W9

Name: Mohammed Alshuaili

Date : 3/10/2025

Include R_s in the exact solution for validation against other methods.

Previously we had the transmission line in state space form as follows:

$$\frac{d}{dx}\begin{bmatrix} v(x,t) \\ i(x,t) \end{bmatrix} = \begin{bmatrix} 0 & -R(x) \\ -G(x) & 0 \end{bmatrix}\begin{bmatrix} v(x,t) \\ i(x,t) \end{bmatrix} - \begin{bmatrix} 0 & -L(x) \\ -C(x) & 0 \end{bmatrix}\frac{d}{dt}\begin{bmatrix} v(x,t) \\ i(x,t) \end{bmatrix}$$

Moving this to Laplace domain we get,

$$\frac{d}{dx}\begin{bmatrix} V(x,s) \\ I(x,s) \end{bmatrix} = \begin{bmatrix} 0 & -Z(x,s) \\ -Y(x,s) & 0 \end{bmatrix}\begin{bmatrix} V(x,s) \\ I(x,s) \end{bmatrix} + \begin{bmatrix} 0 & L(x) \\ C(x) & 0 \end{bmatrix}\begin{bmatrix} V(x,0) \\ I(x,0) \end{bmatrix}$$

where $Z\,(x,s) \;=\; R(x) \;+\; sL(x)$, and $Y\,(x,s) \;=\; G(x) \;+\; sC\,(x,s)$ are series impedance.

Let,

$$W(x,s) = \begin{bmatrix} V(x,s) \\ I(x,s) \end{bmatrix}, M = \begin{bmatrix} 0 & -Z(x,s) \\ -Y(x,s) & 0 \end{bmatrix}, and\ N = \begin{bmatrix} 0 & L(x) \\ C(x) & 0 \end{bmatrix}$$

So,

$$\frac{dW(x,s)}{dx} = M\,W(x,s) + N\,W(x,0)$$

Then, W at the end of the transmission line ($W\,(l\,,s)$) should equal to,

$$Let\ \Phi = e^{Ml}, so\ \phi = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix}, W(l,s) = \Phi\,\mathrm{W}(0,\mathrm{s}) + \int_0^l e^{M(l-x)} * N * W(x,0)dx$$

If the initial conditions are zero, (i.e. at W(x,0) = 0) then,

$$W(l,s) = \Phi\,\mathrm{W}(0,\mathrm{s})$$

Or matrix format,

$$W(l,s) = \begin{bmatrix} V(l,s) \\ I(l,s) \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix}.\begin{bmatrix} V(0,s) \\ I(0,s) \end{bmatrix} \tag{1}$$

If I(l,s) = 0 , Then,

$$\begin{bmatrix} V(l,s) \\ 0 \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix}.\begin{bmatrix} V(0,s) \\ I(0,s) \end{bmatrix}$$

$$0 = \Phi_{21}V(0,s) + \Phi_{22}I(0,s)$$

$$I(0,s) = \Phi_{21}V(0,s) - \Phi_{22}^{-1} \tag{2}$$

$$V(l,s) = \Phi_{11}V(0,s) + \Phi_{12}\,I(0,s)$$

$$V(l,s) = \Phi_{11}V(0,s) - \Phi_{12}\,\Phi_{21}\,\Phi_{22}^{-1}V(0,s) \tag{3}$$

now, the voltage source ($V(0,s)$) will have different value to account for R_s:

$$V(0,s) = V_s - R_s I(0,s)$$

By subbing in equation 2 we get:

$$V(0,s) = V_s - R_s(-\Phi_{22}^{-1}\Phi_{21}V_s)$$

Solve for $V(0,s)$,

$$V(0,s) = \frac{V_s}{1 - R_s\Phi_{21}\Phi_{22}^{-1}}$$

Finally, the load voltage is equal to:

$$V(l,s) = \mathrm{V}(0,\mathrm{s})\,(\Phi_{11} - \Phi_{12}\,\Phi_{21}\,\Phi_{22}^{-1})$$

