

Blocked Adaptive Cartesian Grid FD-TD Method for Electromagnetic Field with Complex Geometries

Hiroshi Abe

Three Wells,
3-56-10 Kamiarai, Tokorozawa, Saitama, Japan

Abstract—The present paper proposes an adaptive Cartesian grid method, namely blocked adaptive Cartesian grid method to solve Maxwell’s equations with complex geometries using FD-TD scheme. The present method could be an alternative to the subgrid method of Yee’s grid. Adaptive Cartesian grid methods were classified as the unstructured grid method and originally introduced in the CFD area to solve complex geometry objects. The grid data structure is octet-tree based one and is different from Yee’s grid. It was introduced to electromagnetics using Riemann solver, which was very complicated and dissipative. Unstructured grids are not convenient to FD-TD scheme so we introduce blocked cell in order to apply FD-TD scheme on the grids. The structure and the procedures are presented in the paper. The accuracy and effectiveness of the present method is confirmed by comparing the conventional FD-TD solutions. The results show the efficiency of the method.

Keywords—Unstructured grid, Adaptive Cartesian Grid, FDTD, Alternate Digital Tree, Complex Geometry

1 Introduction

To simulate the transient solution of the electromagnetic field, finite difference-time domain (FD-TD) scheme is the most popular algorithm. The Yee spatial lattice and Leap-Frog time-marching scheme gives excellent numerical characteristics and the popular boundary conditions, Dirichlet and Neumann, are quite straightforwardly to be implemented.

The FD-TD scheme can deal with the complex geometries of the perfect electric conductive object by the staircase approximation[1] or the contour-path method. But if the finest part of the geometry is much smaller than the wave length, the total grid number would be too large to compute within a reasonable computer memory and time. Subgrid methods may be one solution of the problem [11].

The unstructured grid would be the another solution for the problem. But the FD-TD scheme basically cannot discretized on the unstructured grid.

Meanwhile, in the computation fluid dynamics (CFD) area, Cartesian grid methods have been developed. Structured orthogonal grid system may extend through solid wall boundaries of geometries within the computational domain. The primitive method flags cells to be internal, intersected or external to the geometries[12]. The external cells are considered to be volume elements for the simulation.

Then, Cartesian grid method fall into two categories with the demand of accurate solutions. One keeps its structured grid nature and introduces embedding structured subgrids within the underlying coarse structured grids. Adaptive Mesh Refinement (AMR) is one of them [4]. Figure 2 shows an example of AMR in two dimension. The intersected cells by a circle in the underlying coarse grids are tagged in blue. The blue-tagged cells are

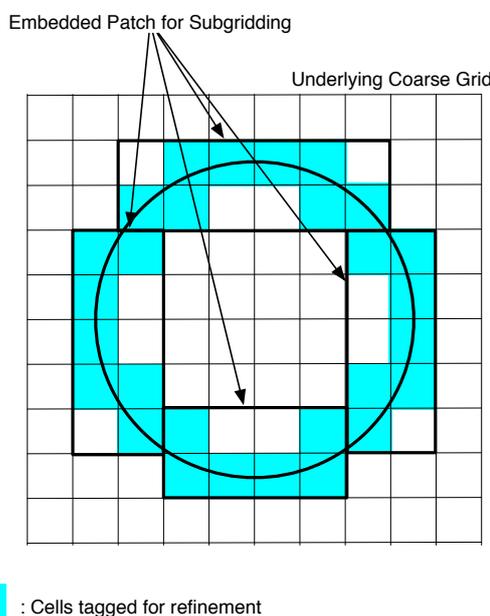


Figure 1: Schematic image of Adaptive Mesh Refinement. Intersecting meshes with a circle are tagged (blue cells) and are to be refined.

to be refined. In the AMR procedure, several embedded rectangle patches are defined so as to contain the blue-tagged cells. Then, the embedded rectangle patch areas are refined.

