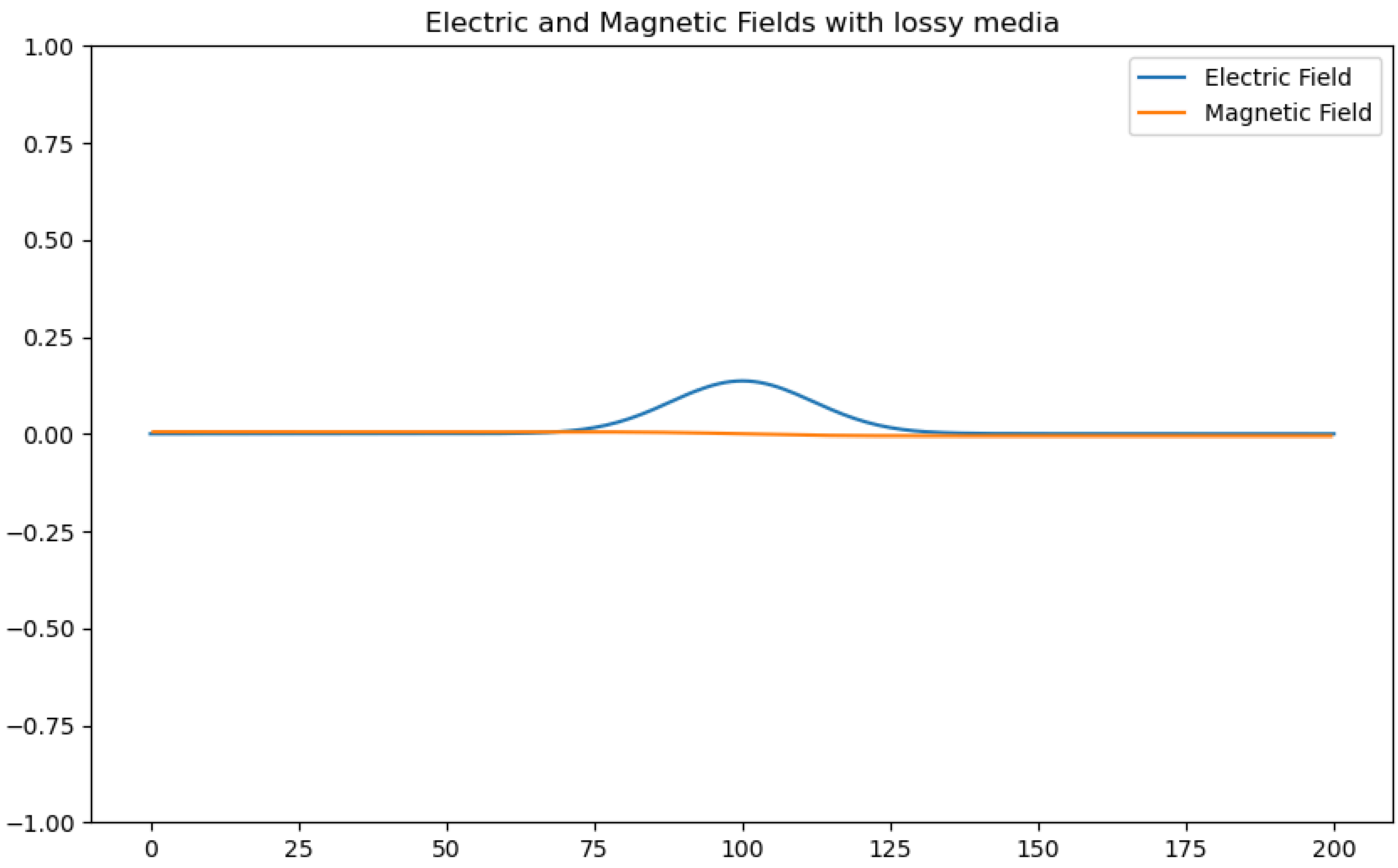
$$E_i^{k+1} = \frac{2\varepsilon - \sigma\Delta t}{2\varepsilon + \sigma\Delta t}E_i^k - \frac{2\Delta t}{(2\varepsilon + \sigma\Delta t)\Delta x}(H_{i+1/2}^{k+1/2} - H_{i-1/2}^{k+1/2})$$

$$H_{i+1/2}^{k+1/2} = H_{i+1/2}^{k-1/2} - \frac{c\Delta t}{\Delta x}(E_{i+1}^{k} - E_{i}^{k})$$

There is the yee method for one-dimensional Maxwell's equations with polarization field.

