IMPLEMENTATION OF FINITE ELEMENT METHODS IN TIME DOMAIN

Milan Motl, Zbyněk Raida
Institute of Radio Electronics
Faculty of Electrical Engineering and Communication
Brno University of Technology

Abstract In this paper, the implementation of finite element method for analysis of electromagnetic structures in time domain is presented. The finite element complex frequency hopping (CFH/FEM) and time-domain finite element (TD-FEM) methods are described. Finally, both methods are used to analyze a homogenous microstrip transmission line and the results are compared.

Formulation

For a homogeneous, isotropic, and a linear medium, starting from Maxwell's equations the following scalar equation can be derived as

$$\nabla^2\Phi - \mu\varepsilon\frac{\partial^2\Phi}{\partial t^2} - \mu\sigma\frac{\partial\Phi}{\partial t} = f(x,y,z,t) \qquad (1)$$

where $\Phi(x, y, z, t)$ represents either the electric or magnetic field components, and ε, μ, σ are the permittivity, permeability, conductivity of the medium respectively. $f(x, y, z, t)$ is related to the external excitation. This expression can be applied to either a diffusion problem where the second term in the left-hand-side of (1) is dropped or to a wave equation where the third term is dropped.

The equations can be transformed using finite element method to [1], [2]

$$\mathbf{A}\Psi + \mathbf{B}\frac{\partial\Psi}{\partial t} + \mathbf{C}\frac{\partial^2\Psi}{\partial t^2} = \mathbf{q} \qquad (2)$$

where $\mathbf{A} = \mathbf{S}$, $\mathbf{B} = \sigma\mu\mathbf{T}$, $\mathbf{C} = \varepsilon\mu\mathbf{T}$, $\Psi = \{\Phi_i\}$, $\mathbf{q} = \mathbf{G}$ is the excitation matrix and **S, T, G** are the well-known matrices of the original finite element method implementation [1].

Finite Element Method with Complex Frequency Hopping (CFH/FEM)

Introduction

CFH uses the concept of moment matching to obtain both frequency- and time-domain responses of large linear networks through a lower order multipoint Padé approximation. It extracts a relatively small set of dominant poles to represent a large network that may contain hundreds to thousands of actual poles.

The main steps involved can be summarized as follows: first, the given problem which is in the form of damped wave equation is formulated using FEM and the resulting ordinary differential equation is transformed to the Laplace domain; second, the Laplace domain output is expanded using Padé approximation at selected frequency points; third, information from each expansion point is used to generate the output frequency response or alternatively a unified set of dominant pole/residue pairs; finally, the results are transformed to the time-domain in either analytical or numerical form.

Taking the Laplace transform of (2) results in

$$(\mathbf{C}s^2 + \mathbf{B}s + \mathbf{A})\mathbf{X}(s) = \mathbf{R}(s) \qquad (3)$$

or

$$\mathbf{Y}(s)\mathbf{X}(s) = \mathbf{R}(s) \qquad (4)$$

where $\mathbf{X}(s) = \mathrm{L}\{\Psi(t)\}$ and $\mathbf{R}(s)$ is given by

$$\mathbf{R}(s) = s\mathbf{C}(\Psi_0 + \Psi_0^{'}) + \mathbf{B}\Psi_0 + \mathbf{Q}(s) \qquad (5)$$

where

$$\mathbf{Q}(s) = \mathrm{L}\{\mathbf{q}(t)\}. \qquad (6)$$

Moment Matching Techniques [1]

In general, moment matching technique approximates the frequency response of a Taylor series expansion in the complex *s*-plane. The moments (coefficients of the expansion) are matched to a lower order transfer function using a rational Padé approximation. This transfer function can be used to obtain the output response. Single Padé approximations are accurate near the point of expansion in the complex *s*-plane and decrease in accuracy with increased distance from the point of expansion. Complex frequency hopping overcomes this problem and is summarized in the following section.

A. Complex Frequency Hopping

Complex frequency hopping [1] is a method by which the frequency response of a system is expanded in multiple Taylor series expansions in the Laplace *s*-domain. The expansion points are chosen on or near the imaginary axis because poles that dominate the transient and frequency response of a system are found there. The moments of the expansion are then matched to a rational Padé approximation. These Padé approximations have several useful properties, one of which is the convergence of the poles of the approximation to the actual poles of a system.

