W9

Name: Mohammed Alshuaili

Date : 3/10/2025

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

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

Moving this to Laplace domain we get,

where , and are series impedance.

Let,

So,

Then, W at the end of the transmission line should equal to,

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

Or matrix format,

(1)

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

(3)

(2)

now, the voltage source will have different value to account for

By subbing in equation 2 we get:

Solve for

Finally, the load voltage is equal to:

Or,

(4)

Let, x = , and factor out from the denominator,

Using the fact that,

We get,

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

(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:

(7)

1. Obtain in equation 19 in the paper using equation 7:
2. Compute the voltages and currents using this .

## Comparison between magic time step and fixed one:

A screen shot of a graph

AI-generated content may be incorrect.A screen shot of a graph

AI-generated content may be incorrect.

Figure : 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.

A graph with lines on it

AI-generated content may be incorrect.

Figure 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\_, Y\_W(1:15) | -1.6969 + 7.7636i  -1.6969 - 7.7636i | -0.9758 - 4.4103i  -0.9758 + 4.4103i |
| AWE\_, Y\_W(16:30) | -4.7717 + 6.3934i  -4.7717 - 6.3934i | -0.4305 - 1.4909i  -0.4305 + 1.4909i |
| AWE\_, Y\_W(31:40) | -0.2437 + 2.3025i  -0.2437 - 2.3025i | -1.1820 + 4.7879i  -1.1820 - 4.7879i |
| w | Poles | Residues |
| AWE\_, Y\_W(1:20) |  | -1.2722 - 4.9377i  -1.2722 + 4.9377i |
| AWE\_, Y\_W(21:30) | -0.1660 + 0.4703i  -0.3265 + 1.2394i | 0.8068 - 5.5236i  -1.3053 - 0.1172i |
| AWE\_, Y\_W(31:40) | -0.2153 + 2.3262i  -0.0700 + 1.7238i | -1.1605 + 5.7069i  0.0035 + 0.0104i |

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 |

A graph of a function

AI-generated content may be incorrect.

Figure : 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\_, Y\_W(1:15) | -1.6969 + 7.7636i  -1.6969 - 7.7636i | -0.9758 - 4.4103i  -0.9758 + 4.4103i |
| AWE\_, Y\_W(16:30) | -0.4040 + 1.3471i  -0.4040 - 1.3471i | -2.0490 - 1.4227i  -2.0490 + 1.4227i |
| AWE\_, Y\_W(31:40) | -0.1593 + 2.2564i  -0.1593 - 2.2564i | -0.7029 + 2.7046i  -0.7029 - 2.7046i |

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 |

A graph of a function

AI-generated content may be incorrect.

Figure : 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\_, Y\_W(1:15) | -1.6969 + 7.7636i  -1.6969 - 7.7636i | -0.9758 - 4.4103i  -0.9758 + 4.4103i |
| AWE\_, Y\_W(16:30) | -0.9679 + 9.8707i  -0.1855 + 8.1118i | -0.4934 - 4.3792i  -0.0087 + 0.0005ix |

Used poles:

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

A graph of a function

AI-generated content may be incorrect.