In [10]:

```
L = 200
N_steps = 800
h, x_E, x_H = yee_mesh_np(0, L, 201)
ey = np.zeros((len(x_E), N_steps + 1), dtype=np.float64)
hz = np.zeros((len(x_H), N_steps + 1), dtype=np.float64)
d = L / 2
tau = 12
c = 1

varepsilon = 1
mu = 1 / (varepsilon * c ** 2)
sigma = 0.005
dt = 1
dx = 1

fig = plt.figure(figsize=(10, 6))

ey[:, 0] = np.exp(-0.5 * ((x_E - d) / tau) ** 2)

for n in range(N_steps):
    hz[:, n] = hz[:, n - 1] -  dt / (mu * dx) * (ey[1:, n] - ey[:-1, n])
    ey[1:-1, n + 1] = (2 * varepsilon - sigma * dt) / (2 * varepsilon + sigma * dt) * ey[1:-1, n] \
                    - (2 * dt / ((2 * mu + sigma * dt) * dx))  * (hz[1:, n] - hz[:-1, n])
    plt.clf()
    plt.plot(x_E, ey[:, n], label='Electric Field')
    plt.plot(x_H, hz[:, n], label='Magnetic Field')
    plt.ylim([-1, 1])
    plt.title('Electric and Magnetic Fields with lossy media')
    plt.legend()
    display(fig)
    clear_output(wait=True)
    time.sleep(0.01)
plt.close()
```




## Variable Permittivity

If we consider a metal cavity filled with materials with different permittivity $\varepsilon(x)$, but each with permiability $\mu$, we can model the situation with a variable coefficient problem, as discussed in [5]:

$$\varepsilon(x)\frac{\partial E_y}{\partial t} + \frac{\partial H_z}{\partial x} = 0$$

$$\mu \frac{\partial H_z}{\partial t} + \frac{\partial E_y}{\partial x} = 0$$

Using the Yee Scheme, we come to the update equations:

$$H_{i+1/2}^{k+1/2} = H_{i+1/2}^{k-1/2} - \frac{1}{\mu}\frac{\Delta t}{\Delta x}\left(E_{i+1}^{k} - E_{i}^{k}\right)$$

$$E_{i}^{k+1} = E_{i}^{k} - \frac{1}{\varepsilon(x)}\frac{\Delta t}{\Delta x}\left(H_{i+1/2}^{k+1/2} - H_{i-1/2}^{k+1/2}\right)$$

For a one-dimensional interface, where the permittivity is given by $\varepsilon_k$ for $k = 1, 2$ on either side of the interface, we define $n_k = \sqrt{\varepsilon_k}$ as the index of refection for each region. Then, the wave number $\omega$ is given by the solution to the equation:

$$-n_2 \tan(n_1 \omega) = n_1 \tan(n_2 \omega)$$

The exact solution to Maxwell's Equations in this interface problem are:

$$E_y(x,t) = -\left[A_k e^{i n_k \omega x} - B_k e^{-i n_k \omega x}\right] e^{i\omega t},$$

$$H_z(x,t) = n_k \left[A_k e^{i n_k \omega x} + B_k e^{-i n_k \omega x}\right] e^{i\omega t},$$

where

$$A_1 = \frac{n_2 \cos(n_2 \omega)}{n_1 \cos(n_1 \omega)}, \qquad A_2 = e^{-i\omega(n_1+n_2)},$$

and

$$B_1 = A_1 e^{-i 2 n_1 \omega}, \qquad B_2 = A_2 e^{i 2 n_2 \omega}.$$

As an application, we consider a domain $x \in [-1, 1]$ where a metal cavity is filled with air on one side, and glass on another, so that

$$\varepsilon(x) = \begin{cases} 1, & \text{if } x \le 0 \\ 2.25, & \text{if } x > 0 \end{cases}$$

with boundary conditions $E_y(-1,t) = E_y(1,t) = 0$ for all $t > 0$. Therefore, from the previous definition, we have $n_1 = 1$, $n_2 = 1.5$.

