

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Using a Low Complexity Numeric Routine for Solving Electromagnetic Transient Simulations

Rafael Cuerda Monzani, Afonso José do Prado, Sérgio Kurokawa,
Luiz Fernando Bovolato and José Pissolato Filho

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/48507>

1. Introduction

Different methods can be applied to accomplish the analysis of transmission lines. Many mathematical tools can be used, the main tools used are: circuits analysis with the use of Laplace or Fourier Transform, State Variables and Differential Equations. These tools can be included in a numeric routine in order to obtain voltage and current values in simulation of electromagnetic transients, at any point of the circuit.

The EMTP (ElectroMagnetic Transient Program) [1] is the main kind of this software. The prototype was developed in 60's by professionals of power system area led by Dr. Hermann Dommel (University of British Columbia, in Vancouver, B.C., Canada), and Dr. Scott Meyer (Bonneville Power Administration in Portland, Oregon, U.S.A.). Currently, EMTP is the basis of electromagnetic transients simulations in power systems.

With EMTP type programs, the following analysis can be done: simulation of switching and lightning surges, transient and temporary overvoltage, electrical machines, resonance phenomena, harmonics, power quality and power electronics applications. The most known programs of EMTP type are:

- MicroTran Power Systems Analysis of the University of British Columbia, Vancouver, Canada, founded in 1987 by: Hermann W. Dommel, Jose R. Marti (University of British Columbia), and Luis Marti (University of Western Ontario, Hydro One Networks Inc.).
- PSCAD®, also known as PSCAD®/EMTDC™ of Manitoba HVDC Research Centre. Commercially available since 1993, PSCAD® is the result of continuous research and development since 1988.
- ATP has been continuously developed through international contributions by Drs. W. Scott Meyer and Tsu-huei Liu, the co-Chairmen of the Canadian/American EMTP User

Group. The birth of ATP dates to early in 1984. For the free software, however there are some rules for using it.

This kind of program, in general, doesn't present an easy interface where data can be included [5]. Thus, many times, undergraduate students may not be interested in initiating works in this area because the transmission lines studies involve complex models and numeric routines which are truly complex if earth effect and line parameters are considered to be frequently dependent ones.

The objective of this chapter is to introduce concepts about power systems, more specifically, in transmission lines, considering a simplified model of monophasic line in order to analyze electromagnetic transients. [2]-[6].

Considering this purpose, a transmission line can be represented as a monophasic circuit and modeled by circuits (Fig. 1). State variables are used to represent this model. The obtained linear system can be solved by using trapezoidal integration techniques.

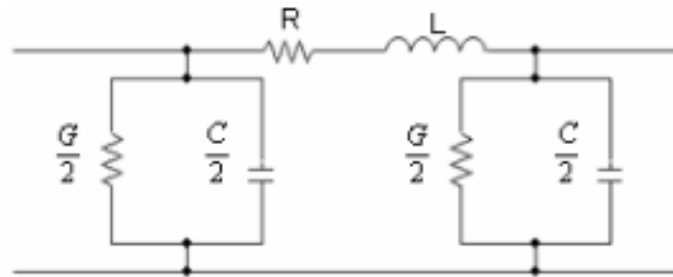


Figure 1. π circuit

Based on these conditions, a simplified numeric routine for a first contact of undergraduate students with the study of travelling waves was obtained. This numeric routine can lead to a satisfactory precision and accuracy for the simulation of electromagnetic transients for a monophasic line transmission representation. The numeric routine was developed with the use of MatLab™.

2. Mathematical model

In order to equate the linear system Kirchhoff's Laws should be used. Nodal analysis is used to calculate the algebraic sum of the currents. Given that the current in the capacitor is defined as the time derivative of the voltage multiplied by the capacitance factor, the first state equation is obtained. Mesh analysis is used to calculate the algebraic sum of the voltages, the voltage across the inductor is defined as the time derivative of the current multiplied by the inductance factor, the second state equation is obtained.

A linear system can be described algebraically by using state equations variables as it follows:

$$\dot{x} = Ax + Bu \quad (1)$$

where: x – vector of state variables;
 u – vector of linear system entries;
 A and B – matrices which feed the system.

The solution of this system can be obtained numerically by trapezoidal rule.

$$x[k+1] - x[k] = \frac{T}{2} (Ax[k+1] + Bu[k+1] + Ax[k] + Bu[k]) \quad (2)$$

On the other hand, to solve the linear system of equations of state variables above, an interactive method can be used, where T is the integration step applied for the system solution. For time domain simulations, the integration step is a time step.

Rearranging equation (2), one can obtain:

$$x[k+1] = x[k] + \frac{T}{2} (Ax[k+1] + Bu[k+1] + Ax[k] + Bu[k]) \quad (3)$$

Solving this equation by using numeric methods, the equation can be rewritten as:

$$\left[I - \frac{T}{2} A \right] x[k+1] = \left[I + \frac{T}{2} A \right] x[k] + \frac{T}{2} B [u[k] + u[k+1]] \quad (4)$$

Verifying the equation (4), there are some constant terms, thus the equation can be simplified to:

$$A'x[k+1] = A''x[k] + B'[u[k] + u[k+1]] \quad (5)$$

A' , A'' e B' are constant matrices and can be described by the following equations:

$$\begin{aligned} A' &= \left[I - \frac{T}{2} A \right] \\ A'' &= A' \cdot \left[I + \frac{T}{2} A \right] \\ B' &= A' \cdot \frac{T}{2} B \end{aligned} \quad (6)$$

In these equations, I is eye matrix of order $(2n \times 2n)$, n is the number of π circuits. A a matrix is based on a cascade of π circuits. B is a matrix that inserts the entry values of the system. If the entrance signal is a voltage source, the inductor will be the component introduced in B

matrix for the correspondent π circuit. If the entrance signal is a current source, the element will be a capacitor.

