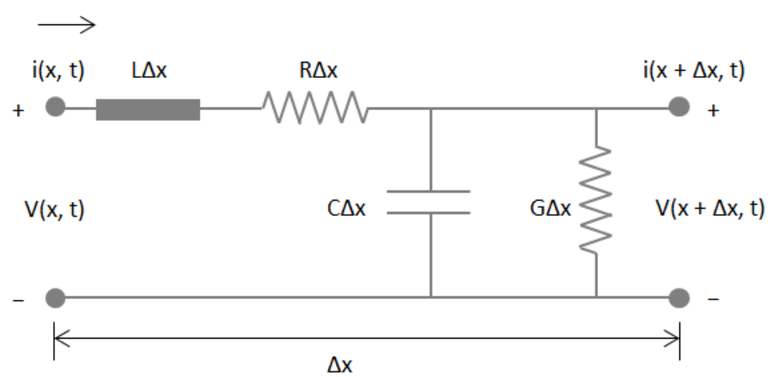
Transmission Line Simulation via Numerical Inversion of Laplace Transforms

Introduction

This application numerically inverts the Laplace transforms that describe the voltage and current in a transmission line.

Consider a lumped parameter model of the transmission line:



Applying Kirchoff's voltage and current laws to this system results in a pair of coupled partial differential equations, commonly called the Telegrapher's Equations:

$$-\frac{\partial}{\partial x} v(x,t) = R_0 \cdot i(x,t) + L_0 \cdot \frac{\partial}{\partial t} i(x,y)$$
$$-\frac{\partial}{\partial x} i(x,t) = G_0 \cdot v(x,t) + C_0 \cdot \frac{\partial}{\partial t} v(x,t)$$

When v(x, 0) = i(x, 0) = 0, the partial differential equations are transformed into Laplace-domain equations, given below (Brancik, 2011).

$$V = V_i(s) \cdot \frac{Z_c(s)}{Z_i(s) + Z_c(s)} \cdot \frac{e^{-\gamma(s)\cdot x} + \rho_2(s) \cdot e^{-\gamma(s)\cdot(2\cdot l - x)}}{1 - \rho_1(s) \cdot \rho_2(s) \cdot e^{-2\cdot\gamma(s)\cdot l}}$$

$$i = V_i(s) \cdot \frac{1}{Z_i(s) + Z_c(s)} \cdot \frac{e^{-\gamma(s) \cdot x} + \rho_2(s) \cdot e^{-\gamma(s) \cdot (2 \cdot l - x)}}{1 - \rho_1(s) \cdot \rho_2(s) \cdot e^{-2 \cdot \gamma(s) \cdot l}}$$

where Z_c and γ(s) are a characteristic impedance and propogation constant, and $\rho_1(s)$ and $\rho_2(s)$ are reflection coefficients:

$$Z_c(s) = \sqrt{\frac{R_0 + s \cdot L_0}{G_0 + s \cdot C_0}} :$$

$$\gamma(s) = \sqrt{\left(R_0 + s \cdot L_0\right) \cdot \left(G_0 + s \cdot C_0\right)} :$$

$$\rho_1(s) = \frac{Z_i(s) - Z_c(s)}{Z_i(s) + Z_c(s)} :$$

$$\rho_2(s) = \frac{Z_2(s) - Z_c(s)}{Z_2(s) + Z_c(s)} :$$

The system is perturbed by a voltage source $V_i(s)$:

$$V_i(s) = \frac{2\pi^2\left(1 - e^{-2 \cdot 10^{-9} s}\right)}{s\left(\left(2 \cdot 10^{-9} s\right)^2 + 4\pi^2\right)} :$$

The Maple code that numerically inverts the Laplace functions is converted from the MATLAB code found in Toutain et al. (2011), which itself is based on the algorithm developed by Iseger (2006).

