AMR-FDTD: A Dynamically Adaptive Mesh Refinement Scheme for the Finite-Difference Time-Domain Technique

Yaxun Liu* and Costas D. Sarris
The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario, M5S 3G4, Canada

Introduction
As the clock and data rates of digital systems increase, signal and power integrity issues become more important. A fast computational electromagnetic (EM) simulator is invaluable, especially for electronics engineers trouble-shooting high-speed Printed Circuit Boards (PCB’s) and packages. However, the latter are generally very complicated and have hundreds or even thousands of traces and vias, and a complete signal path may include several boards and connectors. Simulating the entire three-dimensional (3D) structure of typical geometries of interest reaches and sometimes surpasses the limits of the capability of most commercial simulators that employ traditional EM methods such as Method of Moments (MoM), Finite Element Method (FEM) and Finite-Difference Time-Domain (FDTD) method.

Recently Moving-Window FDTD (MWFDTD) method [1, 2] was proposed for simulating 2D EM wave propagation in large terrain environment. This method takes advantages of the localized nature of a broadband electromagnetic pulse during propagation. Since most EM energy only exists in a small region during the propagation along a long distance, only that region needs to be meshed and updated. As the pulse moves, the mesh moves with it.

When a pulse propagates along a trace on a PCB, usually it also shows a localized nature. Therefore, a modified MWFDTD can become applicable to this type of problems as well. However, when a discontinuity is encountered, multiple pulses may appear and spread in space, therefore multiple windows are potentially needed. Also, the field components in different regions may have different gradient, therefore a variable grid size is desired.

Adaptive Mesh Refinement (AMR) [3] has been developed for parabolic partial differential equations. The method uses a hierarchical mesh. It has a root mesh covering the entire computational domain. After certain time steps, cells needing refinement are detected based on error estimates (if such can be deduced) or field gradients and a clustering algorithm is then used to organize them into boxes, which are also possible to refine. If necessary, the refinement can be recursive. Previously refined regions may also be coarsened by removing the refined mesh. Meshes with different cell sizes use different time steps and are coupled at their boundaries.

FDTD sub-gridding [4-7] has been proposed for simulating structures with small-scale details. The entire structure is meshed by a coarse grid, while regions having details are meshed with a fine grid. The early development of an FDTD sub-gridding algorithm was inspired by AMR [4], yet latter FDTD sub-gridding algorithms focused on improving stability and decreasing reflection at the coarse/dense mesh boundaries. However, in all previous studies the sub-grid was fixed and therefore did not take advantage of the localized nature of the fields.
In this paper, an AMR-FDTD technique is proposed for the simulation of large complex structures. The root mesh is refined hierarchically and forms a tree structure. The ratio between the cell sizes of adjacent levels is fixed. The mesh tree is re-created after fixed time steps. The ratio between the time steps of adjacent levels is the same. At each time step of the root mesh, the fields of the upper level are updated first, and then lower levels are updated. In each level of the mesh, PEC, ABC and source conditions are enforced. Except for grid points on perfect electric conductors (PEC) and absorbing boundary conditions (ABCs), the boundary value of a child mesh is interpolated from its upper level mesh. The lower level mesh is updated multiple times until it catches up its upper level mesh. After the entire tree is updated, from bottom up, each lower level mesh updates the fields of its upper level mesh.

**Method**

This method uses a tree structure to represent the hierarchical mesh. Consider node A of the tree, that meshes a box defined by \( x_0^A \leq x \leq x_1^A, y_0^A \leq y \leq y_1^A, z_0^A \leq z \leq z_1^A \) into \( N_x^A \times N_y^A \times N_z^A \) cells (Fig.1). The cell dimensions are \( \Delta x \times \Delta y \times \Delta z \). The cell can be indexed by a triplet \((i,j,k)\). The electric fields are defined on the edges of the cells while the magnetic fields are defined on face centers, following the convention of the Yee-cell [8]. For simplicity, it is assumed that each cell has uniform material and each face of a cell is subject to uniform boundary conditions (BC). Consider a subset of the cells in a rectangular region, denoted by the indices of the cells \( \{ (i,j,k) | i_1 \leq i \leq i_2, j_1 \leq j \leq j_2, k_1 \leq k \leq k_2 \} \) (shaded region in Fig.1). We can subdivide this region using a smaller cell size \( \frac{\Delta x}{M} \times \frac{\Delta y}{M} \times \frac{\Delta z}{M} \), where \( M \) is a predefined integer. Then we can create a node B and define fields, materials and BC’s for it, in the same way we did for node A. Node B is a child of node A. Node A can have other children but they should not overlap.

