 cmes, vol.5, no.5, pp.383-393, 2004

Parallel 3D Time Domain Electromagnetic Scattering Simulations on Unstructured Meshes

O. Hassan[1], K. Morgan, J. Jones, B. Larwood and N. P. Weatherill

Abstract: A numerical procedure for the simulation of 3D problems involving the scattering of electromagnetic waves is presented. As practical problems of interest in this area often involve domains of complex geometrical shape, an unstructured mesh based method is adopted. The solution algorithm employs an explicit finite element procedure for the solution of Maxwell's curl equations in the time domain using unstructured tetrahedral meshes. A PML absorbing layer is added at the artificial far field boundary that is created by the truncation of the physical domain prior to the numerical solution. The complete solution procedure is parallelised and several large scale examples are included to demonstrate the computational performance that may be achieved by the proposed approach.

keyword: Time domain EM scattering, finite element method, tetrahedral meshes, parallelisation, large scale simulations.

1 Introduction

The accurate simulation of 3D electromagnetic scattering problems of current industrial interest, in realistic time scales, poses major computational challenges. We will address some of these challenges in the context of problems involving the interaction between waves, generated by a source in the far field, and a scatterer of general shape. Difficulties associated with mesh generation are reduced by adopting the unstructured mesh approach, with a fully automatic unstructured mesh generation procedure [George (1991), Peraire, Peiró, and Morgan (1999), Weatherill and Hassan (1994)]. This is confirmed by experiences in the aerospace industry, where the unstructured mesh approach has already been widely embraced, particularly in the area of computational aerodynamics [Hills (1996)]. Following this philosophy requires the identification of a suitable unstructured mesh based solution algorithm and we employ a low order 3D time domain procedure. Several algorithms of this type have been proposed [Petitjean and Löhner (1992), Cioni, Fezoui, and Steve (1993), Darve and Löhner (1997), Morgan, Brookes, Hassan, and Weatherill (1998)], but the method followed is based upon the application of an explicit linear Taylor–Galerkin finite element procedure [Donéa (1984)] to Maxwell's curl equations. With this method, both the electric and magnetic fields are assumed to vary in a continuous piecewise linear fashion [Morgan, Hassan, and Peraire (1994), Morgan, Hassan, and Peraire (1996), Morgan, Hassan, Pegg, and Weatherill (2000)]. The non–reflective boundary condition, that must be imposed at the truncated far field boundary that is created to enable numerical simulation, is handled by surrounding the computational domain by an artificial perfectly matched layer (PML). The parameters in the PML equations are defined in such a manner that the amount of reflection from the far field boundary is decreased [Berenger (1994), Bonnet and Poupaud (1997)]. The use of the PML is found to lead to a significant reduction in computational costs compared to those associated with the use of traditional local absorbing boundary condition approximations. It is recognised that, with an algorithm of this type, care will need to be taken when modelling problems involving singularities [Mur (1994), Sun, Manges, Yaun, and Cendes (1995)] and that meshing requirements will need to take account of the possible effects of pollution error [Deraemaeker, Babuska, and Bouillard (1999)].

To enable the solution of large scale problems on current computer platforms, the complete simulation process is parallelised. The computational performance that can be achieved by the resulting capability is demonstrated by including the results of a number of scattering simulations involving plane single frequency incident waves.

[1] Civil & Computational Engineering Centre, University of Wales, Swansea SA2 8PP, Wales, U.K.

2 The Scattering Problem

2.1 The Governing Equations

Consider the simulation of scattering of single frequency plane incident electromagnetic waves by an obstacle that is surrounded by free space. It is assumed that the incident waves are produced by a general source located in the far field. In three dimensions, Maxwell's curl equations for a general linear isotropic material, of relative permittivity ε and relative permeability μ, can be written, using the summation convention, in the dimensionless form

$$\mu\frac{\partial H_j}{\partial t} = -\varepsilon_{jk\ell}\frac{\partial E_\ell}{\partial x_k} \qquad \varepsilon\frac{\partial E_j}{\partial t} = \varepsilon_{jk\ell}\frac{\partial H_\ell}{\partial x_k} \tag{1}$$

where the subscripts j,k,ℓ can take the values 1,2,3, the alternating tensor is denoted by $\varepsilon_{jk\ell}$ and $E = (E_1,E_2,E_3)^T$ and $H = (H_1,H_2,H_3)^T$ denote the electric and magnetic field intensity vectors respectively. The total E and H fields are split into incident and scattered components according to

$$E = E^i + E^s \qquad\qquad H = H^i + H^s \tag{2}$$

The incident fields, $E^i = \left(E_1^i,E_2^i,E_3^i\right)^T$ and $H^i = \left(H_1^i,H_2^i,H_3^i\right)^T$ are specified by the problem definition and automatically satisfy equations (1) in free space. A formulation expressed in terms of the scattered fields $E^s = (E_1^s,E_2^s,E_3^s)^T$ and $H^s = (H_1^s,H_2^s,H_3^s)^T$ is achieved by the substitution of equations (2) into equation (1). The resulting equations may be combined to produce the single equation

$$\frac{\partial U}{\partial t} + \frac{\partial F^k}{\partial x_k} = S \tag{3}$$

where

$$U = \begin{bmatrix} \mu H^s \\ \varepsilon E^s \end{bmatrix} \qquad S = \begin{bmatrix} (1-\mu)\dfrac{\partial H^i}{\partial t} \\ (1-\varepsilon)\dfrac{\partial E^i}{\partial t} \end{bmatrix} \tag{4}$$

and

$$F_j^k = \begin{cases} \varepsilon_{jk\ell}E_\ell^s & j = 1,2,3 \\ -\varepsilon_{(j-3)k\ell}H_\ell^s & j = 4,5,6 \end{cases} \tag{5}$$

It is apparent that $S \equiv 0$ in free space.

