Utilization of MATLAB in Simulation of Linear Hybrid Circuits

Lubomír BRANČÍK

Dept. of Theoretical and Experimental Electrical Engineering, Brno University of Technology, Purkyňova 118, 612 00 Brno, Czech Republic

brancik@feec.vutbr.cz

Abstract. *In the paper a MATLAB–based method for simulating transient phenomena in linear hybrid circuits containing parts with both lumped and distributed parameters is presented. Distributed parts of the circuit are multiconductor transmission lines, which can generally be nonuniform, with frequency–dependent parameters, and under nonzero initial voltage and/or current distributions. In principle a solution is formulated using the modified nodal analysis method in the frequency domain. Subsequently an improved fast method of the numerical inversion of Laplace transforms in the vector or matrix form is applied to obtain solution in the time domain*

Keywords

Linear hybrid circuit, Matlab language, time-domain simulation, transmission line.

1. Introduction

A possible method which is general enough to be used for the analysis of hybrid circuits is the modified nodal analysis method (MNA) [1, 2]. In [3] this technique is used for the time–domain simulation of multiconductor transmission line (MTL) systems using the Matlab language. The circuit configuration under consideration can be illustrated by the block diagram in Fig.1.

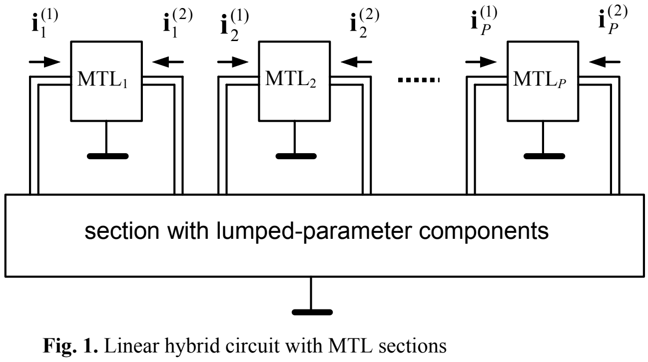


Fig. 1. Linear hybrid circuit with MTL sections

In [3], an MNA matrix equation describing MTL systems under nonzero initial conditions is formulated and the effectiveness of the solution in terms of the Matlab program environment is shown. Unlike in [1, 2], where the modal analysis technique is used, admittance matrices of the MTLs are computed by means of chain matrices. In this way the inhomogeneities of MTLs can easily be considered if necessary. To incorporate nonzero initial conditions into MTLs, matrix convolution integrals must be solved. In the Matlab language, this can effectively be made by the FFT when three-dimensional arrays are utilized [4]. From a general point of view, the solution is performed in the frequency domain and then a fast NILT method in the vector or matrix form is used to obtain the solution in the time domain. Unlike in [3], an improved NILT method based on the FFT and a special quotient-difference algorithm is used here to ensure both high speed of computation and the necessary precision of simulation. Where possible, the Matlab language capabilities to process multidimensional arrays in parallel are utilized with advantage.

2. MNA Matrix Equation Formulation

As shown, for example in [1, 2], a modified nodal analysis matrix equation in the time domain can be written

$$\mathbf{C}_M \frac{d\mathbf{v}_M(t)}{dt} + \mathbf{G}_M \mathbf{v}_M(t) + \sum_{k=1}^{P} \mathbf{D}_k \mathbf{i}_k(t) = \mathbf{i}_M(t) \ , \qquad (1)$$

where $\mathbf{C}_M$ and $\mathbf{G}_M$ are the $N\times N$ constant matrices with entries determined by the lumped memory and memoryless components, respectively, $\mathbf{v}_M(t)$ is the $N\times 1$ vector of node voltages appended by currents of independent voltage sources and inductors, $\mathbf{i}_M(t)$ is the $N\times 1$ vector of source waveforms, $\mathbf{i}_k(t)$ is the $n_k\times 1$ vector of currents entering the k-th MTL, and $\mathbf{D}_k$ is the $N\times n_k$ selector matrix with entries $d_{i,j}\in\{0,1\}$ mapping the vector $\mathbf{i}_k(t)$ into the node space of circuit. To get a frequency-domain representation of the last equation the Laplace transform is applied

$$[\mathbf{G}_M + s\mathbf{C}_M]\mathbf{V}_M(s) + \sum_{k=1}^{P} \mathbf{D}_k \mathbf{I}_k(s) = \mathbf{I}_M(s) + \mathbf{C}_M \mathbf{v}_M(0) \ . \qquad (2)$$

MTLs consist of $N_k=n_k/2$ active conductors, i.e. they can be regarded as $2N_k$-ports. Then $\mathbf{I}_k(s)$ in (2) is formed to contain vectors of currents entering the input and output ports as $\mathbf{I}_k(s) = [\ \mathbf{I}_k^{(1)}(s), \mathbf{I}_k^{(2)}(s)\ ]^T$, and they result from the basic MTL matrix equation as follows.