The other considers the Cartesian mesh as an unstructured collection of h-refined meshes. The data structure is not the same as structured grids but the same as unstructured grids. Adaptive Cartesian grid method was introduced as an unstructured Cartesian grid method

and has shown the great success in simulating complex geometries[2],[5]. Figure 2 shows a case of two-

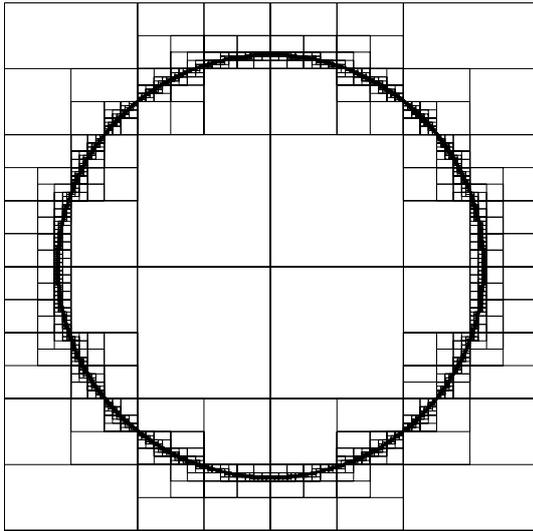


Figure 2: 2D case of Adaptive Cartesian grid method.

dimensional adaptive Cartesian grid method. Beginning with a root cell covering whole domain, the intersected cells by the circle are recursively bisected. This simple procedure finally gives Fig.2.

The actual data structure (quad-tree) of a two dimensional adaptive Cartesian grid method is shown in Fig.3. The case shows a quad-tree structure. In this two dimensional case a tree node (a cell) may have four child nodes. In the case of three dimension, a node may have eight children. Level of a node is defined as the depth of the nest in the tree structure. The root node is specified as 'level 0' (Fig.3).

As for the recent situation of Computational Electromagnetics (CEM), subgrid methods are popular [6],[8] and many commercial softwares declare their support of subgridding. Liu and Sarris [7] proposes AMR-FDTD subgridding algorithm. The method generates meshes with solution adaptive. When electric field changes steeply in some places, the method automatically refines the area. When the field doesn't need fine meshes anymore in certain area, the method coarsens the fine meshes.

The adaptive Cartesian grid method was applied to CEM [10],[13]. The discretization scheme was the Finite volume method with a Riemann solver. The scheme well-used in CFD area because of the good stability and resolution for discontinuity. The scheme is complicated and was numerically dissipative [14]. To control the dissipation, certain flux-limiting filter may be needed.

Boundary conditions of perfect-conducting (PEC) wall is very difficult to formulate in a case of sharp point of PEC. The difficulty comes from the grid system and the discretization scheme.

The present paper describes an adaptive Cartesian grid method, namely blocked adaptive Cartesian (BAC) grid method, which is capable of the FD-TD scheme. In the first part of the paper, the algorithm of the blocked adap-

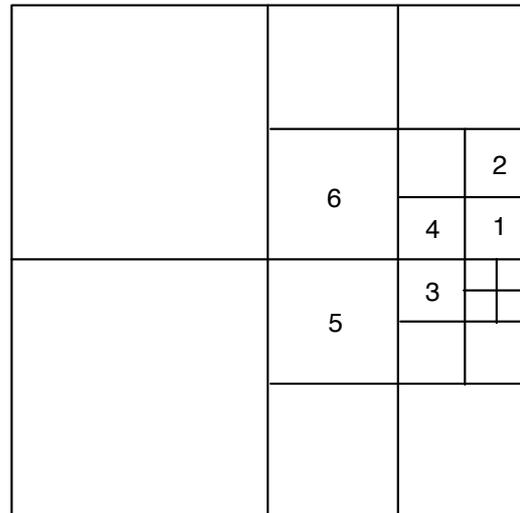
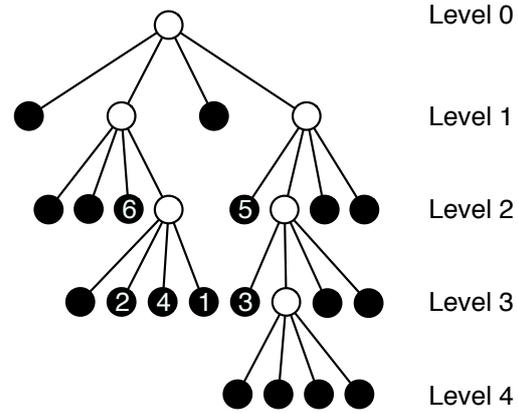


Figure 3: Octree data structure of Adaptive Cartesian grid method. Each black circles indicates leaf nodes in the tree structure and they correspond to the cells as is shown with the numbers.

tive Cartesian grid method is described. Then an evaluation case is presented to show the ability of the method in accuracy and in effectiveness.

