DYNAMICALLY ADAPTIVE MESH REFINEMENT FDTD: A STABLE AND EFFICIENT TECHNIQUE FOR TIME-DOMAIN SIMULATIONS

Yaxun Liu and Costas D. Sarris
The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto, Toronto, ON, M5S 3G4, Canada
E-mail: cds@waves.utoronto.ca

INTRODUCTION
The Finite-Difference Time-Domain (FDTD) technique [1] has been extensively employed in the modeling of microwave and optical structures, due to its simplicity and versatility. However, these FDTD qualities are partially compensated by the stability and numerical dispersion limitations on the choice of the cell size and the time step of the method, that render its application to complex and/or electrically large structures computationally expensive. In the past, a variety of static subgridding techniques [2]-[5] have been proposed, aimed at accelerating the conventional FDTD technique for structures with localized fine geometric features. According to such approaches, local mesh refinement is pursued in a priori defined regions of a computational domain, as dictated by physical considerations. For example, the presence of metallic edges, or high dielectric permittivity inclusions, would call for a locally dense mesh, embedded in a coarser global one. The use of local mesh refinement typically results in significant computational savings compared to the conventional FDTD, despite the fact that its implementation is associated with additional interpolation and extrapolation operations in both space and time.

However, static mesh refinement ignores the dynamic nature of time-domain field simulations. In fact, techniques such as FDTD and the transmission-line matrix (TLM) essentially register the evolution of a broadband pulse propagating in a device under test, along with its retro-reflections. Hence, a localized discontinuity in a simulated domain is only illuminated for a (potentially small) fraction of the total simulation time, during which a local mesh refinement around it is needed. Therefore, static mesh refinement, which is widely employed in frequency-domain simulations and has been incorporated in commercial finite-element tools, is only a sub-optimal solution to the mesh refinement problem in the framework of time-domain analysis. In addition, statically refined meshes are associated with the problem of late time instability, which stems from the accumulation of numerical error at dense/coarse grid interfaces.

In the context of computational fluid dynamics, the technique of Adaptive Mesh Refinement (AMR) was introduced in [6], for the solution of hyperbolic partial differential equations. The application of AMR is based on the use of a hierarchical mesh, recursively developed through the refinement of a coarse root mesh, which covers the entire computational domain. The regions of the computational domain that need further mesh refinement are detected via error estimates or indicators such as gradients of the quantity to be solved for.

Recently, the AMR technique was coupled with FDTD to produce a dynamically mesh adaptive FDTD algorithm, that was successfully applied to microwave integrated circuit [7] and optical waveguide problems [8]. Instead of applying local mesh refinement in a priori defined regions of a computational domain, the dynamic AMR-FDTD uses sub-grids, which are adaptively defined according to the spatio-temporal evolution of field distributions. As a result, significant execution time savings, up to two orders of magnitude, are attainable for large-scale open-domain problems. In this paper, the dynamic AMR-FDTD approach is explained and realistic applications, demonstrating the salient features of the method are provided.

TIME-STEPPING, AND MESH REFINEMENT IN THE DYNAMIC AMR-FDTD
A. Update process
Throughout an AMR-FDTD simulation the computational domain is comprised of hierarchically defined meshes, which form the structure of a “mesh tree”. The mesh tree is re-generated every $N_{\text{AMR}}$ time steps through a recursive mesh refinement procedure. An example is shown in Fig. 1. There is only one level-1 mesh, which is also called the root mesh and covers the entire computational domain. Each level-$m+1$ mesh is created by refining a subset of the cells of a level-$m$ mesh in a rectangular region by a factor $N_s$ (in each direction). Thus, the two meshes form a child-parent relation. The child meshes of the same parent may share an edge, but they may not overlap otherwise. In Fig. 1(b) a solid line corresponds to a child-parent relation, while a dashed line corresponds to a boundary shared by meshes of the same level. Each mesh includes Yee’s cells, where discrete field components are sampled.