Suppose a generally nonuniform MTL of length l, with per-unit-length matrices $\mathbf{R}(x)$, $\mathbf{L}(x)$, $\mathbf{G}(x)$, and $\mathbf{C}(x)$. In the time domain a MTL matrix equation has the form [5]

$$\frac{\partial}{\partial x}\begin{bmatrix}\mathbf{v}(x,t)\\ \mathbf{i}(x,t)\end{bmatrix}=\begin{bmatrix}\mathbf{0} & -\mathbf{R}(x)\\ -\mathbf{G}(x) & \mathbf{0}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{v}(x,t)\\ \mathbf{i}(x,t)\end{bmatrix}-\begin{bmatrix}\mathbf{0} & \mathbf{L}(x)\\ \mathbf{C}(x) & \mathbf{0}\end{bmatrix}\cdot\frac{\partial}{\partial t}\begin{bmatrix}\mathbf{v}(x,t)\\ \mathbf{i}(x,t)\end{bmatrix}, \quad (3)$$

and, after the Laplace transform has been used, (3) leads to

$$\frac{d}{dx}\begin{bmatrix}\mathbf{V}(x,s)\\ \mathbf{I}(x,s)\end{bmatrix}=\begin{bmatrix}\mathbf{0} & -\mathbf{Z}(x,s)\\ -\mathbf{Y}(x,s) & \mathbf{0}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{V}(x,s)\\ \mathbf{I}(x,s)\end{bmatrix}+\begin{bmatrix}\mathbf{0} & \mathbf{L}(x)\\ \mathbf{C}(x) & \mathbf{0}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{v}(x,0)\\ \mathbf{i}(x,0)\end{bmatrix}. \quad (4)$$

Here $\mathbf{V}(x,s) = \mathsf{L}[\mathbf{v}(x,t)]$ and $\mathbf{I}(x,s) = \mathsf{L}[\mathbf{i}(x,t)]$ are column vectors of the Laplace transforms of voltages and currents at distance x from MTL's left end, respectively, $\mathbf{v}(x,0)$ and $\mathbf{i}(x,0)$ are column vectors of initial voltage and current distributions, respectively, and $\mathbf{0}$ means zero matrix. $\mathbf{Z}(x,s)=\mathbf{R}(x)+s\mathbf{L}(x)$ and $\mathbf{Y}(x,s)=\mathbf{G}(x)+s\mathbf{C}(x)$ are series impedance and shunting admittance matrices, respectively. In the compact matrix form (4) changes to

$$\frac{d}{dx}\mathbf{W}(x,s)=\mathbf{M}(x,s)\mathbf{W}(x,s)+\mathbf{N}(x)\mathbf{w}(x,0)\,. \quad (5)$$

Taking then $\mathbf{W}(0,s)$ as the solution for x=0 (MTL input) the solution for $x=l$ (MTL output) can be written as [6]

$$\mathbf{W}(l,s)=\mathbf{\Phi}_0^l(s)\mathbf{W}(0,s)+\int_0^l\mathbf{\Phi}_\xi^l(s)\mathbf{N}(\xi)\mathbf{w}(\xi,0)d\xi\,, \quad (6)$$

where $\mathbf{\Phi}_0^l(s)$ is the integral matrix (matrizant) defined with an infinite series of matrix integrals or with so-called product-integral, see, for example, [6]. In the case of a uniform MTL the matrix exponential function is used for its exact calculation as

$$\mathbf{\Phi}_\mathbf{0}^l(s)\Big|_{\mathbf{M}(x,s)=\mathbf{M}(s)}=e^{\mathbf{M}(s)\cdot l}\,. \quad (7)$$

In general, however, only an approximate integral matrix can be calculated. This can be done by dividing the MTL into a sufficiently large number m of sections assuming that $\mathbf{M}(s)$ is constant in each of them. Taking then into account the basic property of matrizant the following recurrent formula holds:

$$\widetilde{\mathbf{\Phi}}_0^{x_j}(s)=e^{\mathbf{M}(\zeta_j,s)\Delta x_j}\cdot\widetilde{\mathbf{\Phi}}_0^{x_{j-1}}(s)\,, \quad (8)$$

with $\widetilde{\mathbf{\Phi}}_0^0(s)=\mathbf{E}$ as the identity matrix, $\Delta x_j=x_j-x_{j-1}$, $j=1,2,\ldots,m$, and $x_0=0$, $x_m=l$, and $\zeta_j\in\langle x_{j-1},x_j\rangle$. In the case of uniform MTL the result is the same as if calculated by (7).