2 Blocked Adaptive Cartesian Grid Method

The adaptive Cartesian grid method is categorized in the unstructured grid as is described in the previous section. The element cell is hexagonal like structured Cartesian grid however, the data structure is quite different from that of the structured grid.

The adaptive Cartesian procedure is described in Fig.5. This procedure generates an Octet-tree (octree) structure of the grid. Because a node may have eight child-nodes in three dimension as is shown in Fig.4. Each leaf-node of the octree indicates the corresponding grid cell (Fig.3).

Because of the unstructured grid nature of the adaptive Cartesian grid method, we find that it is difficult to apply the FD-TD method to the adaptive Cartesian method. Because the FD-TD simply needs Yee's grid, a structured

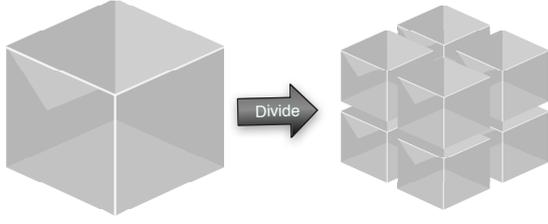


Figure 4: Cell division in three dimensional case. The data structure becomes octet-tree based because each node may have 8 children.

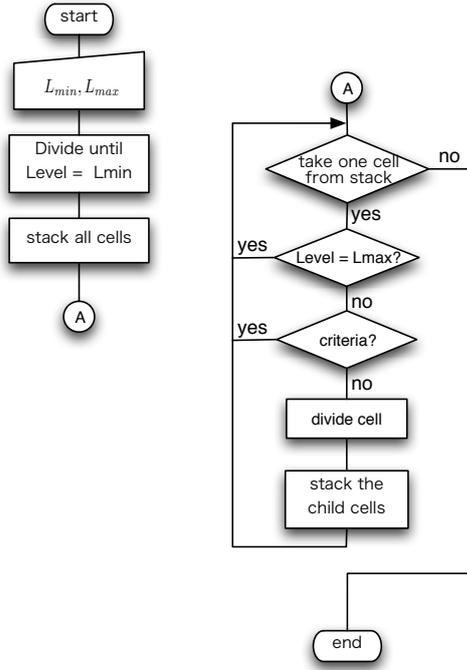


Figure 5: Cell division flow chart. The level of cells are bounded by L_{min}, L_{max} . The cut-cells with a geometry are divided until a certain criteria is satisfied.

grid.

As a solution of applying FD-TD to the octree adaptive Cartesian method, we introduce a block cell which consists of a structured grids instead of a single grid cell [4],[9]. Each block cell contains equi-distance grids with the same number in each coordinate direction. Figure 6 shows the example of the grids and the block cells. The coloured blocks indicate the block cells and the solid lines indicate the grids in the each block cells. The outermost grids of a blocked cell are overlapped between the adjacent cells. A normal FD-TD calculation can be performed on the each block cell. The boundary values in the block cell are interchanged with the adjacent block cells in each time step. An interpolation is performed in the interchange if necessary.

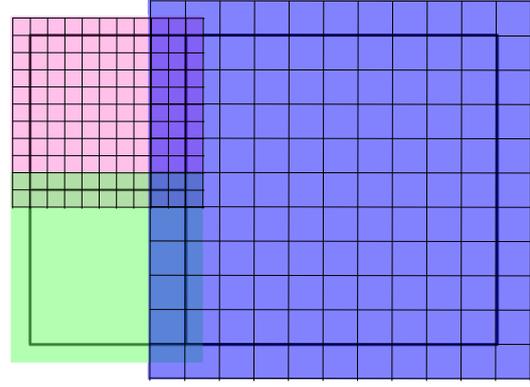


Figure 6: The grids in the block cell and relation of adjacent cells.

