



ELSEVIER

Computer Physics Communications 148 (2002) 17–29

Computer Physics  
Communications

www.elsevier.com/locate/cpc

# A FV-TD electromagnetic solver using adaptive Cartesian grids

Z.J. Wang<sup>a,\*</sup>, A.J. Przekwas<sup>b</sup>, Yen Liu<sup>c</sup>

<sup>a</sup> Department of Mechanical Engineering, Michigan State University, East Lansing, MI 48824, USA

<sup>b</sup> CFD Research Corporation, 215 Wynn Drive, Huntsville, AL 35805, USA

<sup>c</sup> Research Scientist, Mail Stop T27B-1, NASA Ames Research Center, Moffett Field, CA 94035, USA

Received 29 August 2001

## Abstract

A second-order finite-volume (FV) method has been developed to solve the time-domain (TD) Maxwell equations, which govern the dynamics of electromagnetic waves. The computational electromagnetic (CEM) solver is capable of handling arbitrary grids, including structured, unstructured, and adaptive Cartesian grids, which are topologically arbitrary. It is argued in this paper that the adaptive Cartesian grid is better than a tetrahedral grid for complex geometries considering both efficiency and accuracy. A cell-wise linear reconstruction scheme is employed to achieve second-order spatial accuracy. Second-order time accuracy is obtained through a two-step Runge–Kutta scheme. Issues on automatic adaptive Cartesian grid generation such as cell-cutting and cell-merging are discussed. A multi-dimensional characteristic absorbing boundary condition (MDC-ABC) is developed at the truncated far-field boundary to reduce reflected waves from this artificial boundary. The CEM solver is demonstrated with several test cases with analytical solutions.

© 2002 Elsevier Science B.V. All rights reserved.

PACS: 02.70.Fj

Keywords: Computational electromagnetics; Finite volume method; Adaptive Cartesian grid

## 1. Introduction

Maxwell's partial differential governing equations for electromagnetics represent a fundamental unification of electric and magnetic fields predicting electromagnetic wave phenomena. This achievement was sometimes viewed as the most outstanding of the 19th century science [1]. Although analytical solutions of Maxwell equations exist for simple geometries, solutions of these equations for a vast majority of engineering problems have to be sought through

computational simulations, i.e. Computational ElectroMagnetics (CEM). Now engineers worldwide are using computers to obtain solutions of Maxwell equations for the purpose of investigating electromagnetic wave scattering, radiation, and guiding. One of the primary computational approach in CEM is the so-called method-of-moments (MM) [2], which involves solving frequency-domain integral equations. One needs to set up and solve dense, full, complex valued systems of linear equations, which is extremely CPU and memory intensive for medium to high frequency problems. Prompted to a significant degree by the limitations of MM, there has been an explosion of interest in direct solutions of the fundamen-

\* Corresponding author.

E-mail address: zjw@egr.msu.edu (Z.J. Wang).

tal Maxwell's equations on space grids in the time domain.

The most popular algorithm used in solving Maxwell's equations in the time domain is undoubtedly the FD-TD scheme developed by Yee [3] and later refined by many researchers [4–7]. FD-TD was originally developed on uniform Cartesian grid, and later was extended to handle body-fitted-curvilinear grids [8–10]. During the last decade, FD-TD has been used to tackle many challenging electromagnetic problems such as radar cross section (RCS) of complete aircraft, phased arrays of antennas, hyperthermia treatment of cancer, etc. In 1989, Shankar et al., developed what is called a FV-TD method [11], which solved the Maxwell's equations using a cell-centered finite volume scheme with a CFD-like Riemann solver approach. Due to its control volume formulation, FV-TD can easily handle body-fitted non-orthogonal grids. More recently, FV-TD was further refined [12–17] and extended to unstructured grid [16,18,24]. With the unstructured grid technology, grid generation for complex geometries can be completely automated. Compared with the original FD-TD scheme on a stair-step Cartesian grid, FV-TD can easily achieve high geometric fidelity.

One particular type of unstructured grids is the so-called adaptive Cartesian grid, which has been used very successfully in Computational Fluid Dynamics (CFD) [19–22]. The adaptive Cartesian grid has several unique advantages over traditional tetrahedral grids. First, Cartesian cells are much more efficient in filling space than tetrahedral cells given a certain grid resolution. This can be easily understood with the fact that at least five tetrahedra are needed to fill a single cube without adding a grid point. Second, it's difficult to generate nearly isotropic tetrahedral cells because an equilateral tetrahedral is not a space-filling topology (i.e. one cannot fill up a 3D space with identical tetrahedra) as Cartesian cells [23]. The skewness of tetrahedral cells can degrade both accuracy and efficiency (by reducing the allowable time step) of the CEM solver. Third, it is straightforward to cluster or decluster cells in a certain region with adaptive Cartesian grid. For example, one can cluster cells near a geometry and de-cluster cells away from it in wave scattering problems. Finally with cell-cutting the adaptive Cartesian grid preserves the fidelity of a given geometry.

It is therefore argued in this paper that adaptive Cartesian grid is the most promising grid topology for a CEM solver considering both efficiency and accuracy. The paper is organized as follows. In the next session, issues concerning adaptive Cartesian grid generation for arbitrary curved geometries are discussed. After that, a second-order CEM solver in both space and time is described. It is also explained why the solver is capable of resolving material interfaces exactly. In addition, a particular absorbing boundary condition suitable for the unstructured FV-TD solver is presented. Then several demonstration cases are presented to showcase the capability of the present method. Finally, several conclusions are made to complete the paper.

## 2. Adaptive Cartesian grid generation

The use of Cartesian grids in solving partial differential equations (PDE) started decades ago because it is trivial to generate the computational grid. The most serious obstacle in applying Cartesian grid technique to realistic problems is the boundary treatment for curved geometries. In the original Yee 1966 paper, uniform Cartesian grids were used to solve the Maxwell's equations. Curved geometries were approximated with stair-stepped Cartesian grids, inevitably introducing errors in geometry definition and also the computed field solution. Although Yee's FD-TD scheme was extended to body-fitted structured grids [8–10], numerical errors were introduced due to the skewness and non-uniformity of the computational grid. As a result, the most widely used computational grid in FD-TD analysis is still the uniform Cartesian grid. This is evident due to the fact that most CEM commercial FD-TD packages use uniform Cartesian grids. Apart from the drawback of non-body conforming, the uniform Cartesian grid has another disadvantage in that fine grid resolutions must be maintained everywhere, even if it is unnecessary, wasting considerable computer resources.

