

# telegraph equations solved in time with FEM library using MFEM

Denis Lachapelle

January 20, 2026

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Theory</b>	<b>1</b>
2.1	From Telegraph PDE to Matrix form . . . . .	1
2.2	Euler Backward . . . . .	4
2.3	MFEM Implementation . . . . .	6
2.4	Some Simulation Results . . . . .	8
2.5	What Next . . . . .	13

## 1 Introduction

This document explains the telegraph equations simulation using finite element method based on MFEM library. The equations are evolved in time by time stepping estimating the partial derivative at each time step of V and I. The stepping method is backward euler in stltfe1d\_04.cpp

## 2 Theory

### 2.1 From Telegraph PDE to Matrix form

The telegraph equations are:

$$\frac{\partial V(x, t)}{\partial x} + L \frac{\partial I(x, t)}{\partial t} + RI(x, t) = 0 \quad (1)$$

$$\frac{\partial I(x, t)}{\partial x} + C \frac{\partial V(x, t)}{\partial t} + GV(x, t) = 0 \quad (2)$$

For now assumes a single line from points a to b. Positive current going from a to b direction.

Going to weak form in x domain using different space for V and I,  $W_i^V(x)$  for V and  $W_i^I(x)$  for I. Note the test functions span the entire domain.

$$\int_a^b \left( \frac{\partial V(x, t)}{\partial x} + L \frac{\partial I(x, t)}{\partial t} + RI(x, t) \right) W_i^V(x) dx = 0 \quad (3)$$

$$\forall W_i^V(x)$$

$$\int_a^b \left( \frac{\partial I(x, t)}{\partial x} + C \frac{\partial V(x, t)}{\partial t} + GV(x, t) \right) W_i^I(x) dx = 0 \quad (4)$$

$$\forall W_i^I(x)$$

and then

$$\int_a^b \frac{\partial V(x, t)}{\partial x} W_i^V(x) dx + \int_a^b \left( L \frac{\partial I(x, t)}{\partial t} + RI(x, t) \right) W_i^V(x) dx = 0 \quad (5)$$

$$\int_a^b \frac{\partial I(x, t)}{\partial x} W_i^I(x) dx + \int_a^b \left( C \frac{\partial V(x, t)}{\partial t} + GV(x, t) \right) W_i^I(x) dx = 0 \quad (6)$$

Restart with equation 5 and 6 and apply integration by part on  $\frac{\partial I(x, t)}{\partial x}$  of equation 26. so that no BC will be added for the V equation (8).

$$\int_a^b \frac{\partial V(x, t)}{\partial x} W_i^V(x) dx + \int_a^b \left( L \frac{\partial I(x, t)}{\partial t} + RI(x, t) \right) W_i^V(x) dx = 0 \quad (7)$$

$$\int_a^b \frac{\partial I(x, t)}{\partial x} W_i^I(x) dx + \int_a^b \left( C \frac{\partial V(x, t)}{\partial t} + GV(x, t) \right) W_i^I(x) dx = 0 \quad (8)$$

Using integration by part on the term  $\frac{\partial I(x, t)}{\partial x}$ .

$$\int_a^b u(x)v'(x) dx = \left[ u(x)v(x) \right]_a^b - \int_a^b u'(x)v(x) dx \quad (9)$$

The I equation becomes ...

$$\left[ W_i^I(x)I(x,t) \right]_a^b - \int_a^b I(x,t) \frac{\partial W_i^I(x)}{\partial x} dx + \int_a^b C \frac{\partial V(x,t)}{\partial t} W_i^I(x) dx + \int_a^b GV(x,t) W_i^I(x) dx = 0 \quad (10)$$

Then we approximate V and I by summations of possibly different basis function for V  $\phi_j$  and I  $\psi_j$  as follow:

$$V(x,t) = \sum_j V_j(t) \phi_j(x), \quad I(x,t) = \sum_j I_j(t) \psi_j(x) \quad (11)$$

And the time derivatives are:

$$\frac{\partial V(x,t)}{\partial t} = \sum_j \frac{dV_j(t)}{dt} \phi_j(x) \quad \frac{\partial I(x,t)}{\partial t} = \sum_j \frac{dI_j(t)}{dt} \psi_j(x) \quad (12)$$

Replacing each of  $V(x,t)$ ,  $I(x,t)$ ,  $\frac{\partial V(x,t)}{\partial t}$  and  $\frac{\partial I(x,t)}{\partial t}$  in equations 7 and 10 we get...