3 Cell Division for a Geometry

This section describes the *criteria* in Fig.5. First, we describe the strategy to decide a cell to be divide into smaller cells which is described in [2]. The strategy gives a criteria which determines whether a intersected cell by triangular facets is to be divided or not.

Next, a method which is a fast algorithm to find intersecting triangular facets with respect to a cell is described.

3.1 Division Criteria

3D geometry is often provided as a CAD file, especially DXF or Stereo Lithography (STL) data are often used. Both data may consist of a set of triangulated facets of the geometry's surface. We need criteria to decide the cells to be divided in order to resolve the geometry through the triangulated facets. We adopt a curvature detection

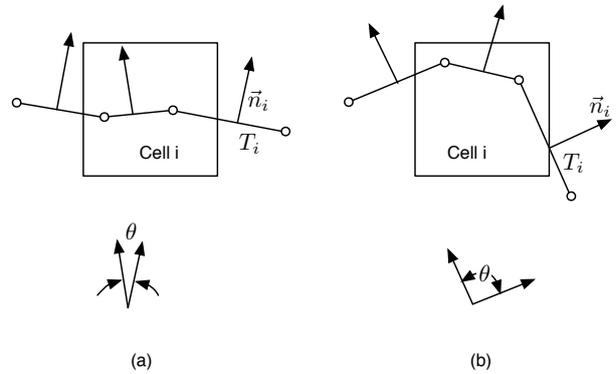


Figure 7: Schematic 2D view of angular variation of normal vector of triangles within cut-cell i. (a) is small variation case and (b) is large variation case.

strategy [2]

Suppose a cell is intersected by triangular facets T_i . \vec{n}_i is the normal vector of the facets (Fig.7).

Angle variation V_j can be defined as,

$$V_j = \max(n_{k_j}) - \min(n_{k_j}), \forall k \in T_i, (j = x, y, z) \quad (1)$$

where k is a running index to sweep over the set of facets T_i . The angle which indicates the curvature of the facets is given by,

$$\cos(\theta_{i_j}) = \frac{V_j}{|\vec{V}|}. \quad (2)$$

If θ in a cell exceeds a predefined angle threshold, then the cell is tagged for division.

This procedure for division is very simple and robust. One can have adaptive Cartesian cells automatically.

3.2 Fast Search for Intersected Triangles

The cell division criteria need the complete set of the triangular facets which are intersecting a block cell. If you simply traverse all the cells and triangles to check their intersection, the computation cost is quite expensive.

Alternating Digital Tree (ADT) algorithm is a quite effective to find intersecting candidates [3]. The ADT maps bounding-box region in R^3 as a point in R^6 . The bounding-box contains a triangular facet. Figure8 shows the case of a region in R^1 .

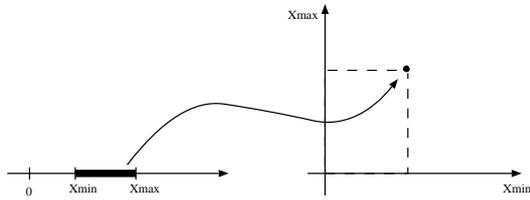


Figure 8: Mapping of a region in R^1 as a point in R^2 .

All the triangular facets are stored as points in hyper-space R^6 in the ADT, which is basically binary-tree in any spatial dimensions. ADT recursively bisects a spatial region in alternate direction as is shown in Fig.9 in order to add a new point in ADT.

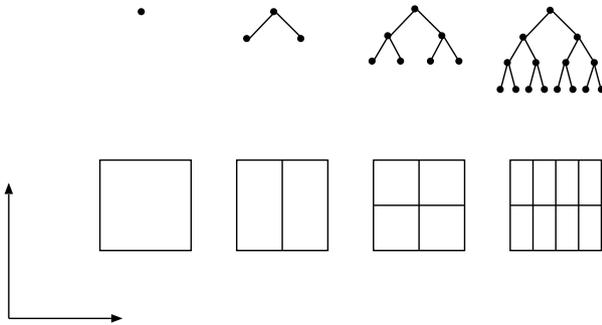


Figure 9: Binary tree and a bisection process in ADT.