With the adaptive Cartesian grid, grid cells can be clustered and de-clustered anywhere based on the geometry and/or the physics of the computational fields if necessary. For example, the computational grid for a wave scattering geometry can be easily clustered near the geometry and de-clustered in the far

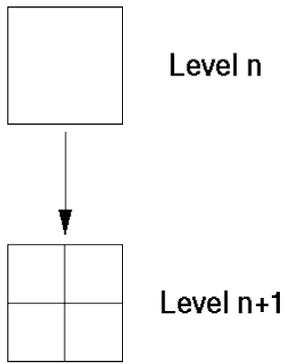


Fig. 1. Schematic of the quadtree data structure.

field to accurately resolve the geometry without wasting computer resources in regions far away from the geometry. An Quadtree (Octree in 3D) data structure is used in generating an adaptive Cartesian grid. In a Quadtree data structure, a parent cell can have four children, as shown in Fig. 1. For the purpose of tree-traversal, each cell stores the pointer to its parent cell and also its pointers to the children (if any). The adaptive Cartesian grid is usually generated through recursive subdivisions of a single Cartesian cell called the root cell covering the entire computational domain. For electromagnetic wave scattering problems by solid bodies, coarse grid cells may be used in the far field, e.g., with about 10 points per wavelength (ppw) (the 10 ppw resolution is used here as an example to show that the adaptive Cartesian grid can easily accommodate any grid resolution). Therefore, given the expected wave frequency  $f$ , one can easily compute the wave length by

$$\lambda = \frac{c}{f}, \quad (1)$$

where  $c$  is the speed of light. The maximum allowed Cartesian cell size is then  $\lambda/10$ . The Cartesian grid can be easily refined everywhere to satisfy this requirement. To fully resolve the geometry, there is a minimum grid resolution which the Cartesian grid must satisfy, i.e.

$$h \leq A, \quad (2)$$

where  $h$  is the Cartesian cell size close to the body,  $A$  is determined from the characteristic length scale of the body being simulated. Furthermore, one can also

require that a 20 ppw grid resolution is used near the body, i.e.

$$h = \frac{\lambda}{20}. \quad (3)$$

From Eqs. (2) and (3), we can determine

$$h \leq \min\left(\frac{\lambda}{20}, A\right). \quad (4)$$

Any Cartesian cell intersecting the body can be easily refined to satisfy Eq. (4). The requirement of Eq. (3) can be enforced for at least a wave length in all directions close to the body.

Another unique advantage of the adaptive Cartesian grid is that the exact geometry is captured with cell-cutting [20,22]. Cell-cutting is the operation of using the body geometry surface to intersect the Cartesian cells, and to divide the cells' interior to the body to the exterior. For all the problems shown in this paper, the geometry formats are connected line segments in 2D, and "water-tight" triangulated surface in 3D. The basic geometric operation in cell-cutting is line–line (in 2D) and face–face (3D) intersections. Before the 3D version of the cell-cutting algorithm is presented, several terminologies are defined. The arbitrarily-shaped polyhedral cells of the Cartesian grid resulting from cell-cutting are named *cut-cells*. Cartesian grid faces which are intersected by the body geometry are called *cut faces*. Note that efficient search operations are critical in the cell-cutting process to ensure that cell-cutting can be performed in reasonable amount of time. For this purpose, the Octree data structure is used to facilitate fast search operations, such as finding the Cartesian cells overlapping the bounding box of a triangular face on the body geometry. In addition, an alternating-digital-tree (ADT) [25] data structure is used to record the bounding boxes of the triangles of the body geometry. The ADT structure can be used to identify all the triangles intersecting a particular Cartesian cell efficiently, i.e. in  $O(\log(N_{\text{triangle}}))$  time rather than in  $O(N_{\text{triangle}})$  time with an exhaustive search. In summary the following algorithm has been used for cell-cutting:

- Generate a point list, an edge list, and face list for the triangulated body surface, and establish their mutual relations;
- Build an ADT structure for the bounding boxes of the triangles;

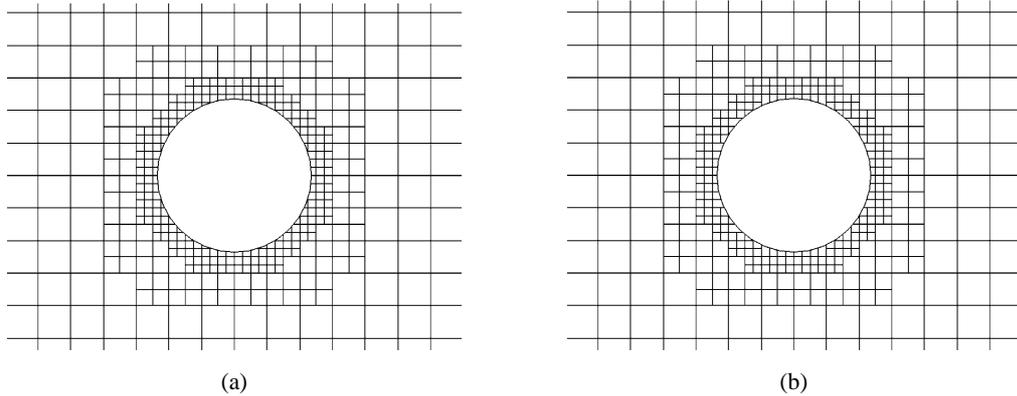


Fig. 2. An example adaptive Cartesian grid before and after cell merging.

- Find the intersection points between each edge in the edge list and the Cartesian grid faces. The Octree is used for efficient search operations;
- Find the intersection points between each edge of the Cartesian grid and the triangulated surface. The ADT is used for fast search operations;
- Identify face–face intersections based on the edge–face intersections identified in the previous two steps;
- Form cut faces for all Cartesian faces cut by the body surface;
- For cut-cells for all Cartesian cells intersected by the body surface.

If the geometry has sharp edges, the cell-cutting operation is capable of preserving them. However, some of the cut-cells may not be convex. In our numerical tests, it appears that the field solver has no difficulty handling non-convex cells. Since cut cells may have arbitrary topology, the field solver must be capable of handling arbitrary polygons or polyhedra. An example adaptive Cartesian grid with cell-cutting is shown in Fig. 2(a). Note that cell-cutting produced many irregular cut cells which can have nearly diminishing cell volumes. The small cut cell can impose very stringent stable time-step limit, which will severely degrade solver efficiency. This problem is eliminated through the so-called cell-merging [21], i.e. merging the small cut cell with one of its bigger neighboring cells. An example of cell-merging is shown in Fig. 2(b). Note that some of the small cut cells near the body are merged with their neighbors to form bigger cells.

### 3. Finite volume discretization

The time-domain Maxwell equations for nonconducting dielectrics can be written in a vector form as

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \times \mathbf{L} = 0, \quad (5)$$

where

$$\mathbf{Q} = \begin{bmatrix} \mathbf{D} \\ \mathbf{B} \end{bmatrix} \quad (6)$$

contains the electric displacement and the magnetic induction vectors and

$$\mathbf{L} = \begin{bmatrix} -\mathbf{H} \\ \mathbf{E} \end{bmatrix} \quad (7)$$

contains the magnetic and electric intensity vectors. For simplicity, we assume linear isotropic constitutive relations, i.e.

$$\mathbf{D} = \varepsilon \mathbf{E}, \quad (8)$$

$$\mathbf{B} = \mu \mathbf{H}, \quad (9)$$

where the permittivity  $\varepsilon$  and permeability  $\mu$  of the material are scalar constants, which determine the wave phase speed  $c = 1/(\varepsilon\mu)^{1/2}$ . In solving (5) with a finite volume scheme, we first need to discretize the computational domain into small control volumes. For geometric flexibility, the control volumes are arranged in an unstructured manner, and can take arbitrary shapes, i.e. arbitrary polygons in two dimensions (2D) and arbitrary polyhedra in three dimensions (3D).