$$\begin{aligned} & \sum_j V_j(t) \int_a^b \frac{\partial \phi_j(x)}{\partial x} W_i^V(x) dx \\ & + L \sum_j \frac{\partial I_j(t)}{\partial t} \int_a^b \psi_j(x) W_i^V(x) dx + R \sum_j I_j(t) \int_a^b \psi_j(x) W_i^V(x) dx \quad (13) \\ & = 0 \end{aligned}$$

and equation 10 becomes ...

$$\begin{aligned} & \left[ W_i^I(x) \sum_j I_j(t) \psi_j(x) \right]_a^b - \sum_j I_j(t) \int_a^b \psi_j(x) \frac{\partial W_i^I(x)}{\partial x} dx \\ & + C \sum_j \frac{\partial V_j(t)}{\partial t} \int_a^b \psi_j(x) W_i^I(x) dx + G \sum_j V_j(t) \int_a^b \phi_j(x) W_i^I(x) dx \quad (14) \\ & = 0 \end{aligned}$$

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

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

$$S_V V(t) + L M_V \frac{\partial I(t)}{\partial t} + R M_V I(t) == 0 \quad (15)$$

$$B^b I(t) - B^a I(t) - S_I I(t) + C M_I \frac{\partial V(t)}{\partial t} + G M_I V(t) = 0 \quad (16)$$

One more step to include our particular boundaries.

$$R_S I_a + V_a = V_S \quad (17)$$

The transmission line point b is loaded with a resistor  $R_L$ .

$$V_b - R_L I_b = 0 \quad (18)$$

The boundaries constraints are required just in the 2nd equation.

$$B^b V(b, t) / R_L - B^a V_s(t) / R_S + B^a V(a, t) / R_S - S_I I(t) + C M_I \frac{\partial V(t)}{\partial t} + G M_I V(t) == 0 \quad (19)$$

Where ...

$$\begin{aligned} S_V &= \int_a^b \frac{\partial \phi_j}{\partial x} W_i^V dx, & M_V &= \int_a^b \psi_j W_i^V dx \\ S_I &= \int_a^b \psi_j \frac{\partial W_i^I}{\partial x} dx, & M_I &= \int_a^b \phi_j W_i^I dx \end{aligned}$$

We select trial space for  $V$  and  $I$  to be  $H1$ ; later on we may try with  $I$  in  $L2$  space.

## 2.2 Euler Backward

$$y_n = y_{n-1} + \Delta t_n f(t_n, y_n) \quad (20)$$

Equation 15 and 16 can be written as a single equation ...

$$A0 U(t) + A1 \frac{\partial U(t)}{\partial t} - A2 h(t) = 0 \quad (21)$$

$$\begin{aligned}
\text{Where } U(t) &= \begin{bmatrix} I(t) \\ V(t) \end{bmatrix} \\
A0 &= \begin{bmatrix} RM_V & S_V \\ -S_I & B^b + B^a + GM_I \end{bmatrix} \\
A1 &= \begin{bmatrix} LM_V & 0 \\ 0 & CM_I \end{bmatrix} \\
A2 &= \begin{bmatrix} 0 & \frac{B^a}{R_s} \end{bmatrix} \\
h(t) &= \begin{bmatrix} 0 \\ V_s(t) \end{bmatrix}
\end{aligned}$$

$$U(t_n) = U(t_{n-1}) + \Delta t A1^{-1} (A2 h(t_n) - A0 U(t_n)) \quad (22)$$

$$A1 U(t_n) = A1 U(t_{n-1}) + \Delta t A2 h(t_n) - \Delta t A0 U(t_n) \quad (23)$$

Term with  $t_n$  on LHS and term with  $t_{n+1}$  on RHS.

$$(A1 + \Delta t A0)U(t_n) = A1 U(t_{n-1}) + \Delta t A2 h(t_n) \quad (24)$$

shift n by +1

$$(A1 + \Delta t A0)U(t_{n+1}) = A1 U(t_n) + \Delta t A2 h(t_n) \quad (25)$$

Now includes the details about the boundaries terms  $B^b$  and  $B^a$ .