References:

- Toutain, J., Battaglia, J., Pradere, C., Pailhes, J., Kusiak, A., Aregba, W., and Batsale, J. (January 13, 2011). "Numerical Inversion of Laplace Transform for Time Resolved Thermal Characterization Experiment." ASME. J. Heat Transfer. April 2011; 133(4): 044504. https://doi.org/10.1115/1.4002777
- Iseger, Peter den. "Numerical Transform Inversion Using Gaussian Quadrature". Probability in the Engineering and Informational Sciences, Vol. 20, pp. 1-44, 2006. Available at SSRN: https://ssrn.com/abstract=1013507
- Lubomir Brancik (October 13th 2011). "Numerical Inverse Laplace Transforms for Electrical Engineering Simulation", MATLAB for Engineers - Applications in Control, Electrical Engineering, IT and Robotics, Karel Perutka, IntechOpen, DOI: 10.5772/19824. Available from: https://www.intechopen.com/books/matlab-for-engineers-applications-in-control-electrical-engineering-it-and-robotics/numerical-inverse-laplace-transforms-for-electrical-engineering-simulation


```
> restart :
  local γ:
```

Procedure to Numerically Invert Laplace Transforms

```
#Deniseger method to numerically invert a Laplace tranform

Deniseger:= proc(Lf, t)

local nt,dt,N,a,li,bi,c,k,kq,liq,s,ft,ftq,ftq2,res, i:

   #Parameter settings
   nt := numelems(t):
   dt := max(t) / (nt - 1):
   N  := 8 * nt:
   a  := 44.0 / N;

   # Numerical values of  and  (see Den Iseger p.29 [2006])
   li := Vector([0., 6.28318530717958, 12.5663706962589, 18.8502914166954, 25.2872172156717, 34.2969716635260,
56.1725527716607, 170.533131190126] ):
   bi := Vector([1., 1.00000000000004,  1.00000015116847,  1.00081841700481,  1.09580332705189,  2.00687652338724,
5.94277512934943,  54.9537264520382] ):


   c      := 2 * I * evalf(Pi) / N;
   li     := a +~ I *~ li;

   k      := [seq(i, i = 0 .. N)]:

   kq     := Matrix(numelems(li), numelems(k), (i, j) -> j - 1 );
   liq    := Matrix(numelems(li), numelems(k), (i, j) -> li[i] ):

   s      := (liq + c*kq) / dt :
   ft     := Re~(Lf~(s)):
   ftq    := Vector[row]([seq(4 * ft[ .. , i ] . bi / dt, i = 1 .. LinearAlgebra:-ColumnDimension(ft))]):
   ftq[1] := 0.5 *(ftq[1] + ftq[N + 1]);

   # Discrete Fourier inversion
   ftq2   := SignalProcessing:-InverseFFT(ftq[1 .. N]/sqrt(N)):
   res    := Re~(exp~(a *~ [seq(0 .. nt - 1)]) *~ ftq2[1 .. nt]):

```

>

Laplace Transforms for a Lossy Transmission Line

Parameter values (all parameters are in per-unit dimensions)

> $L := 1.:$

$R_0 := 10^{-3}:$

$L_0 := 600 \cdot 10^{-9}:$

$G_0 := 2 \cdot 10^{-3}:$

$C_0 := 80 \cdot 10^{-12}:$

$Z_i := 10.:$

$Z_2 := 1000:$

$x := 0.5 \cdot L:$

Equations

$$> V := s \to V_i(s) \cdot \frac{Z_c(s)}{Z_i(s) + Z_c(s)} \cdot \frac{e^{-\gamma(s) \cdot x} + \rho_2(s) \cdot e^{-\gamma(s) \cdot (2 \cdot L - x)}}{1 - \rho_1(s) \cdot \rho_2(s) \cdot e^{-2 \cdot \gamma(s) \cdot L}} :$$

$$> i := s \to V_i(s) \cdot \frac{1}{Z_i(s) + Z_c(s)} \cdot \frac{e^{-\gamma(s) \cdot x} + \rho_2(s) \cdot e^{-\gamma(s) \cdot (2 \cdot L - x)}}{1 - \rho_1(s) \cdot \rho_2(s) \cdot e^{-2 \cdot \gamma(s) \cdot L}} :$$