Then integrating Eq. (5) in an arbitrary control volume with  $N$  planar polygonal faces, we obtain

$$\frac{\partial \mathbf{Q}}{\partial t} dV + \sum_{i=1}^N \mathbf{n}_i \times \mathbf{L}_i dS_i = 0, \quad (10)$$

where  $dV$  is the volume of the control volume,  $\mathbf{n}_i$  is the unit normal of face  $i$  of the control volume,  $dS_i$  is the face area of face  $i$ . Each control volume has a cell-averaged  $\mathbf{Q}$  vector, which is assumed to be the point  $\mathbf{Q}$  vector at the cell centroid (which is correct up to second-order). It is easy to see that the face tangential components of the electric and magnetic fields determine the time variation of the volume averaged electromagnetic fields. It is well known that a simple central difference-type method for (10) results in odd–even decoupling. Instead, CFD-type upwind schemes based on a Riemann solver or intensity-vector splitting [11–15] are implemented for unstructured grids. The basic method can be divided into the following three components: reconstruction, intensity-vector computation and time integration, which are presented in the following sections.

### 3.1. Reconstruction

In a cell centered finite volume procedure, field variables are known in a cell-average sense. No indication is given as to the distribution of the solution over the control volume. In order to evaluate the intensity vector at a face, field variables are required at both sides of the face. This task is fulfilled by reconstruction. A least squares reconstruction method is selected in this study. This reconstruction is capable of preserving a linear function on an arbitrary grid. Given an arbitrary field variable  $q$ , the gradients of  $q$  are constructed by the following least squares reconstruction

$$\begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = \frac{1}{\Delta} L \begin{bmatrix} \sum_n (q_n - q_c)(x_n - x_c) \\ \sum_n (q_n - q_c)(y_n - y_c) \\ \sum_n (q_n - q_c)(z_n - z_c) \end{bmatrix}, \quad (11)$$

where:

$$\Delta = I_{xx}(I_{yy}I_{zz} - I_{yz}^2) + I_{xy}(2I_{xz}I_{yz} - I_{xy}I_{zz}) - I_{xz}^2I_{yy}, \quad (12)$$

$$\begin{aligned} I_{xx} &= \sum_n (x_n - x_c)^2, \\ I_{yy} &= \sum_n (y_n - y_c)^2, \\ I_{zz} &= \sum_n (z_n - z_c)^2, \end{aligned} \quad (13)$$

$$\begin{aligned} I_{xy} &= \sum_n (x_n - x_c)(y_n - y_c), \\ I_{yz} &= \sum_n (y_n - y_c)(z_n - z_c), \\ I_{xz} &= \sum_n (x_n - x_c)(z_n - z_c) \end{aligned} \quad (14)$$

and

$$L = \begin{bmatrix} I_{yy}I_{zz} - I_{yz}^2 & I_{xz}I_{yz} - I_{xy}I_{zz} & I_{xy}I_{yz} - I_{xz}I_{yy} \\ I_{xz}I_{yz} - I_{xy}I_{zz} & I_{xx}I_{zz} - I_{xz}^2 & I_{xy}I_{xz} - I_{xx}I_{yz} \\ I_{xy}I_{yz} - I_{xz}I_{yy} & I_{xy}I_{xz} - I_{xz}I_{yz} & I_{xx}I_{yy} - I_{xy}^2 \end{bmatrix}, \quad (15)$$

where subscript  $n$  indicates the supporting neighbor cells, and subscript  $c$  denotes the current cell,  $x$ ,  $y$ ,  $z$  are cell centroid coordinates. It can be observed that matrix  $L$  and  $\Delta$  are dependent on the geometry only. If one stores  $I_{xx}$ ,  $I_{yy}$ , etc. the reconstruction can be performed efficiently with one loop over the face list.

### 3.2. Intensity-vector-computation

After the cell-wise reconstruction, the field variables at the left and right side of any face can be determined based on a simple Taylor expansion, i.e.

$$\mathbf{Q}_{fL} = \mathbf{Q}_L + \nabla \mathbf{Q}_L \bullet (\mathbf{r}_f - \mathbf{r}_L), \quad (16)$$

$$\mathbf{Q}_{fR} = \mathbf{Q}_R + \nabla \mathbf{Q}_R \bullet (\mathbf{r}_f - \mathbf{r}_R), \quad (17)$$

where  $\mathbf{r}_f$  is the position vector of the face center,  $\mathbf{r}_L$  and  $\mathbf{r}_R$  are the position vectors of the left and right cell centroids. Then the intensity vector at the face is computed based on a Riemann solver [11]. Given the left and right field variables, the intensity vector at the face can be expressed as

$$\mathbf{L}_f = \begin{bmatrix} -\frac{(\mu c)_R \mathbf{H}_R + (\mu c)_L \mathbf{H}_L - \mathbf{n} \times (\mathbf{E}_R - \mathbf{E}_L)}{(\mu c)_L + (\mu c)_R} \\ \frac{(\epsilon c)_R \mathbf{E}_R + (\epsilon c)_L \mathbf{E}_L + \mathbf{n} \times (\mathbf{H}_R - \mathbf{H}_L)}{(\epsilon c)_L + (\epsilon c)_R} \end{bmatrix}. \quad (18)$$

### 3.3. Time integration

An explicit two-stage scheme is used to integrate (10) in time with second-order time accuracy, i.e.

$$\mathbf{Q}^* = \mathbf{Q}^n - \frac{0.5 \times \text{Res}(\mathbf{Q}^n)}{dV}, \quad (19)$$

$$\mathbf{Q}^{n+1} = \mathbf{Q}^n - \frac{\text{Res}(\mathbf{Q}^*)}{dV}, \quad (20)$$

where

$$\text{Res}(\mathbf{Q}) = \sum_{i=1}^N \mathbf{n}_i \times \mathbf{L}_i dS_i. \quad (21)$$

In order to analyze the accuracy and stability of the above finite volume method, let's consider the following one-dimensional linear wave equation

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0, \quad (22)$$

where  $u$  is a state variable, and  $c$  is a positive constant representing the wave speed. Assume that a uniform mesh  $x_i = i \Delta x$  is used to solve (22), and  $u_i$  is the cell-averaged state-variable at the  $i$ th cell  $[x_i, x_{i+1}]$ . The semi-discrete finite-volume scheme for (22) with a linear reconstruction and Roe's Riemann solver can be written as:

$$\frac{\partial u_i}{\partial t} + c \left( \frac{u_{i+1} + 3u_i - 5u_{i-1} + u_{i-2}}{4\Delta x} \right) = 0. \quad (23)$$

Without the loss of second-order accuracy, the cell-averaged state variable can be taken to be the state variable at the centroid of the cell. Then it is easy to show using a Taylor expansion that

$$\frac{u_{i+1} + 3u_i - 5u_{i-1} + u_{i-2}}{4\Delta x} = \frac{\partial u_i}{\partial x} + O(\Delta x^2). \quad (24)$$

Therefore, the space discretization is second-order accurate. Since the two-stage Runge–Kutta scheme is second order accurate in time, the overall numerical scheme is second-order accurate in space and time. Using a von Neumann stability analysis, it can be shown that the explicit two-stage scheme is stable when the CFL number  $c \Delta t / \Delta x$  is less than 1.