For this scheme, from [5] we know the CFL condition is:

$$\Delta t \leq \frac{10}{13c_{max}}h,$$

Where $c_{max} = \max \frac{1}{\sqrt{\varepsilon_k}}$. In this case, $c_{max} = 1$, so we have the CFL condition $\Delta t \leq \frac{10}{13}h$.

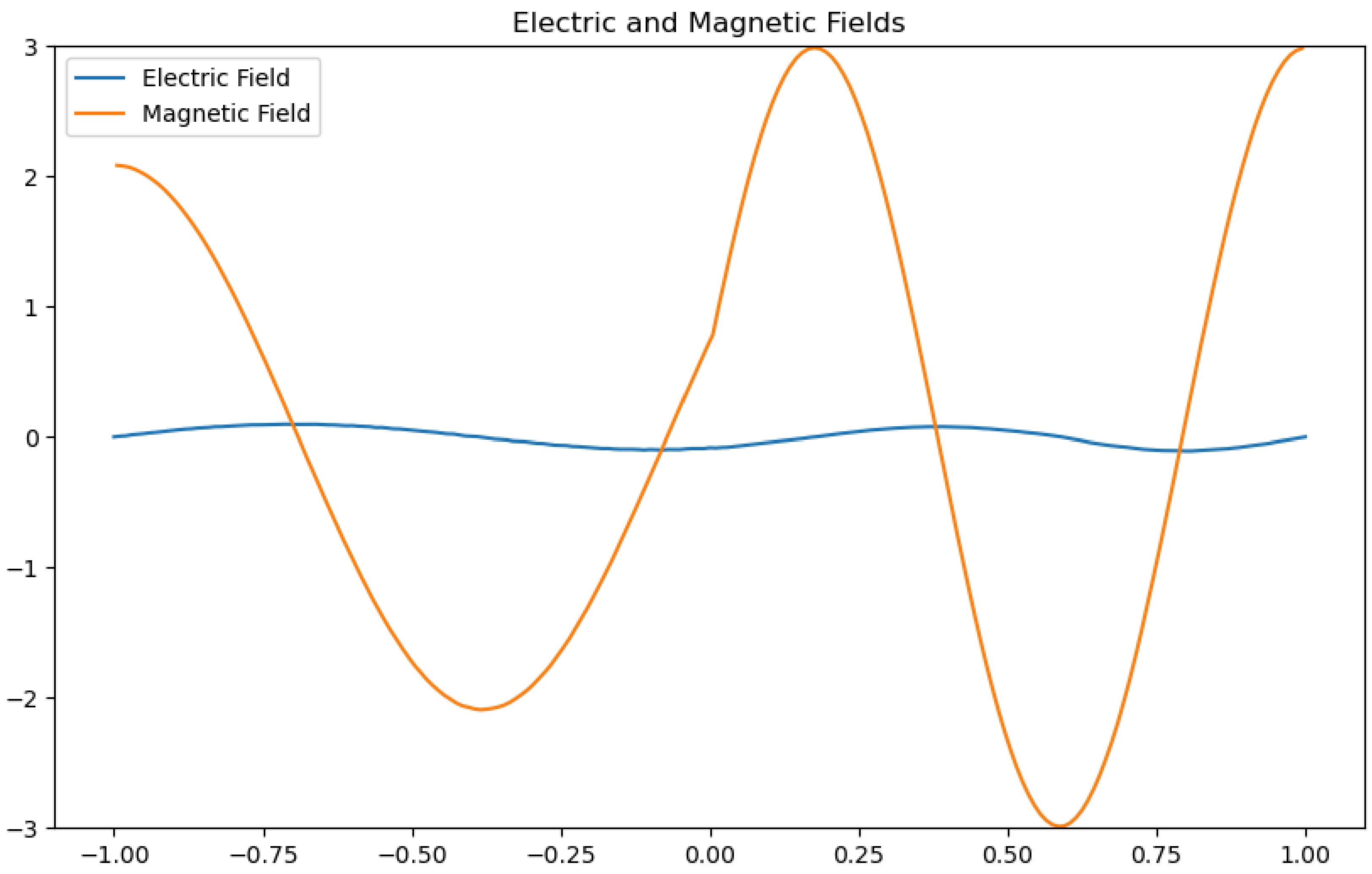
To numerically compute $\omega$, we use Newton's method. Since the function $f(\omega) = \tan(1.5\omega) + 1.5\tan(\omega)$ has discontinuities at $x = \frac{(2k+1)\pi}{2}, \frac{(2k+1)\pi}{3}$ for all $k \in \mathbb{Z}$, we set our initial guess for the root of this function between the discontinuities at $x = \frac{3\pi}{2}$ and $x = \frac{5\pi}{3}$.