In N -dimensional space, a triangular facet bounded by $x_{k,min}, x_{k,max}$ the point in $2N$ hyperspace \vec{x}_k is described as,

$$\vec{x}_k = [x_{k,min}^1, \dots, x_{k,min}^N, x_{k,max}^1, \dots, x_{k,max}^N]^T, \quad (3)$$

where k is a running index for the triangular facets of a geometry in R^N . The intersection condition with a cell

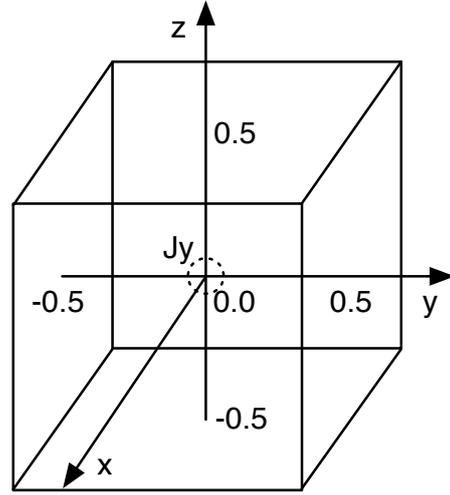


Figure 10: Simulation Model

can be described simply as

$$a^i \leq x_k^i \leq b^i \text{ for } i = 1, 2, \dots, 2N, \quad (4)$$

where \vec{a}, \vec{b} are

$$\vec{a} = [0, \dots, 0, x_{cell,min}^1, \dots, x_{cell,min}^N]^T, \quad (5)$$

$$\vec{b} = [x_{cell,max}^1, \dots, x_{cell,max}^N, 1, \dots, 1]^T, \quad (6)$$

$x_{cell,min}^i, x_{cell,max}^i$ are the minimum and maximum coordinates of the cell, respectively. Note that $[0, \dots, 0]^T$ and $[1, \dots, 1]^T$ are the coordinate limits of the target geometry.

ADT only gives intersection candidates, so precise geometric intersection checks must be applied to obtain a complete list of the intersected facets.

4 Evaluation

The section evaluate the accuracy and effectiveness of the proposed method. And then, the actual speed of grid generation with a certain DXF geometry is evaluated.

4.1 Accuracy

As an evaluation case, we demonstrate wave excitation by centre-located current as shown in Fig.10 and Eq.(7).

$$J_y(x, t) = e^{-\frac{(x-x_0)^2}{2\Delta x^2}} e^{-\frac{(t-t_0)^2}{2\Delta t^2}} \cos \omega(t - t_0), \quad (7)$$

We demonstrate two normal FD-TD with coarse grid and fine grid. Also we demonstrate the blocked Adaptive Cartesian grid with third level, the depth of the octree, around the current distribution and second level on the other area. The distribution of the blocked cells are shown in Fig.11. Each bounding-box contains $8 \times 8 \times 8 = 512$ cells.

The grid width of coarse FD-TD is the same with that of the second level and that of fine FD-TD is the same with that of the third level.



Figure 11: The distribution of blocked Cells.

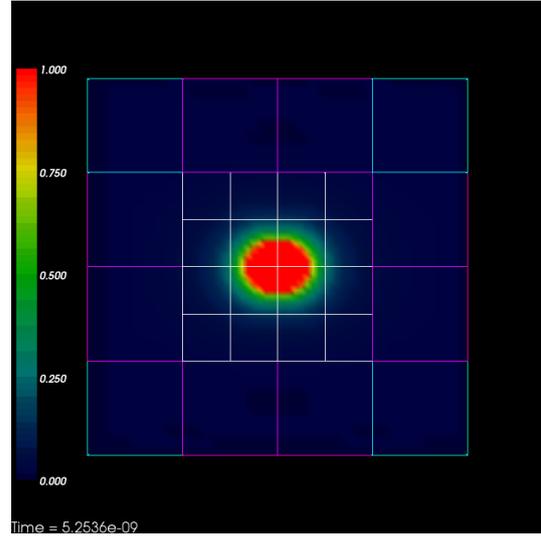


Figure 13: Electric field strength distribution on the blocked adaptive Cartesian cells.