The line transmission model for analysis without frequency influence over longitudinal parameters is based on a cascade of π circuits, which is constituted of a branch of resistor and inductor in series with a branch of capacitor and conductor in parallel. The precision of system is higher as the amount of π circuits increase.

For the transmission line represented in Fig. 2, the A matrix is described in Eq. 7. In this case, the A matrix has a special format; it's a sparse matrix, with elements non null in the three diagonals as seen below.

$$A = \begin{bmatrix} -\frac{G}{C} & -\frac{2}{C} & 0 & \dots & \dots & 0 \\ \frac{1}{L} & -\frac{R}{L} & -\frac{1}{C} & 0 & \ddots & \vdots \\ 0 & \frac{1}{C} & -\frac{G}{C} & -\frac{1}{L} & 0 & \vdots \\ \vdots & 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 0 & \frac{1}{L} & -\frac{R}{L} & -\frac{1}{C} \\ 0 & \dots & \dots & 0 & \frac{2}{C} & -\frac{G}{C} \end{bmatrix} \quad (7)$$

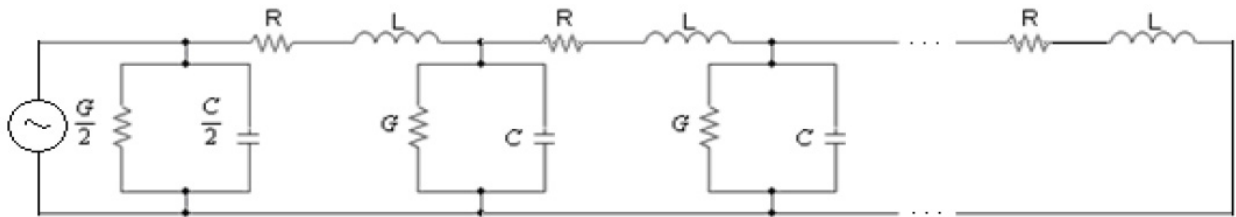


Figure 2. Transmission Line Model

In the main diagonal of the A matrix, there are the negative values of the parameters alternating between $-G/C$ and $-R/L$. In the upper diagonal, the element values alternate between $-1/C$ and $-1/L$. Considering the lower diagonal the element values are positive and similar to the upper diagonal.

The A matrix is a matrix of order $(2n \times 2n)$, if the transmission line is considered opened for sending and receiving in the terminal of the line.

On the other hand, the modeling of frequency influence over longitudinal parameters can be done when introduced branches in parallel of resistor and inductor associated in series with the main branch in series of an inductor and a resistor, in each π circuit [5]. The Fig. 3 shows this description.

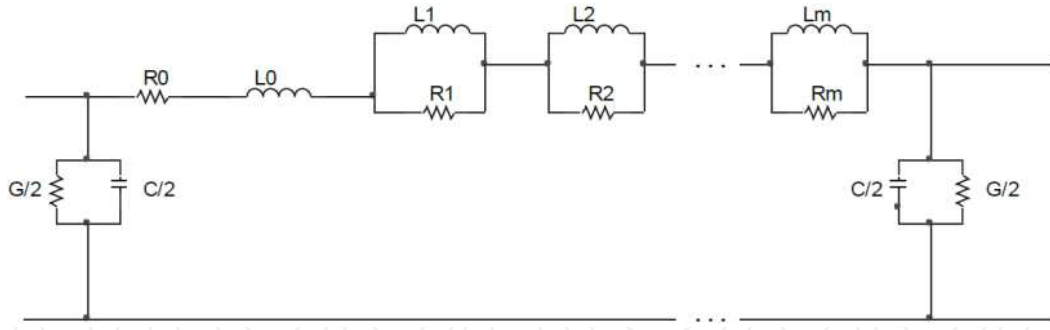


Figure 3. π circuit considering frequency dependency

The parameters R_0 , L_0 refers to resistance and inductance values in series, respectively. The parameters R_m , L_m , with m being the number of branches in series, refers to value of resistance inductance of each one of the m branches in series of each π circuit, respectively. The parameters C and G , refer to the value of capacitance and conductance, respectively.