Or,

$$\frac{V(l,s)}{V_s} = \frac{2e^{\left(l(YZ)^{\frac{1}{2}}\right)}(YZ)^{\frac{1}{2}}}{(YZ)^{\frac{1}{2}} - R_sY + e^{\left(2l(YZ)^{\frac{1}{2}}\right)}(YZ)^{\frac{1}{2}} + R_sYe^{\left(2l(YZ)^{\frac{1}{2}}\right)}} \tag{4}$$

Let, x = $\sqrt{YZ}$, and factor out e^{lx} from the denominator,

$$\frac{V(l,s)}{V_s} = \frac{2xe^{lx}}{e^{lx}\left((x+R_sY)e^{-lx} + (x+R_sY)e^{lx}\right)}$$

Using the fact that,

$$2\cosh(lx) = e^{-lx} + e^{lx}, 2\sinh(lx) = -e^{lx} + e^{lx}$$

We get,

$$\frac{V(l,s)}{V_s} = \frac{\sqrt{YZ}}{\sqrt{YZ}\cosh\left(l\sqrt{YZ}\right) + R_sYsinh\left(l\sqrt{YZ}\right)} \tag{5}$$

The following code can be used to obtain the exact solution output using niltcv.

```
% exact solution
clear
clc
R = 1200;          % Resistance per unit length (Ω/m)
L = 250e-9;        % Inductance per unit length (H/m)
C = 1e-10;         % Capacitance per unit length (F/m)
Rs = 10;
G = 0;
l = 150e-6;        % Length of the transmission line
f_max = 100e9;    % Maximum frequency (100 GHz)
w = 2*pi*f_max;
```

```
vs_sine = @(s) w./(s.^2 + w^2);% sin wave input
Tr = 1e-12;  % 1 ps rise/fall
Tp = 5e-12;  % 5 ps high
Amp = 1;     % 1 V amplitude
% Laplace transform of the trapezoid
vpulse = @(s) (Amp./(Tr*s.^2)).*(1 - exp(-Tr*s))- (Amp./(Tr*s.^2)).*(exp(-(Tr+Tp)*s) - exp(-(2*Tr+Tp)*s));
% Z = R+sL, Y = G+sC, so sqrt(YZ) = sqrt(s*C*(R+s*L))
vo = @(s)
sqrt(s*C.*(R+s*L))./(sqrt(s*C.*(R+s*L)).*cosh(l*sqrt(s*C.*(R+s*L)))+Rs*s*C.*sinh(l*sqrt(s*C.*(R+s*L)
)));
vo_step = @(s) vo(s).*1./s;
vo_sin = @(s) vo(s).*vs_sine(s);
vo_pulse = @(s) vo(s).*vpulse(s);
[y,t] = niltcv(vo_pulse,10e-12);
plot(t, y)
xlabel('time (s)');
ylabel('Vo');
grid on;
```

FDTD

Adding R_s to FDTD can be used using equation 10a in the reference paper, as follows:

$$V_1^{n+1} = \frac{(R_s \frac{C}{2}\frac{\Delta z}{\Delta t} - \frac{1}{2})V_1^n - R_s\left(I_1^{n+\frac{1}{2}}\right) + \frac{(V_s^{n+1} + V_s^n)}{2}}{R_s \frac{C}{2}\frac{\Delta z}{\Delta t} + \frac{1}{2}} \tag{6}$$