In [11]:
```
#use Newton's method to find the solution to -1.5tan(w) = tan(1.5w) between 3pi/2 and 5pi/3
x = 0
x1 = 19*np.pi/12
while abs(x1-x)>1e-12:
    x = x1
    x1 = x - (np.tan(1.5*x)+1.5*np.tan(x))/(1.5/(np.cos(1.5*x)**2)+1.5/(np.cos(x)**2))
x1
```

Out[11]:
```
5.072181161825157
```

In [12]:
```
N_steps = 800
h, x_E, x_H = yee_mesh_np(-1, 1, 201)
ey = np.zeros((len(x_E), N_steps + 1), dtype=np.float64)
hz = np.zeros((len(x_H), N_steps + 1), dtype=np.float64)

def eps(x, eps1, eps2):
    #inputs:
    # x: a vector containing meshnodes
    # eps1: the permittivity for x<=0
    # eps2: the permittivity for x>0
    ##################################
    #outputs:
    # eps: a vector containing the value of permittivity at each node
    eps = np.where(x<=0, eps1, eps2)
    return eps

ep = eps(x_E, 1, 2.25)
cfl = 0.9
```

```python
mu = 1
n1 = 1
n2 = 1.5
w = 5.0721811618
A1 = (n2*np.cos(n2*w))/(n1*np.cos(n1*w))
B1 = A1*np.exp(-2*n1*w*1j)
A2 = np.exp(-w*(n1+n2)*1j)
B2 = A2*np.exp(2*n2*w*1j)

def e_exact(x, t, n1, n2, A1, B1, A2, B2):
    e1 = -(A1*np.exp(n1*w*x*1j)-B1*np.exp(-n1*w*x*1j))*np.exp(w*t*1j)
    e2 = -(A2*np.exp(n2*w*x*1j)-B2*np.exp(-n2*w*x*1j))*np.exp(w*t*1j)
    e_ex = np.where(x<=0, e1, e2)
    return e_ex

def h_exact(x, t, n1, n2, A1, B1, A2, B2):
    h1 = n1*(A1*np.exp(n1*w*x*1j)+B1*np.exp(-n1*w*x*1j))*np.exp(w*t*1j)
    h2 = n2*(A2*np.exp(n2*w*x*1j)+B2*np.exp(-n2*w*x*1j))*np.exp(w*t*1j)
    h_ex = np.where(x<=0, h1, h2)
    return h_ex

ey[:,0] = e_exact(x_E, 0, n1, n2, A1, B1, A2, B2)
hz[:,0] = h_exact(x_H, -cfl*h/2, n1, n2, A1, B1, A2, B2)

fig = plt.figure(figsize=(10, 6))

for n in range(1,N_steps):
    hz[:,n] = hz[:,n-1] - 1/mu * cfl * (ey[1:,n-1]-ey[:-1,n-1])
    ey[1:-1,n] = ey[1:-1,n-1] - 1/ep[1:-1] * cfl * (hz[1:,n]-hz[:-1,n])
    plt.clf()
    plt.plot(x_E, ey[:, n], label='Electric Field')
    plt.plot(x_H, hz[:, n], label='Magnetic Field')
    plt.ylim([-3, 3])
    plt.title('Electric and Magnetic Fields')
    plt.legend()
    display(fig)
    clear_output(wait=True)
    time.sleep(0.01)
plt.close()
```