From analysis of Kirchhoff's Laws of the current in the inductor and the voltage in the capacitor, the linear system of state variables is obtained.

$$\begin{aligned} \frac{di_{10}}{dt} &= \frac{i_{10}}{L_0} \left(-\sum_{j=1}^m R_j \right) + \frac{1}{L_0} \left(\sum_{j=1}^m R_j i_{1j} \right) + \frac{1}{L_0} u(t) - \frac{1}{L_0} v_1(t) \\ \frac{di_{11}}{dt} &= \frac{R_1}{L_1} i_{10} - \frac{R_1}{L_1} i_{11} \\ \frac{di_{12}}{dt} &= \frac{R_2}{L_2} i_{10} - \frac{R_2}{L_2} i_{12} \\ \frac{di_{1m}}{dt} &= \frac{R_m}{L_m} i_{10} - \frac{R_m}{L_m} i_{1m} \\ \frac{dv_1(t)}{dt} &= \frac{2}{C} i_{10} - \frac{G}{C} v_1(t) \end{aligned} \quad (8)$$

n is the number of π circuit and m is the number of series branch. In order to represent these equations, the A matrix is square of order $(m + 2)$ and it's described by:

$$A = \begin{bmatrix} [A_{11}] & [A_{12}] & \cdots & [A_{1n}] \\ [A_{21}] & [A_{22}] & \cdots & [A_{2n}] \\ \vdots & \vdots & \ddots & \vdots \\ [A_{n1}] & [A_{n2}] & \cdots & [A_{nn}] \end{bmatrix} \quad (9)$$

where, each A_{kk} matrix is represented by:

$$A' = \begin{bmatrix} -\frac{\sum_{j=0}^m R_j}{L_0} & \frac{R_1}{L_0} & \frac{R_2}{L_0} & \dots & \frac{R_m}{L_0} & -\frac{1}{L_0} \\ \frac{R_1}{L_1} & -\frac{R_1}{L_1} & 0 & \dots & 0 & 0 \\ \frac{R_2}{L_2} & 0 & -\frac{R_2}{L_2} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ \frac{R_m}{L_m} & 0 & 0 & \dots & -\frac{R_m}{L_m} & 0 \\ \frac{2}{C} & 0 & 0 & \dots & 0 & -\frac{G}{C} \end{bmatrix} \quad (10)$$

For $n \pi$ circuits, the A matrix is formed in the diagonal line by A' matrices, in the upper diagonal these matrices have just one element not null which is in the first column and in the last row and it's described by $-1/C$.

The lower matrices contain just one non null element which is in the last column in first line and it's described by $1/L_0$. Both matrices, upper and lower are $(m+2)$ order one. The B vector is an $n(m+2)$ order with the first element non null described by $1/L_0$. A generic vector x is shown below:

$$x^T = [x_1 \dots x_n] \quad (11)$$

Each element of x has the following structure:

$$x_k^T = [i_{k0} \ i_{k1} \ \dots \ i_{km} \ v_{kn}] \quad (12)$$

This state equation describes the transmission line represented by $n \pi$ circuits, by using numeric methods.

3. Routine development

The routine used for introducing the proposed model shown in previous items is implemented in MatLab™ software. For this, only basic notions of programming are necessary to make the development of this routine easy for undergraduate students. Initially, the source values, the number of π circuits, the line length and the line parameters per unit length in the numeric routine are introduced. By using the line parameters per unit length, the parameter values for each π circuit it is determined by using the following definitions:

$$R = \frac{R' \cdot d}{n} \quad (13)$$

$$L = \frac{L' \cdot d}{n}$$

$$C = \frac{C' \cdot d}{n}$$

$$G = \frac{G' \cdot d}{n}$$

Where R' , L' , C' , G' are the line parameters per unit length, d is the length of the line and n is the number of π circuits. Fig. 4 shows a window program that applies the proposed numeric routine.

In the case of frequency influence the user may choose the number of branches and also specify the value for each resistor and inductor. These values can be calculated by using any routine that considers the frequency influence in transmission line parameters. After this step, the simulation time (t), the time step (T) and the sources that are connected at the line are defined. Then, the equation that describes each source should be introduced in the numeric routine by using a specific MatLab™ tool. Simulations of electromagnetic transient of many different input signals can be obtained just by inserting their functions in MatLab™ program routine as seen in Fig. 4. After specifying all the input values, the routine generates the A and B matrices and also the input vector U in discrete time. Then, the equation (6) terms are calculated numerically by using the chosen time step and the simulation time. The chosen current and voltage output results are uploaded in a vector file. Finally, a graph is plotted with the results of the simulation.

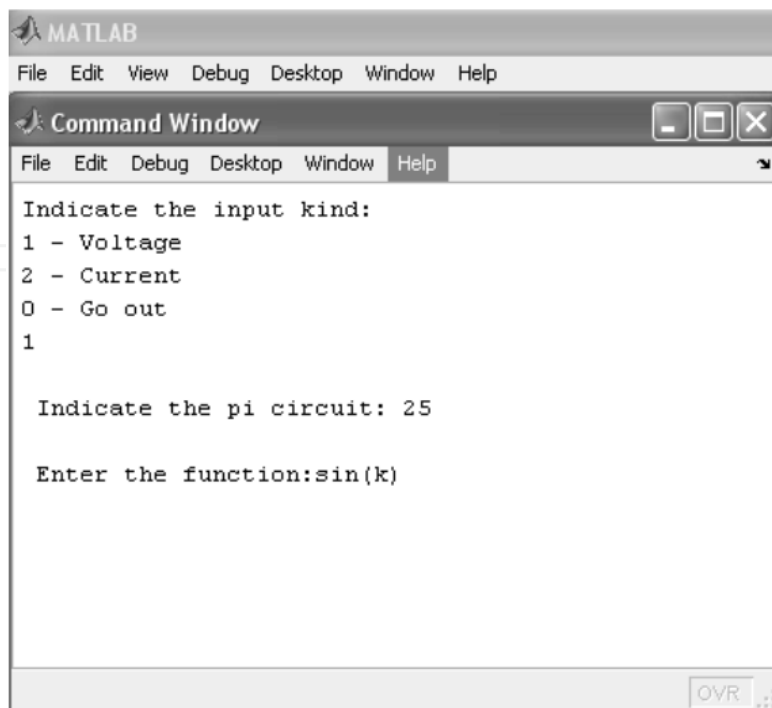


Figure 4. MatLab window.

