telegraph equations solved with MFEM library

Denis Lachapelle

February 2025

1 Introduction

This document explains the telegraph equations simulation using finite differences method based on MFEM library.

2 Theory, Finite Differences

The telegraph equations are:

$$\frac{\partial V}{\partial x} + L\frac{\partial I}{\partial t} + RI = 0 \tag{1}$$

$$\frac{\partial I}{\partial x} + C\frac{\partial V}{\partial t} + GV = 0 \tag{2}$$

1. Assume the TL is divided in N segments with node 0 to N. There is N+1 nodes. So nodes are located at $n\Delta x$.
2. Time is sampled at $k\Delta T$.
3. Assume the $\frac{\partial V(t,x)}{\partial x}$ and $\frac{\partial I(t,x)}{\partial x}$ are constant over each segment.

So at a given time $k\Delta T$ we can express the $\frac{\partial V(t,x)}{\partial x}$ and $\frac{\partial I(t,x)}{\partial x}$ as difference $\frac{V(k\Delta T,(n+1)\Delta x)-V(k\Delta T,(n-1)\Delta x)}{2\Delta x}$ and $\frac{I(k\Delta T,(n+1)\Delta x)-I(k\Delta T,(n-1)\Delta x)}{2\Delta x}$.
We can write matrix equations....

$$\frac{\partial I}{\partial t} = \begin{bmatrix} Dv & Ri \end{bmatrix} \begin{bmatrix} V^k \\ I^k \end{bmatrix} \tag{3}$$

$$\frac{\partial V}{\partial t} = \begin{bmatrix} Gv & Di \end{bmatrix} \begin{bmatrix} V^k \\ I^k \end{bmatrix} \tag{4}$$

The two equations above can be written as a single matrix equation...

$$\begin{bmatrix} \frac{\partial V}{\partial t} \\ \frac{\partial I}{\partial t} \end{bmatrix} = \begin{bmatrix} GvDi \\ DvRi \end{bmatrix} \begin{bmatrix} V^k \\ I^k \end{bmatrix} \tag{5}$$

$$\begin{bmatrix} V^{k+1} \\ I^{k+1} \end{bmatrix} = \begin{bmatrix} V^k \\ I^k \end{bmatrix} + \Delta t \begin{bmatrix} \frac{\partial V}{\partial t} \\ \frac{\partial I}{\partial t} \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} V^{k+1} \\ I^{k+1} \end{bmatrix} = \begin{bmatrix} V^k \\ I^k \end{bmatrix} + \Delta t \begin{bmatrix} GvDi \\ DvRi \end{bmatrix} \begin{bmatrix} V^k \\ I^k \end{bmatrix} \tag{7}$$

Where Dv and Di are the same derivative matrix multiplied by different terms -1/L for Dv and -1/C for Di.

$$\begin{bmatrix} -1/h & 1/h & 0 & 0 & ... & 0 & 0 \\ 0 & -1/h & 1/h & 0 & ... & 0 & 0 \\ 0 & 0 & -1/h & 1/h & ... & 0 & 0 \\ 0 & 0 & 0 & -1/h & 1/h & ... & 0 \\ 0 & 0 & 0 & ... & 0 & -1/h & 1/h \\ 0 & 0 & 0 & ... & 0 & -1/h & 1/h \end{bmatrix} \tag{8}$$

Gv and Ri are identity matrix scale by either -G/C and -R/L.
For Dv the multiplier value will be $\frac{\Delta t}{Lh}$ which is in our case 1e-12/(250e-9 1e-3) = 0.004

2.1 MFEM Implementation

The program is named stltfd.cpp for Single Transmission Line Transient Finite Difference.
No way to get the stable, signal do not make sense. Try with a step, impulse and a sine wave 1 Ghz.
I try with central difference and same problem.
I read that backward Euler and Crank-Nicolson are more stable.
Should I try with python? I have doubt about MFEM.
I made some test and the Dv and Di matrix within the block matrix works fine.