Approach for generating the response of a system is given as follows: a set of rational transfer functions are generated at a minimized set of expansion points. It is the value of these transfer functions that is compared at points intermediate to the expansion points rather than a search for same poles. If two transfer functions are found to give the same frequency response at an intermediate frequency point between the two generating hops then these transfer functions are considered accurate. If this is not the case, then another hop is chosen between the two expansion points and an expansion is performed there.

B. Padé Approximation [1]

Consider the response vector X(s) represented in (4). Taylor's series expansion of the output X(s) about a complex frequency point $s = s_0$, is given by

$$\mathbf{X}(s) = \sum_{n=0}^{\infty} \mathbf{M}_n (s - s_0)^n \qquad (7)$$

where M$_n$ is the nth moment vector of the Taylor's expansion about s_0 and is given by

$$\mathbf{M_n} = \frac{\partial^n [\mathbf{Y}^{-1}(s)\mathbf{R}(s)]_{s=s_0}}{\partial s^n} / n! \qquad (8)$$

A recursive equation for the evaluation of the moments can be obtained in the form

$$[\mathbf{Y}(s_0)]\mathbf{M}_n = -\sum_{r=1}^{n} [\mathbf{Y}]^r \mathbf{M}_{n-r} / r! \qquad (9)$$

where

$$[\mathbf{Y}]^r = \frac{\partial^r}{\partial s^r} \mathbf{Y}\Big|_{s=s_0} \qquad (10a)$$

$$\mathbf{Y}(s_0)\mathbf{M}_0 = \mathbf{R} \tag{10b}$$

For each expansion point, the moments $m_n = [\mathbf{M}_n]_{(i)}$; n=0,1,2, . . . ($2q$ – 1) of an output i are matched to a lower order frequency-domain function in the form

$$H(s) = X_{[i]}(s) = \frac{P_L(s)}{Q_M(s)} = \frac{\sum_{j=0}^{L} a_j s^j}{1 + \sum_{j=1}^{M} b_j s^j} \tag{11}$$

for $0 < s \leq j\omega_n$.

For given L and M, the coefficients of the numerator and denominator of the transfer function are related to the moments by

$$\begin{bmatrix} m_{L-M+1} & m_{L-M+2} & \cdots & m_L \\ m_{L-M+2} & m_{L-M+3} & \cdots & m_{L+1} \\ \cdot & \cdot & \cdot & \cdot \\ m_L & m_{L+1} & \cdots & m_{L+K-1} \end{bmatrix} \begin{bmatrix} b_M \\ b_{M-1} \\ \cdot \\ b_1 \end{bmatrix} = - \begin{bmatrix} m_{L+1} \\ m_{L+2} \\ \cdot \\ m_{L+M} \end{bmatrix}$$

$$a_r = \sum_{j=0}^{r} m_{r-j} b_j \tag{12}$$

where r= 0,l,..., L and m_j=0 if $j < 0$.

C. *Binary Search Algorithm [1]*

Padé approximation is very accurate near the point of expansion i.e., $s = s_0$. However, the accuracy of Padé approximation decreases as we move further away from the point of expansion similar to the case of a Taylor's series expansion.

In order to check the accuracy of an approximation, two expansion points are necessary. The two expansion points can be verified for their accuracy by finding the value of the transfer function at a point intermediate to these two expansion points [22]. Number of expansions required to obtain a fairly accurate set of transfer functions over a specified frequency range is controlled by a binary search algorithm.

The steps involved in the binary search algorithm are summarized below.

Step 1) Set $f_L = 0$ and $f_H = f_{max}$.

Step 2) Expand the system's response at frequency $f = f_L$. Determine the coefficients of the corresponding transfer function $H_L(s)$ using (10).

Step 3) Expand the system's response at frequency $f = f_H$. Determine the coefficients of the corresponding transfer function $H_H(s)$.

Step 4) Set $f = (f_L + f_H)/2$. Calculate $H_L(j2\pi f)$ and $H_H(j2\pi f)$. If, $|H_H(j2\pi f) - H_L(j2\pi f)| < \varepsilon$, where ε represents pre-specified threshold for relative error, GO TO Step 5. Otherwise expand at $f_{mid} = (f_L + f_H)$ and determine $H_{mid}(s)$.