Which result in the following code,

```
%FDTD
clear
clc
L_total = 150e-6;  % Total length of the line (m)
R = 1200;
L = 250e-9;
C = 1e-10;
Rs = 10;
NDZ = 50;  % Number of spatial steps
dz = L_total / NDZ;  % Spatial step delta z
dt = 1e-17;  % Time step delta t
t_max = 10e-12;
t_steps = round(t_max / dt);  % Number of time steps
% allocate voltage and current arrays
time = (0:t_steps-1)*dt;
V = zeros(NDZ+1, t_steps);
I = zeros(NDZ, t_steps);
% 1.Step input (1V source)
Vs = 1 * ones(1, t_steps);
% 2. Sine wave (100 GHz)
%freq = 100e9; % Frequency in Hz
%Vs = sin(2*pi*freq * time);
% 3. Trapezoidal pulse (custom function)
%for i=1:length(time)
    %Vs(i) = trapezoidalPulse(time(i));
%end

% FDTD Loop for Time Stepping
for n = 1:t_steps-1
```

```
    V(1, n+1) = (Rs*C/2* dz/dt + 0.5)^-1* ((Rs *C/2 *dz/dt - 0.5) * V(1, n) - Rs * I(1, n) + 0.5 *
(Vs(n+1) + Vs(n)));
    for k = 1:NDZ
    if k>1
        V(k,n+1) = V(k,n) + dt/(dz *C)* (I(k-1,n) - I(k,n));  % Update voltag
        dV_k = V(k-1,n) - V(k,n);  % Voltage difference between points
        I(k-1,n+1) = I(k-1,n) + dt/(dz *L) * (dV_k-R*dz*I(k-1,n));
    end
    end
    V(NDZ,n+1) =V(NDZ,n)+dt*(I(NDZ-1,n)/(C*dz));
end
y_FDTD = V(NDZ,:);
% Plot the results for the voltage at the load
figure(1)
plot((0:t_steps-1)*dt/1e-12, V(NDZ,:));
xlabel('Time (ps)');
ylabel('V Load (Volts)');
title('FDTD Simulation of Transmission Line with unit step input');
%title('FDTD Simulation of Transmission Line with 100 GHz Sine Wave Input');
%title('FDTD Simulation of Transmission Line with Trapezoidal Pulse Input');
grid on
```

Further investigation regarding FDTD:

If the time step size in FDTD is calculated using equation 19 in the paper instead of predetermined as in the above code, FDTD will show much better results and accuracy following the following steps:

1. Calculate the Phase velocity as:

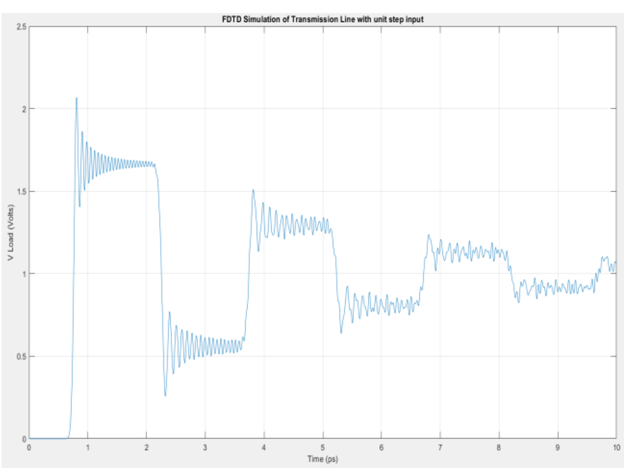
$$Z = \sqrt{\frac{L}{C}} = vL, so\ v^2L^2 = \frac{L}{C}, \quad v = \frac{1}{\sqrt{LC}} \tag{7}$$

2. Obtain $\Delta t\ as$ in equation 19 in the paper using equation 7:

$$\Delta t = \frac{\Delta z}{v}$$

3. Compute the voltages and currents using this Δt.

Comparison between magic time step and fixed one:



