

Lecture III

1D FDTD - a transmission line connecting a generator and a load

In this lecture, we continue to study the Finite-Difference Time-Domain (FDTD) method on staggered grid. We apply this method to a terminated transmission line with a realistic generator/load. For a single-conductor transmission line system shown in Fig. 1, the corresponding boundary conditions read

$$\begin{aligned} v(t, z=0) &= V_G(t) - R_G i(t, z=0) \\ v(t, z=L) &= R_L i(t, z=L) \end{aligned} \quad (1)$$

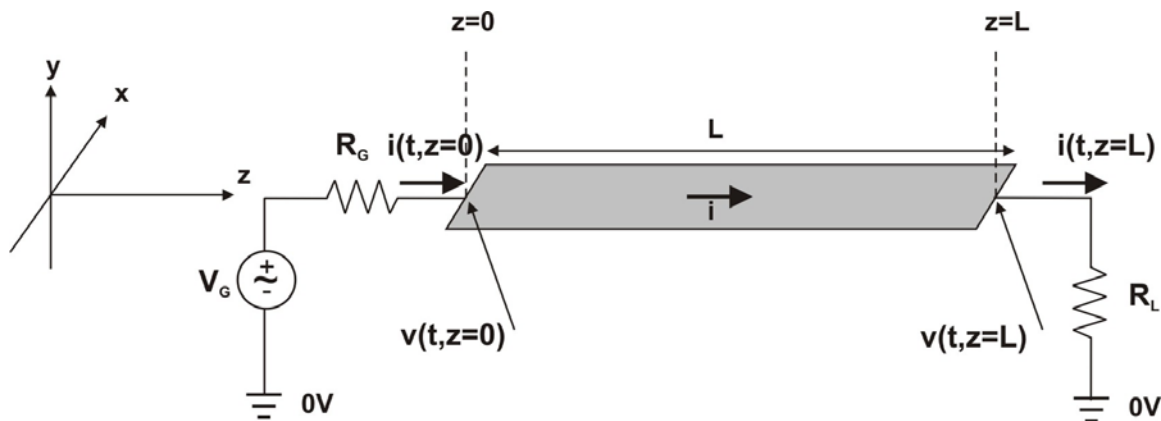


Fig. 1. A generic terminated non-coupled transmission line. Left - line source; right - line termination.

Variations of boundary conditions given by Eqs (1) are possible; all of them will be covered by the method given in this lecture.

1. Concept

One hidden difficulty with the FDTD method for transmission lines considered in lecture #2 is in satisfying the boundary conditions at the boundary voltage nodes labeled BC in Fig. 2.

A straightforward but naive attempt to use extra current/voltage nodes at the boundary and then forward- and backward one-side differences will result in a less-accurate solution. It may indicate a significant waveform distortion, especially for a piecewise-linear initial profile such as the ramp response.