#### 4. Multi-dimensional characteristic boundary condition

A widely used absorbing boundary condition (ABC) for open boundaries is the so-called characteristic boundary condition, in which one-dimensional characteristic theory is derived and applied on the open boundary. Consider the one-dimensional Maxwell's equation in an arbitrary direction  $\mathbf{l} = (l_x, l_y, l_z)$ ,

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial l} = 0, \quad (25)$$

where

$$F = \begin{bmatrix} -\mathbf{l} \times \mathbf{H} \\ \mathbf{l} \times \mathbf{E} \end{bmatrix}. \quad (26)$$

The Jacobian matrix of  $F$  is then

$$G = \frac{\partial F}{\partial Q}. \quad (27)$$

Matrix  $G$  can be diagonalized as

$$G = R \Lambda R^{-1}, \quad (28)$$

where  $R$  is composed of right eigen-vectors of  $G$ , and  $\Lambda$  is a diagonal matrix including the eigenvalues. Eq. (25) can be further written as

$$R^{-1} \frac{\partial Q}{\partial t} + \Lambda R^{-1} \frac{\partial Q}{\partial l} = 0. \quad (29)$$

It can be easily shown that  $R$  is a function of the direction,  $\varepsilon$  and  $\mu$ , therefore Eq. (29) can be decoupled into

$$\frac{\partial W_i}{\partial t} + \lambda_i \frac{\partial W_i}{\partial l} = 0, \quad (30)$$

where  $W_i$  is one of the characteristic variables computed from

$$W = R^{-1} Q, \quad (31)$$

and  $\lambda_i$  is the  $i$ th eigenvalue. Eq. (30) says that  $W_i$  is a constant along the characteristic defined by

$$\frac{\partial l}{\partial t} = \lambda_i. \quad (32)$$

Traditionally, characteristic boundary conditions are implemented in the face normal or a coordinate direction. If a propagating wave is aligned with the face normal direction, the characteristic boundary condition generates nearly no reflection at the truncated boundary [15]. In this paper, a truly multi-dimensional characteristic boundary condition is developed which track

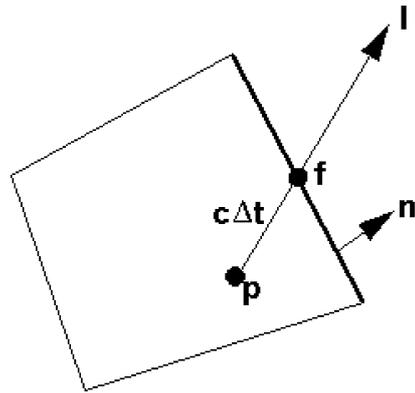


Fig. 3. Schematic of the multi-dimensional characteristic absorbing boundary condition.

the characteristics in the wave propagating direction, i.e.  $\mathbf{l} = \mathbf{E} \times \mathbf{H} / |\mathbf{E} \times \mathbf{H}|$ . This boundary condition is thus coined multi-dimensional characteristic boundary condition. A schematic of this boundary condition is shown in Fig. 3. We want to compute the field variable  $Q_f^{n+1}$  from the field variable  $Q^n$  in the interior domain. The characteristic variables associated with positive eigenvalue  $c$  are interpolated from the interior, i.e.

$$W_{i,p}^n = W_{i,f}^{n+1}, \quad (33)$$

where the position vector is computed from

$$\mathbf{r}_p = \mathbf{r}_f - c\Delta t \cdot \mathbf{l}, \quad (34)$$

where  $\Delta t$  is the time-step, and  $W_{i,p}^n$  is computed based on the linear cell-wise reconstruction presented earlier. The characteristic variables associated with  $-c$  are 0, indicating no incoming waves is present. The static waves associated with eigenvalue 0 are simply computed with  $Q_f^n$ . Then  $Q_f^{n+1}$  are obtained from

$$Q_f^{n+1} = RW_f^{n+1}. \quad (35)$$

Numerical tests indicated that the new multi-dimensional boundary condition performed much better than the traditional one-dimensional characteristic boundary condition. One demonstration example is shown in Fig. 4, which shows a plane wave propagating in free space after 20 cycles. The initial condition is an analytical plane wave at  $t = 0$ . Note that the multi-dimensional characteristic boundary condition produced far superior computational results.

## 5. Demonstration cases

### 5.1. Wave scattering by a conducting cylinder

Since analytical solutions exist for this problem, it is chosen as the first test case. The simulated electric sizes of the cylinder are  $ka = 1, 10,$  and  $50$  (where  $k$  is the wave number,  $a$  is the radius of the cylinder). For the case of  $ka = 1$ , a near body grid resolution of 60 ppw is used since a resolution of 20 ppw with 5 points per quadrant is not sufficient in resolving the geometry. The grid is then declustered to about 15 ppw near the open boundary. For the case of  $ka = 10$ , the near body grid resolution is 20 ppw, and the grid is then declustered to 10 ppw near the open boundary. For the case of  $ka = 50$ , the near body grid resolution is 26 ppw, and again the grid is declustered to 13 ppw near the open boundary. For all cases, the open boundary is located two wave lengths away from the geometry. The size of the computational domain and grid resolution were determined through extensive testing. The goal is to obtain accurate (error  $< 5\%$ ) near field data so that RCS can be extracted accurately. The computational grids and the computed surface currents are displayed in Figs. 5–7. In all the computations, a scattered wave formulation is employed instead of the total wave formulation. Because of that, the grid can be declustered away from the body without significant loss of accuracy in the near field solution. A constant

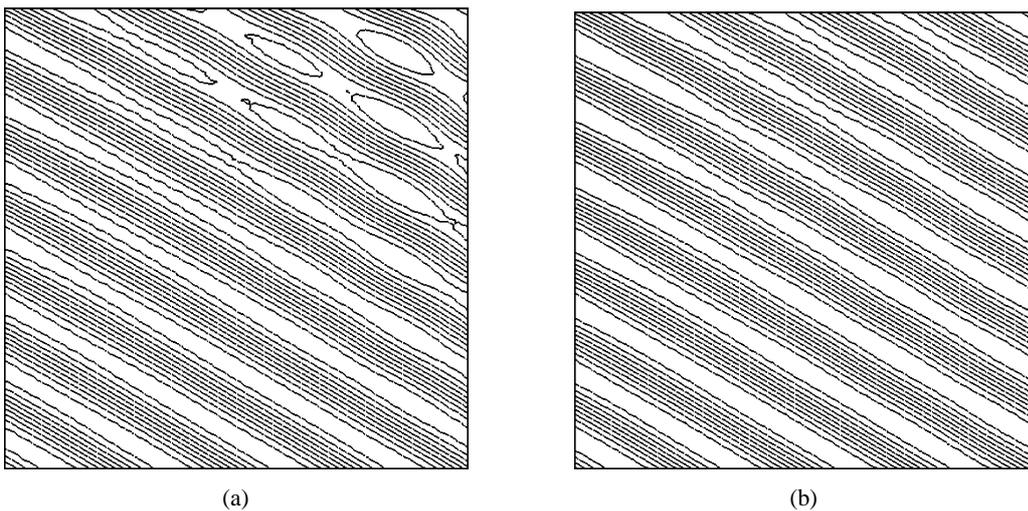


Fig. 4. Comparison of plane wave propagation with a one-dimensional (a) and multi-dimensional (b) characteristic boundary conditions.