In the following, the simulation of a microstrip-based circuit is considered. For simplicity, Mur’s 1\(^{st}\) order ABC is used for truncation of the board. A Gaussian pulse excitation is imposed near the end of the feeding trace [9]. A two-level mesh tree is used here. The sub-division ratio is 2, but it is straightforward to extend to other ratios.

The method is described as follows:

1. Create the root mesh which covers the entire computational domain. Denote the cell size as \( \Delta x, \Delta y, \Delta z \) and time step as \( \Delta t \).
2. Create the initial mesh tree by refining a box surrounding the source region. The box should be larger than the source region at least by \( c\Delta t \) so that the fields generated by the source won’t propagate outside of the refined mesh at the next time step.
3. Denote current time as \( t \). Back-up the current electric field. Update the magnetic field of root mesh, which is for the time \( t + \Delta t/2 \). Update the interior electric field of root mesh, which is for the time \( t + \Delta t \). Impose PEC BC by forcing tangential electric field zero. Update electric field on ABC. Add the excitation to the \( E_z \) in the source region.
4. For each child mesh, copy the electric field of its parent mesh for time \( t \) at the interface between them, excluding PEC BC and ABC. Update the magnetic field using halved time step, so the magnetic field obtained is for time \( t + \Delta t/4 \). Update
the interior electric field, which is for $t + \Delta t/2$. Impose PEC BC, ABC and source on the electric field. Backup the magnetic field.

5. Continuing with step 4, for each child mesh, copy the electric field of its parent mesh interpolated for $t + \Delta t/2$ at the interface between them, excluding PEC BC and ABC. Update the magnetic field, which is for $t + 3\Delta t/4$. Update the interior electric field, which is for $t + \Delta t$. Impose PEC BC, ABC and source on the electric field.

6. For each child mesh, copy its electric field at $t + \Delta t$ to its parent, excluding the field on their interface. Copy the magnetic field of the child mesh interpolated for $t + \Delta t/2$ to its parent.

7. In the root mesh, detect the cells which need refinement, cluster them into boxes, and re-create the child meshes. Copy the electric field of the root mesh for $t + \Delta t$ and the magnetic field of the root mesh extrapolated for $t + 3\Delta t/4$ to the child mesh. If the new child meshes overlap with some old child meshes, copy overlapping fields of the old child meshes to the new child meshes.

8. Go to Step 3, until the pre-defined end time is reached.

It should be noted that when copying fields between parent and child meshes, interpolation is necessary. We found linear interpolation is sufficient. The exact field positions should be used in the interpolation, otherwise there may be instability.

**Numerical Results**

As an example, a microstrip line is simulated by using AMR-FDTD. The thickness of the substrate is 0.5mm and its dielectric constant is 2.2. The computational domain is 40mm x 20mm x 4mm. The root mesh has 40x20x8 cells and CFL number 0.7 is used for determining the time step. Gaussian pulse $f(t) = \exp\left[-(t-t_0)^2/T^2\right]$ is imposed at 5mm from the left end of the microstrip line. $T = 15ps$, $t_0 = 3T$. At $t = 0$, an initial child mesh is created which covers the source region. After each 10 time steps, the child mesh is re-created by first detecting cells where $|E_z|$ is greater than threshold value $E_{th} = 0.2$, and then expanding the bounding box by 5 grids. Fig.2 shows the mesh and $E_z$ at cross-section $z = 0.25mm$ at different times. From this figure we can see that the child mesh can track the moving pulse. However, more advanced remeshing techniques based on the Fast Wavelet Transform are currently under investigation. Fig.3 shows the time curve of $E_z$ at $z = 0.25mm$ and 10mm from the right end of the microstrip line. The result of the coarse mesh is deduced by using 40x20x8 mesh without AMR. The result of dense mesh is deduced by using an 80x40x16 mesh without AMR. From the figure, we can see that AMR-FDTD can achieve almost the same accuracy as the dense mesh.

**References**


Fig.1 A mesh and its child mesh.  Fig.2 Mesh and Ez at z=0.25mm, t=105Δt (left) and t=135Δt (right).

Fig.3 Comparison of AMR-FDTD with FDTD.