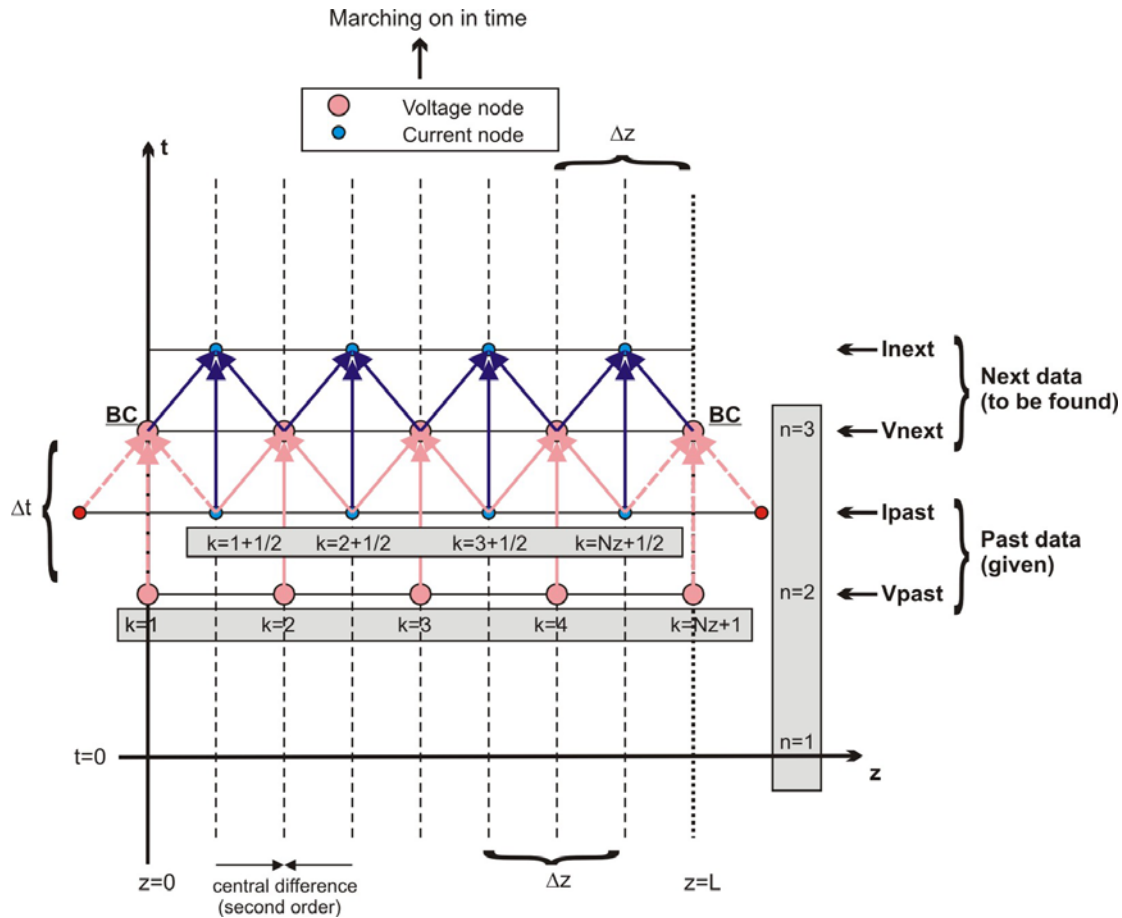


Fig. 2. The second-order finite-difference scheme for transmission-line equations on a staggered grid including two past and two next voltage/current vectors, and a schematic for boundary terminations.

2. Idea

We will use an approach developed in Ref. [C. R. Paul, "Incorporation of terminal constraints in the FDTD analysis of transmission lines", *IEEE Trans. Electromagnetic Compatibility*, vol. 36, no. 2, May 1994, pp. 85-91]. That approach implies a continuation of the line toward negative z , by $-\Delta z/2$, and then physically imposing the condition of zero current on such a continuation.

This approach is widely used today - see for example [A. Taeb, A. Abdipour, and A. Mohhamadi, "FDTD analysis of the lossy coupled transmission lines loaded by nonlinear devices," *Asia-Pacific Microwave Conference Proceedings*, 2005, vol. 5, 4-7 Dec. 2005, 4 pages.]. Its idea is shown in Fig. 2 above. This approach is exact for a two-conductor line (a microstrip line) and provides a very good approximation for a multi-conductor line.

3. Realization

A line is extended by $-\Delta z/2$ to the left at the source and by $+\Delta z/2$ at the termination. Since the approach becomes identical in both cases, only the source cell will be considered further. The first step is to replace the generator and the load by its Norton

equivalent - see Fig. 3a. Then, the new section of the line with the total length Δz , centered at $z = 0$, will have the form shown in Fig. 3b.