3 Numerical Solution Algorithm

An approximate solution to the scattering problem is obtained by using a two–step finite element Taylor–Galerkin procedure [Morgan, Hassan, and Peraire (1994)]– [Morgan, Hassan, Pegg, and Weatherill (2000)]. This procedure is notionally second order accurate in both time and space [Donéa (1984)].

3.1 Discretisation in time

The solution of equation (3) is advanced over one time step, from time level $t = t_m$ to time level $t = t_{m+1} = t_m + \Delta t$, in a two step fashion. In the first step, the solution at time $t_{m+1/2} = t_m + \Delta t/2$ is obtained by employing the forward difference approximation

$$\frac{U^{\{*\}} - U^{\{m\}}}{\Delta t/2} = S^{\{m\}} - \left.\frac{\partial F^k}{\partial x_k}\right|^{\{m\}} \tag{6}$$

where the superscript $\{m\}$ denotes an evaluation at time $t = t_m$ and the superscript $\{*\}$ denotes an evaluation at time $t = t_{m+1/2}$. The second step determines the solution at the end of the time step, by using the central difference approximation

$$\frac{U^{\{m+1\}} - U^{\{m\}}}{\Delta t} = S^{\{*\}} - \left.\frac{\partial F^k}{\partial x_k}\right|^{\{*\}} \tag{7}$$

The quantities required at time level $t_{m+1/2}$ are to be computed using the values obtained for $U^{\{*\}}$ in the first step of equation (6). In practice, following the spatial discretisation of the domain, equation (6) is applied directly, while the solution of equation (7) is obtained from an approximate variational formulation.

3.2 Discretisation in space

The computational domain is discretised into a general unstructured mesh of linear tetrahedral elements, using a Delaunay generator [Weatherill and Hassan (1994)]. Over each element E in the mesh, the solution $U^{\{m\}}$, the fluxes $F^{k\{m\}}$ and the source term $S^{\{m\}}$ are linearly interpolated, between the element nodal values, in the form

$$\begin{aligned} U_E^{\{m\}} &= \sum_{J\in E} N_J U_J^{\{m\}} \\ F_E^{k\{m\}} &= \sum_{J\in E} N_J F_J^{k\{m\}} \\ S_E^{\{m\}} &= \sum_{J\in E} N_J S_J^{\{m\}} \end{aligned} \tag{8}$$

Here N_J denotes the linear finite element shape function associated with node J and the summations extend over all nodes J of element E. In the computational implementation, equation (6) is used first to compute, for each element E in the mesh, an approximate solution at time $t = t_{m+1/2}$ according to

$$\frac{U_E^{\{*\}} - U_E^{\{m\}}}{\Delta t/2} = S_E^{\{m\}} - \left.\frac{\partial F_E^k}{\partial x_k}\right|^{\{m\}} \tag{9}$$

This results in a piecewise linear discontinuous approximation to the solution at the time level $t_{m+1/2}$. The solution at the time level t_{m+1} is obtained following a Galerkin approximate variational formulation [Morgan and Peraire (1998)] of equation (7). At a general node I, in a general material M, the resulting equation takes the form

$$\begin{aligned}\sum_{J\in I} M_{IJ}\frac{(U_J^{\{m+1\}} - U_J^{\{m\}})}{\Delta t} \\ = \int_M S^{\{*\}} N_I \mathrm{d}\Omega - \int_{\Gamma_M} \overline{F}_n^{\{*\}} N_I \mathrm{d}\Omega \\ + \int_M F^{k\{*\}} \frac{\partial N_I}{\partial x_k}\mathrm{d}\Omega\end{aligned} \tag{10}$$

where the summation is restricted to nodes $J \in M$. Here $\overline{F}_n$ denotes a normal flux on Γ_M, which is computed according to the boundary condition being simulated, and M_{IJ} denotes the entries in the consistent linear finite element mass matrix. Equation (10) may be readily solved by either lumping this matrix or by explicit iteration [Donéa and Giuliani (1981)].

3.3 Boundary conditions

3.3.1 Perfect Electrical Conductor and Material Interfaces

At the surface of a perfect electrical conductor (PEC), the boundary condition that is imposed is that the tangential component of the total electrical field should vanish. This requirement may be expressed in the form

$$n \wedge E^s = -n \wedge E^i \tag{11}$$

Here $\wedge$ denotes the vector product and n is the unit normal vector to the PEC surface.

Across a material interface, the tangential component of the total electric and the total magnetic field must both be continuous. These conditions can be expressed as

$$\begin{aligned} n \wedge (E^s_{(**)} - E^s_{(*)}) = 0 \\ n \wedge (H^s_{(**)} - H^s_{(*)}) = 0 \end{aligned} \tag{12}$$

In these equations, the subscripts $(*)$ and $(**)$ denote the field values on either side of the interface and n now represents the unit normal vector to the interface surface, in the direction from $(*)$ to $(**)$.

Both the PEC and the material interface boundary conditions are applied through the boundary integral term in equation (10), which means that they are imposed in a weak sense only. The approach that is adopted is to determine the entries in the normal flux $\overline{F}_n$ by employing a characteristic decomposition in the direction normal to the boundary [Morgan, Hassan, Pegg, and Weatherill (2000), Shankar, Hall, Mohammadian, and Rowell (1993)].