In order of results, both routines, with and without frequency influence, will be shown and discussed step by step. The objective of these routines is to show how to use MatLab™ by using simple functions or commands to present the wave propagation of current or voltage to undergraduate students. The functions and commands used into routine will be explained; also the loops and conditions used into it will also be explained below the routine.

The first routine doesn't consider the frequency influence.

MatLab code

```
%Clear variables and command window
clear;
clc;
%Close windows
Close all;
%Definitions of Entrance Elements
disp('Transmission Line Analyze')
P = input('Enter the Number of Pi Circuits = ');
D = input('Enter the line length (km) = ');
R = input('Enter the value of distributed resistance (ohm/km) = ');
L = input('Enter the value of distributed inductance (H/km) = ');
C = input('Enter the value of distributed capacitance (F/km) = ');
G = input('Enter the value of distributed conductance (S/km) = ');
T = input('Set time step [s]: ');
t = input('Enter the simulation total time [s]: ');
%Distributed Elements
R=R*D/P;
L=L*D/P;
G=G*D/P;
C=C*D/P;
%e = Pi circuits which will receive entries
cont=1;
while (i ~= 0)
    clc;
    disp('Indicate the input kind:');
    disp('1 - Voltage');
    disp('2 - Current');
    disp('0 - Exit');
    i = input('');
    if (i==1)
        e(cont)= input ('\n Indicate the pi circuit = ');
        fun = input('\n Enter the function:');
        fun = inline(fun);
```

```

    u(2*e(cont)-1,1) = fun;
end
if (i==2)
    e(cont)= input ('\n Indicate the pi circuit= ');
    fun = input('\n Enter the function:');
    fun = inline(fun);
    u(2*e(cont),1) = fun;
end
cont=cont+1;
end
%A matrix
for j=1:(2*P-1)
    h = rem(j,2);
    if h==1
        A(j,j)= -(R/L);
        A(j,j+1)= -(1/L);
        A(j+1,j)= 1/C;
    else
        A(j,j)= -(G/C);
        A(j,j+1)= -(1/C);
        A(j+1,j)= 1/L;
    end
end
end
%A (2*P,2*P)
A(2*P,2*P)=-(G/C);
%B matrix
B(2*P,1)=0;
for j=1: numel(u)
    h = rem(j,2);
    if h==1
        B(j,j)=1/L;
    else
        B(j,j)=1/C;
    end
end
end
%Constant terms
A1=inv(eye(2*P)-(T/2)*A);
A2=(eye(2*P)+(T/2)*A);
A2=A1*A2;
B1=(T/2)*B;
B2=A1*B1;
x(2*P,1)=0;
%Iterations to solve the linear system

```

```

j=1;
for q= 0: T: t
    u1=feval(u,q);
    u2=feval(u,q+T);
    x=A2*x+B2*(u1+u2);
    y(j,1)=q;
    y(j,2)=x(2*P,1);
    y(j,3)=x(2*P-1,1);
    j=j+1;
end
plot(y(:,1),y(:,2),'b')
title('Voltage in the end of transmission line');
ylabel('Voltage [kV]');
xlabel('Time [s]');
pause
plot(y(:,1),y(:,3),'r')
title('Current in the end of transmission line');
ylabel('Current [A]');
xlabel('Time [s]');

```

In the beginning of the routine some commands are used like:

- **clear all:** cleans all the variables existents in MatLab™.
- **clc:** cleans the command window.
- **close all:** closes all the graphics opened.

The command **disp** shows a message in the command window for users to know what happens in the routine, that way no one thinks the routine is not working, because sometimes, it does take a lot of time to finish all the procedures, thus, a message is important to enlighten that.

The command **input** asks the user to insert a value for a variable, instead of defining a constant value inside the routine, this command makes it more interactive, so the user can put any value wanted and analyze the answer for different values inserted.

The loop **while** is used with the purpose to insert voltage or current sources as many as the user wants. The user will specify if the source is of voltage (1) or current (2), after specifying the kind of source, it shall specify the π circuit that will receive the source, and finally specify the function that represents the source. A loop while is used, to mount a vector of entries. The user will get out of the loop inserting the value (0). The function of entry (voltage or current) must be inserted as a string, as explained above. At least one source for the routine works correctly is necessary, as an example, the user can insert a step function in the first circuit. Entering the following:

Indicate the input kind:

1 – Voltage

2 - Current

0 - Exit

1

Indicate the pi circuit = **1**

Enter the function: '**1**'

For mounting the A matrix a loop **for** is used because the number of interactions is known, inside the loop the command **if** is used to construct the mainly, upper and lower diagonals as shown in Eq. 7. The same is done in the B matrix.

Finally, a loop **for** is used to solve the trapezoidal rule considering the time step and the total time elapsed. The variable *y* retains in the first column the value of steps of time, the second column retains the value of voltage, and the third column retains the value of current in the terminal of the line transmission.