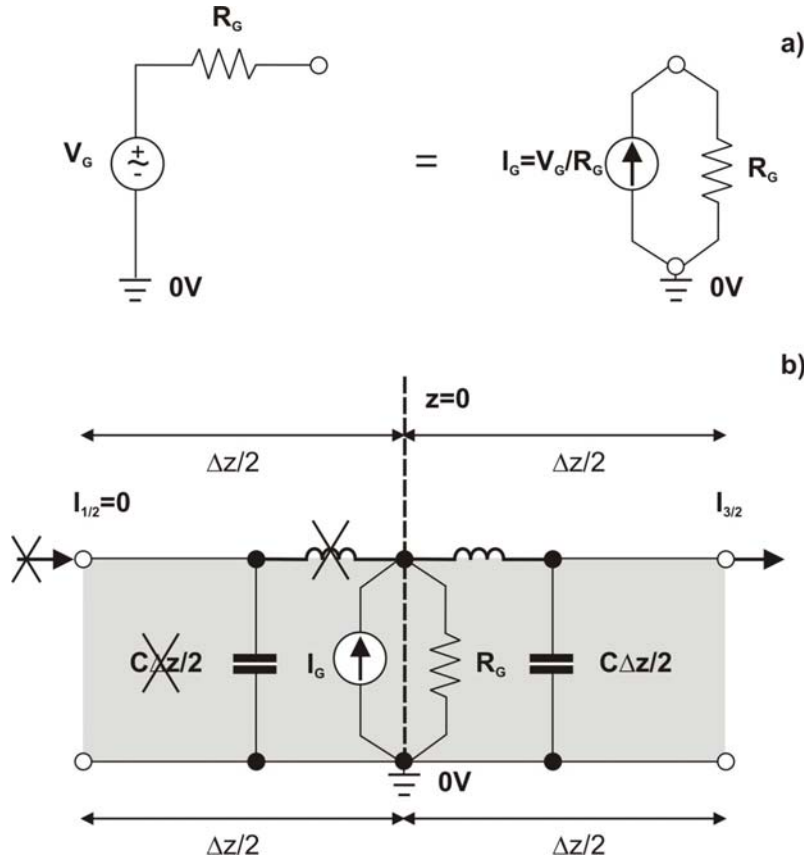


Fig. 3. Source boundary termination section of the TL extended into an imaginary line section of the length $\Delta z/2$.

It follows from Fig. 2 that, for the "extended" unit cell of length L , the distributed capacitance C and distributed inductance L could be replaced by their "average" values:

$$\begin{aligned} C &\rightarrow C/2 \\ L &\rightarrow L/2 \end{aligned} \quad (2)$$

since the distributed reactance values for the imaginary line continuation of length $\Delta z/2$ are zero.

The current source and the shunt resistance imposed at the center of the cell are then considered as distributed loads over the entire unit cell. Eq. (6) of Lecture 2 (the KCL on a line) is now rewritten in the form

$$C \frac{\partial v(t, z)}{\partial t} + \frac{\partial i(t, z)}{\partial z} = 0 \rightarrow \frac{C}{2} \frac{\partial v(t, z)}{\partial t} + \frac{\partial i(t, z)}{\partial z} + Gv = I_F \quad (3)$$

where

$$G = \frac{1}{R_G \Delta z}, \quad I_F = \frac{V_G}{R_G \Delta z} \quad (4)$$

are distributed conductance and the distributed density of current sources, respectively. Eq. (3) could be derived using the KCL for the top node in the circuit shown in Fig. 1b from lecture#2.

4. Finite differences

Only one voltage node needs to be recalculated, only at $k=1$ in Fig. 2. The standard staggered voltage grid introduced in lecture #2 (Eqs. (11) of lecture 2)

$$C \frac{\partial v(t, z)}{\partial t} + \frac{\partial i(t, z)}{\partial z} \Rightarrow C \frac{V_k^{n+1} - V_k^n}{\Delta t} + \frac{I_{k+1/2}^{n+1/2} - I_{k-1/2}^{n+1/2}}{\Delta z} \quad (5)$$

was not evaluated at the first node, i.e. at $k=1$. Instead, a boundary condition of zero voltage (or a given sine voltage, thanks to Saad) has been used in lecture #2. Now, we need to correct this step.