3.3.2 Far Field Boundary and the PML

The incident wave is assumed to be generated by a source in the far field. This means that, when the infinite physical domain is truncated to enable numerical simulation of the scattering problem, the correct boundary condition to be imposed at the outer computational boundary is the requirement that the scattered field should only consist of outgoing waves only. The modelling of this truncated far field boundary condition is achieved by the addition of a PML [Berenger (1994)] to the exterior of the computational domain. In the examples presented here, the truncated outer boundary is always taken to be a regular hexahedron and the PML is discretised using a structured mesh of tetrahedral elements. The formulation which is implemented follows the work of Bonnet and Poupaud [Bonnet and Poupaud (1997)], in which the governing equations in the PML are considered in the form

$$\begin{aligned}\varepsilon\frac{\partial E_j^s}{\partial t} = \varepsilon_{jk\ell}\frac{\partial H_\ell^s}{\partial x_k} - \sigma_{[|j+1|]}E_j^s \\ - \left(\sigma_{[|j+2|]} - \sigma_{[|j+1|]}\right)\overline{E}_j^s\end{aligned} \tag{13}$$

$$\begin{aligned}\mu\frac{\partial H_j^s}{\partial t} = -\varepsilon_{jk\ell}\frac{\partial E_\ell^s}{\partial x_k} - \sigma_{[|j+1|]}H_j^s \\ - \left(\sigma_{[|j+2|]} - \sigma_{[|j+1|]}\right)\overline{H}_j^s\end{aligned} \tag{14}$$

The new variables, $\overline{H}_j$ and $\overline{E}_j$ for $j = 1,2,3$, are obtained by the solution of the additional equations

$$\varepsilon \frac{\partial \overline{E}_j^s}{\partial t} = -\sigma_{[|j+2|]} \overline{E}_j^s - \frac{\partial \overline{H}_{[|j+1|]}^s}{\partial x_{[|j+2|]}} \tag{15}$$

$$\mu \frac{\partial \overline{H}_j^s}{\partial t} = -\sigma_{[|j+2|]} \overline{H}_j^s + \frac{\partial \overline{E}_{[|j+1|]}^s}{\partial x_{[|j+2|]}} \tag{16}$$

In these equations

$$[|j|] = \begin{cases} j & \text{for } j = 1,2,3 \\ j-3 & \text{for } j = 4,5 \end{cases} \tag{17}$$

Based upon the results of numerical testing, the variation of the material parameter σ_j through the PML is defined as

$$\sigma_j = \frac{18 X_j^3}{\lambda^4} \tag{18}$$

where X_j denotes the distance, in the x_j direction, from a point in the PML to the truncated far field boundary and λ is the wavelength of the incident wave. Following appropriate redefinition of the quantities U, F^k and S, the form of equation (3) remains valid and the solution within the PML can again be obtained by using equations (9) and (10).

3.4 Computational details

This algorithm is stable provided that the selected time step size satisfies a standard CFL type condition [Morgan and Peraire (1998)]. For the single frequency incident waves considered here, the solution is advanced in time through a prescribed number of cycles of the incident wave until steady periodic conditions are achieved. A closed surface, completely enclosing the scatterer, is constructed and a further cycle is then computed during which time the variation of the solution at nodes lying on this surface is monitored. The amplitude and phase of the scattered electric and magnetic field components at these nodes are recorded for use in the computation of the radar cross section (RCS) [Balanis (1989), Morgan, Hassan, and Peraire (1994)]. It should be noted that the algorithm allows for the appearance of a spurious steady non–zero mode, which can be removed by a simple post–processing of the computed solution [Kangro and Nicolaides (1997)].

4 Parallel Implementation

This basic algorithm has already been validated for a number of different scattering problems. However, the nature of the algorithm means that the required mesh size will increase rapidly when the method is applied to the solution of problems involving the electrically large scatterers which arise when realistic frequencies and geometries are considered. Such simulations will require the use of significant computational resources and, in this case, the use of parallel computers becomes essential. It should be noted that the success of this route will require not only a parallel implementation of the basic Maxwell equation solver but, in addition, the effective parallelisation of the mesh generation and solution visualisation stages.

4.1 Mesh generation

The approach adopted for parallel mesh generation is based upon a geometrical partitioning of the domain [Weatherill, Hassan, Morgan, Jones, and Larwood (2001)]. The complete domain is divided into a set of smaller sub–domains and a mesh is generated independently in each sub–domain. The combination of the sub–domain meshes produces the mesh for the complete domain. A manager/worker model is employed in which the initial work is performed by the manager, before distributing the mesh generation tasks to the workers. The entire procedure can be divided into four separate stages: [1] Starting from a surface triangulation, the domain is partitioned using a geometrical partitioning scheme into N sub–domains. [2] The sub–domain boundaries are mapped into two dimensions and a 2D advancing front generator is employed to mesh the mapped surfaces. The triangulations of the inter–domain boundaries are mapped back to three dimensions. [3] A dynamic load balancing scheme is employed to generate meshes in each sub–domain. This scheme uses the fact that the proposed strategy involves no communication between processors so that more than one domain may be allocated, in turn, to each available processor. The number of domains given to each processor depends on the workload required for each sub–domain. [4] The mesh is post–processed, by node smoothing on inter–domain boundaries, and the inter–domain communication table is constructed.