In terms of the multiport theory the integral matrix acts as a chain matrix $\mathbf{\Phi}(s)$. Thus after denoting

$$\mathbf{W}(0,s)=\mathbf{W}^{(1)}(s)=[\mathbf{V}^{(1)}(s),\mathbf{I}^{(1)}(s)]^T\,, \quad (9)$$

$$\mathbf{W}(l,s)=\mathbf{W}^{(2)}(s)=[\mathbf{V}^{(2)}(s),-\mathbf{I}^{(2)}(s)]^T\,, \quad (10)$$

$$\int_0^l\mathbf{\Phi}_\xi^l(s)\mathbf{N}(\xi)\mathbf{w}(\xi,0)d\xi=\mathbf{W}^{(0)}(s)=[\mathbf{V}^{(0)}(s),\mathbf{I}^{(0)}(s)]^T \quad (11)$$

the MTL is described by (6) in the decomposed form:

$$\begin{bmatrix}\mathbf{V}^{(2)}(s)\\ -\mathbf{I}^{(2)}(s)\end{bmatrix}=\begin{bmatrix}\mathbf{\Phi}_{11}(s) & \mathbf{\Phi}_{12}(s)\\ \mathbf{\Phi}_{21}(s) & \mathbf{\Phi}_{22}(s)\end{bmatrix}\cdot\begin{bmatrix}\mathbf{V}^{(1)}(s)\\ \mathbf{I}^{(1)}(s)\end{bmatrix}+\begin{bmatrix}\mathbf{V}^{(0)}(s)\\ \mathbf{I}^{(0)}(s)\end{bmatrix}. \quad (12)$$

After some manipulations and taking into account nonzero initial conditions the admittance equations have the form:

$$\begin{bmatrix}\mathbf{I}^{(1)}(s)\\ \mathbf{I}^{(2)}(s)\end{bmatrix}=\begin{bmatrix}\mathbf{Y}_{11}(s) & \mathbf{Y}_{12}(s)\\ \mathbf{Y}_{21}(s) & \mathbf{Y}_{22}(s)\end{bmatrix}\cdot\begin{bmatrix}\mathbf{V}^{(1)}(s)\\ \mathbf{V}^{(2)}(s)\end{bmatrix}-\begin{bmatrix}\mathbf{Y}_{12}(s) & \mathbf{0}\\ \mathbf{Y}_{22}(s) & \mathbf{E}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{V}^{(0)}(s)\\ \mathbf{I}^{(0)}(s)\end{bmatrix}, \quad (13)$$

where submatrices $\mathbf{Y}_{11}(s)=-\mathbf{\Phi}_{12}^{-1}(s)\mathbf{\Phi}_{11}(s)$, $\mathbf{Y}_{12}(s)=\mathbf{\Phi}_{12}^{-1}(s)$, $\mathbf{Y}_{22}(s)=-\mathbf{\Phi}_{22}(s)\mathbf{\Phi}_{12}^{-1}(s)$, and $\mathbf{Y}_{21}(s)=\mathbf{Y}_{12}^T(s)$, because of the reciprocity of the MTL. In the case of a uniform MTL the equality $\mathbf{Y}_{11}(s)=\mathbf{Y}_{22}(s)$ is also valid. Considering the k–th MTL Eqn. (13) is written in the compact matrix form

$$\mathbf{I}_k(s)=\mathbf{Y}_k(s)\mathbf{V}_k(s)-\mathbf{X}_k(s)\mathbf{\Gamma}_k(s)\,. \quad (14)$$

Finally, after substituting (14) into basic MNA equation (2) the resultant MNA equation can be written in the form:

$$\mathbf{V}_M(s)=\left[\mathbf{G}_M+\mathbf{sC}_M+\sum_{k=1}^P\mathbf{D}_k\mathbf{Y}_k(s)\mathbf{D}_k^T\right]^{-1}\cdot$$
$$\cdot\left[\mathbf{I}_M(s)+\mathbf{C}_M\mathbf{v}_M(0)+\sum_{k=1}^P\mathbf{D}_k\mathbf{X}_k(s)\mathbf{W}_k^{(0)}(s)\right]. \quad (15)$$

To solve voltages and currents at the x coordinate from the beginning [(1)] of the MTL, Eqn. (12) can be written as

$$\begin{bmatrix}\mathbf{V}(x,s)\\ \mathbf{I}(x,s)\end{bmatrix}=\begin{bmatrix}\mathbf{\Phi}_{11}(x,s) & \mathbf{\Phi}_{12}(x,s)\\ \mathbf{\Phi}_{21}(x,s) & \mathbf{\Phi}_{22}(x,s)\end{bmatrix}\cdot\begin{bmatrix}\mathbf{V}^{(1)}(s)\\ \mathbf{I}^{(1)}(s)\end{bmatrix}+\begin{bmatrix}\mathbf{V}^{(0)}(x,s)\\ \mathbf{I}^{(0)}(x,s)\end{bmatrix}, \quad (16)$$

where $\mathbf{\Phi}(x,s)$ is the partial chain matrix computed via (8) and $[\mathbf{V}^{(0)}(x,s),\mathbf{I}^{(0)}(x,s)]=\mathbf{W}^{(0)}(x,s)$ is expressed by matrix integral (11) while replacing indices l by x. As this integral expression is of the convolution type, the method based on the FFT can be used for its calculation. In [4] it is proposed to use three-dimensional arrays when the Matlab capabilities to treat multidimensional arrays in parallel are utilized. The necessary voltage $\mathbf{V}^{(1)}(s)$ and current $\mathbf{I}^{(1)}(s)$ pertaining to the k–th MTL can be extracted from the equation

$$\mathbf{V}_k(s)=\mathbf{D}_k^T\mathbf{V}_M(s) \quad (17)$$

and from Eqn. (14), respectively.