The second-order discretization of Eq. (3) over the staggered grid gives

$$\frac{C}{2} \frac{V_1^{n+1} - V_1^n}{\Delta t} + \frac{I_{3/2}^{n+1/2} - \underbrace{I_{1/2}^{n+1/2}}_{=0!}}{\Delta z} + G \frac{V_1^{n+1} + V_1^n}{2} = I_F = \frac{V_G^{n+1} + V_G^n}{2 R_G \Delta z} \quad (6)$$

One notes that time averaging has been used to calculate the voltage term and the right-hand side of Eq. (3). Such an averaging preserves the second-order accuracy of the entire numerical routine.

From Eq. (6), the boundary voltage at $k=1$ can be expressed. After combining the similar terms, one has

$$V_1^{n+1} = \left[\frac{\Delta z}{\Delta t} R_G C + 1 \right]^{-1} \left\{ \left[\frac{\Delta z}{\Delta t} R_G C - 1 \right] V_1^n - 2 R_G I_{3/2}^{n+1/2} + V_G^{n+1} + V_G^n \right\} \quad (7)$$

The load boundary condition at $z = L$ is identical to Eq. (7), except for the following two replacements to be made in Eq. (7):

$$\begin{aligned} R_L &\rightarrow -R_G \\ V_G &\rightarrow 0 \end{aligned} \quad (8)$$

The minus sign appears due to the opposite voltage drop through the load resistance versus ground as compared to the source resistance.

Example 1: How to realize the boundary conditions given by Eqs. (7, 8)?

Solution: A MATLAB code given below is to be inserted in the main loop of the TL solver from example 1 in lecture 2:

```
B1inv = inv(RG*C/delta + 1); % BC inverse -left
B1dir = (RG*C/delta - 1); % BC direct -left
B2inv = inv(RL*C/delta + 1); % BC inverse -right
B2dir = (RL*C/delta - 1); % BC direct -right

VG = [Vg(n) + Vg(n+1)];
VL = [0];
Vnext(1) = B1inv*( B1dir*Vpast(1) - 2*RG*Ipast(1) + VG);
Vnext(Nz+1) = B2inv*( B2dir*Vpast(Nz+1) + 2*RL*Ipast(Nz) + VL);
```

We use the MATLAB function `inv` since we intend to extend the above algorithm to multiconductor transmission lines and to line coupling in future. The same algorithm will then be converted to a matrix form.

Note: An accurate implementation of an absorbing boundary or a voltage source in FDTD for Maxwell equations in 2D or 3D can be done in a similar fashion.

Note: Different types of the terminal load, e.g. a nonlinear (Shockley) diode, or a diode in series with a capacitor (the envelope detector AM circuit), or etc. could be treated in a similar fashion.

Example 2: Write a MATLAB script that solves the terminated transmission line problem.

Solution: We start with the MATLAB script of example 1 in lecture #2 and modify it accordingly - see example 1 of this lecture discussed above. We again consider a transmission line with the total length of $D=300\text{mm}$, with the capacitance per unit length of 122 pF/m , and inductance per unit length of $0.302\text{ }\mu\text{H/m}$ (a 50Ω transmission line on 62mil FR4). The line is subject to the applied generator voltage in the form

$$V_G(t) = 2V\sin(2\pi ft), \quad f = 1\text{GHz} \quad (9)$$

The MATLAB code that follows solves the problem. Both the load and the source are matched to 50 ohms . The underlined lines are those added to the code of example 1 from lecture #2.