The structure of the parallel grid generator is a single pro-

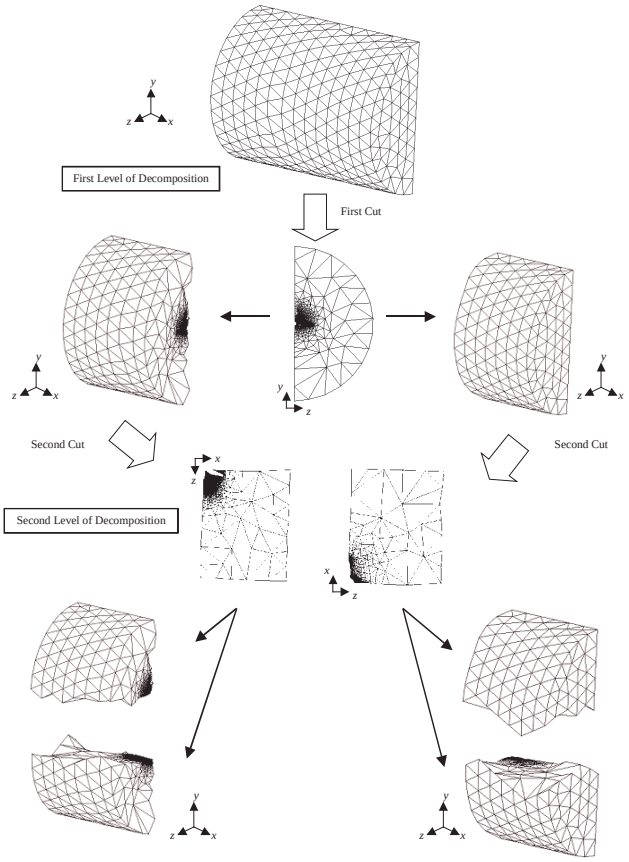


Figure 1 : Illustration of the parallel isotropic mesh generation process using four partitions

gram multiple data (SPMD) model. Sub–domain boundary data has to be passed from the manager to worker processors and sub–domain meshes passed back to the manager. MPI is used as the message passing library. Figure 1 illustrates the generation of an isotropic mesh following the decomposition of a domain into four partitions.

The structured mesh within the PML region is constructed using an advancing layer technique [Hassan, Morgan, Probert, and Peraire (1996)], which is a variant of the advancing front method and which generates successive layers of elements. For the first layer, the surface mesh at the truncated far field boundary forms the initial front. Lines normal to the surface are constructed at each point in the front and new points are generated on these lines. Tetrahedra are formed by connecting these points to the surrounding triangular faces. For this particular geometrical configuration, it is apparent that no checking for front intersection is required. Following the generation of the first layer of elements, succeeding layers are then generated similarly. As the only information required to generate a new element, and thus a new layer, is the front, or the top surface of the previous layer, it is prudent not to store the element connectivity matrix. This allows the memory required to be reduced to simply that corresponding to the size of the current front. Indeed, with careful programming, it is possible to evaluate the element connectivity, on the fly, by storing the node from which each new node is generated. This implementation ensures that there are no bottle necks in the process of mesh generation.

4.2 Mesh Partitioning

There are a number of different approaches available for serially decomposing a given unstructured mesh. However, for the current application, it is envisaged that the mesh data sets will be too large to load onto one processor. Therefore, the partitioning process has to be parallelised and distributed amongst the processors at all times. The present implementation utilises the ParMetis library for the partitioning [Karypis and Kumar (1998)]. This procedure produces high quality partitions in a fast, robust and parallel manner. However ParMetis operates on an edge based data structure, which means that the mesh had to be represented as a set of edges. One method of doing this is to create the dual of the mesh, where the nodes represent the elements and the edge (E_1, E_2) is present if the two elements E_1 and E_2 are adjacent. An alternative method is to create an edge based representation of the original element edges. Using the dual of the mesh has the advantage of automatically producing an element based partitioning, whereas the edge based representation of the mesh produces elements split across partitions. However, in the edge based representation, the number of edges is approximately the same as the number of elements, whereas the mesh dual approach results in approximately twice the number edges as elements. For this reason, it is more efficient to use the edge based representation in the partitioning process. This results in a partition number being assigned to each node, with the partition number for each element taken to be the lowest partition number of the all the nodes of the element.

In order to minimise memory usage during the partitioning process, the elements are read from disk twice. The first time is for the construction of the ParMetis edge based data structure and the element information is dis-

carded before ParMetis is executed. When ParMetis has finished, the elements, vertices and boundary faces are read from disk again and placed in their respective partitions. Before the equation solver is run on these data sets, each processor performs a bandwidth minimisation procedure on its sub–domain in order to improve cache use and, hence, to increase the performance of the solver. This is achieved by the application of a reverse Cuthill–McKee procedure [Cuthill and McKee (1969)].

4.3 Parallel Solution Procedure

In the parallel implementation of the solution algorithm, elements are owned by only one domain and are not duplicated, while points are owned by one domain and are duplicated. This strategy enables data locality to be achieved during the gather process, from points to elements, and the scatter process, from elements to points, and hence there is no need to communicate. For each time step, the interface nodes obtain contributions from more than one domain. One strategy for achieving this is to form two groups of elements, the first group containing all the interface elements and the second group containing all the interior elements. The element contributions from the first group are evaluated first. At the end of this stage, communication of these contributions takes place as the contributions from the elements in the second group are evaluated. The updated solution is sent back before the start of the next time step. However, for large meshes, the element grouping will result in inefficient use of the cache. It is, therefore, deemed to be more efficient to keep the elements in one group and to communicate the element contribution to the interface nodes at the end of the element loop. In this case, the parallel implementation of the solver is broken down into three main steps: [1] Compute the element contribution to all nodes in the each sub–domain. [2] Initiate non–blocking sends and receives to communicate the values of the inter–domain nodes. [3] Compute the updated solution for nodes in each sub–domain.