The main difference between standard subgridded FDTD methods and AMR-FDTD is the dynamic mesh generation which is pursued in the latter, every $N_{\text{AMR}}$ time steps. In order to define the time step of AMR-FDTD, the following observations need to be made. For a level $M$-mesh, Yee cell dimensions are: $\Delta x/2^{M-1}$, $\Delta y/2^{M-1}$, $\Delta z/2^{M-1}$, where $\Delta x$, $\Delta y$, $\Delta z$ are the root mesh Yee cell dimensions. Furthermore, the Courant number is fixed to a constant value $s$ in all meshes. This implies that the root mesh time step $\Delta t$ (from now on referred to, as AMR-FDTD time step) is given as:

$$\Delta t = s \frac{1}{u_{p,\text{max}}} \frac{1}{\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}}}.$$

where $u_{p,\text{max}}$ is the maximum phase velocity in the computational domain. Applying (1) for a mesh of level $M$, keeping $s$ fixed, yields a time step $\Delta t_M$ for this mesh, equal to:

$$\Delta t_M = \Delta t/2^{M-1}.$$

For example, level 2 meshes are updated twice as many times as the root mesh. Thus, another shortcoming of the conventional FDTD is addressed; the minimum time step of the algorithm is only used for the update of regions of large field variations, as opposed to the whole domain, a salient feature that is also part of the fixed subgridding algorithms of [2], [5].

The loop of the AMR-FDTD operations is as follows:

1) Check the number of time steps executed. If it is an integer multiple of $N_{\text{AMR}}$, perform adaptive mesh refinement to create a new mesh tree, and carry the field values from the old mesh tree to the new mesh tree.
2) Update fields of the root mesh.
3) Copy fields from the root mesh to the boundary of the level-2 meshes. Update fields of the level-2 meshes twice through a recursive procedure. Copy fields from level-2 meshes back to the root mesh, for the time steps of the latter. The update of meshes of level 3 and up proceeds similarly in a recursive manner.
4) Check whether the maximum time step has been reached. If so, terminate the simulation, otherwise return to step 1.

B. Adaptive Mesh Refinement

The criteria that the dynamic AMR-FDTD approach applies (every $N_{\text{AMR}}$ time steps) in order to decide whether to refine a Yee cell of the root mesh, are presented in section IV of [7]. For completeness, they are briefly outlined here. Assuming a current time step $n$, the electromagnetic energy $W_{i,j,k}^n$ of each $(i,j,k)$ cell of the root mesh is approximated via the expression:

$$W_{i,j,k}^n = \int_{V_{i,j,k}} \varepsilon \left| \mathbf{E}(\mathbf{r}, n\Delta t) \right|^2 + \mu \left| \mathbf{H}(\mathbf{r}, n\Delta t) \right|^2 \, dv$$

$$\approx \frac{1}{2} \left\{ \varepsilon_{i,j,k} \left| \mathbf{E}_{i,j,k}^n \right|^2 + \mu_{i,j,k} \left| \mathbf{H}_{i,j,k}^n \right|^2 \right\} V_{i,j,k},$$

where $V_{i,j,k}$ is the volume of the cell $(i,j,k)$. For a level $M$ mesh, the energy is approximated as:

$$W_{i,j,k}^n = \sum_{m=0}^{M} \int_{V_{i,j,k}^m} \varepsilon \left| \mathbf{E}(\mathbf{r}, n\Delta t) \right|^2 + \mu \left| \mathbf{H}(\mathbf{r}, n\Delta t) \right|^2 \, dv$$

$$\approx \frac{1}{2} \sum_{m=0}^{M} \left\{ \varepsilon_{i,j,k} \left| \mathbf{E}_{i,j,k}^n \right|^2 + \mu_{i,j,k} \left| \mathbf{H}_{i,j,k}^n \right|^2 \right\} V_{i,j,k}^m.$$
where: $V_{i,j,k}$ is the volume of cell $(i,j,k)$, and $E_{i,j,k}$, $H_{i,j,k}$ are vector electric and magnetic field values at the center of the cell at time step $n$ (approximated by space/time averaging). Then, the gradient of the energy is approximated by a second order finite-difference expression, as:

$$\nabla W_{i,j,k}^n = \frac{W_{i+1,j,k}^n - W_{i,j,k}^n}{2\Delta x} + \frac{W_{i,j,k+1}^n - W_{i,j,k-1}^n}{2\Delta y} + \frac{W_{i,j,k+1}^n - W_{i,j,k-1}^n}{2\Delta z}. \quad (4)$$

Defining thresholds $\theta_g$ and $\theta_e$, a cell $(i,j,k)$ is marked for refinement if both of the following criteria are met:

$$\left| \nabla W_{i,j,k}^n \right| > \theta_g G^n, \quad W_{i,j,k}^n > \theta_e Q^n, \quad (5)$$

where:

$$G^n = \max_{i,j,k} \left| \nabla W_{i,j,k}^n \right|, \quad (6)$$

$$Q^n = \max_{0 \leq m \leq n} W_{i,j,k}^m, \quad (7)$$

$$W_{i,j,k}^m = \frac{1}{N_x N_y N_z} \sum_{i,j,k} W_{i,j,k}^m. \quad (8)$$

Once the mesh refinement algorithm is set and recursively performed, the non-uniform FDTD mesh that its application produces, can be handled according to any static subgridding technique [2]-[5]. This paper follows the methodology outlined in [7], [8]. For the multilevel case, the same algorithm is recursively applied at each mesh level.