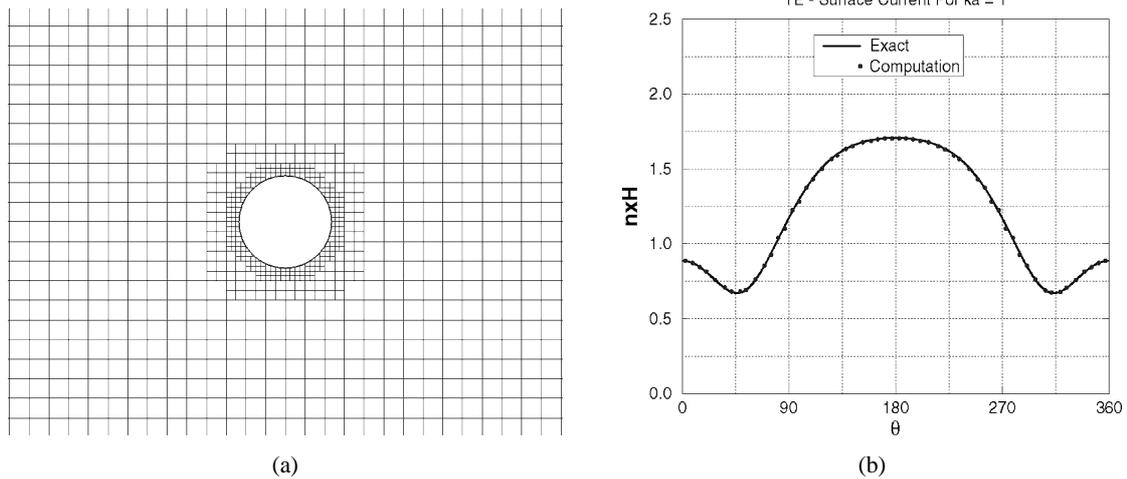


Fig. 5. Computational grid and surface current for TE  $ka = 1$ . (a) Computational grid. (b) Comparison of surface current.

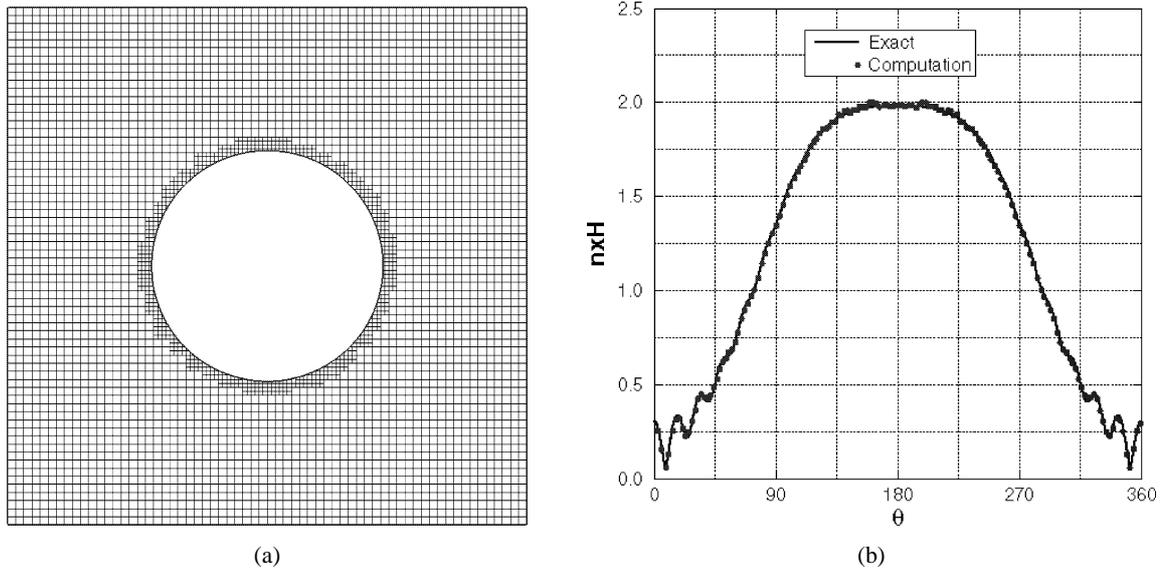


Fig. 6. Computational grid and surface current for TE  $ka = 10$ . (a) Computational grid. (b) Comparison of surface current.

CFL number of 1 was used in all the computations. The computations usually reached a periodic “steady state” in about 4–6 cycles. Note from Figs. 5–7 that the agreement between the analytical and computational surface currents was shown to be very good, indicating that the computational results with the current grid resolution are acceptable.

The case of  $ka = 10$  was also simulated with a triangular grid for comparison purpose. In order to make a fair comparison with the adaptive Cartesian

grid, the grid resolutions for the triangular grid at the cylinder surface and the outer boundary are similar. Such a triangular grid is shown in Fig. 8(a). The grid has 7133 points, 20,961 faces and 13,828 cells. In comparison, the adaptive Cartesian with similar grid resolutions has 4482 points, 8442 faces and 3960 cells. The problem with the triangular grid has 13,828 degrees-of-freedom (DOFs), while it only has 3960 DOFs on the Cartesian grid. It is expected the solution accuracy on both grids be similar. This is confirmed