$$\begin{aligned}
&\begin{bmatrix} (L + \Delta t R)M_V & \Delta t S_V \\ -\Delta t S_I & (C + \Delta t G)M_I + \Delta t B^b/R_L + \Delta t B^a/R_S \end{bmatrix} * \begin{bmatrix} I(t_{n+1}) \\ V(t_{n+1}) \end{bmatrix} \\
&= \begin{bmatrix} LM_V & 0 \\ 0 & CM_I \end{bmatrix} * \begin{bmatrix} I(t_n) \\ V(t_n) \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta t B^a V_s(t_n)/R_S \end{bmatrix}
\end{aligned} \quad (26)$$

$$\begin{aligned}
&\begin{bmatrix} (L/\Delta t + R)M_V & S_V \\ -S_I & (C/\Delta t + G)M_I + B^b/R_L + B^a/R_S \end{bmatrix} * \begin{bmatrix} I(t_{n+1}) \\ V(t_{n+1}) \end{bmatrix} \\
&= \begin{bmatrix} L/\Delta t M_V & 0 \\ 0 & C/\Delta t M_I \end{bmatrix} * \begin{bmatrix} I(t_n) \\ V(t_n) \end{bmatrix} + \begin{bmatrix} 0 \\ B^a V_s(t_n)/R_S \end{bmatrix}
\end{aligned} \quad (27)$$

The above formulae shall be used to time step. Rewrite below to match the naming used in the software.

$$\begin{aligned}
 & \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} * \begin{bmatrix} I(t_{n+1}) \\ V(t_{n+1}) \end{bmatrix} \\
 &= \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix} * \begin{bmatrix} I(t_n) \\ V(t_n) \end{bmatrix} + \begin{bmatrix} 0 \\ C_2 \end{bmatrix}
 \end{aligned} \tag{28}$$

## 2.3 MFEM Implementation

Below are listed the bilinear form and their respective integrators, and the linear form require for signal injection.

```

A11 = new BilinearForm(FESpace);
ConstantCoefficient A11Coeff(L/deltaT);
A11->AddDomainIntegrator(new MassIntegrator(A11Coeff));

A22 = new BilinearForm(FESpace);
ConstantCoefficient A22Coeff(C/deltaT);
A22->AddDomainIntegrator(new MassIntegrator(A22Coeff));

B11 = new BilinearForm(FESpace);
ConstantCoefficient B11Coeff(L/deltaT + R);
B11->AddDomainIntegrator(new MassIntegrator(B11Coeff));

B12 = new BilinearForm(FESpace);
B12->AddDomainIntegrator(new DerivativeIntegrator(one, 0));

B21 = new MixedBilinearForm(FESpace, FESpace);
B21->AddDomainIntegrator(new MixedScalarWeakDerivativeIntegrator(one));
// weak derivative is (-lU, dv/dx)

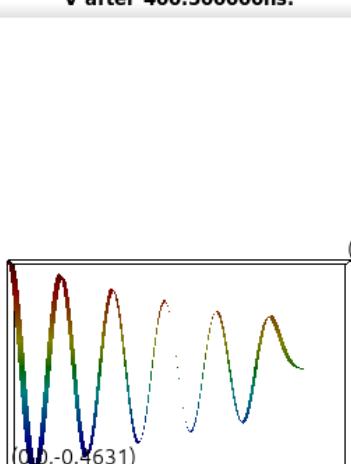
B22 = new BilinearForm(FESpace);
ConstantCoefficient B22Coeff(C/deltaT + G);
B22->AddDomainIntegrator(new MassIntegrator(B22Coeff));
ConstantCoefficient oneOverRsCoeff(1.0/Rs);
ConstantCoefficient oneOverRlCoeff(1.0/Rl);

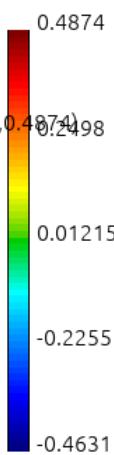
```

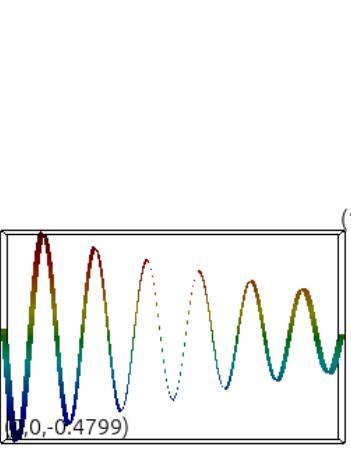
```
B22->AddBoundaryIntegrator(new BoundaryMassIntegrator(oneOverRsCoeff), *inputBdrM  
B22->AddBoundaryIntegrator(new BoundaryMassIntegrator(oneOverRlCoeff), *outputBdm  
  
VsRs = new VsRsCoefficient(1.0 / (Rs), SourceFunction);  
C2 = new LinearForm(FESpace);  
C2->AddBoundaryIntegrator(new BoundaryLFIntegrator(*VsRs), *inputBdrMarker);
```