Step 5) If no middle frequency $\mathrm{f_{mid}}$ is generated between any two other frequency expansions, STOP, ELSE, repeat Steps 24 between every two consecutive frequency points (e.g., between f_L & f_{mid} and f_{mid} & f_H).

At the completion of the binary search algorithm, a set of transfer functions is generated. The frequency response at a particular frequency point is computed by choosing a transfer function valid for that region. This is repeated for all other frequency points and the system's frequency response is computed. The time-domain response is obtained as a closed form function of the generated poles and residues or alternatively by using inverse fast fourier transform (IFFT).

Implementation in MATLAB:

```
f_lw = f_min;

f_hg = f_max;
id  = 1;                             % for storing approx.coefs

f(id) = f_lw;
[a(id,1:iL),b(id,1:iM)] = app3b( A, C, exc, f(id), out, L, M);
id = id + 1;

f(id) = f_hg;
[a(id,1:iL),b(id,1:iM)] = app3b( A, C, exc, f(id), out, L, M);
id = id + 1;

i_lw = id - 2;
i_hg = id - 1;

while f_hg>0
  f_md = 0.5*(f_lw + f_hg);
  h_hg = h( a(i_hg,1:iL), b(i_hg,1:iM), f(i_hg), f_md);
  h_lw = h( a(i_lw,1:iL), b(i_lw,1:iM), f(i_lw), f_md);
  acter= abs(h_hg - h_lw)/(abs(h_hg) + abs(h_lw));    % actual error

  while acter>err                              % high error
    f_hg  = f_md;
    f(id) = f_hg;
    i_hg  = id;
    [a(id,1:iL),b(id,1:iM)] = app3b( A, C, exc, f(id), out, L, M);
    id    = id + 1;
    f_md  = 0.5*(f_lw + f_hg);
    h_hg  = h( a(i_hg,1:iL), b(i_hg,1:iM), f(i_hg), f_md);
    h_lw  = h( a(i_lw,1:iL), b(i_lw,1:iM), f(i_lw), f_md);
    acter = abs(h_hg - h_lw)/(abs(h_hg) + abs(h_lw));
  end

  sup = [];
  sup(1) = 0;
  xx = 1;
  i_p(1) = 1;
  for n=1:(id-1)
    if f(n) > f_hg
      sup(xx) = f(n);
      i_p(xx) = n;
      xx = xx + 1;
    end
  end


  f_lw = f_hg;
  i_lw = i_hg;
  [f_hg, ptr] = min( sup);
  i_hg = i_p(ptr);

end

up = 10*f_max;                 %sorting coefficients according to freq.
id = id-1;
for n=1:id
  [fc(n), ptr] = min(f);
  f(ptr) = up;
  ac(n,1:iL) = a(ptr,1:iL);
  bc(n,1:iM) = b(ptr,1:iM);
end
```

D. Moments Generation [1]

To derive an expression for moments $\mathbf{M}_m$, we can rewrite (4) using (6) and (7) at any arbitrary complex frequency point s = s_0, and expanding the right hand side of (4) using Taylor's series, we get

$$\left[(s-s_0)^2\mathbf{C}+(s-s_0)\mathbf{B}+\mathbf{A}+(s-s_0)2s_0\mathbf{C}+\mathbf{B}s_0+\mathbf{C}s_0{}^2\right]\sum_{n=0}^{\infty}\mathbf{M}_n(s-s_0)^n =$$

$$\left[\mathbf{R}(s_0)+\frac{(s-s_0)}{1!}\mathbf{R}'(s_0)+\frac{(s-s_0)^2}{2!}\mathbf{R}''(s_0)+........+\frac{(s-s_0)^n}{n!}\mathbf{R}^{(n)}(s_0)\right] \qquad (13)$$

where $\mathbf{R}(s_0)$ = $\mathbf{B}\Psi_0$ + $\mathbf{Q}(s_0)$, $\mathbf{R}^{`}(s_0)$ = $\mathbf{C}(\ \Psi_0 + \Psi_0^{`})$ + $\mathbf{Q}^{`}(s_0)$, $\mathbf{R}^{``}(s_0)$ = $\mathbf{Q}^{``}(s_0)$ . . . and $\mathbf{R}^{(n)}(s_0) = \mathbf{Q}^{(n)}(s_0)$.