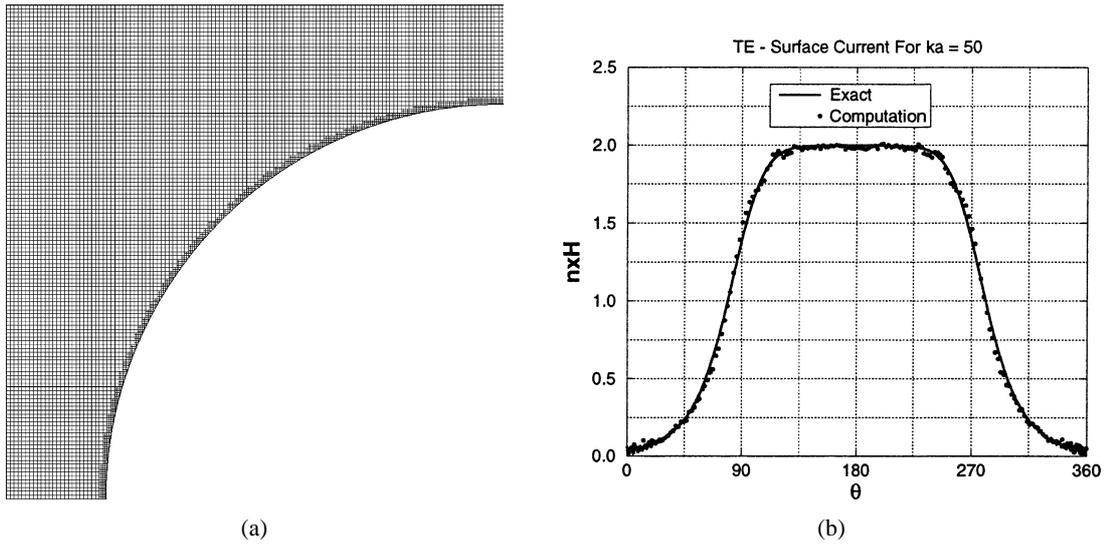


Fig. 7. Computational grid and surface current for TE  $ka = 50$ . (a) Computational grid. (b) Comparison of surface current.

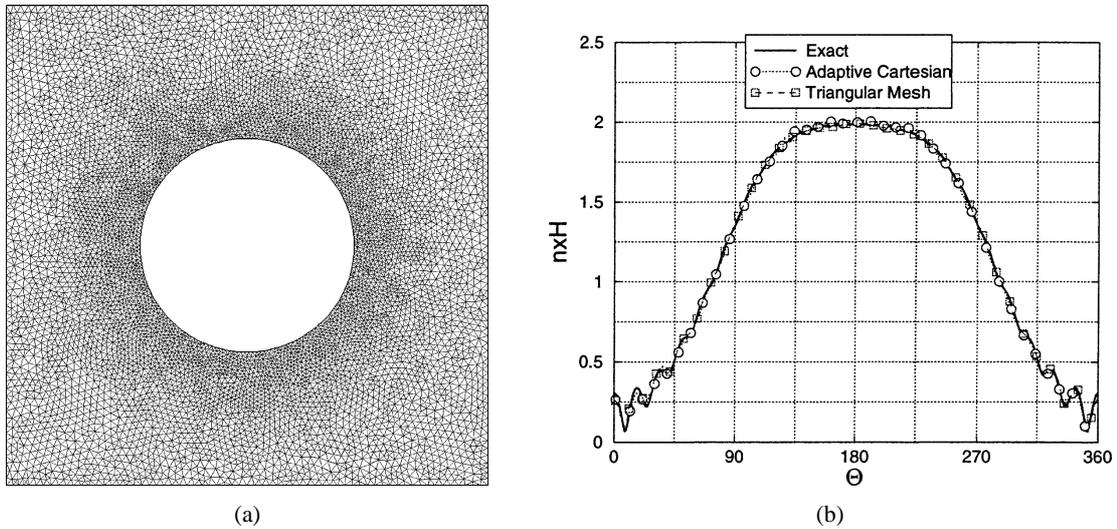


Fig. 8. Triangular grid and comparison of computed surface currents for the case of  $ka = 10$ . (a) Triangular grid. (b) Comparison of surface currents.

in Fig. 8(b), which presents the computed surface currents on both grids and the exact solution. Note that the computed solutions have similar quality. However, the simulation on the triangular grid took 3.3 times the CPU time on the adaptive Cartesian grid.

Another test was performed to see whether the simulation is CFL number dependent. The adaptive Cartesian grid shown in Fig. 6 was used with CFL = 0.5 and 1. The simulations were carried out for

six cycles. The histories of the  $z$  component of the magnetic induction vector at a near field point were plotted for CFL = 0.5 and 1 in Fig. 9, which clearly shows that the computed field is CFL-independent, and that cyclic “steady state” solutions were indeed obtained after about 4–5 cycles.

Finally the effectiveness of the MDC-ABC boundary condition was tested. In this test, two quadrilateral grids with different far-field locations were used. One

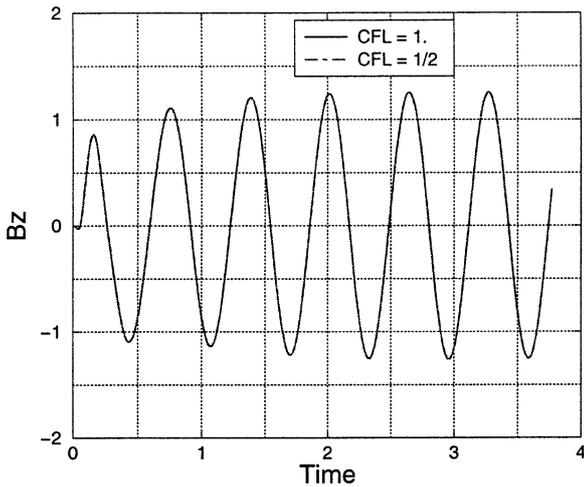


Fig. 9.  $B_z$  time histories computed with different CFL numbers at a near field point.

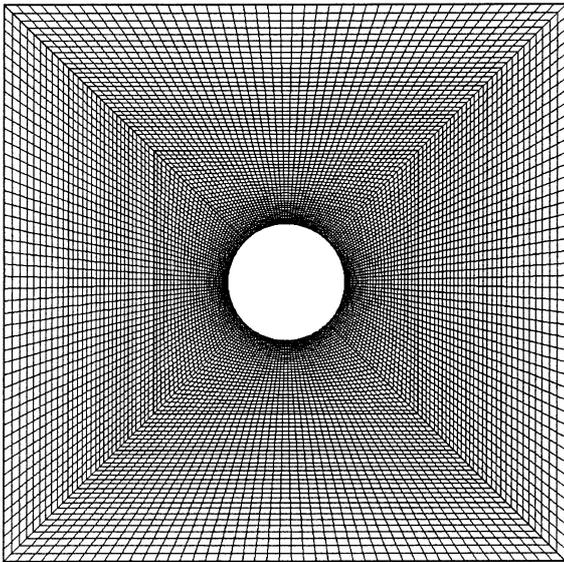


Fig. 10. A quadrilateral grid used to test the multi-dimensional ABC boundary condition.

grid (called Grid 1) has a far field boundary 2 wavelengths away from the cylinder surface, and the other one (Grid 2) has a far field boundary 6 waves away, as shown in Fig. 10. The computations were performed for 6 cycles on both grids. It is obvious that the wave reflections from the far field boundary of Grid 1 should reach the cylinder after 4 cycles, while the reflections on the far field boundary of Grid 2 cannot reach the