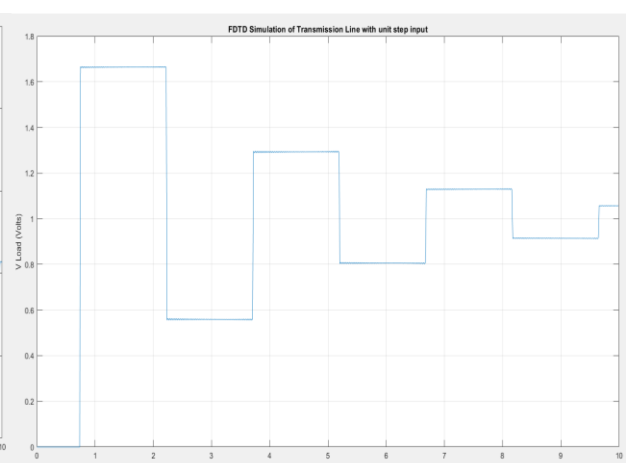


Figure 1 : shows FDTD with fixed time step size compared to the magic time step.

This gives more accurate results using a smaller number of sections with RMSE of 0.0801 using only 50 sections taking only 0.003999 seconds.

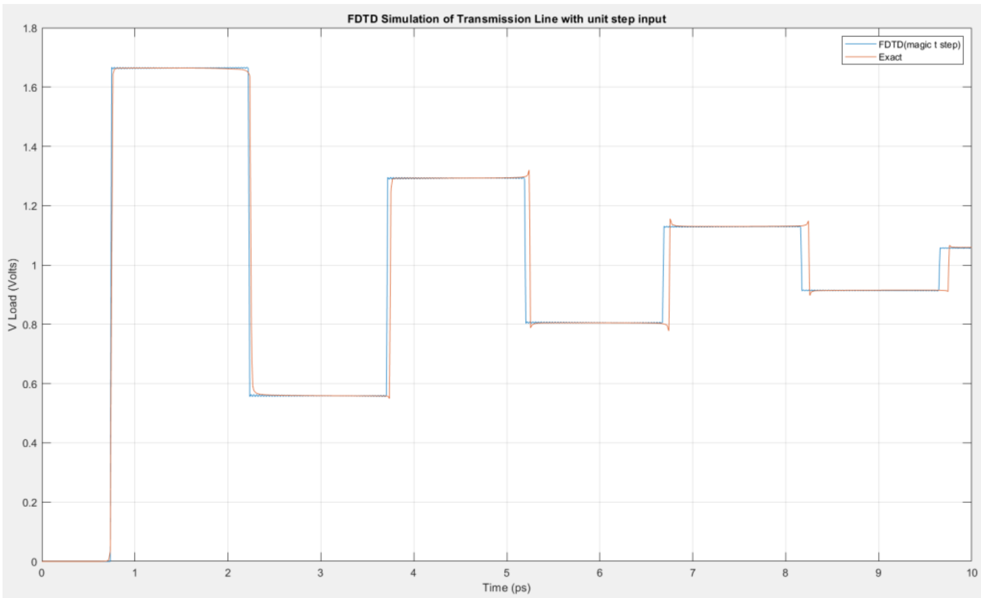


Figure 2 shows FDTD with 50 sections compared to the exact solution.