3. Advanced FFT-Based NILT Method

The original $f(t)$ to a Laplace transform $F(s)$ can be expressed by the Bromwich integral

$$f(t)=\frac{1}{2\pi j}\int_{c-j\infty}^{c+j\infty}F(s)e^{st}ds\,, \quad (18)$$

on the basic assumption $|f(t)|\leq Ke^{\alpha t}$, K real positive, α as an exponential order of the real function $f(t)$, $t\geq 0$, and $F(s)$ defined for $\mathrm{Re}[s]>\alpha$. Integrating (18) numerically an approximate formula in the discrete form $\tilde{f}_k=\tilde{f}(kT)$, $k=0,\ldots,N-1$, can be derived [7]:

$$\widetilde{f}_k = C_k\{2\,\mathrm{Re}[\sum_{n=0}^{N-1} F_n z_k^n + \sum_{n=0}^{\infty} G_n z_k^n] - F_0\}\,, \tag{19}$$

where

$$C_k = \frac{\Omega}{2\pi} e^{ckT},\ z_k = e^{-jkT\Omega}\,,\ F_n = F(c - jn\Omega)\,,\ G_n = F_{N+n}\,, \tag{20}$$

with T and $\Omega=2\pi/(NT)$ as the sampling periods in the original and the transform domains, respectively. The error analysis has resulted in an approximate formula for c:

$$c \approx \alpha - \Omega/2\pi \cdot \ln E_r\,, \tag{21}$$

where E_r denotes the desired relative error. The finite sum in (19) is evaluated by the FFT, assuming $N=2^m$, m is an integer. This enables obtaining a set of N points in a single calculation step. Consequently, the required maximum time is taken as $t_m=(M-1)T$, with $M=N/2$ as the number of resultant computed points. To make the error come closer to its theoretical value E_r the infinite sum in (19) must be evaluated as accurately as possible. For this purpose just the quotient-difference algorithm is used to accelerate its convergence. Thus taking into account only the first $2P+1$ terms of this sum the continued fraction is found as [8]

$$v(z_k, P) = d_0/(1 + d_1 z_k/(1 + \cdots + d_{2P} z_k))\ ,\ \forall k\,, \tag{22}$$

which corresponds to the Padé rational approximation of power series. The q-d algorithm is illustrated in Fig.2.

$$\begin{array}{ccccc}
e_0^{(0)} & & & & \\
 & q_1^{(0)} & & & \\
e_0^{(1)} & & e_1^{(0)} & & \\
 & q_1^{(1)} & & q_2^{(0)} & \\
e_0^{(2)} & & e_1^{(1)} & & e_2^{(0)} \\
 & q_1^{(2)} & & q_2^{(1)} & \\
e_0^{(3)} & & e_1^{(2)} & & \\
 & q_1^{(3)} & & & \\
e_0^{(4)} & & & &
\end{array}$$

Fig. 2. Quotient-difference algorithm diagram

The first two columns are formed as

$$e_0^{(i)} = 0\ ,\qquad i = 0,\cdots,2P\,, \tag{23}$$

$$q_1^{(i)} = G_{i+1}/G_i\ ,\ i = 0,\cdots,2P-1\,, \tag{24}$$

and the successive columns are given by the rules as follows:
for $r=1,\ldots,P$,

$$e_r^{(i)} = q_r^{(i+1)} - q_r^{(i)} + e_{r-1}^{(i+1)}\ ,\ i = 0,\cdots,2P-2r\,, \tag{25}$$

for $r=2,\ldots,P$,

$$q_r^{(i)} = q_{r-1}^{(i+1)} e_{r-1}^{(i+1)}/e_{r-1}^{(i)}\ ,\qquad i = 0,\cdots,2P-2r-1\,. \tag{26}$$

Then the coefficients d_n, $n=0,\ldots,2P$, are given by

$$d_0 = G_0\ ,\ d_{2m-1} = -q_m^{(0)}\ ,\ d_{2m} = -e_m^{(0)}\ , \tag{27}$$

$m=1,\ldots,P$. The evaluation of the continued fraction (22) can also be based on a recurrent formula. For any z_k it is valid [8]

$$A_n(z_k) = A_{n-1}(z_k) + d_n z_k A_{n-2}(z_k)\ , \tag{28}$$

$$B_n(z_k) = B_{n-1}(z_k) + d_n z_k B_{n-2}(z_k)\ ,\ \forall k\,, \tag{29}$$

where $n=1,\ldots,2P$, with the initial values

$A_{-1} = 0$, $B_{-1} = 1$, $A_0 = d_0$, and $B_0 = 1$.

Then the continued fraction (22) can be expressed as

$$v(z_k, P) = A_{2P}(z_k)/B_{2P}(z_k)\ ,\ \forall k\,. \tag{30}$$

Finally, $v(z_k,P)$ is used in (19) instead of the original infinite sum.