$$> Z_c := s \to \sqrt{\frac{R_0 + s \cdot L_0}{G_0 + s \cdot C_0}} :$$

$$> \gamma := s \to \sqrt{(R_0 + s \cdot L_0) \cdot (G_0 + s \cdot C_0)} :$$

$$> \rho_1 := s \to \frac{Z_i(s) - Z_c(s)}{Z_i(s) + Z_c(s)} :$$

$$> \rho_2 := s \to \frac{Z_2(s) - Z_c(s)}{Z_2(s) + Z_c(s)} :$$

$$> V_i := s \to \frac{2\,\pi^2 \left(1 - e^{-2 \cdot 10^{-9}\, s}\right)}{s \left(\left(2 \cdot 10^{-9}\, s\right)^2 + 4\,\pi^2\right)} :$$

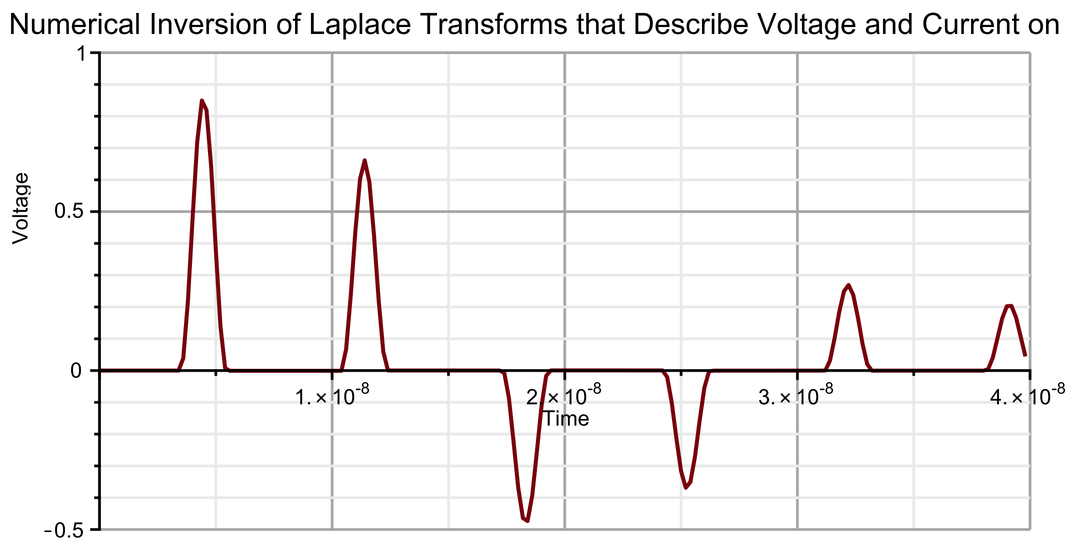
Numerical Inversion of Laplace Functions

$> t := [seq(0.2 \cdot 10^{-9} \cdot (i - 1), i = 1..200)]:$

```
> tt := time[real]( ) :
   volt_results := Deniseger(V, t) :
  time[real]( ) - tt;
```

2.622 **(4.1)**

```
> plot(t, volt_results, view = [0..4. 10^-8, -0.5..1], gridlines, labels = ["Time", "Voltage"],
     labeldirections = [horizontal, vertical], title
     = "Numerical Inversion of Laplace Transforms that Describe Voltage and
     Current on a Transmission Line", titlefont = [Arial, 18], size = [800, 400], axesfont
     = [Arial], labelfont = [Arial])
```



> tt := time[real]( ) :
curr_results := Deniseger(i, t) :
time[real]( ) − tt;

2.762 **(4.2)**

> plot(t, curr_results, view = [0 ..4. 10^−8, default], gridlines, labels = ["Time", "Current"], labeldirections = [horizontal, vertical], title = "Numerical Inversion of Laplace Transforms that Describe Voltage and Current on a Transmission Line", titlefont = [Arial, 18], size = [800, 400], axesfont = [Arial], labelfont = [Arial])