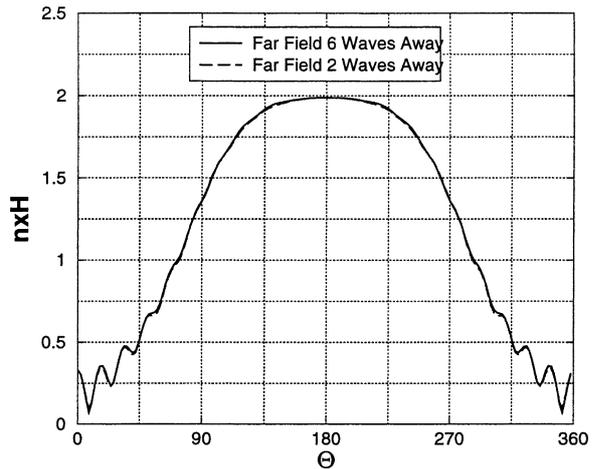


Fig. 11. Comparison of computed surface currents with two different locations of the far field boundary.

cylinder until the 12th cycle. The computed surface currents for both grids are compared in Fig. 11. Note that the difference is very small. In fact, the maximum relative difference between the computed surface currents is about 1.5%, which indicates that the MDC-ABC performs very well for this type of wave scattering problems.

### 5.2. Wave scattering by a conducting sphere

The three-dimensional validation case is plane wave scattering by a perfectly conducting sphere, for which an analytical solution also exists. The incident wave is of TE mode, with  $ka = 1$ , and propagates in the positive  $x$ -direction. The computational grid is shown in Fig. 12. The grid has a total of 48,267 cells. A grid resolution of 20 ppw is too coarse to resolve the sphere geometry. Therefore, a grid resolution of 64 ppw is used near the sphere, and the grid is declustered away from the sphere to about 8 ppw near the open boundary. The open boundary is two wavelengths away from the surface of the sphere. In the simulation, a scattered wave formulation is employed instead of the total wave formulation. Since the scattered fields approach zero away from the body, one can use coarse grid cells near the open boundary without compromising the accuracy of the computed fields near the body. A constant CFL number of 1 is used in the simulation, which corresponds to about 342 time steps per cycle. The com-

putation reached a periodic “steady state” after 6 cycles. The analytical electric intensity field is compared with the current prediction in Fig. 13 on several cutting planes. Note that the agreement between the computational and analytical solutions is very good.

### 5.3. Wave scattering by a missile

As a demonstration of the current method in handling complex geometries, the case of wave scattering by a conducting cruciform missile was simulated. The geometry of the missile is quite complex, having four fins. The missile geometry was originally defined in trimmed NURBS patches. The geometry surface was

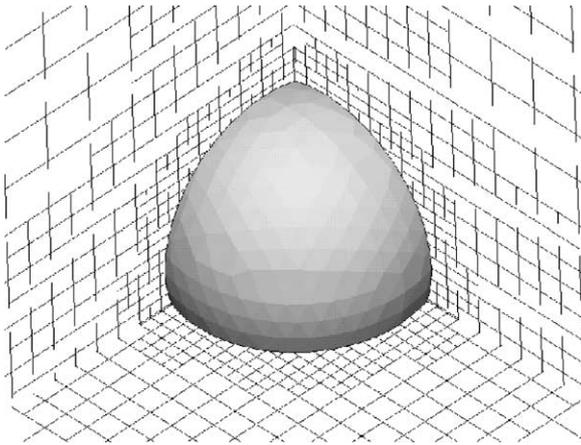
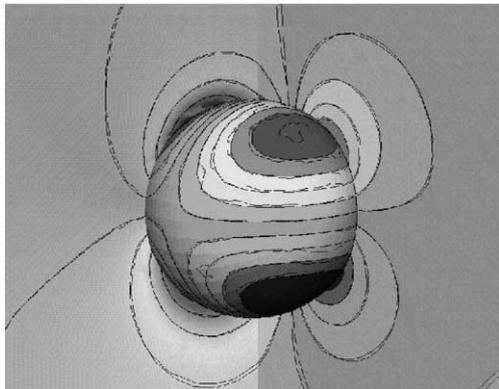
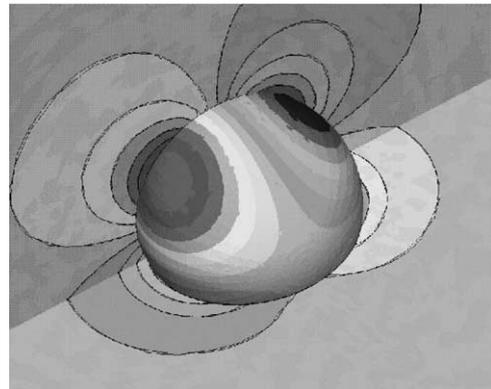


Fig. 12. Adaptive Cartesian grid around a sphere ( $ka = 1$ , TE polarization).



(a)



(b)

Fig. 13. Comparison of computed and analytical electric field components. (a)  $x$ -component. (b)  $z$ -component.

automatically triangulated given a surface grid resolution. The generation of the volume adaptive Cartesian grid was nearly automatic. All the user needs to input are: the size of the Cartesian grid domain, the minimum grid cell size near the geometry, and the maximum grid cell size near the open boundary. A computational grid was then generated without any user interferences, with automatic cell-cutting and cell-merging. The adaptive Cartesian grid and the surface geometry for the missile is shown in Fig. 14. The frequency of the incoming wave is 72 MHz. The length of the missile is about 11 wavelengths. The plane wave is of TE polarization, and propagates in positive  $x$ -direction (missile length direction). The grid resolution near the geometry is 20 ppw, and the grid is gradually declustered away from the body. The grid has a total of 139,484 cells. The unsteady simulation reached a periodic “steady” state after only four to five cycles, with about 4–5 hours of CPU time on a DEC Alpha machine. The computed  $z$ -component of the magnetic field is shown in Fig. 15. Other useful information such as radar cross sections can be extracted from the field solutions if necessary.

## 6. Conclusions

A FV-TD CEM solver supporting arbitrary grid including structured, unstructured and adaptive Cartesian grids has been developed. It is argued that adaptive Cartesian grid is the optimum grid topology to handle complex geometries considering both accuracy

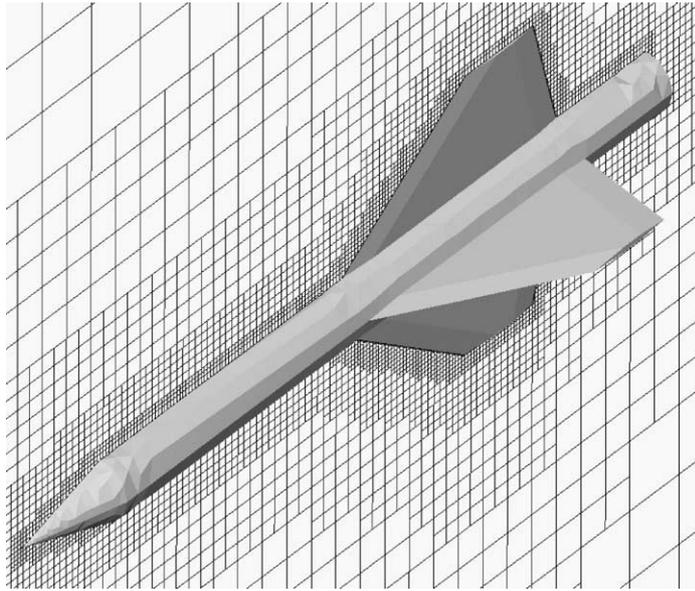


Fig. 14. Computational grid for plane wave scattering by a cruciform missile at 72 MHz.