To get the time-domain solution of (15) the fast vector version of the NILT method is used as follows [9]: If a transform is a vector $\mathbf{F}^J(s)=[F_1(s),F_2(s),\ldots,F_J(s)]^T$ then an NILT formula in the matrix form can be written as

$$\widetilde{\mathbf{f}}^{J\times M} = \mathbf{C}^{J\times M} \circ \{2\,\mathrm{Re}[\mathsf{R}^{J\times M}\{\underset{<2>}{FFT}(\mathbf{F}^{J\times N})\} + \mathbf{V}_P^{J\times M}] - \mathbf{F}_0^{J\times M}\}\,, \tag{31}$$

where all the terms are matrices of superscribed sizes computed according to (20), but formed for all the vector components. The subscript <2> means that the FFT operation runs along the 2nd dimension (columns) but in parallel for all the rows. $\mathbf{V}_p^{J\times M}$ is the matrix resulting from (29), $\mathsf{R}^{J\times M}\{\ \}$ denotes the operator of reducing the matrix dimension $N\to M$ and the symbol $\circ$ means the Hadamard product of matrices. Similarly, to find the time-domain solution of (16) the matrix version of the NILT method is the most effective to use [10]. The formula is

$$\widetilde{\mathbf{f}}^{J\times M\times L} = \mathbf{C}^{J\times M\times L} \circ \{2\,\mathrm{Re}[\mathsf{R}^{J\times M\times L}\{\underset{<2>}{FFT}(\mathbf{F}^{J\times N\times L})\} + \mathbf{V}_P^{J\times M\times L}] - \mathbf{F}_0^{J\times M\times L}\}\,, \tag{32}$$

with terms as three–dimensional arrays of superscribed sizes.

4. MATLAB Functions Definition

Below the Matlab listings are presented for both the vector and the matrix version of the NILT functions. They are called with three parameters: ´F´ - the function defining the Laplace transform, tm - the maximum time, ´pl´ - the plotting function.

```
%NILTV-FUNCTION DEFINITION (vector version)%
function [ft,t]=niltv(F,tm,pl);
global ft t;
alfa=0; M=256; P=3; Er=1e-10;   % adjustable
N=2*M; qd=2*P+1; t=linspace(0,tm,M);
NT=2*tm*N/(N-2); omega=2*pi/NT;
c=alfa-log(Er)/NT;
s=c-i*omega*(0:N+qd-1); Fsc=feval(F,s);
ft=fft(Fsc,N,2);
ft=ft(:,1:M); delv=size(Fsc,1);
d=zeros(delv,qd); e=d;
q=Fsc(:,N+2:N+qd)./Fsc(:,N+1:N+qd-1);
d(:,1)=Fsc(:,N+1); d(:,2)=-q(:,1);
for r=2:2:qd-1
  w=qd-r;
```

```
  e(:,1:w)=q(:,2:w+1)-q(:,1:w)+e(:,2:w+1);
  d(:,r+1)=-e(:,1);
  if r>2
    q(:,1:w-1)=q(:,2:w).*e(:,2:w)./e(:,1:w-1);
    d(:,r)=-q(:,1);
  end
end
A2=zeros(delv,M); B2=ones(delv,M);
A1=repmat(d(:,1),[1,M]); B1=B2;
z=repmat(exp(-i*omega*t),[delv,1]);
for n=2:qd
  Dn=repmat(d(:,n),[1,M]); A=A1+Dn.*z.*A2;
  B=B1+Dn.*z.*B2;
  A2=A1; B2=B1; A1=A; B1=B;
end
ft=2*real(ft+A./B)-repmat(Fsc(:,1),[1,M]);
ft=repmat(exp(c*t)/NT,[delv,1]).*ft;
ft(:,1)=2*ft(:,1); feval(pl);
```

```
%****** PLOT1 - FUNCTION DEFINITION ******%
function pl1  % multiple plotting into a
single figure
global ft t;
plot(t,ft); xlabel('t'); ylabel('f(t)');
grid on;
```

```
%****** PLOT2 - FUNCTION DEFINITION ******%
function pl2 %plotting into separate figures
global ft t;
for k=1:size(ft,1)
  figure; plot(t,ft(k,:)); xlabel('t');
  ylabel('f(t)'); grid on;
end
```

```
%*NILTM-FUNCTION DEFINITION(matrix version)%
function [ft,t,x]=niltm(F,tm,pl);
global ft t x;
alfa=0; M=256; P=3; Er=1e-10;   % adjustable
N=2*M; qd=2*P+1; t=linspace(0,tm,M);
NT=2*tm*N/(N-2); omega=2*pi/NT;
c=alfa-log(Er)/NT;
s=c-i*omega*(0:N+qd-1); Fsc=feval(F,s);
ft=fft(Fsc,N,2);
ft=ft(:,1:M,:); dim1=size(Fsc,1);
dim3=size(Fsc,3); d=zeros(dim1,qd,dim3);
e=d;
q=Fsc(:,N+2:N+qd,:)./Fsc(:,N+1:N+qd-1,:);
d(:,1,:)=Fsc(:,N+1,:); d(:,2,:)=-q(:,1,:);
for r=2:2:qd-1
  w=qd-r;
  e(:,1:w,:)=q(:,2:w+1,:)q(:,1:w,:)+e(:,2:w+1,:);
  d(:,r+1,:)=-e(:,1,:);
  if r>2
    q(:,1:w-1,:)
    =q(:,2:w,:).*e(:,2:w,:)./e(:,1:w-1,:);
    d(:,r,:)=-q(:,1,:);
  end
end
A2=zeros(dim1,M,dim3); B2=ones(dim1,M,dim3);
A1=repmat(d(:,1,:),[1,M]); B1=B2;
z=repmat(exp(-i*omega*t),[dim1,1,dim3]);
for n=2:qd
  Dn=repmat(d(:,n,:),[1,M]); A=A1+Dn.*z.*A2;
  B=B1+Dn.*z.*B2;
  A2=A1; B2=B1; A1=A; B1=B;
end
ft=ft+A./B;
ft=2*real(ft)-
repmat(real(Fsc(:,1,:)),[1,M]);
ft=repmat(exp(c*t)/NT,[dim1,1,dim3]).*ft;
ft(:,1,:)=2*ft(:,1,:);feval(pl);
```

```
%***** PLOT3 - FUNCTION DEFINITION ********%
function pl3
global ft t x;
m=length(t);tgr=[1:m/64:m,m];%65 time points
for k=1:size(ft,3)
  figure; mesh(t(tgr),x(k,:),ft(:,tgr,k));
  xlabel('t'); ylabel('x');
  zlabel(strcat('f_{‘,num2str(k),’}’));
end
```