4.4 RCS Computation

The computation of the RCS requires that an integral over a closed surface enclosing the scatterer be evaluated for each viewing angle. In the present implementation, the surface is taken to be the surface of the scatterer and, consequently, only information on this surface is required. Furthermore, the computation of the RCS for one viewing angle is independent of the computation for any other viewing angle, so that the RCS computation can also be paralleized in the following manner: [1] Discard the volume mesh and recombine the surface mesh

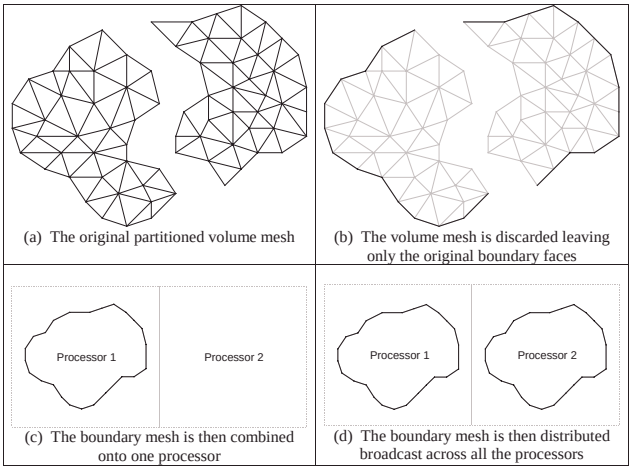


Figure 2 : Illustration of the process of evaluating the RCS in parallel

data into a global surface mesh. [2] Communicate the recombined global surface mesh to each processor. [3] The number of viewing angles required is then divided equally between the available processors.

A schematic illustration of the major steps employed in the process of performing the RCS computation in parallel is given in Figure 2.

5 Numerical Examples

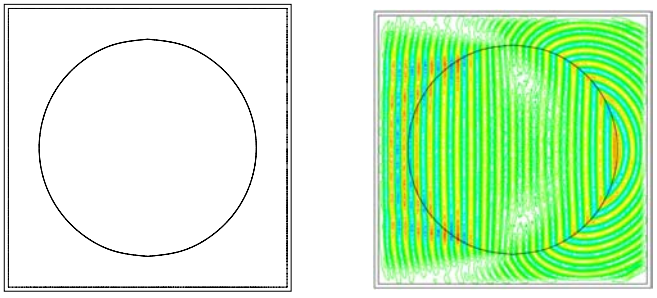
5.1 PEC Sphere

The first example involves the scattering of a plane single frequency incident wave by a perfectly conducting sphere of diameter $D = 15\lambda$. This case is used as a bench mark to study the effect of mesh resolution, and the distance of the PML from the scatterer, on the quality of the numerical results. In addition, the effect of utilising an explicit iteration scheme to solve the discretised equation system is also studied. Table 1 provides details of the various meshes employed for this example. In this table, row *a* gives the number of nodes per wavelength on the sphere for each mesh, while row *b* gives the corresponding number of nodes per wavelength at the far field. Rows *c* and *d* give, for each mesh, the distance, in wavelengths, of the PML from the sphere and the number of layers in the PML respectively. Rows *e* and *f* indicate,

Table 1 : Details of the meshes used for the simulation of scattering of a plane wave by a perfectly conducting sphere of diameter $D = 15\lambda$

	Mesh Number							
	1	2	3	4	5	6	7	8
a	25	25	25	20	20	15	15	10
b	25	15	10	15	10	15	10	10
c	2	1	1	1	0.5	0.5	0.5	1
d	25	15	10	15	10	15	10	10
e	230	123	90	71	57	60	32	8
f	3.5	1.9	1.4	1.1	0.9	1.	0.7	0.6

in millions, the number of elements and the number of nodes in each mesh. The solution was advanced through 30 cycles of the incident wave. Figure 3(a) shows the domain used for the simulation and the computed contours of E_2^s are displayed in Figure 3(b). A comparison between the computed and the exact RCS distributions is shown in Figure 4. This study indicates that, for this geometry and this frequency, a mesh spacing corresponding to 15 nodes per wavelength at the surface of the scatterer



(a) (b)

Figure 3 : Scattering by a PEC sphere of diameter $D = 15\lambda$ showing (a) the computational domain and (b) the computed contours of E_2^s

and varying linearly to a mesh spacing of 10 nodes per wavelength at the inner surface of the PML, with a PML region made of 10 layers and located at a distance of half a wavelength from the scatterer, is adequate to obtain excellent agreement with the analytical solution. Furthermore, it can be seen in Figure 5 that the use of mass matrix iteration provides a better quality solution than that computed with the lumped mass form of the scheme on a given mesh. For this reason, the remaining simulations in this paper are performed using two mass matrix iterations per time step. It should be noted, however, that this iterative procedure imposes a more stringent stability limitation which results in an increase in the execution time compared with the lumped mass implementation. For problems of this type, the PML is found to be

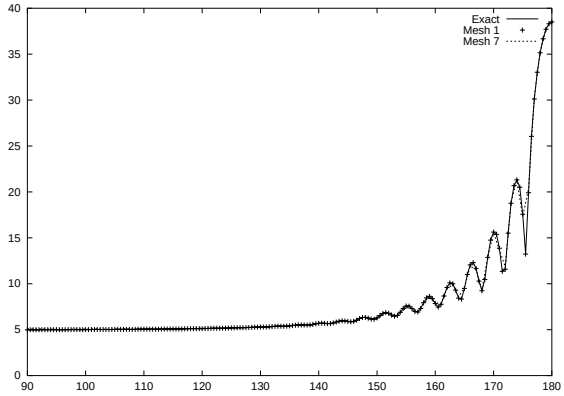


Figure 4 : Scattering of a plane wave by a PEC sphere of diameter $D = 15\lambda$ showing the comparison between the exact and the computed RCS distributions

considerably more efficient than methods based upon the use of a traditional local absorbing boundary condition, resulting here in a reduction, by about a factor of 5, in the number of nodes required.

