Combining the FDTD Algorithm with Signal Processing Techniques for Performance Enhancement

Kapil Sharma, K. Panayappan, Ravi K. Arya and R. Mittra

EMC Lab, The Pennsylvania State University
rajmittra@ieee.org
Conventional FDTD algorithm

- Time domain numerical convergence (e.g., below 30 dB level) is used as the termination criterion for conventional FDTD algorithm.

- The termination criterion used in conventional FDTD results in high computational cost when a given problem is solved for low frequencies, or when high-Q structures and dispersive dielectric medium are involved.

<table>
<thead>
<tr>
<th>Frequency</th>
<th>10 MHz</th>
<th>1 MHz</th>
<th>1 Hz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Steps</td>
<td>6.9582e+005</td>
<td>6.9582e+006</td>
<td>6.9582e+012</td>
</tr>
</tbody>
</table>
The proposed efficient FDTD algorithm utilizes signal processing techniques to check convergence in the frequency domain (as opposed to time domain), and thus differs from the conventional FDTD algorithm.

First, we wait until the first overshoot of the time signature has occurred.

Then we compute the DFT of these samples at the lowest frequency of interest at the end of every 100 time steps.

The FDTD algorithm is terminated when the difference between the successive DFT values becomes negligible.
Proposed Solution – Low Frequency Processing

- Region 3: High frequency regime - Use DC Gaussian pulse results.
- Region 1: Low frequency regime - Use the proposed method.
- Region 2: Validation region – Use the Single Frequency results in this region to validate the smoothed DC Gaussian results.

Frequency Definitions:
- \( f_L \) - User Input
- \( f_1 \) - 500 – 1000 MHz
- \( f_2 \) - \( 2 \frac{f_1}{3} f_1 \)
- \( f_H \) - User Input
Smooth the DC Gaussian Results.

Fit the curve from $f_L$ to $f_1$ with the DC values, using a quadratic curve. The choice of $f_1$ can be fine tuned based on the quality of the resulting fit.

Validate the smoothed “DC Gaussian” results in region 2 by comparing them with those generated by “Single Frequency” simulations at a few points (typically 2 or 3).

Frequency Definitions:
- $f_L$ - User Input
- $f_1$ - 500 – 1000 MHz
- $f_2 = 2 \frac{f_1}{3}$
- $f_H$ - User Input
RF Filter

$f @ 0 \text{ Hz to } 1.5 \text{ GHz}$

$\varepsilon_r = 2$

Variation of $S_{11}$

Variation of $S_{12}$

Graphs showing the magnitude in dB vs. frequency in MHz for different models: DC Gaussian, Single Frequency, Universal GEMS, and FEKO.
Connector

$\text{f @ 10 MHz to 1.5 GHz}$
Connector (Contd.)

Variation of $S_{11}$

Variation of $S_{12}$

Variation of $S_{13}$

Variation of $S_{14}$
Vias in a Printed Circuit Board Geometry

Trace: 17.7 micron thick, 3.557 mm wide

Dielectric Substrates:
FR4: $\varepsilon_r = 4.47$, $\tan\delta = 0.016$, height: 1.905 mm

Taconic: $\varepsilon_r = 10$, $\tan\delta = 0.0035$, height: 0.508 mm

Via diameter: 4.8 mm
Pad diameter: 5.2 mm
Comparison of simulated return loss ($S_{11}$)

Same accuracy is obtained in return loss ($S_{11}$) using conventional FDTD and efficient FDTD algorithms.

Results obtained using FDTD compare well against those obtained using commercial FEM solver.
Comparison of computational resources utilized

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Commercial FEM</th>
<th>Conventional FDTD</th>
<th>FDTD using signal processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>36 min. 36 sec</td>
<td>33 min. 15 sec</td>
<td>29 min. 30 sec</td>
</tr>
<tr>
<td>Memory</td>
<td>207 MB</td>
<td>68 MB</td>
<td>68 MB</td>
</tr>
</tbody>
</table>

Use of signal processing along with conventional FDTD algorithm reduces the computation time as compared to conventional FDTD algorithm.

Also, computational resources utilized by both FDTD algorithms are less than those utilized by the commercial FEM solver.
Conclusions

• An efficient FDTD algorithm has been proposed in this work, which utilizes signal processing techniques to reduce the overall computational cost.
• Results for different test examples demonstrated show that the FDTD algorithm used along with signal processing reduces computational cost while maintaining the accuracy of the results. Cost savings vary depending on the nature of the problem and the frequency range of interest. The savings can range from a few percent to a large factor.
References


Thank You

Questions ?