5. Examples

As the first example, a linear circuit with 3 uniform (2+1)-conductor transmission lines is shown in Fig. 3 [1].

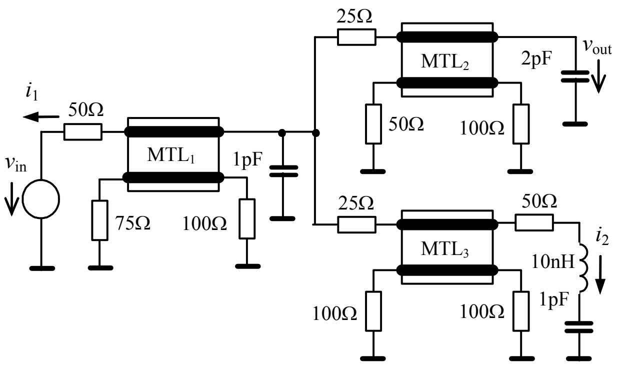


Fig. 3. Linear hybrid circuit containing three MTLs

The per-unit-length matrices are given in [1]. The MTL lengths are: l_1=0.05 m, l_2=0.04 m, l_3=0.03 m. An input 1V pulse with 1.5 ns rise/fall times and 7.5 ns width is applied.

To obtain the waveforms of nodal voltages or branch currents the Matlab function describing solution (15) is called by the `niltv` function. In Fig. 4 the input and output voltages and the current i_2 are given as examples.

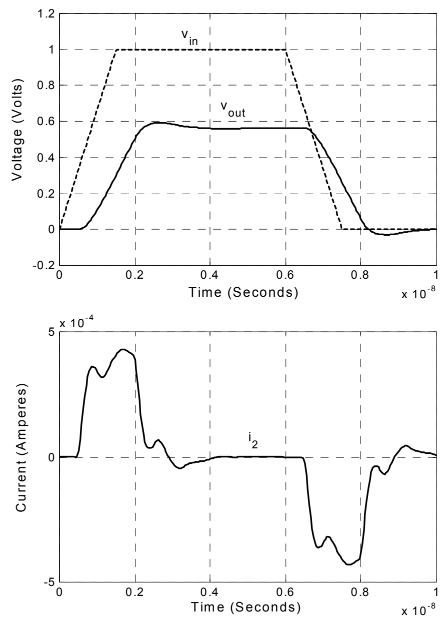
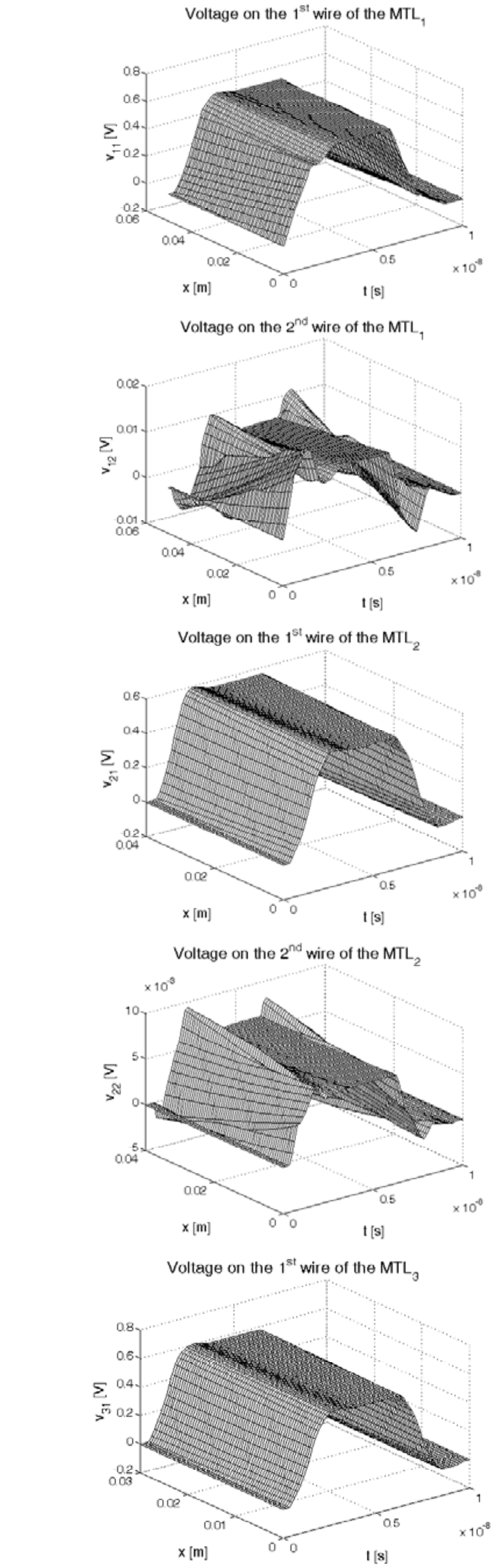