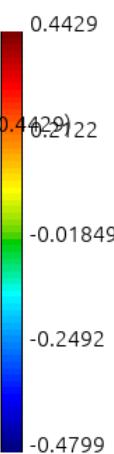
## 2.4 Some Simulation Results

The next two figure are 13MHz sine wave in a 100m rg-58 cable driven 1 Vp-p 50 ohm and a 50 ohm load, we see that after 400ns the signal did not travel to the cable end, after 1000ns the signal has reach the end since quite long time, we can also see the signal attenuation.

**V after 400.500000ns.** 

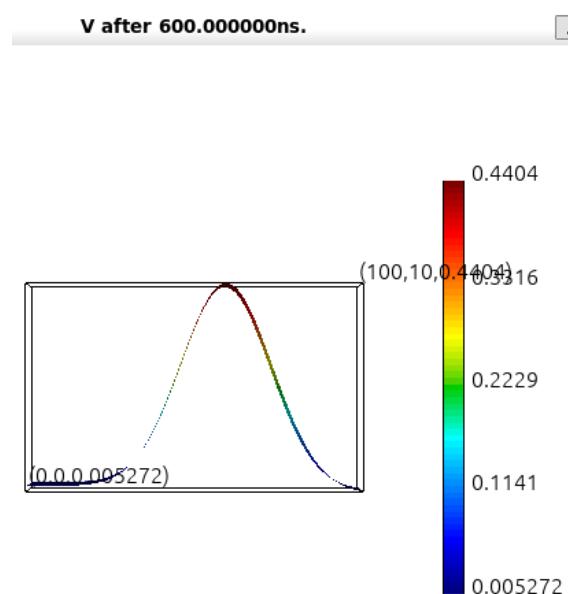
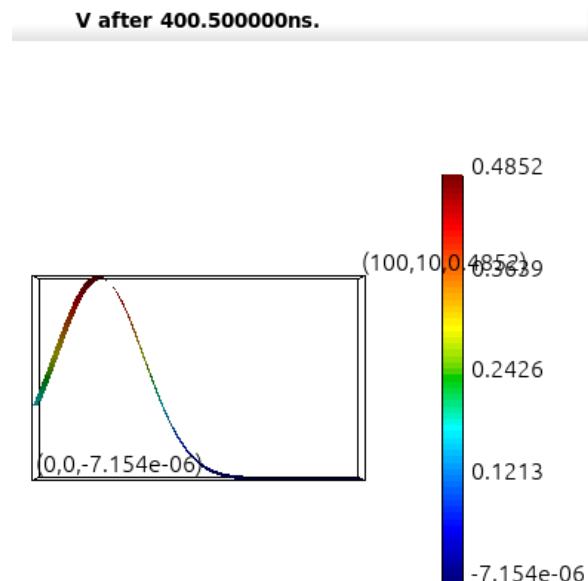


**V after 1000.000000ns.** 

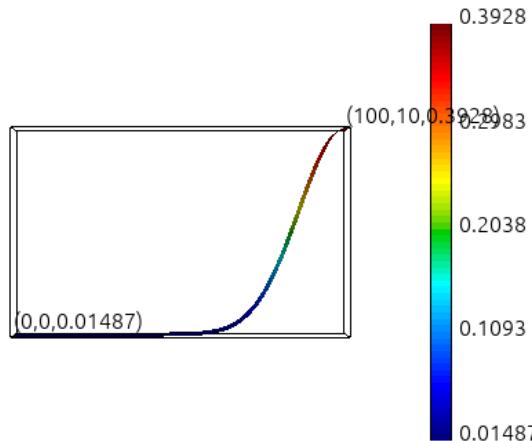




The following three figure are showing a gaussian pulse delay by 300ns and tau of 100ns propagating in a 100m rg-58 cable.