Example 2 (cont.):

```

% TL_03 - 1D FDTD solver for a terminated TL line with a generator/load;
% it implements the second-order central differences in time/space including BCs.
% ECE539 - Spring 2010
clear all;
%-----
D      = 300.0e-3;          % Trace trace length, m
Nz     = 400; dz = D/Nz;    % Number of spatial cells and spatial step
zkv    = [0:dz:D];         % Z-coordinates for the voltage nodes, Nz+1 total
zki    = [dz/2:dz:D-dz/2]; % Z-coordinates for the current nodes, Nz total
%-----
C      = 1.22e-010;         % Line capacitance per unit length
L      = 3.02e-007;         % Line inductance per unit length
%-----
cph    = 1/sqrt(L*C);       % phase speed on the main line
dt     = 1.00*(1/cph)*dz;   % magic time step
T      = 15e-9;            % observation time
NT     = round(T/dt); t = [0: dt: NT*dt]; % Number of time steps and time vector
%-----
% FDTD - marching on in time
% Initialization
Inext   = zeros(1, Nz+0); % fourth layer in the series of four (n+3/2)
Vnext   = zeros(1, Nz+1); % third layer in the series of four (n+1)
Ipast   = zeros(1, Nz+0); % second layer in the series of four (n+1/2)
Vpast   = zeros(1, Nz+1); % first layer in the series of four (n+0)
indv    = 2:Nz;           % batch process in space (voltage)
indi    = 1:Nz;           % batch process in space (current)
delta   = dt/dz;          % standard ratio

% Main loop - "bootstrapping"
n = 1;
%-----
% BCs
f = 1e9; Vg = 2*sin(2*pi*f*t); % Generator
RG = 50; RL = 1000; % Generator and a load
B1inv = inv(RG*C/delta + 1); % BC inverse -left
B1dir = (RG*C/delta - 1); % BC direct -left
B2inv = inv(RL*C/delta + 1); % BC inverse -right
B2dir = (RL*C/delta - 1); % BC direct -right

while n <= NT-1
    VG = [Vg(n) + Vg(n+1)]; % BCs
    VL = [0]; % BCs
    Vnext(1) = B1inv*( B1dir*Vpast(1) - 2*RG*Ipast(1) + VG); % BCs
    Vnext(Nz+1) = B2inv*( B2dir*Vpast(Nz+1) + 2*RL*Ipast(Nz) + VL); % BCs

    Vnext(indv) = Vpast(indv) + delta*(-1/C)*(Ipast(indv+0) - Ipast(indv-1));
    Inext(indi) = Ipast(indi) + delta*(-1/L)*(Vnext(indi+1) - Vnext(indi-0));
    Vpast = Vnext;
    Ipast = Inext;
    n = n + 1;
%-----
    string = strcat(num2str(1e9*t(n)), ' ns');
    x = [min(zkv) max(zkv)]; y = [1 1];
    plot(zkv, Vnext(1, :), 'r', 'LineWidth', 2); grid on;
    line(x, y, 'LineWidth', 2); line(x, -y, 'LineWidth', 2);
    axis([min(zkv), max(zkv) -3 3]); title (strcat('Line voltage at t=',string));
    xlabel('TL length, m'); ylabel('Voltage instant. profile'); drawnow;
%-----
end

```

Note: Using the above method we could evaluate four basic examples:

1. Matched load/generator
2. Load mismatch - a low-impedance load of 1Ω ;
3. Load mismatch - a high-impedance load of 1Ω ;
4. Generator impedance mismatch: $R_G = 1k\Omega$.

5. Bit (ramp) response and numerical dispersion

Now, we consider a digital voltage signal in the form of a single bit as shown in Fig. 4.