Fig. 4 shows the previous version of the routine, in that routine the command **syms** was used. This command creates a symbol as a variable to solve this. The command **subs** must be used, in order to substitute the variable in the function with the value requested. In this case, the function entry of current or voltage source didn't need to be inserted as a string, but the user always had to insert the function by using the variable specified, what was not always done. The second purpose used the command **inline** which gets a string and converts it into a function; with this command the user can use any variable, to solve this in the trapezoidal rule. It's used the command **feval**, which gets the function and substitutes the values with the time step.

The second routine considers the frequency influence. Almost all the steps used in this routine were the same as shown in the first one. Only the different steps done here will be described.

MatLab code

```
%Clear variables and command window
clear all;
clc;
%Close all graphic windows
close all;
disp('Transmission Line Analysis');
sprintf('\n');
P = input('Enter the Number of Pi Circuits = ');
D = input('Enter the line length (km) = ');
%It's defined R(1) and L(1) which represents R0 and L0 respectively,
%values of R1, L1, ... Rm, Lm, will be with a plus number.
R(1) = input('Enter the value of distributed resistance (ohm/km) = ');
```

```

L(1) = input ('Enter the value of distributed inductance (H/km) = ');
C = input ('Enter the value of distributed capacitance (F/km) = ');
G = input ('Enter the value of distributed conductance (S/km) = ');
m = input ('Enter the number of coupling circuits = ');
for i=2:m+1
    R(i) = input (['Enter the value of resistance of the branch ' int2str(i-1) ' [ohm/km] = ']);
    %Distributed Resistance
    R(i) = R(i)*D/P;
    L(i) = input (['Enter the value of inductance of the branch ' int2str(i-1) ' [H/km] = ']);
    %Distributed Inductance
    L(i) = L(i)*D/P;
end
%Time step and total time
T = input ('Set time step [s]: ');
t = input ('Enter the simulation total time [s]: ');
%Distributed Elements
R(1)=R(1)*D/P;
L(1)=L(1)*D/P;
G=G*D/P;
C=C*D/P;
%A matrix
A=zeros(size(P*(m+2)));
c=0;
for i=1:m+1
    A(i,i)=-R(i)/L(i);
    A(1,i)=R(i)/L(1);
    A(i,1)=-A(i,i);
end
%First term as positive
A(1,1) = -A(1,1);
for i=1:m+1
    %c variable will get the values to put in A(1,1), uses recall
    c=A(1,i)+c;
end
A(1,1)=-c;
%Terminal elements of matrix
A(1,m+2)=-1/L(1);
A(m+2,1)=1/C;
A(m+2,m+2)=-G/C;

```

```

%Mainly matrix
AP=A;
%Putting the matrix in all the positions
for j=1:(P-1)
    AP(j*(m+2)+1:(j+1)*(m+2),j*(m+2)+1:(j+1)*(m+2))=A;
end
%Lower matrix
AI(1,m+2)=1/L(1);
AI(m+2,m+2)=0;
%Upper matrix
AS(m+2,1)=-1/C;
AS(m+2,m+2)=0;
%Putting the lower and upper matrices in their places
for j=1:(P-1)
    AP(j*(m+2),j*(m+2)+1)=-1/C;
    AP(j*(m+2)+1,j*(m+2))=1/L(1);
end
%Element of the first matrix
A(m+2,1)=2/C;
%Element of last matrix
A(P*(m+2),1)=2/C;
%e = Pi circuits which will receive entries
%cont sets which pi circuit will receive the entry
cont=1;
while (i ~= 0)
    clc;
    disp('Indicate the input kind:');
    disp('1 - Voltage');
    disp('2 - Current');
    disp('0 - Exit');
    i = input('');
    if (i==1)
        e(cont)= input ('\n Indicate the pi circuit = ');
        fun = input('\n Enter the function:');
        fun = inline(fun);
        u(e(cont)*(m+2)-(m+1),1) = fun;
    end
    if (i==2)
        e(cont)= input ('\n Indicate the pi circuit = ');

```

```

        fun = input('\n Enter the function:');
        fun = inline(fun);
        u(e(cont)*(m+2),1) = fun;
    end
    cont=cont+1;
end
%Clear the command window
clc;
disp('PROCESSING...');
%B matrix
B(P*(m+2),1)=0;
for j=1:m+2: numel(u)
    h = rem(j,2);
    if h==1
        B(j,j)=1/L(1);
    else
        B(j,j)=1/C;
    end
end
end
A1=inv(eye(P*(m+2))-(T/2)*AP);
A2=(eye(P*(m+2))+(T/2)*AP);
A2=A1*A2;
B1=(T/2)*B;
B2=A1*B1;
%x matrix
x(P*(m+2),1)=0;
j=1;
for q = 0: T: t
    u1=feval(u,q);
    u2=feval(u,q+T);
    x=A2*x+B2*(u1+u2);
    y(j,1)=q;
    y(j,2)=x(P*(m+2),1);
    y(j,4)=x(P*(m+2)/2,1); %middle of line
    y(j,3)=x(P*(m+2)-(m+1),1);
    y(j,5)=x((P*(m+2)/2)-(m+1),1); %middle of line
    if (q==30*T)
        for kk = 1:P
            z(kk,1)=kk*D/P;