Equating the coefficients of the powers of (s – s_0) on both sides, we get

$$\left[\mathbf{C}s_0{}^2+\mathbf{B}s_0+\mathbf{A}\right]\mathbf{M}_0=\mathbf{R}(s_0) \qquad (14)$$

$$\left[\mathbf{C}s_0{}^2+\mathbf{B}s_0+\mathbf{A}\right]\mathbf{M}_1=-\left[\mathbf{B}+2s_0\mathbf{C}\right]\mathbf{M}_0+\mathbf{R}'(s_0) \qquad (15)$$

Generalizing we have

$$\left[\mathbf{C}s_0{}^2+\mathbf{B}s_0+\mathbf{A}\right]\mathbf{M}_n=-\mathbf{B}\mathbf{M}_{n-1}-\mathbf{C}\left[2s_0\mathbf{M}_{n-1}+\mathbf{M}_{n-2}\right]+\frac{\mathbf{R}^{(n)}(s_0)}{n!} \qquad (16)$$

for $n \geq 2$.

In MATLAB notation:

```
function [a,b] = app3b( A, C, exc, f0, out, L, Mm);

nd = L + Mm + 1;
[s,t] = size(A);

s0 = j*2*pi*f0;
Y  = A + s0*s0*C;                    % equation (3)
Y1 = 2*s0*C;                         % first derivation of Y
Y2 = 2*C;                            % second derivation of Y
iY = inv(Y);                         % inverse Y


M(1:s,1) = + iY(1:s,exc);            % equation (14)
M(1:s,2) = - iY*(Y1*M(1:s,1));       % equation (15)
M(1:s,3) = - iY*(Y1*M(1:s,2) + 0.5*Y2*M(1:s,1));       % equation (16)
if nd > 3
    M(1:s,4) = - iY*(Y1*M(1:s,3) + 0.5*Y2*M(1:s,2));   % equation (16)
end
if nd > 4
    M(1:s,5) = - iY*(Y1*M(1:s,4) + 0.5*Y2*M(1:s,3));   % equation (16)
end
if nd > 5
    M(1:s,6) = - iY*(Y1*M(1:s,5) + 0.5*Y2*M(1:s,4));   % equation (16)
end
if nd > 6
    M(1:s,7) = - iY*(Y1*M(1:s,6) + 0.5*Y2*M(1:s,5));   % equation (16)
```

```
end
if nd > 7
    M(1:s,8) = - iY*(Y1*M(1:s,7) + 0.5*Y2*M(1:s,6));  % equation (16)
end
if nd > 8
    M(1:s,9) = - iY*(Y1*M(1:s,8) + 0.5*Y2*M(1:s,7));  % equation (16)
end
if nd > 9
    M(1:s,10) = - iY*(Y1*M(1:s,9) + 0.5*Y2*M(1:s,8)); % equation (16)
end

m(1:nd) = M(out,1:nd);

ind = 0;                                    % generating of system (12)
for p=(L-Mm+1):L
  ind = ind + 1;
  for q=1:Mm
    left(q,ind)=m(p+q);
  end
end
right(1:Mm) = - m((L+2):(L+Mm+1));   % generating of right side of (12)
right = conj(right');

b(1:Mm) = left\right;                       % solving of system (12)
b(Mm+1) = 1;

for p=0:L
  in    = L+1-p;
  a(in) = 0;
  for q=0:p
    a(in) = a(in) + m(q+1)*b(in+q+1);       % computation of a(in)
  end
end
```

Time-Domain Finite-Element Method (TD-FEM)

Introduction

This method is based on general time-domain formulation. The time response of the structure is produced, so the fast Fourier transform must be used to convert it to the spectral domain consequently. The method is based on "step-by-step" solution. In the first step, the structure is excited. Then, the field components are computed from the values in previous two steps.