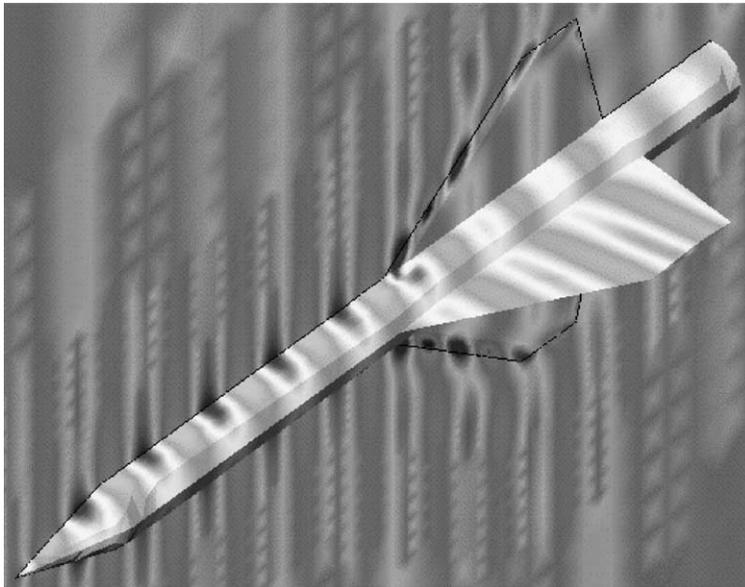


Fig. 15. Computed  $z$ -component of the magnetic field.

and efficiency. A new multi-dimensional characteristic boundary condition was developed, which was shown to be far superior than the conventional one-dimensional counterpart. Several validation cases confirm the capability and accuracy of the current CEM solver.

#### **Acknowledgement**

Helpful discussions with Dr. Joe Shang of Wight Lab., Drs. N. Hariharan and R. Chen are gratefully acknowledged.

## References

- [1] A. Taflove, *Computational Electrodynamics—The Finite Difference-Time-Domain Method*, Artech House, Boston, 1995.
- [2] R.F. Harrington, *Field Computation by Moments Methods*, MacMillan, New York, 1968.
- [3] K.S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations isotropic media, *IEEE Trans. Antennas Propagation* AP-14 (4) (1966) 302–307.
- [4] A. Taflove, M.E. Brodwin, Numerical solution of steady-state electromagnetic scattering problems using the time-dependent Maxwell's equations, *IEEE Trans. Microwave Theory Techn.* MT-T-23 (8) (1975) 623–630.
- [5] R. Rolland, L. Simpson, K. Kunz, Finite-difference analysis of EMP coupling to lossy dielectric structures, *IEEE Trans. Electromagnetic Comp.* EMC-22 (3) (1980) 203–209.
- [6] D.E. Merewether, R. Fisher, F.W. Smith, On implementing a numeric Huygen's source scheme in a finite difference program to illuminate scattering bodies, *IEEE Trans. Nuclear Sci.* NS-27 (6) (1980) 1829–1833.
- [7] K.S. Kunz, R.J. Luebbers, *The Finite Difference Time Domain Method for Electromagnetics*, CRC Press, Boca Raton, FL, 1993.
- [8] R. Holland, Finite difference solutions of Maxwell's equations in generalized nonorthogonal coordinates, *IEEE Trans. Nuclear Sci.* (1983) 4589–4591.
- [9] M. Fusco, FDTD algorithm in curvilinear coordinates, *IEEE Trans. Antennas Propagation* (1992) 1068–1075.
- [10] K.S. Yee, J.S. Chen, A.H. Chang, Conformable finite-difference time-domain (FDTD) with overlapping grids, *IEEE Trans. Antennas Propagation* (1992) 1068–1075.
- [11] V. Shankar, W. Hall, A.H. Mohammadian, A CFD-based finite-volume procedure for computational electromagnetics-interdisciplinary applications of CFD methods, *AIAA-89-1987-CP*, 1989.
- [12] A.H. Mohammadian, V. Shankar, W. Hall, Computations of electromagnetic scattering and radiation using a time-domain finite volume discretization procedure, *Comput. Phys. Comm.* 68 (1991) 175–196.
- [13] J.S. Shang, R.M. Fithen, A comparative study of numerical algorithms for computational electromagnetics, *AIAA-94-2410*, 1994.
- [14] V. Shankar, W.F. Hall, A. Mohammadian, C. Rowell, Science and supercomputing in the aerospace industry, *AIAA-93-4674*, 1993.
- [15] J.J. Shang, Characteristic-based algorithms for solving the Maxwell equations in the time domain, *IEEE Antennas Propagation Magazine* 37 (3) (1995).
- [16] Y. Liu, A generalized finite volume algorithm for solving the Maxwell equations on arbitrary grids, in: *Proceedings of 10th Annual Review of Progress in Applied Computational Electromagnetics*, 1994.
- [17] K.S. Yee, J.S. Chen, The finite-difference time-domain (FDTD) and the finite-volume time-domain (FVTD) methods in solving Maxwell's equations, *IEEE Trans. Antennas Propagation* 45 (3) (1997).
- [18] W.F. Hall, V. Shankar, S. Palaniswamy, Algorithmic aspects of wave propagation through stretched unstructured cells for problems in computational electromagnetics, *AIAA-97-2089*, 1997.
- [19] M.J. Berger, R.J. LeVeque, An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries, *AIAA-89-1930*, in: *9th Computational Fluid Dynamics Conference Buffalo, New York, June 1989*.
- [20] D. DeZeeuw, K. Powell, An adaptively refined Cartesian mesh solver for the Euler equations, *AIAA-91-1542*, 1991.
- [21] S.A. Bayyuk, K.G. Powell, B. van Leer, A simulation technique for 2D unsteady inviscid flows around arbitrarily moving and deforming bodies of arbitrarily geometry, *AIAA Paper 93-3391-CP*, 1993.
- [22] Z.J. Wang, A quadtree-based adaptive Cartesian/quad grid flow solver for Navier–Stokes equations, *Comput. Fluids* 27 (4) (1998) 529–549.
- [23] Y. Liu, Fourier analysis of numerical algorithms for the Maxwell equations, *J. Comput. Phys.* 124 (1996) 396–416.
- [24] Y. Liu, A 3D high-order unstructured finite-volume algorithm for solving Maxwell's equations, in: *1995 Digest, USNC/URSI Radio Science Meeting*, 1995.
- [25] J.A. Bonet, J. Peraire, An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problem, *Internat. J. Num. Methods Engrg.* 31 (1991) 1–17.