In [13]:

```
def yee1D_varicoeff(x_E, x_H, h, eps, mu, E0, H0, t0, T, cfl):
    #inputs:
    # x_E: the meshnodes for the electric field
    # x_H: the meshnodes for the magnetic field
    # h: the spacing between meshnodes
    # eps: a vector containing the value of permittivity at each electric field node
    # mu: a scalar, the value of the magnetic permiability
    # E0: the initial condition for the electric field
    # H0: the initial condition for the magnetic field
```

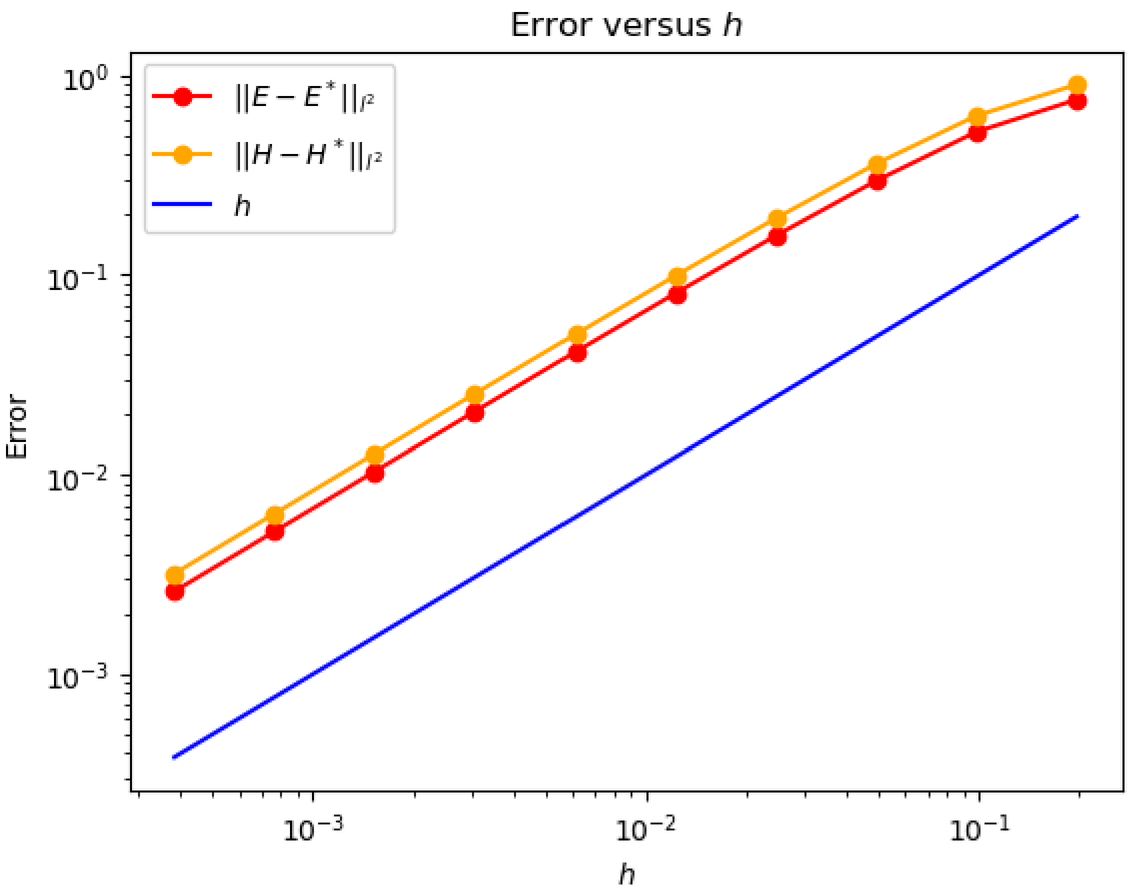
```python
    # t0: the initial time
    # T: the final time to compute the electric field
    # cfl: the mesh ratio
    ################################################################################
    #outputs:
    # Hz: the computed magnetic field at time t - 1/2 dt
    # Ey: the computed electric field at time t
    # t: the computed final time

    Ey = np.copy(E0)
    Hz = np.copy(H0)
    t = t0
    while t < T:
        Hz = Hz - 1/mu*cfl*(Ey[1:] - Ey[:-1])
        Ey[1:-1] = Ey[1:-1] - 1/eps[1:-1]*cfl*(Hz[1:] - Hz[:-1])
        t += cfl*h
    return Hz, Ey, t
```