Updated FDTD code

```
%FDTD
clear
clc
L_total = 150e-6;  % Total length of the line (m)
R = 1200;
L = 250e-9;
C = 1e-10;
Rs = 10;
NDZ = 50;  % Number of spatial steps
dz = L_total / NDZ;  % Spatial step delta z
v = 1/sqrt(L*C); % Phase velocity (m/s)
dt = dz / v;   % Magic time step (dt = dz/v)
%dt = 1e-17;  % Time step delta t
t_max = 10e-12;
t_steps = round(t_max / dt);  % Number of time steps
% allocate voltage and current arrays
time = (0:t_steps-1)*dt;
V = zeros(NDZ+1, t_steps);
I = zeros(NDZ, t_steps);
% 1.Step input (1V source)
Vs = 1 * ones(1, t_steps);
% 2. Sine wave (100 GHz)
%freq = 100e9; % Frequency in Hz
%Vs = sin(2*pi*freq * time);
% 3. Trapezoidal pulse (custom function)
%for i=1:length(time)
    %Vs(i) = trapezoidalPulse(time(i));
%end

% FDTD Loop for Time Stepping
for n = 1:t_steps-1
```

```
    V(1, n+1) = (Rs*C/2*dz/dt+0.5)^-1*((Rs *C/2 *dz/dt-0.5)*V(1,n)-Rs*I(1,n)+0.5*(Vs(n+1)+Vs(n)));
    for k = 1:NDZ
    if k>1
        V(k,n+1) = V(k,n) + dt/(dz *C)* (I(k-1,n) - I(k,n));  % Update voltag
        I(k-1,n+1) = I(k-1,n)-(dt/(L*dz))*(V(k,n+1)-V(k-1,n+1))-(R*dt/L)*I(k-1,n);% Update current
    end
    end
    V(NDZ,n+1) =V(NDZ,n)+dt*(I(NDZ-1,n)/(C*dz));
end
y_FDTD = V(NDZ,:);
% Plot the results for the voltage at the load
figure(1)
plot(time/1e-12, V(NDZ,:));
xlabel('Time (ps)');
ylabel('V Load (Volts)');
title('FDTD Simulation of Transmission Line with unit step input');
%title('FDTD Simulation of Transmission Line with 100 GHz Sine Wave Input');
%title('FDTD Simulation of Transmission Line with Trapezoidal Pulse Input');
grid on
```

CFH

In the current version of AWE, I can only obtain complex conjugates when the expansion point is fixed at 0, across different frequency ranges. Examples

w	Poles	Residues
AWE_$W(1)=0$, Y_W(1:15)	-1.6969 + 7.7636i -1.6969 - 7.7636i10^5	-0.9758 - 4.4103i -0.9758 + 4.4103i10^5
AWE_$W(1)=0$, Y_W(16:30)	-4.7717 + 6.3934i -4.7717 - 6.3934i10^5	-0.4305 - 1.4909i -0.4305 + 1.4909i$x10^6$
AWE_$W(1)=0$, Y_W(31:40)	-0.2437 + 2.3025i -0.2437 - 2.3025i$x10^6$	-1.1820 + 4.7879i -1.1820 - 4.7879i$x10^5$
w	Poles	Residues
AWE_$W(1)=0$, Y_W(1:20)	$-1.9378 + 8.0523i$ $-1.9378 - 8.0523ix10^5$	-1.2722 - 4.9377i -1.2722 + 4.9377i$x10^5$
AWE_$W(21)=1.142x10^6$, Y_W(21:30)	-0.1660 + 0.4703i -0.3265 + 1.2394i$x10^6$	0.8068 - 5.5236i -1.3053 - 0.1172i$x10^5$
AWE_$W(26)=1.713x10^6$, Y_W(31:40)	-0.2153 + 2.3262i -0.0700 + 1.7238i10^6	-1.1605 + 5.7069i 0.0035 + 0.0104i$x10^5$

Considering the first 2 models, and even number of poles with the moment from the first model we get.