```

```

        z(kk,2)=x(kk*(m+2),1);
    end
end
if (q==60*T)
    for kk = 1:P
        z(kk,3)=x(kk*(m+2),1);
    end
end
if (q==90*T)
    for kk = 1:P
        z(kk,4)=x(kk*(m+2),1);
    end
end
if (q==120*T)
    for kk = 1:P
        z(kk,5)=x(kk*(m+2),1);
    end
end
if (q==150*T)
    for kk = 1:P
        z(kk,6)=x(kk*(m+2),1);
    end
end
if (q==180*T)
    for kk = 1:P
        z(kk,7)=x(kk*(m+2),1);
    end
end
if (q==210*T)
    for kk = 1:P
        z(kk,8)=x(kk*(m+2),1);
    end
end
if (q==240*T)
    for kk = 1:P
        z(kk,9)=x(kk*(m+2),1);
    end
end
if (q==260*T)

```



```

        for kk = 1:P
            z(kk,10)=x(kk*(m+2),1);
        end
    end
    if (q==290*T)
        for kk = 1:P
            z(kk,11)=x(kk*(m+2),1);
        end
    end
    if (q==310*T)
        for kk = 1:P
            z(kk,12)=x(kk*(m+2),1);
        end
    end
    if (q==330*T)
        for kk = 1:P
            z(kk,13)=x(kk*(m+2),1);
        end
    end
    if (q==360*T)
        for kk = 1:P
            z(kk,14)=x(kk*(m+2),1);
        end
    end
    if (q==390*T)
        for kk = 1:P
            z(kk,15)=x(kk*(m+2),1);
        end
    end
    j=j+1;
end
plot(y(:,1),y(:,2),'b')
title('Voltage in the end of transmission line');
ylabel('Voltage [kV]');
xlabel('Time [s]');
pause
plot(y(:,1),y(:,3),'r')
title('Current in the end of transmission line');
ylabel('Current [A]');

```

```

xlabel('Time [s]');
pause
plot(z(:,1),z(:,2),'b',z(:,1),z(:,5),'r',z(:,1),z(:,11),'g',z(:,1),z(:,15),'k')
title('Voltage in line path');
ylabel('Voltage [kV]');
xlabel('Length (km)');

```

This routine uses the loop **for** in the beginning to obtain all the values series branches. The user specifies the amount of series branches and enters the resistance and inductance values for each branch. In the last loop **for**, a sequence of **if** is used to obtain the wave propagation in different time instants. This is a very important part of the routine, because it shows how the voltage or current waves propagates into the line. This representation shows to undergraduate students that a signal put in the beginning of line will not appear instantaneously in the end of line. It takes a little time, like milliseconds to arrive at the end, because the length of the line is considered, differently from a bipole, the signal put in a terminal is at the other terminal at the same time. Another observation is that, with the EMTP type programs, the user can only analyze a specific point of the circuit and in this case in a time range. The routine shows how the wave propagates into the line for different line points in the time instant.

4. Obtained results

For all simulations the following values were used: the number of π circuits was 100 and the transmission line has 10 kilometers. The resistance value was $0.05 \Omega/Km$. The inductance value was $1 mH/Km$. The capacitance values was $11.11 nF/Km$. The conductance value was $0.556 \mu S/Km$. The time step was $50 ns$ and the period of simulation was $600 \mu s$.

Resistors (Ω)		Inductors (mH)	
R ₀	0,026	L ₀	2,209
R ₁	1,470	L ₁	0,74
R ₂	2,354	L ₂	0,12
R ₃	20,149	L ₃	0,10
R ₄	111,111	L ₄	0,05

Table 1. Longitudinal parameters values using the routine considering the frequency influence

A simulation was made for voltage input unitary step signal. It was a step function with a unitary step after $t = 0s$. The result of this simulation can be seen in Fig. 5. the voltage output at the receiving line end is shown. By using the routine without frequency influence, from the results of Fig. 5, it is observed that there is a period time related to the propagation time of the signal through the line. So, it represents a time delay between the input signal and the output signal. After the time delay, there are oscillations associated to wave reflections on the sending and receiving line ends that compose the shown voltage output.

By using the routine with frequency influence at longitudinal parameters, it is obtained Fig. 6. In this figure, it is noticed that are not so many transients as in Fig. 5, because in this case the series branches dampen the signal.

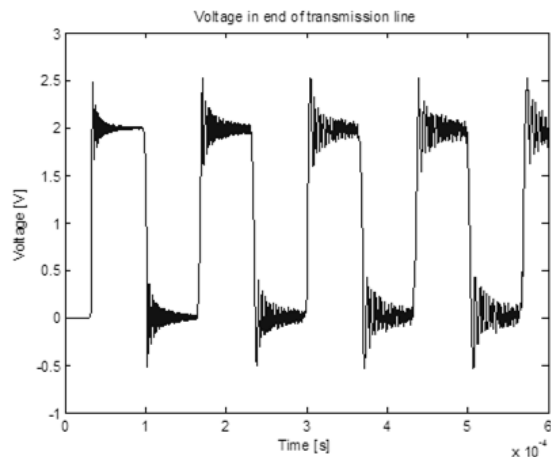


Figure 5. Voltage in the end of transmission line by using routine without frequency influence.