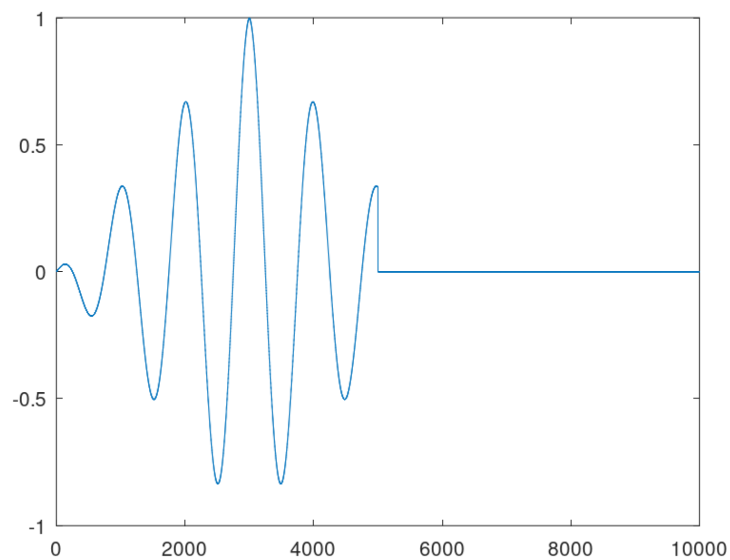

Figure 1: Test pulse

For Dv the multiplier value will be $\frac{\Delta t}{Lh}$ which is in our case 1e-12/(250e-9 1e-3) = 0.004.
For Di the multiplier value will be $\frac{\Delta t}{Ch}$ which is in our case 1e-12/(100e-12 1e-3) = 10.

3 Another way, Finite Difference Discretization

3.1 Discretizing the first equation

V_i^n is meaning $V_{ih}^{n\Delta t}$

$$\frac{V_{i+1}^n - V_i^n}{\Delta x} + L\frac{I_i^{n+1} - I_i^n}{\Delta t} + RI_i^n = 0 \tag{9}$$

Rearranged for I_i^{n+1}:

$$I_i^{n+1} = I_i^n - \frac{\Delta t}{L}\left(\frac{V_{i+1}^n - V_i^n}{\Delta x} + RI_i^n\right) \tag{10}$$

3.2 Discretizing the second equation

$$\frac{I_{i+1}^n - I_i^n}{\Delta x} + C\frac{V_i^{n+1} - V_i^n}{\Delta t} + GV_i^n = 0 \tag{11}$$

Rearranged for V_i^{n+1}:

$$V_i^{n+1} = V_i^n - \frac{\Delta t}{C}\left(\frac{I_{i+1}^n - I_i^n}{\Delta x} + GV_i^n\right) \tag{12}$$

4 Conclusion (for now: April 6, 2025)

I never succeed in any way; I may be confused with the expected result or the stability. I start trying with finite method and then finite difference, none succeeded.

Given my many reading, the finite element method is one of the best with the finite volume method. I suspect the code I made for finite element method using MFEM is mostly right, but I do not provide valid initial and boundary conditions and my understanding of the output was not good.

I shall also understand the notion of elliptic, parabolic and hyperbolic equation because each seems to call for a different approach.

5 Finite Element Method with RK4

The file name is stltferk4.cpp for single transmission transient line finite fe runge kutta 4. stltferk4.cpp is the continuation of stltfe.cpp.

MFEM provides a class for RK4 solver I'll try to use it.
odesolver need like rk4solver need a time dependent operator that is used to compute the various slopes required by RK4.

The telegraph equations are:

$$\frac{\partial V}{\partial x} + L\frac{\partial I}{\partial t} + RI = 0 \tag{13}$$

$$\frac{\partial I}{\partial x} + C\frac{\partial V}{\partial t} + GV = 0 \tag{14}$$

with the $\frac{\partial}{\partial t}$ on the left side.

$$\frac{\partial I}{\partial t} = -\frac{RI}{L} - \frac{1}{L}\frac{\partial V}{\partial x} \tag{15}$$

$$\frac{\partial V}{\partial t} = -\frac{GV}{C} - \frac{1}{C}\frac{\partial I}{\partial x} \tag{16}$$

With this formulation it look easier to use finite difference....let give it a try in the same file using option.

The following search on google "transmission line simulation with finite differences" returns many interesting results.