General Implicit Two-Step Recurrence Algorithms [2]

To formulate implicit TDFEM's that involve the solution of a matrix equation, we start with the semi-discrete equation

$$\mathbf{T}\frac{\partial^2 u}{\partial t^2} + \mathbf{B}\frac{\partial u}{\partial t} + \mathbf{S}u + f = 0\,, \tag{17}$$

where

$$\begin{aligned} \mathbf{T}_{ij} &= \frac{1}{c^2}\int_{\Omega} W_i \bullet \varepsilon_r W_j \, d\Omega \\ \mathbf{B}_{ij} &= \frac{1}{c}\int_{\Gamma} W_i \bullet \left(W_j \times n\right) d\Omega \\ \mathbf{S}_{ij} &= \int_{\Omega} \nabla \times W_i \bullet \frac{1}{\mu_r} \nabla \times W_j \, d\Omega \end{aligned} \tag{18}$$

$$f_i = \mu_0 \int_\Omega W_i \bullet \frac{\partial J}{\partial t} d\Omega$$

Also, $u^{(-1)}$=u(-δt) and $u^{(0)}$=u(0) are assumed to be given either from the initial excitation or from previous iteration. Even though the implicit schemes involve solving a matrix equation for every time step, it is possible to derive implicit TDFEM's that are unconditionally stable. The unconditional stability is very desirable when solving problems with very small features that lead to large variations in element size across the problem domain. Without loss of generality, we shall concentrate on deriving general two-step algorithms to obtain using the weighted residual principle. The approach begins by approximating the function $u(t)$, t⊂(-δt,δt) by a second-order polynomial expansion

$$u(t) = \frac{1}{(\delta t)^2}\left[u^{(-1)} \cdot \left(\frac{t^2}{2} - \frac{t\delta t}{2} \right) - u^{(0)} \cdot (t^2 - \delta t^2) + u^{(1)} \cdot \left(\frac{t^2}{2} + \frac{t\delta t}{2} \right) \right]. \tag{19}$$

From (19), the first- and second-time derivatives are approximated as

$$\frac{du}{dt} = \frac{1}{\delta t^2}\left[u^{(-1)} \cdot \left(t - \frac{\delta t}{2} \right) - 2u^{(0)} \cdot t + u^{(1)} \cdot \left(t + \frac{\delta t}{2} \right) \right]$$

$$\frac{d^2 u}{dt^2} = \frac{1}{\delta t^2}\left[u^{(-1)} - 2u^{(0)} + u^{(1)} \right]. \tag{20}$$

Substituting (19) and (20) into the semi-discrete equation (17), weighting the residual by a test function *W(t)*, and defining Θ$_1$, Θ$_2$ according to [2] yields

$$u^{(1)}\left[\mathbf{T} + \delta t\left(2\Theta_1 + \frac{1}{2} \right)\mathbf{B} + \delta t^2 (\Theta_1 + \Theta_2)\mathbf{S} \right] + u^{(0)}\left[-2\mathbf{T} - 4\delta t\Theta_1 \mathbf{B} - \delta t^2 (4\Theta_2 - 1)\mathbf{S} \right]$$

$$u^{(-1)}\left[\mathbf{T} + \delta t\left(2\Theta_1 - \frac{1}{2} \right)\mathbf{B} + \delta t^2 (2\Theta_2 - \Theta_1)\mathbf{S} \right] + \delta t^2 \overline{f} = 0 \tag{21}$$

where

$$\overline{f} = \frac{\int_{-\delta t}^{\delta t} W(t) f dt}{\int_{-\delta t}^{\delta t} W(t) dt} \tag{22}$$

Once $u^{(1)}$ has been obtained from the above equation, a new time step can be started. Implicit algorithms of various kinds can be obtained from (21) by employing different Θ$_1$ and Θ$_2$ values. A particular pair Θ$_1$ = 0 and Θ$_2$ = 1/8 is recommended in [2]. This results in an unconditionally stable implicit algorithm with truncation error O(δt^2).
Substituting these values into (21) we obtain

$$u^{(1)}\left[\mathbf{T} + \frac{\delta t}{2}\mathbf{B} + \frac{\delta t^2}{4}\mathbf{S} \right] + u^{(0)}\left[-2\mathbf{T} + \frac{\delta t^2}{2}\mathbf{S} \right] + u^{(-1)}\left[\mathbf{T} - \frac{\delta t}{2}\mathbf{B} + \frac{\delta t^2}{4}\mathbf{S} \right] + \delta t^2 \overline{f} = 0. \tag{23}$$