5.2 Coated PEC Sphere

The second example involves scattering of a plane single frequency incident wave by a coated perfectly conducting sphere. The sphere diameter $D = 3\lambda$ and the dielectric coating is of thickness $t = 0.25D$. The coating is characterised by the material properties $\varepsilon = 2.56, \mu = 1$. The mesh employed in the region between the sphere and the far field boundary consists of 3 296 694 tetrahedra and 599 399 nodes. The structured PML region contains 820 380 tetrahedra and 149 160 nodes. A cut through this mesh is shown in Figure 6(a). The solution is advanced through 60 cycles of the incident wave and the computed contours of E_2^s are displayed in Figure 6(b). The exact

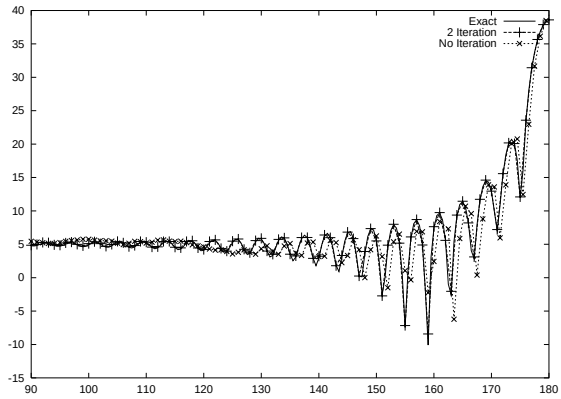
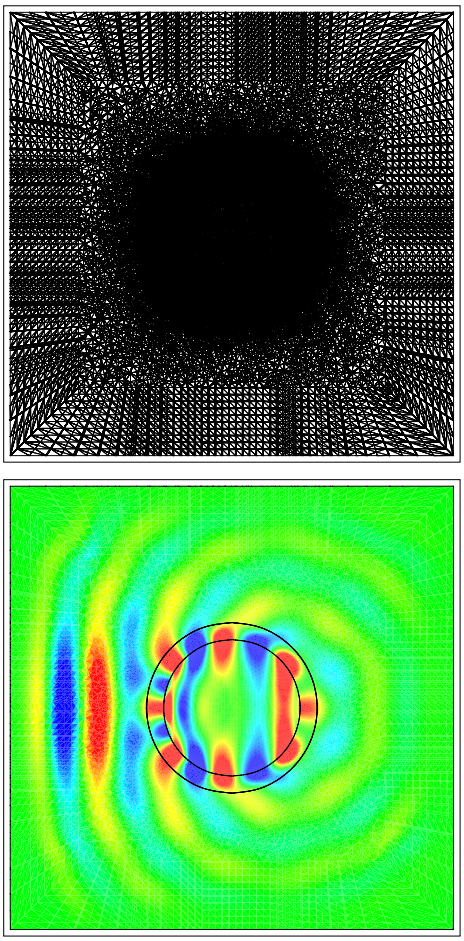


Figure 5 : Scattering by a PEC sphere of diameter $D = 15\lambda$ showing the comparison between the exact and the computed RCS distributions achieved using both the lumped mass and the mass matrix iteration schemes

and the computed distribution of the scattering width are seen to be in very good agreement in Figure 7.

5.3 PEC Trihedral Object

To illustrate the predictive capability of the procedure, two additional examples are considered. The first involves the simulation of scattering by a PEC trihedral object. Geometrically, the object dimensions are 1.36*0.6*0.2 *m* and the thickness of the walls is 0.01 m. This example has been used to evaluate the performance of the parallel implementation of the system. Two meshes were generated using various number of partitions and various number of processors. The first mesh is considered as being suitable for a 10GHz calculation and consists of approximately 100 million elements and 18 million nodes. The mesh was generated using four *R*14000 processors, with the domain divided initially into 8 partitions. The time taken to perform the complete mesh generation was 4 hours with a peak memory requirement of 1.5Gb. The mesh was generated again by partitioning the domain into 16 partitions. For this case, the time taken to generate the mesh was 1.4 hours with a peak memory requirement of 0.8Gb. The second mesh is considered as being suitable for a 20GHz simulation and consists of approximately 980 million elements and 178 million nodes. The domain was split into 64 and 128 partitions and the mesh was generated using 16 processors. The mesh generation required 8 hours and the



(a)

(b)

Figure 6 : Scattering of a plane wave by a coated PEC sphere of diameter $D = 3\lambda$ showing (a) a cut through the mesh and (b) the computed contours of E_2^s

maximum memory requirement was 1.8 Gb for the 64 partition mesh. With 128 partitions, the mesh generation time was reduced to 6 hours and the maximum memory requirement was 1Gb. Since the main drive behind the parallel implementation is to allow large meshes to be generated on moderate computers, it can be seen that the maximum amount of memory required for the generation is in fact decreased by increasing the number of partitions.

A 10GHz simulation has been attempted and the solution is advanced through 180 cycles of the incident wave. The computed contours of H_3^s are shown in Figure 8 and the computed distribution of the scattering width is displayed in Figure 9. To evaluate the performance of the parallel

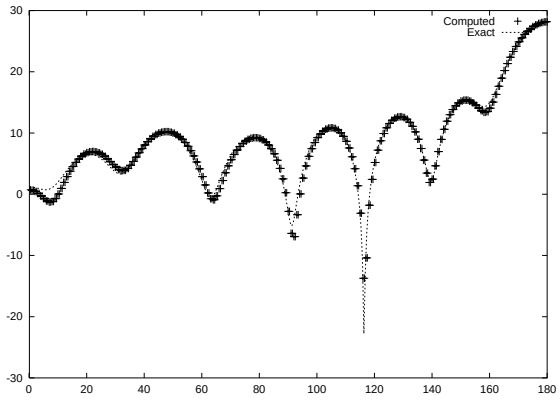


Figure 7 : Scattering of a plane wave by a coated sphere of diameter $D = 3\lambda$ showing the comparison between the exact and computed distributions of the scattering width

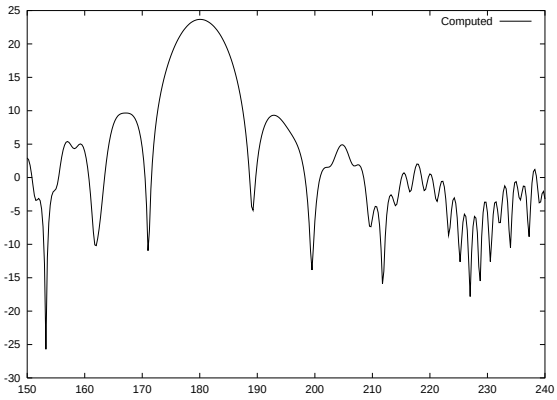


Figure 9 : Scattering of a 10GHz plane wave by a PEC trihedral object showing the predicted distribution of the RCS

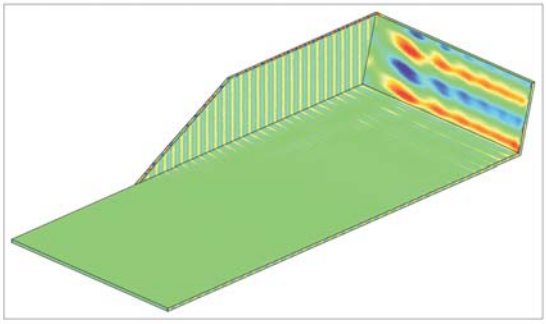


Figure 8 : Scattering of a 10Ghz plane wave by a PEC trihedral object showing contours of H_3^s

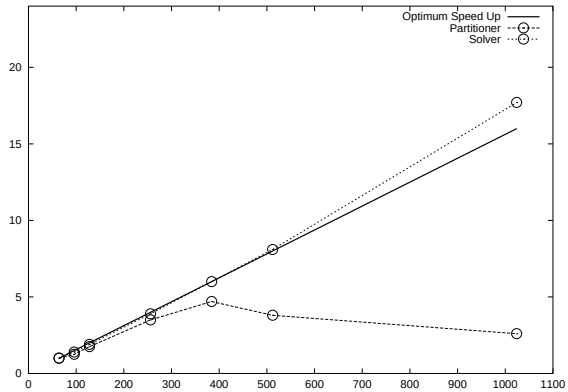


Figure 10 : Performance of the parallel preprocessor and the parallel solver on a CRAY T3E

solution algorithm, the scalability of the implementation on a CRAY T3E was investigated for this case, with the number of processors utilised being gradually increased from 64 to 1024. For each computation, the parallel preprocessor and the solver were executed on the same number of processors. It can be seen from Figure 10 that super linear speed up is achieved for the parallel solver. However, for the preprocessor, the speed up achieved degrades when more than 256 processors are used. This is due to the fact that the time required to partition the mesh on a large number of processors is very small and the process is then dominated by the I/O time.

5.4 PEC Aircraft

The final example uses the procedure to simulate the scattering of a plane wave by a PEC aircraft. The aircraft length is 10 wavelengths and the mesh employed consists of approximately 7.2 million elements and 1.35 million nodes. The PML is located at a distance of one half wavelength from the aircraft and has a total thickness equal to one wavelength. The PML region, consisting of 10 layers of elements, has approximately 1.4 million elements and 0.27 million nodes. The solution was advanced for 40 cycles and the computed contours of H_3^s on a cut through the mesh and on the aircraft surface are shown in Figure 11. The computed distribution of the RCS is displayed in Figure 12.

6 Conclusions

A numerical procedure that enables the parallel simulation of three dimensional electromagnetic scattering problems using automatically generated unstructured tetrahedra meshes has been presented. The solution algorithm employs a scattered field formulation and a two step Taylor–Galerkin time stepping scheme. The truncated far field boundary condition is imposed by the addi-

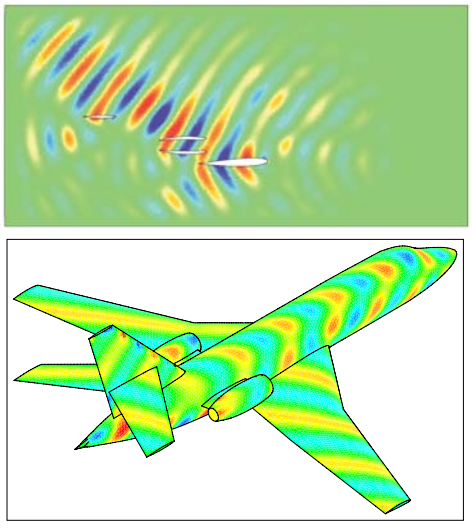


Figure 11 : Scattering of a plane wave by a PEC aircraft showing computed contours of H_3^s on (a) a cut through the mesh and (b) on the aircraft surface

tion of a PML. Parallel mesh generation is accomplished by a Delaunay procedure, following a geometrical partitioning of the domain. A number of computationally challenging examples have been included to demonstrate the numerical performance of the proposed procedure.

Acknowledgement: The authors acknowledge the support provided by the USARDSGUK (Technical Monitor: Dr R. Namburu, US Army Research Laboratory).