-1.9378 - 8.0523i -1.9378 + 8.0523i -4.3238 - 7.0043i -4.3238 + 7.0043i

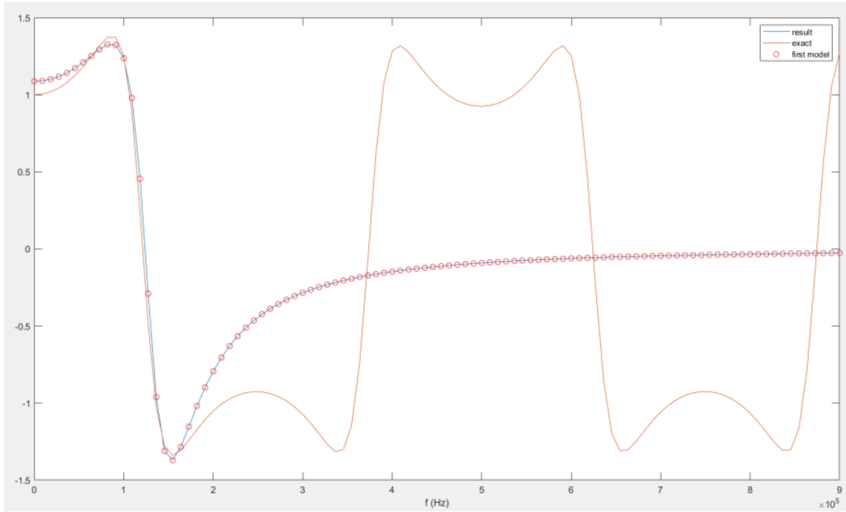


Figure 3 : shows the resultant model Vs exact Vs the first model

Consider evaluating the second model as the difference between the exact values and the first model at that range of frequency.

w	Poles	Residues
AWE_$W(1) = 0$, Y_W(1:15)	-1.6969 + 7.7636i -1.6969 - 7.7636i10^5	-0.9758 - 4.4103i -0.9758 + 4.4103i10^5
AWE_$W(1) = 0$, Y_W(16:30)	-0.4040 + 1.3471i -0.4040 - 1.3471i10^6	-2.0490 - 1.4227i -2.0490 + 1.4227i$x10^5$
AWE_$W(1) = 0$, Y_W(31:40)	-0.1593 + 2.2564i -0.1593 - 2.2564i$x10^6$	-0.7029 + 2.7046i -0.7029 - 2.7046i$x10^5$

The first 2 models :

-0.1697 - 0.7764i -0.1697 + 0.7764i -0.4040 - 1.3471i -0.4040 + 1.3471i

New residues:

-0.9758 - 4.4103i -0.9758 + 4.4103i   0.0000 + 0.0000i   0.0000 + 0.0000i

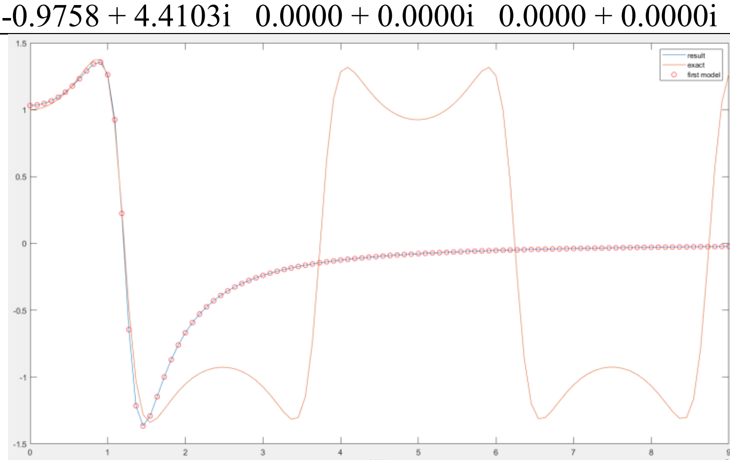


Figure 4: shows the resultant model Vs exact Vs the first model with H_diff.

Consider evaluating the second model about the imaginary part of the first one.

w	Poles	Residues
AWE_$W(1) = 0$, Y_W(1:15)	-1.6969 + 7.7636i -1.6969 - 7.7636i10^5	-0.9758 - 4.4103i -0.9758 + 4.4103i10^5
AWE_$W(15) = 7.9968x10^5$, Y_W(16:30)	-0.9679 + 9.8707i -0.1855 + 8.1118i10^5	-0.4934 - 4.3792i -0.0087 + 0.0005ix10^4

Used poles:

-1.6969 - 7.7636i -1.6969 + 7.7636i -0.9679 - 9.8707i -0.1855 - 8.1118i