Further, **B** = **0** for shielded and lossless structures:

$$u^{(1)}\left[\mathbf{T} + \frac{\delta t^2}{4}\mathbf{S} \right] + u^{(0)}\left[-2\mathbf{T} + \frac{\delta t^2}{2}\mathbf{S} \right] + u^{(-1)}\left[\mathbf{T} + \frac{\delta t^2}{4}\mathbf{S} \right] + \delta t^2 \overline{f} = 0\ . \tag{24}$$

Expressing the $u^{(1)}$ yields

$$u^{(1)}\left[\mathbf{T}+\frac{\delta t^2}{4}\mathbf{S}\right]=u^{(0)}\left[2\mathbf{T}-\frac{\delta t^2}{2}\mathbf{S}\right]-u^{(-1)}\left[\mathbf{T}+\frac{\delta t^2}{4}\mathbf{S}\right]-\delta t^2\,\overline{f}\,. \qquad (25)$$

The u are row vectors. Now it is necessary to modify (25) to the form of equation system. The u^{T} will then be column vectors. Taking advantage of the **T** a **S** symmetry (which is provided by the FEM) we obtain

$$\left[\mathbf{T}+\frac{\delta t^2}{4}\mathbf{S}\right](u^{(1)})^T=\left[2\mathbf{T}-\frac{\delta t^2}{2}\mathbf{S}\right](u^{(0)})^T-\left[\mathbf{T}+\frac{\delta t^2}{4}\right](u^{(-1)})^T-\delta t^2\,\overline{f}\,. \qquad (26)$$

MATLAB Implementation

The notation is following: $u^{(1)}$=uc, $u^{(0)}$=ub, $u^{(-1)}$=ua. The variable names are different for three successive steps, in order to eliminate the excessive copying.

```
dt=dx/(c*sqrt(2));                     % for linear approximation
%dt=dx/(c*sqrt(32));                   % for quadratic approximation

Y=T+((dt^2)/4)*S;                      % matrix on left side (26)
Z=2*T-((dt^2)/2)*S;                    % first matrix on right side (26)

F=sparse(k(1),1);
F(exc)=1;                              % vector of excitation

clear S;
clear T;

vys=zeros(krok+2,1);

ua=zeros(k(1),1);
ub=zeros(k(1),1);

B = Z*ub - Y*ua;                       % first step- right side of (26)
uc = Y\B;                              % solving eq. (26)
vys(1)=uc(out);                        % saving the results
ua=ub;                                 % time shift
ub=uc;

B = Z*ub - Y*ua ;                      % second step-right side of (26)
uc = Y\B;                              % solving eq. (26)
vys(2)=uc(out);                        % saving the results
ua=ub;                                 % time shift
ub=uc;

B = Z*ub - Y*ua - (dt^2)*F;        % excitation step – right side of (26)
uc = Y\B;                              % solving eq. (26)
vys(3)=uc(out);                        % saving the results
ua=ub;                                 % time shift
ub=uc;

clear F;
r=round((krok-3)/3);

for i=1:r,                             % over all other time steps
   l=3*(i-1);
   B = Z*ub - Y*ua;                    % next step-right side of (26)
   uc = Y\B;                           % solving eq. (26)
   vys(l+4)=uc(out);                   % saving the results
   B = Z*uc - Y*ub;                    % next step-right side of (26)
   ua = Y\B;                           % solving eq. (26)
```

```
    vys(l+5)=ua(out);                   % saving the results
    B = Z*ua - Y*uc;                    % next step-right side of (26)
    ub = Y\B;                           % solving eq. (26)
    vys(l+6)=ub(out);                   % saving the results
end

figure(2);
plot(vys);                              % time response

figure(3);
spe=abs(fft(vys));                      % computing of spectrum

pt=round((length(spe))/2);
krok=length(vys);
df=1/(krok*dt);
f=0:df:((pt-1)*df);                     % frequency range
plot(f,spe(1:pt));                      % frequency response
```