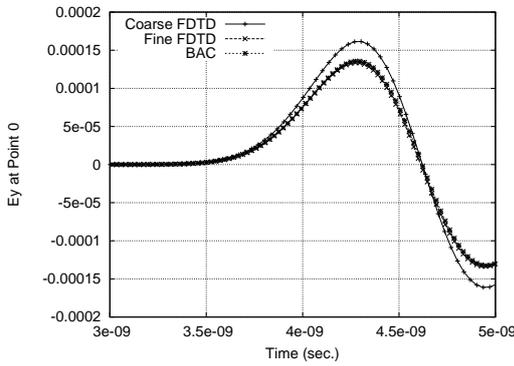


Figure 12: Electric Field Variation.

The results is shown in Fig.12 and the result of the fine FD-TD and BAC FD-TD shows the good agreement. The (effective) number of grids in the case is shown in Tab.4.1. The number of grids corresponds to the amount of computation of the schemes. The table shows the BAC is quite effective in computation to get high accuracy.

4.2 Grid Generation

The most important algorithms in the grid generation is ADT. According to [3], the computation cost to generate unstructured finite element method grids are proportional to $N \log(N)$, where N is the number of elements. The

	The Grid Num.
Coarse FD-TD	35,937
Fine FD-TD	274,625
BAC FD-TD	159,720

Table 1: The number of grids of the case.

actual cost in the proposed method in a certain case is evaluated here.

A fixed geometry of 616 triangular facets is used for evaluation. The upper limit of level of ADT node is changed from 3 to 6 for the fixed geometry. The actual computation times are obtained as Fig. 14. The com-

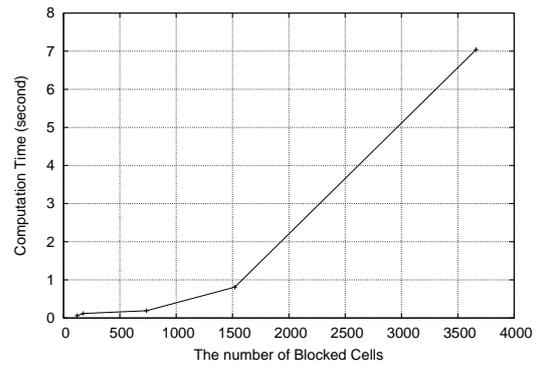


Figure 14: Computation time of the blocked cells generation with PowerPC 1.8GHz MacOS X 10.5. The number of triangular facets of the model is 616.

putation time doesn't include the cut-cell search in each blocked cell.

As a typical case of the best performance of BAC method, we generate mesh cells to solve a small PEC sphere (radius=0.01m) centred in an area (1x1x1m). BAC method generates blocked cells concentrated around the sphere, the density is controlled by the upper level of oct-tree node. Figure 15 show the comparison of BAC method with equi-distanced FDTD in the case of giving the same resolution. The spatial width of the mesh cells (δx) of the FDTD is equal to the smallest δx of the mesh grids of BAC method in the Fig.15. The total mesh cell number of the BAC methods is much smaller than that of the

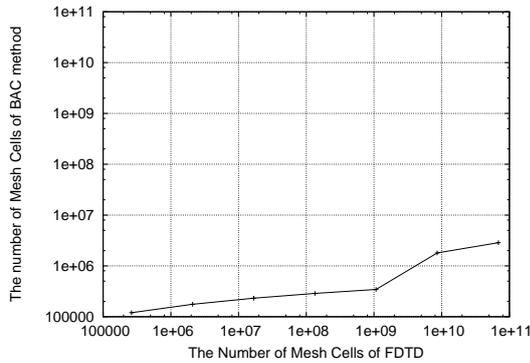


Figure 15: The number of mesh cells of BAC method automatically generated in the different level of octree and that of the equivalent equi-distanced FDTD whose mesh width is equal to the smallest width of the corresponding BAC method.

equi-distanced FDTD in the case. The time step width is limited by CFL condition, so the both methods use the same time step width. This means they require the same computational costs in the time integration per cell. The mesh cell number indicates simply the cost of computation and memory.

If single precision of a set of physical values, electric and magnetic field, is stored in a mesh cell, it is required $4 \times 6 = 24$ bytes memory in the both methods. The octree data of BAC method is a difference between the two methods. But the octree data only stores the nodes connectivity so usually negligible. Especially, we adopts blocked cells so the number of nodes is small. In the biggest case in Fig.15, the number of blocked cells are 2864.

The computation cost in a mesh cell is defined by the adopted scheme. The both methods adopts Leap-Frog method so they are no difference in computational cost per cell. BAC method requires an additional interpolation of physical values to interchange the values between the adjacent cells in different tree node level. The interpolation cost should not be so large as the cost of computation for a solution inside a block cell but isn't estimated yet.

BAC method shows an excellent memory efficiency in a certain case. The computation cost also indicates very good efficiency.

5 Conclusion

The present paper proposes an adaptive Cartesian grid method, namely blocked adaptive Cartesian grid method to solve Maxwell's equations with complex geometry. The procedures to generate the octree-based Cartesian grids for the complicated geometries are described. Alternate Digital Tree algorithm is applied to the proposed method in order to search speedily the intersected facets of a geometry. The accuracy and effectiveness of the present method is confirmed by comparing the conventional FD-TD solutions. The computation cost of grid generation is also shown as reasonable. A simple case is

demonstrated to compare the number of mesh cells to give a same resolution in both BAC method and equi-distanced FDTD. The results shows a very good efficiency in memory and computation of the proposed method.

References

- [1] Hiroshi Abe and Ken Hayami. Three dimensional electromagnetic field simulator - an application to a microwave device -. In *The 14th JSST Conference*, pages 117–122. JSST, 1992.
- [2] M.J. Aftosmis. Solution adaptive cartesian grid methods for aerodynamic flows with complex geometries, 1997.
- [3] Javier Bonet and Jaime Peraire. An alternating digital tree (adt) algorithm for 3d geometric searching and intersection problems. *Int. J. for Numerical Methods in Engineering*, 31:1–17, 1991.
- [4] D. DeZeeuw and Kenneth G. Powell. An adaptively refined cartesian mesh solver for the euler equations. *AIAA-91-1542-CP*, 1991.
- [5] Kozo Fujii Keiichiro Fujimoto and Z.J.Wang. Improvements in the reliability and efficiency of body-fitted cartesian grid method. In *AIAA Aerospace Sciences Meeting*, 2009.
- [6] I. S. Kim and W. J. R. Hoefer. A local mesh refinement algorithm for the time domain-finite difference method using Maxwell's curl equations. *IEEE Transactions on Microwave Theory and Techniques*, 38(6):812–815, June 1990.
- [7] Y. Liu and C. D. Sarris. Efficient modeling of microwave integrated-circuit geometries via a dynamically adaptive mesh refinement-FDTD technique. *IEEE Transactions on Microwave Theory and Techniques*, 54(4):689–703, April 2006.
- [8] M. Okoniewski, E. Okoniewska, and M. A. Stuchly. Three-dimensional subgridding algorithm for FDTD. *IEEE Transactions on Antennas and Propagation*, 45(3):422–429, March 1997.
- [9] Kenneth G. Powell, Philip L. Roe, Timur J. Linde, Tamas I. Gombosi, and Darren L. De Zeeuw. A solution-adaptive upwind scheme for ideal magnetohydrodynamics. *J. Comput. Phys.*, 154(2):284–309, 1999.
- [10] V. Shankar, W.Hall, and A.H. Mohammadian. A cfd-based finite-volume procedure for computational electromagnetics-interdisciplinary applications of cfd methods. *AIAA-89-1987-CP*, 1989.
- [11] A. Taflove and S. Hagness. *Computational Electrodynamics: The Finite-Difference Time-Domain Method*, 3 ed. Artech House, Boston, MA, 2005.

- [12] K. Tsuboi, K. Miyakoshi, and K. Kuwahara. Incompressible flow simulations with solid models in rectangular grid system. In J. Huser A. S. Arcilla and J. F. Thompson P. R. Eiseman, editors, *Proceedings of Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, 1991.
- [13] Z. J. Wang, A. J. Przekwas, and Yen Liu. A FV-TD electromagnetic solver using adaptive Cartesian grids. *Computer Physics Communications*, 148(1):17–29, October 2002.
- [14] Z.J. Wang. private communication. 2008.