# Stability of Transmission Line Models

**Lucas plant**
Georgia Institute of Technology
lucasplant@gatech.edu

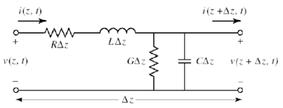

Figure 1: Discrete element model of transmission lines from Dr. Tenzeris' class on electromagnetic applications ECE 4350

## 0.1 Simulation of Transmission Lines with FDTD

This semester, I was interested in simulating transmission lines and the wave equation.

To start off, I want to derive a finite-dimensional state space model for a transmission line. To start, we can model transmission lines as a cascade of an infinite number of resistors, inductors, and capacitors, or we can use the Telegrapher's equations, which describe transmission lines as a partial differential equation:

$$\frac{\partial v(z,t)}{\partial z} = -Ri(z,t) - L\frac{\partial i(z,t)}{\partial z}$$
$$\frac{\partial i(z,t)}{\partial z} = -Gv(z,t) - L\frac{\partial v(z,t)}{\partial z}$$

R is the resistance per unit length, G is the conductance per unit length, L is the inductance per unit length, and C is the capacitance per unit length. We can expand out our spatial derivatives as limits using the Yee Lattice. The Yee lattice discretizes our problem in which voltages/electric fields and currents/magnetic fields are defined at different alternating points in space (Oskooi et.al). This allows us to define our derivative between our 2 end points, giving us second-order accuracy with respect to $\Delta z$. Re-arranging the equation and expanding out the derivative in this way:

$$\frac{\partial i}{\partial t}(t, z + \tfrac{\Delta z}{2}) =$$
$$-\frac{R}{L}i(t, z + \tfrac{\Delta z}{2}) - \frac{1}{L}\lim_{\Delta z \to 0}\left(\frac{v(t, z+\Delta z) - v(t,z)}{\Delta z}\right)$$
$$\frac{\partial v}{\partial t}(t,z) =$$
$$-\frac{G}{C}v(t,z) - \frac{1}{C}\lim_{\Delta z \to 0}\left(\frac{i(t, z+\frac{\Delta z}{2}) - i(t, z-\frac{\Delta z}{2})}{\Delta z}\right)$$

If we take $\Delta z$ to be non-zero zero we now have a finite-dimensional continuous-time system. To derive the A matrix for our system, consider the following setup of our state vector:

$$\vec{x}(t) = \begin{bmatrix} v(t,0) \\ v(t,\Delta z) \\ ... \\ v(t, N\Delta z) \\ i(t, \frac{\Delta z}{2}) \\ i(t, \frac{\Delta z}{2} + \Delta z) \\ ... \\ i(t, \frac{\Delta z}{2} + N\Delta z) \end{bmatrix}$$

We can now define our state space A matrix as a partitioned matrix:

$$\dot{\vec{x}}(t) = Ax(t)$$
$$A = \begin{bmatrix} \mathcal{G} & \mathcal{C} \\ \mathcal{L} & \mathcal{R} \end{bmatrix}$$

$\mathcal{G}$ represents the conductance of the system, $\mathcal{C}$ represents the capacitance of the system, $\mathcal{L}$ represents the inductance of the system, and $\mathcal{R}$ represents the resistance of the system. We define these submatrices as follows:

$$\mathcal{G} = -\frac{G}{C}\mathcal{I}_N$$
$$\mathcal{R} = -\frac{R}{L}\mathcal{I}_N$$
$$\mathcal{C} = \frac{1}{C\Delta z}\begin{bmatrix} 0 & 0 & 0 & ... & 0 & 0 & 0 \\ -1 & 1 & 0 & ... & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & ... & 0 & 0 \\ & & & ... & & & \\ 0 & 0 & 0 & ... & 0 & -1 & 1 \end{bmatrix}$$
$$\mathcal{L} = \frac{1}{C\Delta z}\begin{bmatrix} -1 & 1 & 0 & ... & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & ... & 0 & 0 \\ & & & ... & & & \\ 0 & 0 & 0 & ... & 0 & -1 & 1 \\ 0 & 0 & 0 & ... & 0 & 0 & 0 \end{bmatrix}$$

Where $\mathcal{I}_N$ is the NxN identity matrix. Now, we can use the forward Euler method to discretize in time to get a discrete-time system of the form:

$$\vec{x}(t + \Delta t) = A_d\vec{x}(t)$$
$$A_d = \begin{bmatrix} \mathcal{G}_d & \mathcal{C}_d \\ \mathcal{L}_d & \mathcal{R}_d \end{bmatrix}$$

$$\mathcal{G}_d = \Delta t \mathcal{G} + I_N, \mathcal{R}_d = \Delta t \mathcal{R} + \mathcal{I}_N, \mathcal{C}_d = \Delta t \mathcal{C}, \text{ and } \mathcal{L}_d = \Delta t \mathcal{L}$$

Simulating this system with Python, we observe that this is unstable: Initially, I thought that this

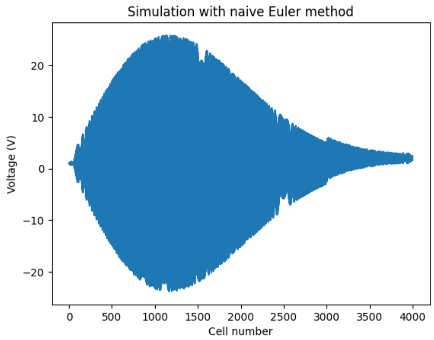


Figure 2: Visualization of instabilities in forward Euler

instability would need to be solved with filtering; however, after further investigation, this error comes from the fact that we used the staggered Yee discretization in space but not time. To solve this, we can implement an algorithm that first calculates the current/magnetic field at a half-time step in the future and then calculates the voltage/electric field:

$$\vec{i}(t + \tfrac{\Delta t}{2}) = \mathcal{L}_d \vec{v}(t) + \mathcal{R}_d \vec{i}(t - \tfrac{\Delta t}{2})$$
$$\vec{v}(t + \Delta t) = \mathcal{G}_d \vec{v}(t) + \mathcal{C}_d \vec{i}(t + \tfrac{\Delta t}{2})$$

We can substitute the current update equation into the voltage equation:

$$\vec{v}(t + \Delta t) = \mathcal{G}_d \vec{v}(t) + \mathcal{C}_d(\mathcal{L}_d \vec{v}(t) + \mathcal{R}_d \vec{i}(t - \tfrac{\Delta t}{2}))$$

We can now formulate the stable discrete A matrix $A_{Yee}$:

$$x(t + \Delta t) = A_{Yee} x(t)$$
$$A_{Yee} = \begin{bmatrix} \mathcal{G}_d + \mathcal{C}_d \mathcal{L}_d & \mathcal{C}_d \mathcal{R}_d \\ \mathcal{L}_d & \mathcal{R}_d \end{bmatrix}$$

Simulating with the following update, we can now see that the simulation is stable:

To explore why forward Euler is unstable but the Yee discretization is stable, we can look at the eigenvalues of the discrete A matrix. In the figure below, we observe that the Yee discretization has all of its eigenvalues on the unit circle corresponding to pure undamped oscillatory behavior. The forward Euler discretization has a line of Eigenvalues that are mostly outside of the unit circle, leading to instability.

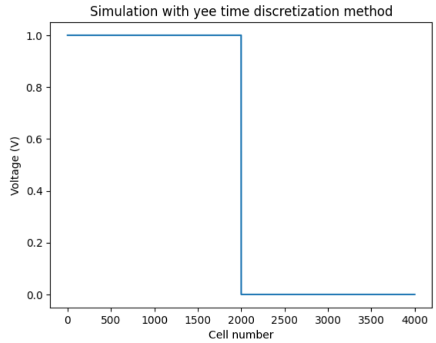


Figure 3: A stable simulation of a step function propagating down a transmission line

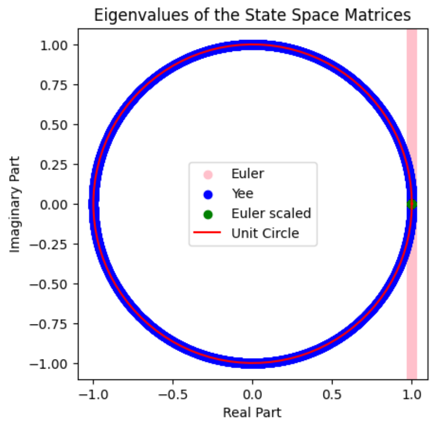


Figure 4: Plot of the eigenvalues of the different time discretization methods