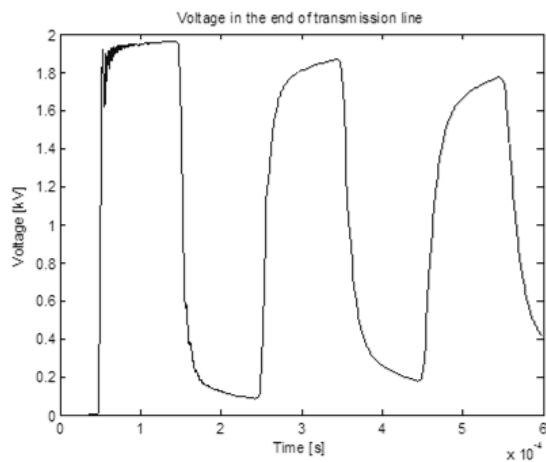


Figure 6. Voltage in the end of transmission line by using routine with frequency influence

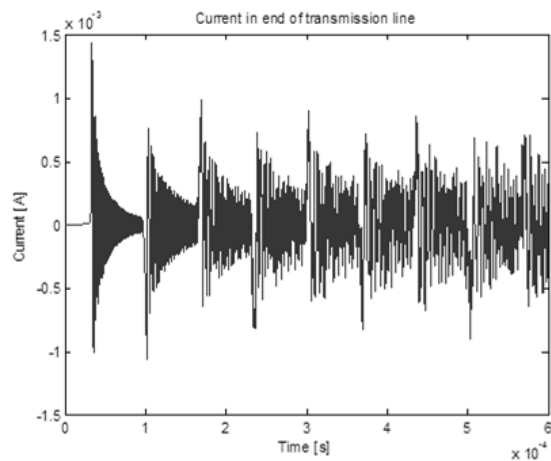


Figure 7. Current in the end of transmission line by using routine without frequency influence.

In Figs. 7 and 8, it's possible to observe the influence of frequency in the current. Without frequency influence the signal is very disturbed. On the other hand, when it is used the routine that considers frequency influence, it is noticed that in the end of the line it should not have any current, supposing an end line opened, but, there are some transients because of the last branch. As in voltage signal, the current signal is damped because of the series branches in circuit. By using the routine of frequency influence, it can also be obtained how the voltage signal goes to the path of line for the time in Fig. 9.

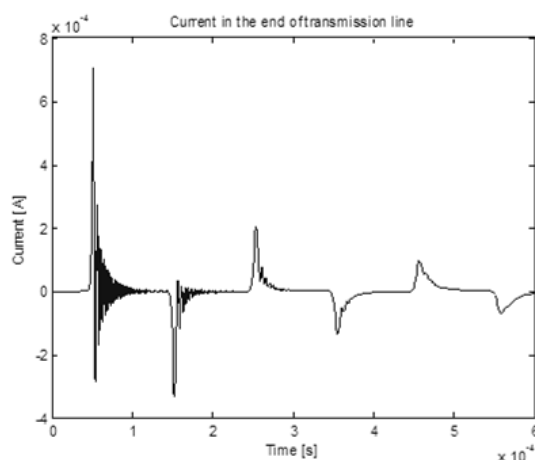


Figure 8. Current in the end of transmission line by using routine with frequency influence.

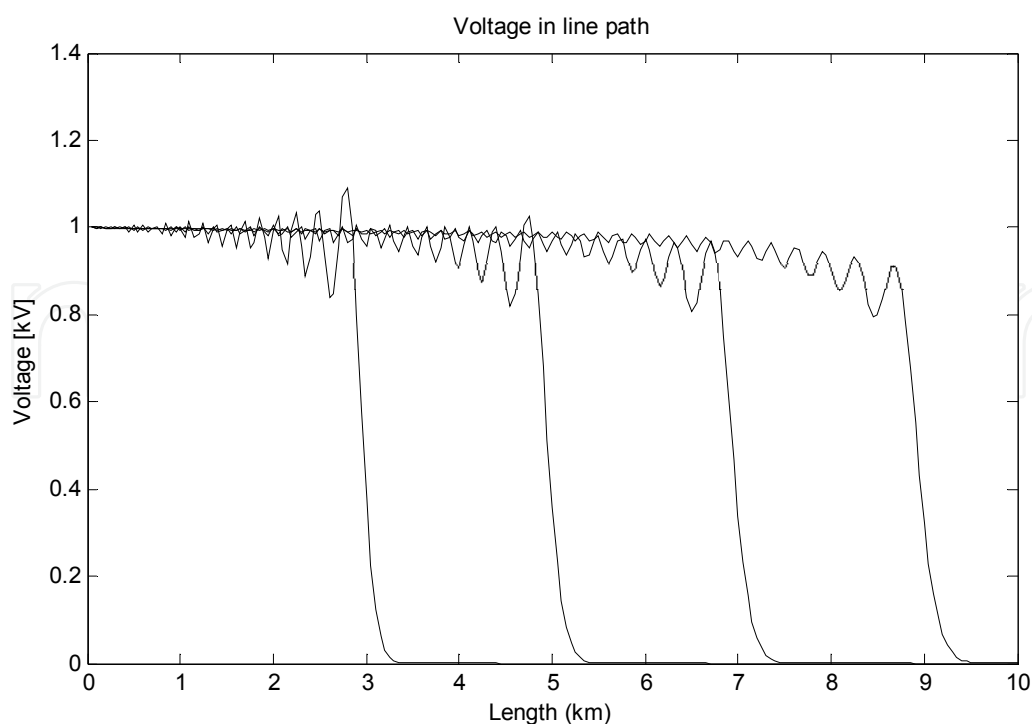


Figure 9. Voltage in line path.