The Comparison

The accuracy of both methods is compared on 2D cross section of a shielded dielectric supported air strip transmission line. The section is treated as a planar resonator where the acquisition of resonant frequencies (represented by wave modes in the real waveguide) is observed. The simplicity is the main reason for choosing the mentioned type of a waveguide. If the microstrip waveguide has no dielectric substrate, the analysis splits into two independent problems, TM and TE. The exact values are computed using common eigenproblem solvers, with $Nx = Ny = N = 400$ mesh density. The experiment showed that the TD-FE method is more accurate than the CFH/FE method. To make the comparison possible, the $Nx = Ny = N = 20$ mesh was employed for TD-FE method while the CFH/FE used the spatial resolution $Nx = Ny = N = 40$. The TD-FEM is significantly more time-consuming, however. The eigenfrequency error for TM mode is displayed in Fig. 1., Fig 2.

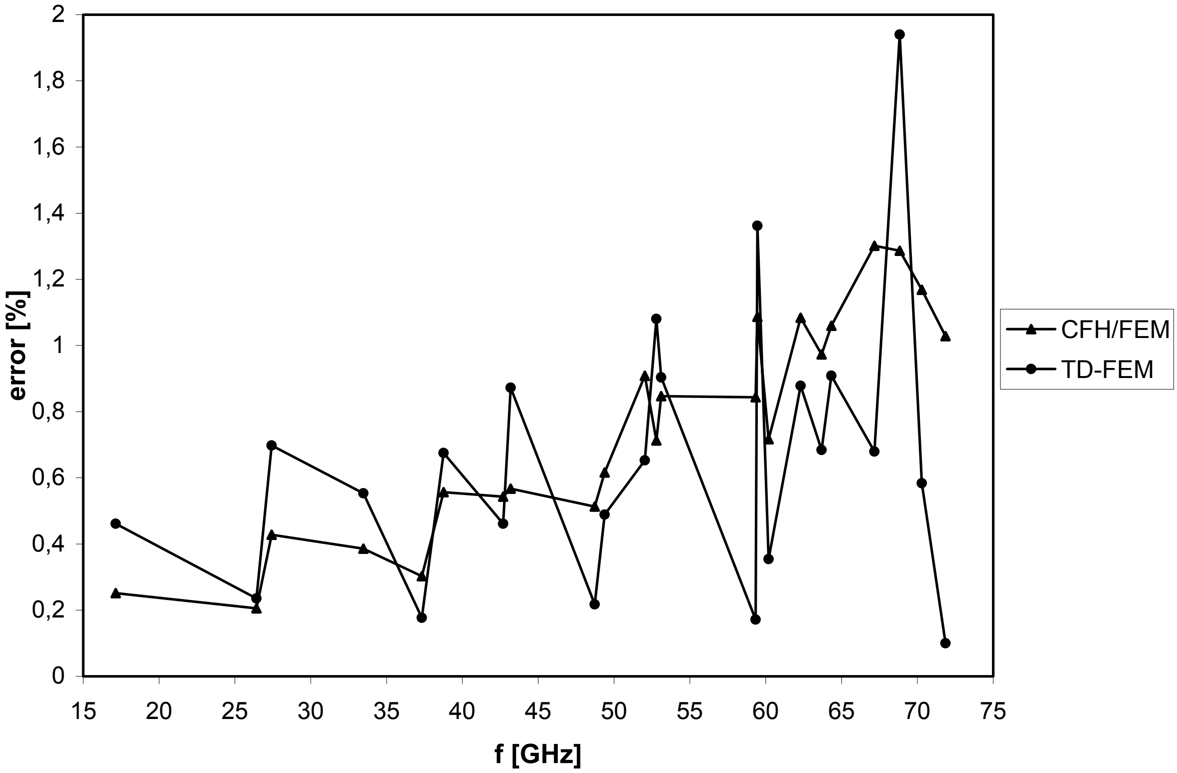

Fig. 1.: *Eigenfrequency error for TM mode (N = 40 for CFH/FEM, N = 20 for TD-FEM)*