Fig. 4. Voltage and current waveforms

However, to obtain voltage or current distribution along the MTL wires a Matlab function describing solution (16) is called by the `niltm` function, for examples see Fig. 5.



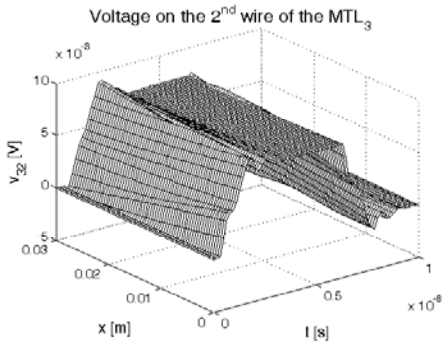


Fig. 5. Voltage distributions along MTL wires (Example 1)

As the second example, consider a linear circuit with 2 identical (2+1)-conductor transmission lines in Fig. 6 [3].

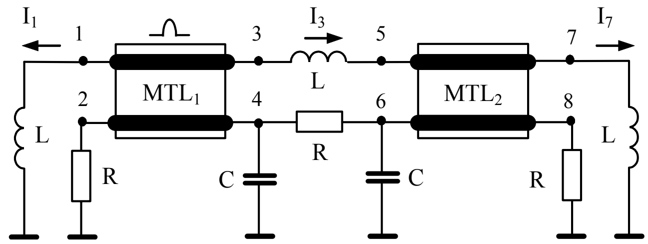


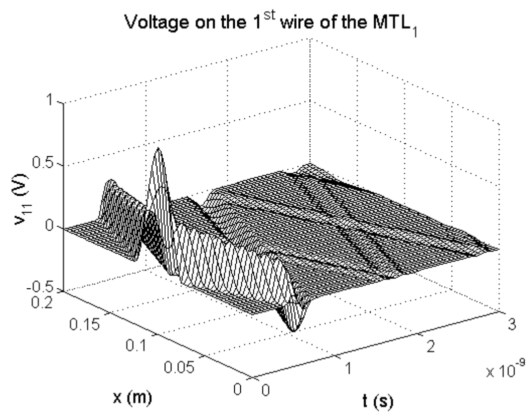
Fig. 6. Linear hybrid circuit with initially excited MTL_1

The MTLs are uniform, of the length l=0.2 m, and with per-unit-length matrices as given in [1]. The lumped-parameter elements have the values: R=10 Ω, C=10 pF, and L=1 nH. On the first MTL_1 wire the initial voltage distribution is non–zero, i.e.

$$v_1(x,0) = \sin^2\left(\pi\left[\frac{4x}{l} - \frac{3}{2}\right]\right) \quad \text{if} \quad \frac{3}{8}l \le x \le \frac{5}{8}l \ , \qquad (33)$$

otherwise $v_1(x,0) = 0$,

while the MTL_2 is considered under zero initial conditions. Here Eqn. (16) will again be used to solve waves on the MTL wires when the matrix version of the NILT method (32), i.e. the `niltm` function, is applied. Examples are shown in Fig. 7.