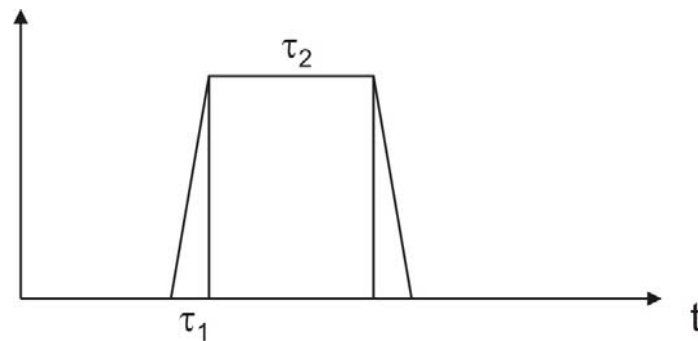


Fig. 4. Ramp excitation instead of the harmonic excitation. One single bit form is shown.

In Fig. 4, the ramp rise time is τ_1 , the flat top width is τ_2 .

Example 3: How to realize the bit form given by Fig. 4 in MATLAB?

Solution: A MATLAB code given below could be used in the main solver:

```
R          = 1.6e9;           % bit rate, bits/sec
T          = 6*(1/R);         % observation time
NT         = round(T/dt);     % Number of times steps
t          = [0: dt: NT*dt];  % time vector
tau1       = 0.10*(1/R)       % ramp rise time
tau2       = 0.80*(1/R)       % flat top time
for n = 1:NT
    time = t(n);
    if time <= tau1
        Vg(n) = time/tau1;
    elseif time < (tau1 + tau2)
        Vg(n) = 1;
    else
        if time <= (2*tau1 + tau2)
            Vg(n) = 1- (time - tau1 - tau2)/tau1;
        else
            Vg(n) = 0;
        end
    end
end
```

Note: With the code provided by example 3 we could test the numerical dispersion for

- the magic time step and;
- for the time step twice as low as the magic time step.

The results of this comparison are impressive; they show that the magic time step is really superior. For example, the following data could be used:

```
Nz      = 400; dz  = D/Nz;           % Number of spatial cells and spatial step
cph      = 1/sqrt(L*C);              % phase speed on the main line
dt       = 1.00*(1/cph)*dz;          % magic time step
```

or

```
Nz      = 400; dz  = D/Nz;           % Number of spatial cells and spatial step
cph      = 1/sqrt(L*C);              % phase speed on the main line
dt       = 0.50*(1/cph)*dz;          % magic time step
```

The corresponding algorithm for the single bit is realized in MATLAB script `tl_04.m` that implements the code from example 3.

Note: Generally, it is also interesting to observe the reflection of a voltage bit from a high-impedance load.

6. Spatial resolution per ramp

With the code provided by example 3 (and realized in the MATLAB script `tl_04.m`) we could also test an effect of the *numerical resolution* per ramp length. For a magic time step, this resolution is the number of spatial discretization steps per ramp length or, which is the same, the number of time steps per ramp rise time.

The results of this test appear to be rather unexpected. As long as the magic time step is employed, the method seems to work well for *any* spatial resolution. On the other hand, lowering the resolution below eight spatial steps per ramp leads to a significant distortion of the bit form for other (smaller) time steps. Please remember that we often do not know a priori the magic time step (the phase speed on a transmission line).

HW#3 exercise

This exercise is based on the MATLAB code `tl_04.m` for one bit propagating along the 300mm long transmission line.

1. Consider a diode load given by a (modified) Shockley diode model,

$$i(t) = I_0 \left(\exp \left(\frac{v(t)}{\eta V_T} \right) - 1 \right)$$

where $v(t)$ is the voltage across the diode and $i(t)$ is the diode current. We use diode parameters (saturation current I_0 and ideality factor η) for a certain RF Schottky Si diode:

$$I_0 = 10^{-3} \text{ A}, \quad V_T = 0.026 \text{ V}, \quad \eta = 1.05$$

2. Modify the boundary condition algorithm given by Eqs. (7), (8) for the diode load instead of the resistance load.
 3. Modify the MATLAB script `tl_04.m` accordingly. Document your modified algorithm.
 4. Observe the solution for the line voltage.
 5. Plot this solution at $t = 3.75 \text{ ns}$.
-