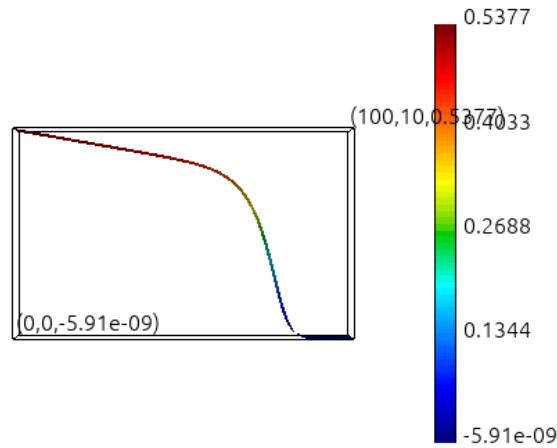


**V after 800.000000ns.**

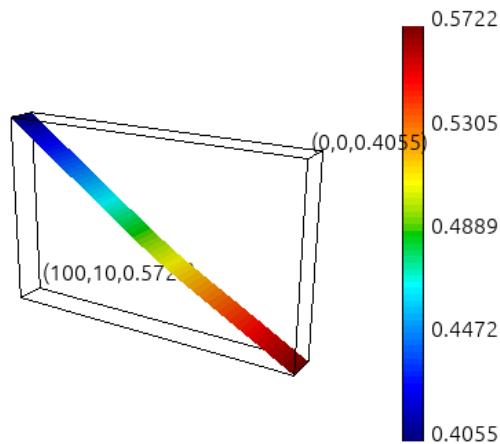


The next two figures are showing a step signal propagating in a 100m rg-58 cable. In the second figure we see the DC signal attenuation.

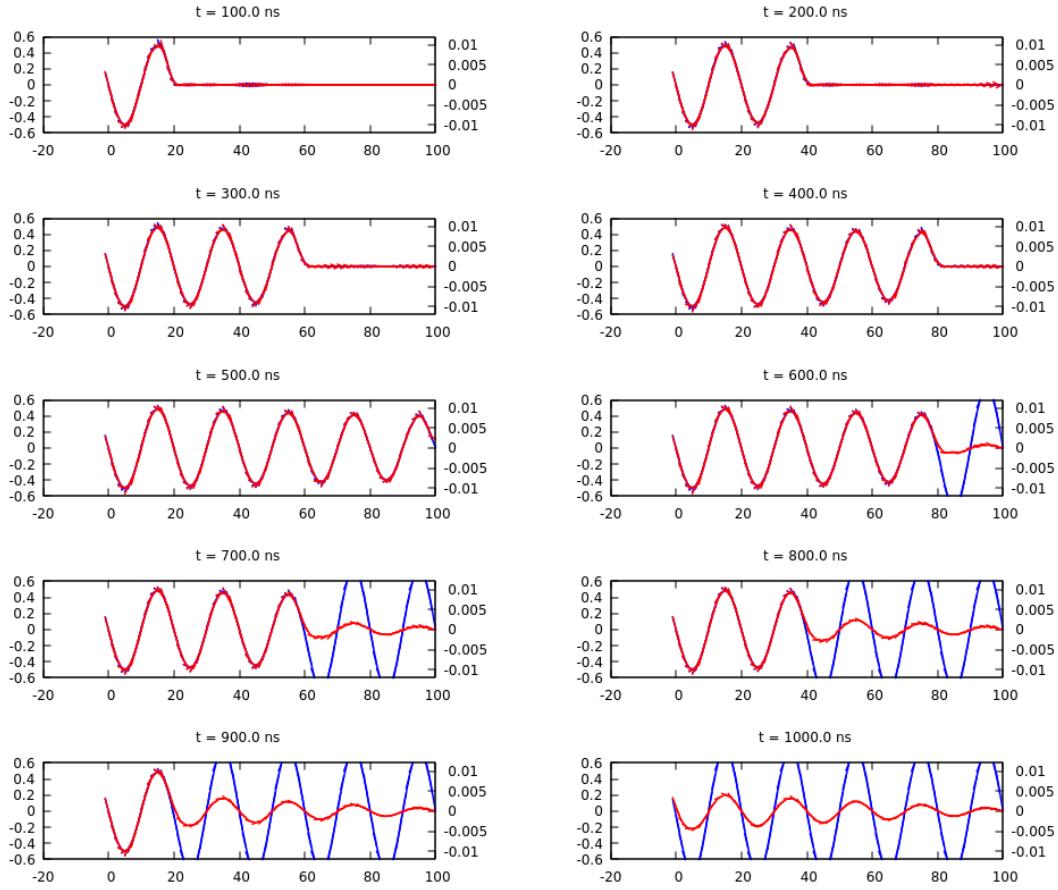
**V after 400.500000ns.**



**V after 800.000000ns.**



The next multiplot shows a sine wave propagating through the transmission line, 100m rg-58 cable. The source is 50ohm the load is 2 ohm, we can see the reflection and the current inverting and the attenuation.



## 2.5 What Next