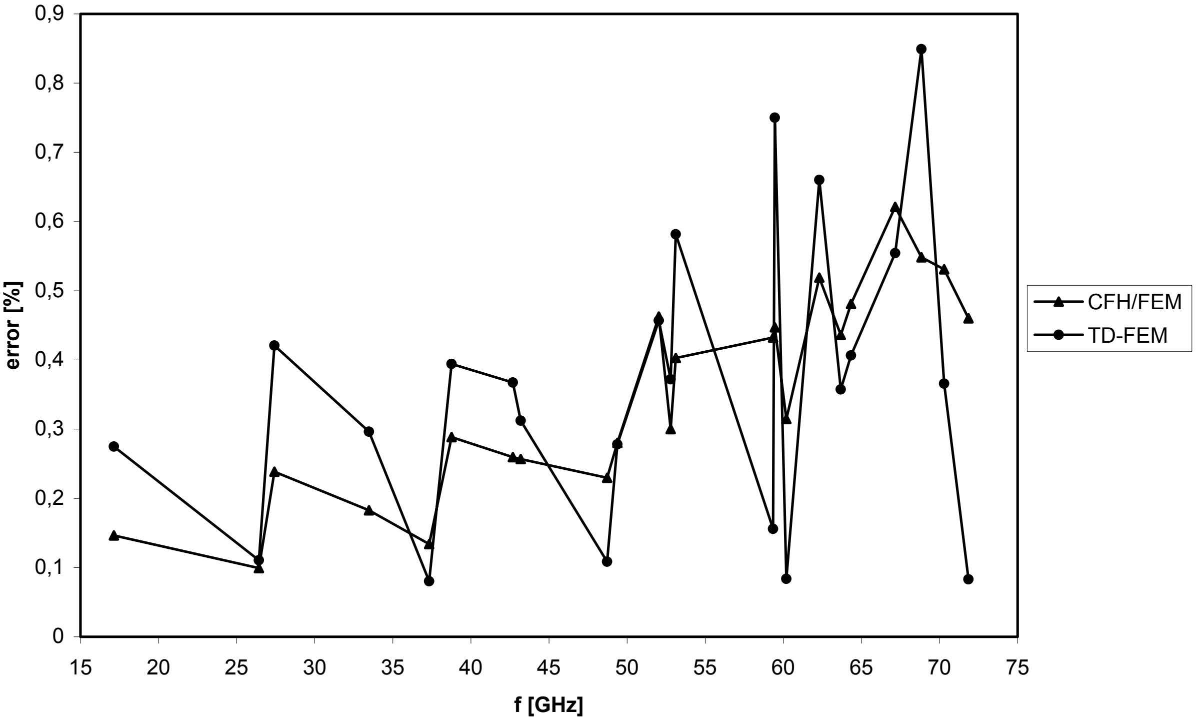

Fig. 2.: *Eigenfrequency error for TM mode (N = 60 for CFH/FEM, N = 30 for TD-FEM)*

Conclusion. The CFH/FE method produces the response in the specified frequency range. Therefore, convenient frequency resolution can be chosen. The computer demands increase with widening the frequency range. Hence, the method is suitable for investigating the narrow-band or highly selective structures.
On the other hand, the TD-FEM method yields the response in wide frequency range without possibility to reduce it and save the execution time. Better frequency resolution is achieved by increasing the number of time steps, but this is often unacceptable with respect to computer resources. For this reason, the TD-FEM method is applied to wide-band structures with advantage.

Acknowledgement. Development, which is described in the presented paper, was financially supported by the grant projects no. 102/01/0571 and 102/01/0573 of the Czech Grant Agency, and by the research program no. MSM 262200011 of the Czech Ministry of Education.

References.

[1] M. A. KOLBEHDARI, M. SRINIVASAN, M. S. NAKHLA, Q. ZHANG and R. ACHAR, “Simultaneous Time and Frequency Domain Solutions of EM Problems Using Finite Element and CFH Techniques,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 44, no. 9, pp. 1526–1533, September 1996.

[2] LEE, J., LEE, R., CANGELLARIS, A. “Time-Domain Finite-Element Methods,” *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 3, pp. 430–441, March 1997.

Contact
motl@feec.vutbr.cz
raida@feec.vutbr.cz

Department of Radio Electronics
Brno University of Technology
Purkyňova 118
612 00 Brno