In [14]:

```python
N = np.array([2**k+1 for k in range(5, 15)])
h_vals = 2*np.pi/(N-1)
cfl = 10/13
mu = 1
n1 = 1
n2 = 1.5
w = 5.0721811618
A1 = (n2*np.cos(n2*w))/(n1*np.cos(n1*w))
B1 = A1*np.exp(-2*n1*w*1j)
A2 = np.exp(-w*(n1+n2)*1j)
B2 = A2*np.exp(2*n2*w*1j)
e = np.zeros((len(N),2))
for i in range(10):
    h, x_E, x_H = yee_mesh_np(-1, 1, N[i])
    ep = eps(x_E, 1, 2.25)
    Ey0 = e_exact(x_E, 0, n1, n2, A1, B1, A2, B2)
    Hz0 = h_exact(x_H, -cfl*h/2, n1, n2, A1, B1, A2, B2)
    H, E, t = yee1D_varicoeff(x_E, x_H, h, ep, 1, Ey0, Hz0, 0, 2*np.pi, cfl)
    E_exact = e_exact(x_E, t, n1, n2, A1, B1, A2, B2)
    H_exact = h_exact(x_H, t-cfl*h/2, n1, n2, A1, B1, A2, B2)
    e[i,0] = np.linalg.norm(E-E_exact, 2)*np.sqrt(h_vals[i])
    e[i,1] = np.linalg.norm(H-H_exact, 2)*np.sqrt(h_vals[i])
```

```
plt.loglog(h_vals, e[:,0], color='red', marker='o', label=r'$||E-E^*||_{l^2}$')
plt.loglog(h_vals, e[:,1], color='orange', marker='o', label=r'$||H-H^*||_{l^2}$')
plt.loglog(h_vals, h_vals, color='blue', label='$h$')
plt.legend()
plt.title("Error versus $h$")
plt.xlabel('$h$')
plt.ylabel('Error')
```

Out[14]: `Text(0, 0.5, 'Error')`




# 7. Conclusion

Yee method has advantage on accuracy, it has the second order convergence. It is a mimetic scheme reflects physical situation and is is a explicit method which is easy to compute. However, it use staggered grid which means we cannot simultaneously enforce boundary conditions on both E and H. It also has the stiff stability condition, Δt cannot greater than Δx. In higher dimensions it may result in increased computational costs, as smaller time steps may be required to maintain stability.

## 8. References

[1] Allen Taflove. Computational Electrodynamics: The Finite-Difference Time-Domain Method. 3rd ed. Boston: Artech House, 2005.

[2] Understanding the Finite-Difference Time-Domain Method , John B. Schneider, www.eecs.wsu.edu/~schneidj/ufdtd, 2010

[3] Umran S. Inan and Robert A. Marshall. Numerical Electromagnetics: The FDTD Method. 1st ed. Cambridge: Cambridge University Press, 2011. doi: https://doi.org/10.1017/cbo9780511921353.

[4] M.S. Min, C.H. Teng, The Instability of the Yee Scheme for the "Magic Time Step", Journal of Computational Physics, Volume 166, Issue 2, 2001, Pages 418-424, ISSN 0021-9991, https://doi.org/10.1006/jcph.2000.6650.

[5] Adi Ditkowski, Kim Dridi, and Jan S. Hesthaven. Convergent Cartesian grid methods for Maxwell's equations in complex geometries. Journal of Computational Physics, 170(1):39–80, 2001.

In [ ]: