The Finite-Difference Time-Domain (FDTD) Algorithm

James R. Nagel

1. OVERVIEW

It is difficult to overstate the importance of simulation to the world of engineering. Simulation is useful because it allows us to test our designs without going through all the hassle of physically constructing any devices. In fact, the market for quality simulation is quite rich, and one could easily make an entire career out of developing cheap, efficient, and accurate tools for simulation.

The finite-difference time-domain (FDTD) algorithm is an especially popular tool because it is simple, robust, and easy to understand. Basically, the algorithm works by taking the telegrapher's equations and approximating all of the derivatives as finite-differences. The system is then incremented by little time steps, and the solution effectively "plays itself out" in time. In fact, virtually any physical system that is governed by a time-dependent partial differential equation (PDE) can be readily simulated through the use of FDTD. Engineers routinely test their designs through FDTD simulations of the heat equation, Maxwell's equations, and even the Schrödinger equation.

2. THE TELEGRAPHER'S EQUATIONS

The telegrapher's equations are basically nothing more than a pair of coupled partial differential equations (PDEs) in space and time. Together, they represent the fundamental governing equations of all transmission line theory, and are given as

$$-\frac{\partial v(z,t)}{\partial z} = R' i(z,t) + L' \frac{\partial i(z,t)}{\partial t} \ , \tag{1}$$

$$-\frac{\partial i(z,t)}{\partial z} = G' v(z,t) + C' \frac{\partial v(z,t)}{\partial t} \ , \tag{2}$$

where $v(z,t)$ and $i(z,t)$ represent the instantaneous voltage and current at position z and time t. The R', L', G', and C' terms represent the resistance, inductance, conductance, and capacitance per unit length.

Because computers only have a finite capacity for memory storage, the first step in applying FDTD is to define a *mesh*, which is a set of discrete points in space and time that will sample our functions. Shown in figure 1, this is done by fixing a grid spacing of Δz in space and Δt in time. The points that lie on the mesh are then defined as

$$z_k = k\Delta z \ , \tag{3}$$

$$t_n = n\Delta t \ . \tag{4}$$

where k and n are integers confined by sets $1 \le k \le K$, and $1 \le n \le N$.

Now that the voltage and current functions have been sampled on a discrete, finite grid, we are ready to define a *stencil*, which is simply a numerical approximation for the derivative that uses neighboring points along

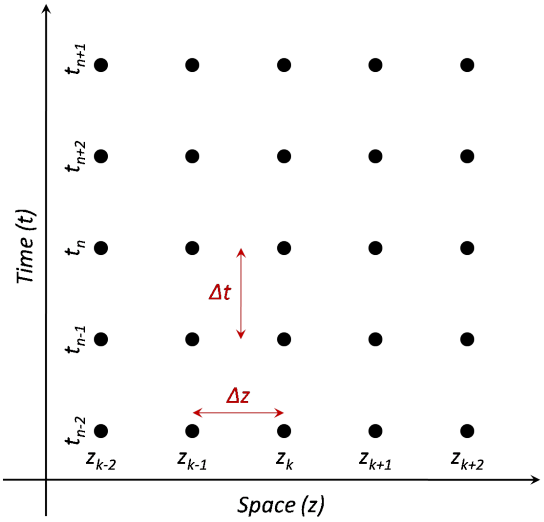

Figure 1. A simple FDTD mesh defined in space and time.

the mesh. For example, the spatial derivative acting on the voltage function can be approximated through the use of a *forward* finite-difference:

$$\frac{\partial v(z,t)}{\partial z} \approx \frac{v(z+\Delta z,t)-v(z,t)}{\Delta z} \ . \tag{5}$$

At this point, it is convenient to introduce a more compact notation for the function $v(z_k,t_n)$ by simply shortening it to v_k^n. In other words, the subscript index represents the spatial grid point, while the superscript index represents the temporal grid point. Using this notation, the numerical derivative is rewritten as

$$\frac{\partial}{\partial z}v_k^n \approx \frac{v_{k+1}^n - v_k^n}{\Delta z} \ . \tag{6}$$

Note that as long as the difference length Δz is very small, this approximation can provide us with reasonably accurate results. Nevertheless, we still would like to minimize the error from this approximation. It is therefore best to approximate all of our derivatives by using the *central-difference* method because it is more accurate than either forward- or backwards-differences. We shall therefore impose the approximation

$$\frac{\partial v(z,t)}{\partial z} \approx \frac{v(z+\Delta z,t)-v(z-\Delta z,t)}{2\Delta z} \ , \tag{7}$$

or equivalently,

$$\frac{\partial}{\partial z}v_k^n \approx \frac{v_{k+1}^n - v_{k-1}^n}{2\Delta z} \ . \tag{8}$$

Similarly, we can apply the same approximation to the time-derivatives. For example, the time-derivative on the current function $i(z,t)$ can be rewritten as

$$\frac{\partial}{\partial t}i_k^n \approx \frac{i_k^{n+1} - i_k^{n-1}}{2\Delta t} \ . \tag{9}$$

Using these new expressions, we are now ready to transform the telegrapher's equations into a pair of coupled, finite-difference equations. In terms of our new stencil notation, this is written as

$$-\frac{v_{k+1}^n - v_{k-1}^n}{2\Delta z} = R' i_k^n + L' \frac{i_k^{n+1} - i_k^{n-1}}{2\Delta t} \ , \tag{10}$$

$$-\frac{i_{k+1}^n - i_{k-1}^n}{2\Delta z} = G' i_k^n + C' \frac{v_k^{n+1} - v_k^{n-1}}{2\Delta t} \ . \tag{11}$$

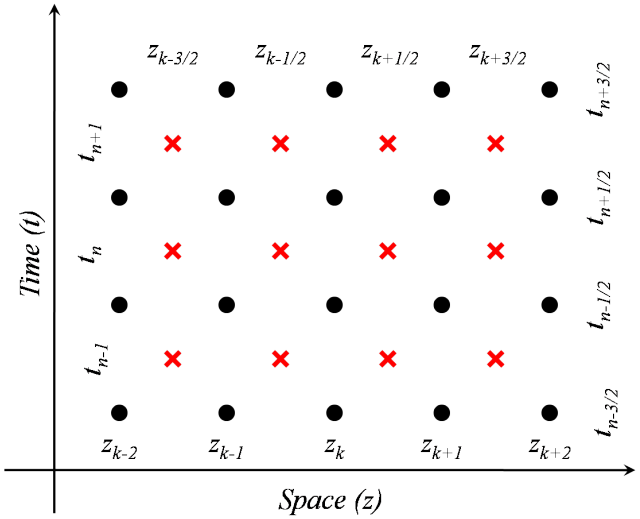

Figure 2. Revised FDTD mesh using the staggered grid. The red X's represent voltage samples while the black circles represent current samples.

Comparing with equations (1) and (2), it is easy to see that all we have done so far is approximate the continuous derivatives of the telegrapher's equations with discrete finite-differences. Although this is a perfectly acceptable approximation, there is still one final trick we can use to improve the algorithm. Mathematically, *there is no reason why the voltage and currents must be defined along the same grid points in space and time.* This may seem a little counter-intuitive at first, but we can exploit it to improve accuracy without sacrificing any computational efficiency.

To see how this works, suppose that we redefine all of our samples to exist at half-step increments away from the each other. Shown in figure 2, the voltage mesh is *staggered* from the current mesh in both space and time by the convention

$$\begin{aligned} v_k^n &\rightarrow v_{k+1/2}^n \\ i_k^n &\rightarrow i_k^{n+1/2} \end{aligned} .$$

The reason for redefining the meshes this way is so that we can approximate the spatial derivative on v as

$$\frac{\partial}{\partial z} v_k^n \approx \frac{v_{k+1/2}^n - v_{k-1/2}^n}{\Delta z} . \tag{12}$$

In other words, *the spatial derivative on v, centered around the point k, is defined in terms of the spatial points $k+1/2$ and $k-1/2$.* Notice however, that the central-difference approximation only spans a length of Δz instead of $2\Delta z$. Similarly, the temporal derivative on the current is approximated by centering around the nth time step using

$$\frac{\partial}{\partial t} i_k^n \approx \frac{i_k^{n+1/2} - i_k^{n-1/2}}{\Delta t} . \tag{13}$$

Putting these ideas together and applying them to equations (1) and (2) then gives a new set of finite-difference equations given by

$$-\frac{v_{k+1/2}^n - v_{k-1/2}^n}{\Delta z} = R'\, i_k^n + L'\, \frac{i_k^{n+1/2} - i_k^{n-1/2}}{\Delta t} , \tag{14}$$

$$-\frac{i_{k+1}^{n+1/2} - i_k^{n+1/2}}{\Delta z} = G'\, v_{k+1/2}^{n+1/2} + C'\, \frac{v_{k+1/2}^{n+1} - v_{k+1/2}^n}{\Delta t} . \tag{15}$$

This is almost the expression we are after. However, the terms i_k^n and $v_{k+1/2}^{n+1/2}$ do not fit in with their proper grid points. To correct this, we simply redefine these terms by the average value between the two nearest temporal points along the stencil. That is,

$$v_{k+1/2}^{n+1/2} \approx \frac{v_{k+1/2}^{n+1} + v_{k+1/2}^{n}}{2} , \tag{16}$$

$$i_k^n \approx \frac{i_k^{n+1/2} + i_k^{n-1/2}}{2} . \tag{17}$$

Finally, plug these expressions back into the telegrapher's equations to find a pair of coupled, finite-difference equations, with all points defined along a staggered mesh. This represents a direct numerical approximation to the telegrapher's equations that can be worked out by a computer:

$$-\frac{v_{k+1/2}^{n} - v_{k-1/2}^{n}}{\Delta z} = \frac{1}{2}R'\left(i_k^{n+1/2} + i_k^{n-1/2}\right) + L'\frac{i_k^{n+1/2} - i_k^{n-1/2}}{\Delta t} , \tag{18}$$

$$-\frac{i_{k+1}^{n+1/2} - i_k^{n+1/2}}{\Delta z} = \frac{1}{2}G'\left(v_{k+1/2}^{n+1} + v_{k+1/2}^{n}\right) + C'\frac{v_{k+1/2}^{n+1} - v_{k+1/2}^{n}}{\Delta t} . \tag{19}$$

3. UPDATE EQUATIONS

The key concept behind the FDTD algorithm is to begin with some given "present" state for a system and then solve for the nearest "future" state. This is accomplished by taking the numerical telegrapher's equations and solving for the "future" terms. For example, the *future* term in equation (18) is $i_k^{n+1/2}$ because it exists at the latest point in time. Similarly, the *future* term in equation (19) is $v_{k+1/2}^{n+1}$. Solving for these two terms therefore gives the *update equations* of the FDTD algorithm, which are

$$\boxed{i_k^{n+1/2} = c_1\left(v_{k+1/2}^{n} - v_{k-1/2}^{n}\right) + c_2\, i_k^{n-1/2}} , \tag{20}$$

and

$$\boxed{v_{k+1/2}^{n+1} = c_3\left(i_{k+1}^{n+1/2} - i_k^{n+1/2}\right) + c_4\, v_{k+1/2}^{n}} , \tag{21}$$

where the constant coefficients are given by

$$c_1 = -\frac{2\Delta t}{\Delta t \Delta z R' + 2\Delta z L'} \tag{22}$$

$$c_2 = \frac{2L' - \Delta t R'}{2L' + \Delta t R'} \tag{23}$$

$$c_3 = -\frac{2\Delta t}{\Delta t \Delta z G' + 2\Delta z C'} \tag{24}$$

$$c_4 = \frac{2C' - \Delta t G'}{2C' + \Delta t G'} . \tag{25}$$

The importance of equations (20) and (21) is that they provide a way to solve for the *future* state of the system in terms of surrounding points in space and time. This is illustrated in figure 3, which indicates the points that determine how a value is updated. The system then "leap frogs" between voltage and current as it is updated in half-step time increments. A summary of the algorithm is provided below:

1. Instantiate the samples of voltage and current over the simulation domain to their initial values.

2. Loop over all values of k and solve for the future state of the current, $i_k^{n+1/2}$.

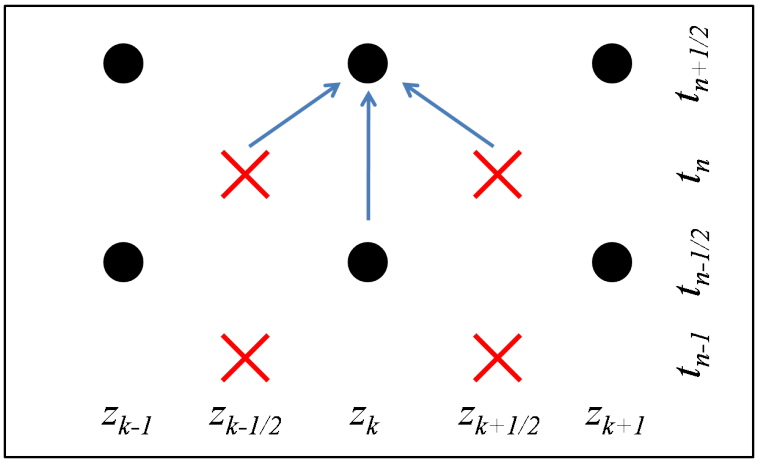

Figure 3. Schematic representation of how a current sample at $i_k^{n+1/2}$ is solved for in terms of $i_k^{n-1/2}$, $v_{k-1/2}^n$, and $v_{k+1/2}^n$.

3. Loop over all values of k and solve for future state of the voltage, $v_{k+1/2}^{n+1}$.
4. Increment n.
5. Process the new state of the system (ie, plot a graph, compute power flow, etc).
6. If n is greater than the desired simulation time N, terminate the loop. Else, return to step 2.

4. STABILITY

If one were to attempt the algorithm defined above for FDTD, one would quickly discover that certain configurations of the physical constants are unstable, and the simulated system rapidly increases without bound. To gain insight into this phenomenon, it helps to study the physical solutions to the telegrapher's equations and see how they behave under the FDTD scheme.

To illustrate, consider a lossless transmission line where $R' = G' = 0$. The general solutions for the voltage and current take on the form of forward- and reverse-traveling waves, given by

$$v(z,t) = V_0^+ \cos(\omega t - \beta z + \phi^+) + V_0^- \cos(\omega t + \beta z + \phi^-) \ , \tag{26}$$

$$i(z,t) = \frac{V_0^+}{Z_0} \cos(\omega t - \beta z + \phi^+) - \frac{V_0^-}{Z_0} \cos(\omega t + \beta z + \phi^-) \ . \tag{27}$$

where $Z_0 = \sqrt{L'/C'}$. For simplicity, let us only consider the case of a basic, forward-traveling wave, such that $V_0^+ = 1.0$ V, and $V_0^- = 0$. Also, we may neglect the phase term so that $\phi^+ = 0$, leaving

$$v(z,t) = \cos(\omega t - \beta z) \ , \tag{28}$$

$$i(z,t) = \frac{1}{Z_0} \cos(\omega t - \beta z) \ . \tag{29}$$

In terms of our FDTD stencil, the solutions to the telegrapher's equations can now be expressed as

$$v_k^n = \cos(\omega n \Delta t - \beta k \Delta z) \ , \tag{30}$$

$$i_k^n = \frac{1}{Z_0} \cos(\omega n \Delta t - \beta k \Delta z) \ . \tag{31}$$