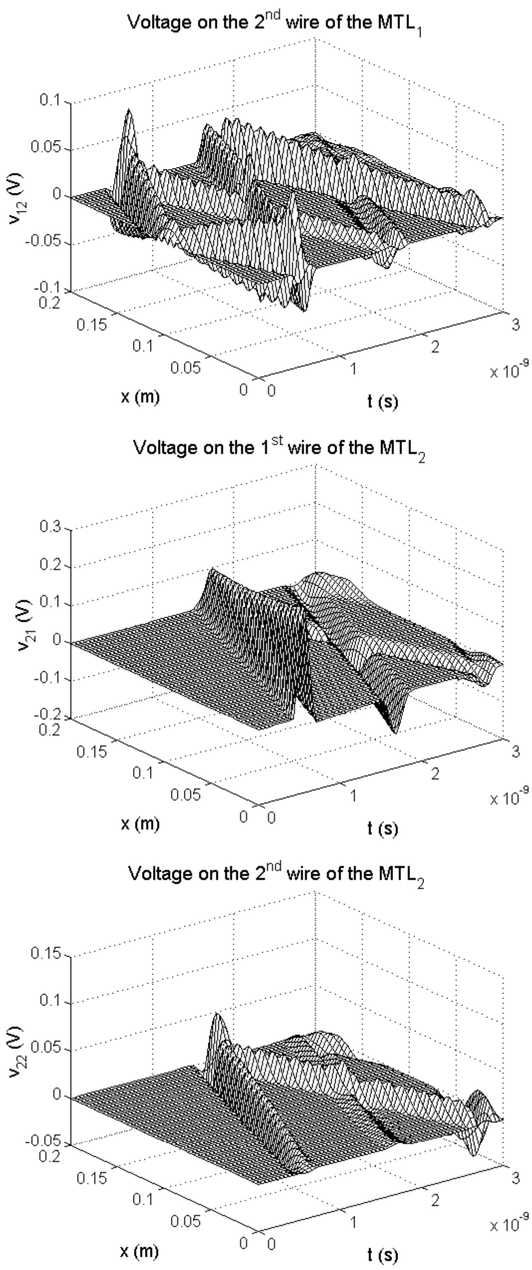


Fig. 7. Voltage distributions along MTL wires (Example 2)

6. Conclusion

The paper deals with a method for simulating transient phenomena in linear hybrid circuits oriented to the application of the universal scientific–technical language Matlab. This program has proved to be a very effective tool for developing and verifying novel methods and algorithms just numerically. All computations were done on a 2GHz/256MB PC. The technique under consideration leads to relatively fast computations, namely, the CPU time was about 0.5 and 6 seconds (Fig. 4 and Fig. 5, respectively) when zero initial conditions were considered, and about 10 seconds for nonzero ones (Fig. 7). The results can also be utilized e. g. to animate voltage/current waves propagating along MTL wires.

Acknowledgements

This work was financially supported by GAČR grant No. 102/03/0241 and by the research programmes MSM 262200011 and MSM 262200022.

References

[1] LUM, S., NAKHLA, M. S., ZHANG, Q. J. Sensitivity Analysis of Lossy Coupled Transmission Lines. *IEEE Trans. on Microwave Theory and Techniques*, 1991, vol. 39, no. 12, p. 2089-2099.

[2] CHIPROUT, E., NAKHLA, M. S. *Asymptotic Waveform Evaluation and Moment Matching for Interconnect Analysis*. Carleton University, Kluwer Academic Publisher, Boston, 1994.

[3] BRANČÍK, L. Time-Domain Simulation of Multiconductor Trans–mission Line Systems under Nonzero Initial Conditions. In *Proc.of the 15th ECCTD'01*. Espoo, Finland, 2001, vol. 1, p. 73-76.

[4] BRANČÍK, L. Time-Domain Simulation of Nonuniform Multicon–ductor Transmission Lines under Nonzero Initial Conditions Using Matlab Language. In *Proc. of the 14th ECCTD'99.* Stresa, Italy, 1999, vol. 2, p. 1135-1138.

[5] PAUL, C. R. *Analysis of Multiconductor Transmission Lines*. John Wiley & Sons, New York, 1994.

[6] GANTMACHER, R *The Theory of Matrices.* Chelsea, N.York, 1977.

[7] BRANČÍK, L. Utilization of Quotient-Difference Algorithm in FFT-based Numerical ILT Method. In *Proc. 11th Radioelektronika'2001.* Brno, Czech Republic, 2001, p. 352-355.

[8] HENRICI, P. *Quotient-Difference Algorithms, Math. Methods for Digital Computers*. Eds. A. Ralston and H. S. Wilf, John Wiley & Sons, New York, 1967, vol. 2, p. 37-62.

[9] BRANČÍK, L. Improved Numerical Inversion of Laplace Transforms Applied to Simulation of Distributed Circuits. In *Proc. XI. ISTET01.* Linz, Austria, 2001, p. 51-54.

[10] BRANČÍK,L. Matlab Oriented Matrix Laplace Transforms Inversion for Distributed Systems Simulation. In CD ROM *Proc. of 12th Radioelektronika'2002.* Bratislava, Slovakia, 2002, paper no. 114.

About Author

Lubomír BRANČÍK was born in Kyjov in 1961. He received his Ing. (MSc.) degree in microelectronics in 1985, and CSc. (PhD.) degree in measuring technique in 1993, both at the Brno University of Technology. Since 2000 he has been working as an associate professor of theoretical electrical engineering at the same university. His main research interests include the application of numerical methods in circuit theory and computer-aided simulation.