telegraph equations solved with MFEM library using MFEM

Denis Lachapelle

June 2025

1 Introduction

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

2 Theory, a second trial

V is approximated in H1 space and I in L2 space.

2.1 From Telegrah PDE to Matrix form

The telegraph equations are:

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

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

For now assumes a single line from points a to b.

Going to weak form using space H1 for V (ϕ) and L2 for I (ψ). Note the test functions span the entire domain and is zero on the border and the equations below have to be true for all test functions. Look at "element_finis.pdf" p. 42.

$$\int_\Omega (\frac{\partial V}{\partial x} + L\frac{\partial I}{\partial t} + RI)\phi_i = 0 \tag{3}$$

$$\int_\Omega (\frac{\partial I}{\partial x} + C\frac{\partial V}{\partial t} + GV)\psi_i = 0 \tag{4}$$

and then

$$\int_\Omega \frac{\partial V}{\partial x}\phi_i + \int_\Omega (L\frac{\partial I}{\partial t} + RI)\phi_i = 0 \tag{5}$$

$$\int_\Omega \frac{\partial I}{\partial x}\psi_i + \int_\Omega (C\frac{\partial V}{\partial t} + GV)\psi_i = 0 \tag{6}$$

Using integration by part on the first term...

$$\int_a^b u(x)v'(x)\,dx = \Big[u(x)v(x)\Big]_a^b - \int_a^b u'(x)v(x)\,dx \tag{7}$$

Each equations becomes ...

$$\Big[\phi_i V\Big]_a^b - \int_\Omega (V\frac{\partial \phi_i}{\partial x}) + \int_\Omega (L\frac{\partial I}{\partial t} + RI)\phi_i = 0 \tag{8}$$

$$\Big[\psi_i I\Big]_a^b - \int_\Omega (I\frac{\partial \psi_i}{\partial x}) + \int_\Omega (C\frac{\partial V}{\partial t} + GV)\psi_i = 0 \tag{9}$$

Then we approximate V and I, V(x) by

$$V(x,t) = \sum_j V_j(t)\phi_j(x) \tag{10}$$

and I(x) by

$$I(x,t) = \sum_j I_j(t)\psi_j(x) \tag{11}$$

Equations (8) becomes ...

$$\Big[\phi_i \sum_j V_j(t)\phi_j(x)\Big]_a^b - \int_{\Omega_i} \sum_j V_j\phi_j\frac{\partial \phi_i}{\partial x} + L\int_{\Omega_i} \frac{\partial}{\partial t}\Big[\sum I_j\psi_j\Big]\phi_i + R\int_{\Omega_i} \sum_j I_j\psi_j\phi_i = 0 \tag{12}$$

Assume $\frac{\partial I}{\partial t}$ is constant on each element so we can exclude $\frac{\partial I}{\partial t}$ from inside the integral and interchange the integral and summation.

$$\left[\phi_i \sum_j V_j(t)\phi_j(x)\right]_a^b - \sum_j V_j \int_{\Omega_i} \phi_j \frac{\partial \phi_i}{\partial x} + L\sum_j \frac{\partial I_j}{\partial t}\int_{\Omega_i} \psi_j\phi_i + R\sum_j I_j \int_{\Omega_i} \psi_j\phi_i = 0 \tag{13}$$

and equation 9 becomes ...

$$\left[\psi_i \sum_j I_j(t)\psi_j(x)\right]_a^b - \sum_j I_j \int_{\Omega} \psi_j \frac{\partial \psi_i}{\partial x} + C\sum_j \frac{\partial V_j}{\partial t}\int_{\Omega} \phi_j\psi_i + G\sum_j V_j \int_{\Omega} \phi_j\psi_i = 0 \tag{14}$$

So now we have two equations (13) and (14) with unknown $V_j(t)$ and $I_j(t)$.

The term $\left[\phi_i \sum_j V_j(t)\phi_j(x)\right]_a^b$ is $\phi_{N-1}(b)V(b,t) - \phi_0(a)V(a,t)$.

The term $\left[\psi_i \sum_j I_j(t)\psi_j(x)\right]_a^b$ is $\psi_{N-1}(b)I(b,t) - \psi_0(a)I(a,t)$.

The j span the trial functions (Approximating Functions) and the i span the elements. The equations can be converted in matrix form...

$$\phi_i(b)V(b,t) - \phi_i(a)V(a,t) - S_V V + LM_V\frac{\partial I}{\partial t} + RM_V I == 0 \tag{15}$$

$$\psi_i(b)I(b,t) - \psi_i(a)I(a,t) - S_I I + CM_I\frac{\partial V}{\partial t} + GM_I V == 0 \tag{16}$$

Where ...

$$S_V = \int_\Omega \phi_j \frac{\partial \phi_i}{\partial x}$$

dimensions VDOFxVDOF

$$M_V = \int_\Omega \psi_j\phi_i$$

dimension VDOFxIDOF

$$S_I = \int_\Omega \psi_j \frac{\partial \psi_i}{\partial x}$$

dimension IDOFxIDOF

$$M_I = \int_\Omega \phi_j \psi_i$$

dimension IDOFxVDOF

We can then isolate $LM_V \frac{\partial I}{\partial t}$ and $CM_I \frac{\partial V}{\partial t}$ on the right.

$$LM_V \frac{\partial I}{\partial t} = -RM_V I + S_V V - \phi_i(b)V(b,t) + \phi_i(a)V(a,t) \tag{17}$$

$$CM_I \frac{\partial V}{\partial t} = -GM_I V + S_I I - \psi_i(b)I(b,t) + \psi_i(a)I(a,t) \tag{18}$$

The combine block matrix will be ...

$$\begin{bmatrix} CM_I & 0 \\ 0 & LM_V \end{bmatrix} \begin{bmatrix} \frac{\partial V}{\partial t} \\ \frac{\partial I}{\partial t} \end{bmatrix} = \begin{bmatrix} -GM_I & S_I \\ S_V & -RM_V \end{bmatrix} \begin{bmatrix} V \\ I \end{bmatrix} + \begin{bmatrix} -Fi \\ -Fv \end{bmatrix} \tag{19}$$

For the source Vs Rs, Fi shall have element 0 (x=a) $-\psi_i(a)I(a,t)$ where I(a, t) is $\frac{V_S - V_0}{Rs}$ so it becomes $-\psi_i(a)\frac{V_S - V_0}{Rs}$ and split in two $-\psi_i(a)\frac{V_S}{Rs} + \psi_i(a)\frac{V_0}{Rs}$. The term in V_S is a forcing function and need to be an added linear form (column vector) in place of Fi. The term in V_0 shall be assemble as a linear form and added to A_{00} row 0.

Just rewrite the equation with named submatrix.

$$\begin{bmatrix} CM_I & 0 \\ 0 & LM_V \end{bmatrix} \begin{bmatrix} \frac{\partial V}{\partial t} \\ \frac{\partial I}{\partial t} \end{bmatrix} = \begin{bmatrix} B0 \\ B1 \end{bmatrix} \tag{20}$$

The equations are then independants:

$$CM_I \frac{\partial V}{\partial t} = B0 \tag{21}$$

$$LM_V \frac{\partial I}{\partial t} = B1 \tag{22}$$

These two equations shall be solved separately. The first one is overdetermined while the second is underdetermined.

$$A^T A x = A^T b$$

DL250625: je suis blocqué la système ne donne pas de bon résultats, et je n'arrive pas à faire un preconditionner. Je dois prendre un break.

Je retourne avec H1 et L2 puis je vais demander de l'aide sur mfem/issues. DL250709: New approach, I will work in 2D one dimensions being x and the other being time, so no need for runge kuta.

2.2 Source Boundary Condition

To add the voltage source Vs with Rs we should add the current I caused by (Vs-Va)/Rs to Va.

Firstly add the current source caused by Vs/Rs, so to the rhs node vector element 0 of I add the contribution Vs/Rs. Units are OK because Vs/Rs gives Ampere.

Secondly add to the coupling matrix (RHS matrix) the contribution of R_s to account for V_0 caused current. So to I_0 row (A10) add to first element $-\frac{1}{R_s}$. Physical unit check; S_V is a weight without unit so $S_V V$ are Volt and $\frac{V}{R_s}$ unit is Volt.

$$\begin{bmatrix} CM & 0 \\ 0 & LM \end{bmatrix} \begin{bmatrix} \frac{\partial V}{\partial t} \\ \frac{\partial I}{\partial t} \end{bmatrix} = \left[\begin{bmatrix} -GM & S_I \\ S_V & -RM \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ -\frac{1}{R_s} & 0 \\ 0 & 0 \\ \vdots & \vdots \end{bmatrix} \right] \left[\begin{bmatrix} V \\ I \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{V_S}{R_s} \\ 0 \end{bmatrix} \right] \tag{23}$$

2.3 Load Boundary Condition

Add to the coupling matrix the current caused by Rl. This will be the element row nbrDof-1, col nbrDof-1 of S_V.

$$
\begin{bmatrix} CM & 0 \\ 0 & LM \end{bmatrix} \begin{bmatrix} \frac{\partial V}{\partial t} \\ \frac{\partial I}{\partial t} \end{bmatrix} = \left[\begin{bmatrix} -GM & S_I \\ S_V & -RM \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ -\frac{1}{R_s} \cdots & 0 \\ 0 & 0 \\ \vdots & \vdots \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \vdots & \vdots \\ \cdots - \frac{1}{R_L} & 0 \\ 0 & 0 \\ \vdots & \vdots \end{bmatrix} \right] \left[\begin{bmatrix} V \\ I \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{V_S}{R_s} \\ 0 \end{bmatrix} \right] \tag{24}
$$

Let check the units to make sure the BC make sense.

2.4 C++/MFEM Implementation

The software is name stltferk4.cpp.
Instead of using the MFEM class to add BC I modify the sparse matrix directly.

2.4.1 MV Matrix

The MV matrix is made this way:
MV = new MixedBilinearForm(IFESpace, VFESpace);
MV-¿AddDomainIntegrator(new MixedScalarMassIntegrator(one));
which made the following operator $(\lambda u, v)$.

2.4.2 MI Matrix

MI = new MixedBilinearForm(VFESpace, IFESpace);
MI-¿AddDomainIntegrator(new MixedScalarMassIntegrator(one));

2.4.3 S_V Matrix

The S_V Matrix is made with:
SV = new BilinearForm(VFESpace);
SV-¿AddDomainIntegrator(new DerivativeIntegrator(one, 0));

Which one implement the following operator: $(\lambda \frac{du}{dx_i}, v)$

2.4.4 S_I Matrix

The S_I Matrix is made with:
SI = new BilinearForm(IFESpace);
SI-¿AddDomainIntegrator(new DerivativeIntegrator(one, 0));

Which one implement the following operator: $(\lambda \frac{du}{dx_i}, v)$