For convenience, it helps to define the constants A, B, and C such that

$$A = \omega n \Delta t - \beta k \Delta z \tag{32}$$
$$B = \omega \Delta t \tag{33}$$
$$C = \beta \Delta z \ . \tag{34}$$

This allows us to rewrite the stencils within the update equations as

$$v_{k+1/2}^{n} = \cos(A - C/2) \ , \tag{35}$$
$$v_{k+1/2}^{n+1} = \cos(A + B - C/2) \ , \tag{36}$$
$$i_{k}^{n+1/2} = \cos(A + B/2) \ , \tag{37}$$
$$i_{k}^{n-1/2} = \cos(A - B/2) \ , \tag{38}$$

and so on. If these expressions are plugged back into an update equation such as (21), we find

$$\cos(A + B - C/2) = \frac{c_3}{Z_0} \left[\cos(A + B/2) - \cos(A - B/2)\right] + \cos(A - C/2) \ , \tag{39}$$

where $c_4 = 1$ under the lossless condition.

The importance of equation (39) is that it places constraints on the available choices for c_3. As long as A, B, and C are real numbers, the left side of the expression is bounded by the interval $[-1, 1]$. However, the constant c_3 is arbitrary, and could easily be chosen such that the amplitude of the cosine terms on the right side exceeds the natural bounds on the left. When this happens, error quickly accumulates in the simulation, and the system is said to be *unstable* or *blow up.*

To prevent the simulation from being unstable, all we need to do is define the system parameters in such a way that equation (39) never exceeds its natural bounds. This is usually accomplished by solving for a value of Δt that guarantees stability under the maximum possible extremes for the cosine terms. To illustrate, suppose that $\cos(A + B - C/2) = 1$ and $\cos(A - C/2) = -1$. We are then left with

$$2 \leq \frac{c_3}{Z_0} \left[\cos(A + B/2) - \cos(A - B/2)\right] \tag{40}$$
$$\leq \frac{-\Delta t}{\Delta z C'} \sqrt{\frac{C'}{L'}} \left[\cos(A + B/2) - \cos(A - B/2)\right] \ . \tag{41}$$

Next, we suppose the cosine terms in the brackets sum together at an extreme value of 2, and we are left with

$$1 \leq \frac{-\Delta t}{\Delta z C'} \sqrt{\frac{C'}{L'}} \ . \tag{42}$$

Finally, substitute the propagation velocity $v_p = 1/\sqrt{L'C'}$ and solve for Δt to find

$$\boxed{\Delta t \leq \frac{\Delta z}{v_p}} \ . \tag{43}$$

The upper limit on Δt is called the *critical time step* Δt_c, and represents the maximum allowable time increment for a stable simulation. It also provides an intuitive physical picture because we cannot increment the simulation any more than the time required for a wave to travel one grid step in space.

5. BOUNDARY CONDITIONS

The final step to implementing any FDTD algorithm is figuring out how to handle the boundaries. The simplest boundary condition (BC) is called the *Dirichlet* boundary, and works by fixing the end points of a simulation to some specified value. For example, the short-circuit load on a transmission line can easily be simulated by enforcing a voltage of zero at the far-right boundary. For a simulation domain of size K, this implies

$$v_{K+1/2}^{n+1} = 0 \quad \text{(Short-Circuit Boundary)} \; . \tag{44}$$

Another special BC is called the *Neumann* boundary, which only fixes the *derivative* of a function to some specified value. This can be used to simulate the open-circuit load by enforcing the condition

$$v_{K+1/2}^{n+1} = v_{K-1/2}^{n+1} \qquad \text{(Open-Circuit Boundary)} \; . \tag{45}$$

In other words, the boundary point is assigned the same value as the first inner point, giving a slope of zero between them.

It should finally be noted that these conditions are complementary between voltage and current. That is, a Dirichlet condition on the voltage acts the same as a Neumann condition on the current, and vice versa.