5. Conclusions

By using a mono-phase circuit representation of transmission lines associated to the state variables and the trapezoidal rule, π circuits are applied for electromagnetic transient simulations. The π circuits represent the mono-phase circuit of transmission lines and considering no frequency influence, they are included in a numeric routine through a sparse matrix where only three diagonal lines have non-null elements. Considering the frequency dependent longitudinal line parameters, the parameters are included in a matrix, by using the first line, the first column and the main diagonal of this matrix. This matrix is introduced in another large matrix, rounded by other matrices with one non-null element each. In the upper matrix, the parameter is introduced $-1/C$. In the lower matrix, the parameter is introduced $1/L_0$. The obtained linear system, which is based on state variables, it is solved by using trapezoidal rule integration method. The numeric solution uses the trapezoidal rule method, which solves the numeric integration by the addition of infinitesimal trapeziums areas. This method increases the accuracy of the calculation, comparing with the Euler method for the same time step. The numeric routine is applied to a mathematical matricial program and, because of this, it is possible to simulate the propagation of electromagnetic transients on transmission lines. Using MatLabTM software, it is possible to describe the input signal through mathematical functions that are included as an initial datum of the numeric routine. It is not included in the structure of the routine, but during the routine running. By using the mentioned routine, examples of a unitary step were applied as voltage input signal and some electromagnetic transients generated from these signals are shown in this chapter. The numeric routine permits the inclusion of any function describing the voltage or current source that represents the simulated transient. The included function can be located on any point of the represented transmission line.

The shown numeric routine is simple and because of this, it can be used by undergraduate students for their first contact with electromagnetic transient phenomena, by traveling wave propagation, transmission line analyses. For the first contact with wave propagation simulations for undergraduate students, the proposed routine is an excellent tool. It is simple and its use is easy. So, by using basic concepts of traveling wave, linear systems and computing, it is possible to manipulate the numeric routine, observing the wave propagation characteristics, such as time delays, wave reflection and refraction, numeric oscillations (Gibbs' oscillations), transient oscillations. For undergraduate students, it is possible to compare the results and the modeling of the electromagnetic transients in transmission lines considering or not frequency dependent line parameters. This is possible by accessing the proposed routine numeric code. In courses related to the transmission line area, the manipulation of this code can make the course more interesting for the undergraduate students. These advantages are added to the other ones that are shown in this chapter and related to the application of the MatLabTM software. The use of this software makes the implementation of the mentioned routine extremely easy.

Author details

Rafael Cuerda Monzani, Afonso José do Prado,

Sérgio Kurokawa and Luiz Fernando Bovolato

Department of Electrical Engineering at FEIS/UNESP – The University of São Paulo State, Brazil

José Pissolato Filho

Department of Electrical Engineering, DSCE/UNICAMP – The State University of Campinas, Brazil

6. References

- [1] Microtran Power System Analysis Corporation, Transients Analysis Program Reference Manual, Vancouver, Canada, 1992.
- [2] R. M. Nelms, Steven, R. Newton, G. B. Sheble, L. L. Grigsby. "Simulation of Transmission Line Transients", IEEE.
- [3] R. M. Nelms, Steven, R. Newton, G. B. Sheble, L. L. Grigsby. "Using a personal Computer to teach power system Transients", IEEE Transaction on Power System, Vol. 4, No. 3, August 1989.
- [4] S. R. Newton, B. B. Reid, G. B. Sheble, R. M. Nelms, L. L. GRIGSBY. "Electromagnetic transient simulator for large-scale spacecraft power systems". IEEE.
- [5] H. W. Dommel, EMTP Theory Book, Department of Electrical Engineering, University of British Columbia, Vancouver, 1996.
- [6] J. R. Marti, "Accurate modeling of frequency-dependent transmission lines in electromagnetic transients simulations", IEEE Trans. on PAS, vol. 101, pp. 147–155, January 1982.
- [7] E. Clarke, Circuit Analysis of AC Power Systems, vol. I, Wiley, New York, 1950.
- [8] S. Kurokawa, F. N. R. Yamanaka, A. J. Prado, L. F. Bovolato, J. Pissolato, "Representação de Linhas de Transmissão por meio de variáveis de Estado Levando em Consideração o Efeito da Frequência Sobre os Parâmetros Longitudinais", SBA. Sociedade Brasileira de Automática, Controle & Automação, v. 18, p. 337-346, 2007.
- [9] S. Kurokawa, F. N. R. Yamanaka, A. J. Prado, J. Pissolato, L. F. Bovolato, "Utilização de variáveis de estado no desenvolvimento de modelos de linhas de transmissão: Inclusão do efeito da frequência nas matrizes de estado", XIX Seminário nacional de produção e transmissão de energia elétrica (XIX SNPTEE), p. 1-8, Rio de Janeiro. 2007.
- [10] S. Kurokawa, F. N. R. Yamanaka, A. J. Prado, J. Pissolato, "Representação de linhas de transmissão por meio de variáveis de estado considerando o efeito da frequência sobre os parâmetros longitudinais", XVI Congresso Brasileiro de Automática, v. 1, p. 268-273, Salvador, 2006.

- [11] F. N. R. Yamanaka, S. Kurokawa, A. J. Prado, J. Pissolato, L. F. Bovolato, “Analysis of longitudinal and temporal distribution of electromagnetic waves in transmission lines by using state-variable techniques”, Sixty Latin-American Congress: Electricity Generation and Transmission – VI CLAGTEE, Mar Del Plata, 2005.

IntechOpen

IntechOpen