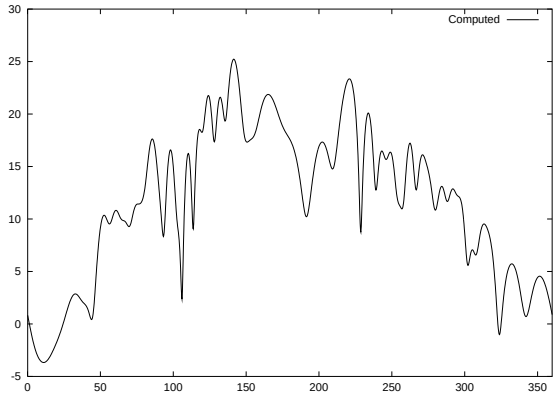


Figure 12 : Scattering of a plane wave by a PEC aircraft showing the predicted distribution of the scattering width

References

Balanis, C. A. (1989): *Advanced Engineering Electromagnetics*, Wiley: New York.

Berenger, J. P. (1994): A perfectly matched layer for free–space simulation in finite–difference computer codes, *Journal of Computational Physics*, 114: 185–200.

Bonnet, F.; Poupaud, F. (1997): Berenger absorbing boundary condition with time finite–volume scheme for triangular meshes, *Applied Numerical Mathematics*, 25: 333–354.

Cioni, J. P.; Fezoui, L.; Steve, H. (1993): A parallel time–domain Maxwell solver using upwind schemes and triangular meshes, *Impact of Computing in Science and Engineering*, 5: 215–247.

Cuthill, E.; McKee, J. M. (1969): Reducing the bandwidth of sparse symmetric matrices, *ACM Publication P69: Proceedings of the 24th National Conference of the Association of Computing Machines*, New York, 157–172.

Darve, E.; Löhner, R. (1997): Advanced structured–unstructured solver for electromagnetic scattering from multimaterial objects, *AIAA Paper 97–0863*.

Deraemaeker, A.; Babuska, I.; Bouillard, P. (1999): Dispersion and pollution of the FEM solution for the Helmholtz equation in one, two and three dimensions, *International Journal for Numerical Methods in Engineering*, 46: 471–499.

Donéa, J. (1984): A Taylor–Galerkin method for convection transport problems, *Interernational Journal for Numerical Methods in Engineering*, 20: 101–119.

Donéa, J.; Giuliani, S. (1981): A simple method to generate high–order accurate convection operators for explicit schemes based on linear finite elements, *International Journal for Numerical Methods in Fluids*, 1: 63–79.

George, P. L. (1991): *Automatic Mesh Generation. Applications to Finite Element Methods*, Wiley: Chichester.

Hassan, O.; Morgan, K.; Probert, E. J.; Peraire, J. (1996): Unstructured tetrahedral mesh generation for three dimensional viscous flows, *International Journal for Numerical Methods in Engineering*, 39: 549–567.

Hills, D. P. (1996): Numerical aerodynamics: past successes and future challenges from an industrial point of view. In *Computational Methods in Applied Sciences '96—Invited Lectures and Special Technological Ses-*

sions at the ECCOMAS Conferences, J–A. Desideri et al (editors). Wiley: Chichester, 166–173.

Kangro, U.; Nicolaides, R. (1997): Spurious fields in time–domain computations of scattering problems, *IEEE Transactions on Antenna and Propagation*, 45: 228–234.

Karypis, G.; Kumar, V. (1998): Multilevel k–way partitioning scheme for irregular graphs, *Journal of Parallel and Distributed Computing*, 48: 96–129.

Morgan, K.; Brookes, P. J.; Hassan, O.; Weatherill, N. P. (1998): Parallel processing for the simulation of problems involving scattering of electromagnetic waves, *Computer Methods in Applied Mechanics and Engineering*, 152: 157–174.

Morgan, K.; Hassan, O.; Pegg, N. E.; Weatherill, N. P. (2000): The simulation of electromagnetic scattering in piecewise homogeneous media using unstructured grids, *Computational Mechanics*, 25: 438–447.

Morgan, K.; Hassan, O.; Peraire, J. (1994): An unstructured grid algorithm for the solution of Maxwell's equations in the time domain, *International Journal for Numerical Methods in Fluids*, 19: 849–863.

Morgan, K.; Hassan, O.; Peraire, J. (1996): A time domain unstructured grid approach to the simulation of electromagnetic scattering in piecewise homogeneous media, *Computer Methods in Applied Mechanics and Engineering*, 134: 17–36.

Morgan, K.; Peraire, J. (1998): Unstructured grid finite–element methods for fluid mechanics, *Reports on Progress in Physics*, 61: 569–638

Mur, G. (1994): Edge elements, their advantages and their disadvantages, *IEEE Transactions on Magnetics*, 37: 3552–3557.

Peraire, J.; Peiró, J.; Morgan, K. (1999): Advancing front grid generation. In *Handbook of Grid Generation*, J. F. Thompson, B. K. Soni and N. P. Weatherill (editors), CRC Press: Boca Raton, 17.1–17.22

Petitjean, B.; Löhner, R. (1992): Finite element solvers for radar cross section RCS calculations, *AIAA Paper 92–0455*.

Shankar, V.; Hall, W. F.; Mohammadian, A.; Rowell, C. (1993): Theory and application of time–domain electromagnetics using CFD techniques, *Course Notes*, University of California Davis.

Sun, D.; Manges, J.; Yaun, X.; Cendes, Z. (1997): Spurious modes in finite element methods, *IEEE Antennas and Propogation Magazine*, 37: 12–24.

Weatherill, N. P.; Hassan, O. (1994): Efficient three–dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints, *International Journal for Numerical Methods in Engineering*, 37: 2005–2040.

Weatherill, N. P.; Hassan, O.; Morgan, K.; Jones, J. W.; Larwood, B. (2001): Towards fully parallel aerospace simulations on unstructured meshes, *Engineering Computations"*, 18: 347–375.