**NUMERICAL RESULTS**

To demonstrate the dynamic AMR-FDTD algorithm, the geometry of a spiral inductor of Fig. 2 is analyzed. The parameters of this geometry are: $A_1 = 60$ mm, $A_2 = 40$ mm, $w_1 = w_2 = 2$ mm, $B_1 = 24$ mm, $B_2 = 20$ mm, $B_3 = 18$ mm, $B_4 = 4$ mm. The thickness of the substrate is 0.8 mm and its dielectric constant is 2.2. The dimensions of the computational domain enclosing the structure are $60$ mm $\times$ $40$ mm $\times$ $4$ mm. The air bridge is 0.8 mm above the substrate. The AMR-FDTD method uses a $60 \times 40 \times 10$ mesh and 8192 time steps. A reference FDTD simulation of a $120 \times 80 \times 20$ mesh is used for comparison. The AMR control parameters are $\theta_g = 0.001$, $\theta_e = 0.1$, $\theta_c = 0.6$, $N_{AMR} = 50$, $\sigma_{AMR} = 2$ [7]. The mesh refinement process is illustrated in Fig. 3, which shows the effective vertical electric field wavefront tracking achieved by the algorithm.

Scattering parameter errors and execution time data are shown in Table I. The (strict) error metric used is:

$$\mathcal{E}_S = \sqrt{\frac{\sum_k \sum_l \sum_m \left| S_{l,m}^i (f_k) - S_{l,m}^{ref} (f_k) \right|^2}{\sum_k \sum_l \sum_m \left| S_{l,m}^{ref} (f_k) \right|^2}}, \quad (9)$$

where, $f_k$ is a discrete frequency within the modeled frequency band (up to 30 GHz), $S_{l,m}$ the $(l,m)$-element of the scattering matrix of the simulated circuit, as determined by the AMR-FDTD or the coarsely meshed FDTD technique and $S_{l,m}^{ref}$ is the same element, determined by the densely meshed FDTD, which is used as a reference code. Despite the highly resonant nature of the spiral inductor, which necessitates the use of a significant number of time steps for the extraction of the $S$-parameters, significant execution time savings (of about 80%) have been achieved.

The time-domain results for this case study demonstrate the absence of late-time instability in the AMR-FDTD. In fact, the number of AMR-FDTD child meshes converges to zero over time, implying that only the root mesh is still present at a late stage of the code. Therefore, no spatial or temporal interpolation operations, which are the primary sources of instabilities in adaptive mesh FDTD codes [5], are applied then. This is an additional advantage of using a dynamically adaptive instead of a statically adaptive mesh in time-domain simulations.

<table>
<thead>
<tr>
<th>Method</th>
<th>Mesh</th>
<th>Time steps</th>
<th>Total time (s)</th>
<th>$\mathcal{E}_S$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDTD</td>
<td>$60 \times 40 \times 10$</td>
<td>8192</td>
<td>411</td>
<td>52.1</td>
</tr>
<tr>
<td>AMR-FDTD</td>
<td>$60 \times 40 \times 10$</td>
<td>8192</td>
<td>5276</td>
<td>3.74</td>
</tr>
<tr>
<td>FDTD</td>
<td>$120 \times 80 \times 20$</td>
<td>16384</td>
<td>29154</td>
<td>-</td>
</tr>
</tbody>
</table>
CONCLUSION

The application of the dynamic AMR-FDTD technique leads to significant execution time savings compared to the conventional FDTD, without significantly compromising the accuracy of dense mesh implementations of the latter. A significant aspect of the method is its late time stability. This is naturally achieved by virtue of the dynamic mesh refinement process, which guarantees that no pulses propagating in a dynamic AMR-FDTD domain encounter dense-to-coarse grid interfaces; instead, they are enclosed in locally refined meshes. Therefore, contrary to its static counterparts, the proposed technique is both efficient and late-time stable.

ACKNOWLEDGMENT

This work has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES