LOCAL ADAPTIVE MULTIGRID FOR FINITE INTEGRALS TECHNIQUE

Cătălin Ciobotaru, Daniel Ioan
Numerical Methods Lab (LMN), Electrical Engineering Department
"Politehnica" University of Bucharest, Spl. Independenței 313, 77206, Bucharest, Romania.
Email:[catalinc, daniel]@lmn.pub.ro

***Abstract**: This paper contains an analysis of the local adaptive approach for Finite Integrals Technique (FIT), suitable and efficient in solving electromagnetic field problems encountered in compact modelling of on-chip passive structures at high frequencies. More scientific applications have the property of high speed varying solution in small surfaces/volumes of the modeled domain. To solve efficiently this kind of problems, the adaptive strategies are used to concentrate grid nodes in regions where probability of errors appearance is. In this way the quality of approximated solution can be improved while the total number of nodes is maintained at a low level. The Adaptive Mesh Refinement (AMR) technique we propose uses Octree data-structure. Theoretical aspects and obtained numerical results are presented.*

Introduction

Dynamic adaptive numerical techniques for solving the equations obtained by integral or differential methods have the advantage of concentrating the computing effort in the interest zones of the problem domain [1]. In the case of hierarchic adaptive refinement methods, are identified domain regions that requires increased solution precision and finer meshes are dynamically added in these regions. AMR based techniques are starting from a rare mesh with a minimum accepted resolution that covers the entire computing domain. While the solution finding process is advancing, the domain regions that are requiring refinement are marked and replaced with a finer mesh in a recursive or iterative process [2]. The obtained mesh constitutes an adaptive dynamic hierarchy.

AMR techniques have been used in a great variety of disciplines like fluids dynamics, astronomy, meteorological simulations, mechanics, electromagnetism and thermodynamics. Rapid mesh generation constitutes a novelty subject. First steps have been made in fluids dynamic problems from aeronautic industry. A fast and stabile method for mesh generation is given in [3]. The algorithm strategy is based on intersection of components in order to find out the external surface followed by a volume refinement. The intersection schema is based on a geometric engine which is using an adaptive error estimator and it is automatically manipulating geometrical degenerations by a tie-breaking routine. In the worst case, the complexity order of the algorithm is $O(N \log N)$.

An important aspect in AMR is to avoid parasite nodes that are generated on transition from finer to rare regions of the simulating domain. An algorithm that studies this problem is proposed in paper [4].

Operations that can be done by using a hierarchic adaptive mesh according to the reference algorithm proposed by Berger and Oliger [5] are as follows:

- **Time integration** is the refresh operation done on each grid on each level from the hierarchic adaptive mesh. The integration is done using an application specific quadrature technique.

- **Inter grid operations** are used to communicate the solutions for hierarchic adaptive mesh. Basic operations are of extension, defined from a rare to a finer mesh, and restriction, defined from a finer to a rare mesh.
- **Grid regeneration** takes into consideration three steps – (1) marking the regions that have to be refined in function of specific criteria; (2) unifying the marked points; (3) generating the refined mesh. Regeneration can consists in building a new refinement level or adding of additional grids on some levels and/or deleting the existing grids.

Parallelization of adaptive methods based on hierarchic AMR consists in partitioning of hierarchic adaptive mesh on a computing network nodes and performing the concurrent operations for the local problem domain regions [6]. Parallel AMR applications are requiring two kinds of communications: inter networks communication and intra network communication.

A 3D hybrid adaptive mesh refinement algorithm with code named Enzo [7] was projected for high resolution simulation of Universe astrophysical structure formation. Parallel implementation of the algorithm does not impose a depth or complexity limit of the adaptive mesh. To lower the complexity order of the algorithm the *Structured Adaptive Mesh Refinement (SAMR)* method was adopted.

At this time there are three main alternative methods that can be used at local mesh refinement [8]. First method consists in recursive refinement applied to all domain cells, thus obtaining dynamical data structures of Quadtree type in 2D and Octree type in 3D [9]. The tree connection provides necessary information for multi grid based methods. Even local refinement can be easily implemented by using this data structure, the disadvantages are the difficult vectoring (on vectorial architectures), and the bandwidth minimization for maintaining local information (on machines that use the processor cache memory). To avoid the overloading due to go through the three, it is used a map of leaf nodes kept in another data structure.

The second alternative is the use of structured Cartesian meshes, usually associated to the Adaptive Mesh Refinement approach (AMR). In this approach, the cells from a refinement level are organized in an orthogonal grid which usually contains a number of some hundreds cells. By using structured arrays in function of connections number allows compact allocation of an entire sequence of grid on almost 20 memory words. The main disadvantage is that of complexity for implementation at programming level.

The third alternative consists in using a data structure in which the connections are explicitly memorized with the grid. The advantages provided by Cartesian grids are resulting in extremely compact data structures. In paper [4] it is used a data structure based on faces in which the grid is discretized in a list of cells faces that have as reference the Cartesian cells form the other face. Adjacent cells from different levels of refinement (that can differ with at most a level) are introduced in this structure and refined faces are indicating to the corresponding cells from a face and the same thick cell from the other side. Cartesian nature of hexahedral allows to faces and cells from a network to be memorized in almost 9 words per cell.

The algorithm proposed in this paper is based on Octree Adaptive Mesh Refinement for Finite Integrals Technique [13]. The local error estimator is computed by the dual Finite Integrals Technique (dFIT) [14].

Algorithm principles

In order to obtain a robust and efficient computing system for auto-adaptive simulations, it was developed an OOP implementation using C++. This approach allows a better code maintenance with a low probability of introducing additional errors. Essentially, the approach is based on the divide-and-conquer principle of the multigrid algorithm [10]. Due its

advantages, for the numerical implementation it was used the Octree data structure. An Octree is a data structure used to automatically represent the objects in the tridimensional space, by grouping them hierarchically and excluding the void space domains. Octree based spatial discretization consists in dividing a cubic cell that covers the interest domain in eight cubic cells (octants). In this way, the computing domain is successively divided into octants [11] until the boundary regions are approximated well enough. For an Octree to become balanced it has to support the 2 to 1 constraint [12]. This constraint imposes the dimension of an edge from a neighbour cell to not be two times bigger than the edge of current cell. Octree data structure is represented as a tree which nodes can have eight leafs. The first octree node is the root cell (Fig. 1), that is an array of eight neighbour elements. Each of these elements can point to a block of neighbour elements where every element can have a reference to another block of eight neighbour elements and so on uintil it is reached a maximum number of levels. The last level is the leaf level on which are placed the leaf elements. Octree cells numbering is done after Ox, Oy and Oz pozitive directions. An Octree node can have 26 possible directions: 6 neighbours along a face, 12 neighbours along a vertex, and 8 neighbours for an edge.

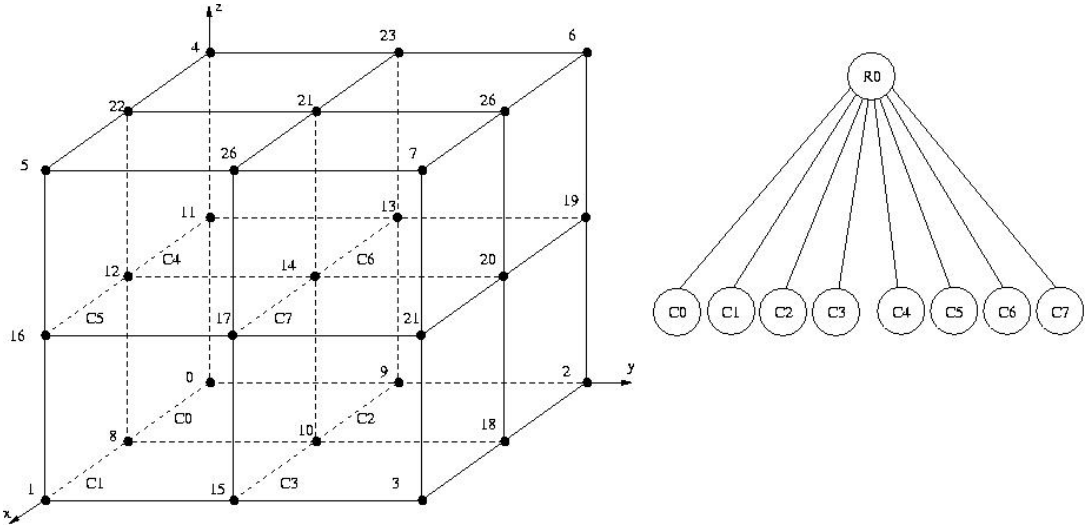

Fig. 1: Parent cell with one refinement level: tree structure with nodes and cells numbering

The solutions obtained from the two dual grids (voltages from the primary grid and currents from the secondary grid) are used to compute the power density in the middle of each each cell (*a*, *b*, *c* being the cells length after Ox, Oy, and Oz directions) in two different ways:

$$p_E = \frac{\sigma}{4}\left[\left(\frac{u_1}{a}\right)^2 + \ldots + \left(\frac{u_4}{a}\right)^2 + \left(\frac{u_5}{b}\right)^2 + \ldots + \left(\frac{u_8}{b}\right)^2 + \left(\frac{u_9}{c}\right)^2 + \ldots + \left(\frac{u_{12}}{c}\right)^2\right], \tag{1}$$

$$p_J = \frac{\rho}{4}\left[\frac{i_1'^2 + i_4'^2}{(bc)^2} + \frac{i_2'^2 + i_5'^2}{(ac)^2} + \frac{i_3'^2 + i_6'^2}{(ab)^2}\right], \tag{2}$$

the **local error estimator** being:

$$\varepsilon = \left| p_F - p_J \right|. \tag{3}$$

As a **global error estimator** is based on resistance computed two times once by primary FIT and second times by secondary (on dual grid) FIT methods:

$$err_{global} = \left(\frac{R_{dFIT} - R_{FIT}}{R_{dFIT} + R_{FIT}} \right)^2 \tag{4}$$

The pseudocode of the implemented algoritm has the following structure:

```
InitialMeshGeneration();
repeat
   FITsolving();// computes the lower boundary for the solution
   dFITSolving();// computes the upper boundary for the solution
   LocalErrorEstimator();
   GloablErrorEstimator();
   MeshRefinement();
until StopingCriteria
```

At each iteration, the mesh refinement method is controlled by the local error estimator. Refinement is applied where the estimator has a relativelly high value, compared to an addaptive thrshold. If the global error estimator is bigger than it was in the previous iteration, then the value of threshold is decreased. If its value becomes closer to zero, an uniform refinement is applied. Therefore, it can be said that multigrid algorithm with uniform grid is a limit case of AMR algorithm for the situation in which the value of this threshold becomes zero.

Mesh refinement process continues until the stoping criteria is achieved, as follows:

- If the resistence computed with dFIT becomes smaller than the resistance computed with FIT;
- If the global error becomes smaller than a referenced value (e.g. 10^{-3});
- If the maximum number of adaptive mesh refinement steps is achieved (by default, 10 steps).

Numerical results

The test problem consists in finding the resistance of "L-shaped" conductor. Geometrical data of this problem (Fig. 2) are as follows: a= b= c= d= e=1μm. The conductivity of metallic region is $\sigma_1 = 10^6$ S/m, while the insulator conductivity is $\sigma_2 = 10^{-2}$ S/m.
The accurate value of conductor resistance is known as being R=2.558523 Ω.

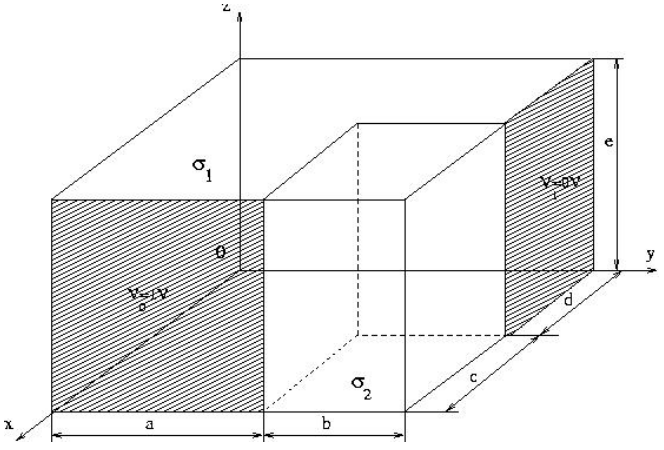

Fig. 2: "L-shaped" test problem

The relative error was determined by using the following relations:

$$\varepsilon_{rE} = \frac{|R - R_E|}{R}, \varepsilon_{rJ} = \frac{|R - R_J|}{R}, \varepsilon_{ra} = \frac{|R - R_a|}{R}, \quad R_a = \frac{R_E + R_J}{2}.$$

Fig. 3: "L-shape" resistance relative error variation in function of node numbers – uniform grid and AMR grid.

In figure 3 it is presented the "L-shape" resistance relative errors versus number of nodes for uniform grid and AMR-Octree grid. Figure 4 presents the mesh obtained with AMR-Octree algorithm for an imposed global error of 10^{-1}.

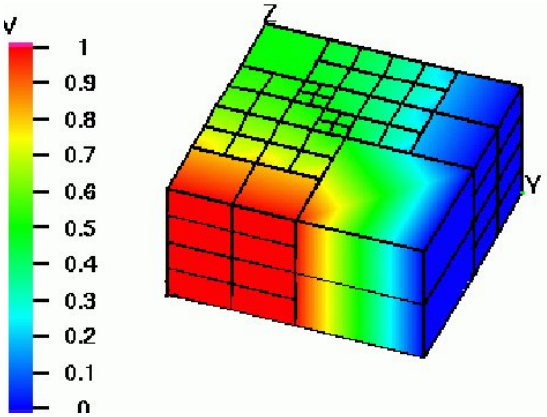

Fig. 4: Refined mesh

Conclusions

Adaptive mesh refinement has been used for nearly two decades in electromagnetics to speed the convergence of different computing methods. This technology has two principle elements: unstructured mesh generation and local error estimation. In this paper is introduced an algorithm for computing the local and global error estimation based on dual Finite Integration Technique. Compared with the traditional multigrid algorithm, the new one requires less total nodes, therefore lower computing cost, in order to obtain same solution accuracy.

Acknowledgements

The research of the authors for results presented in this paper was supported by European Union FP5/IST/Codestar project.

References

[1]. Manish Parshar and James C. Browne. *Distributed Dynamic Data-Structures for Parallel Adaptive Mesh-Refinement*, 1995.

[2]. Istvan Bardi, Gaunghen Peny, and Zoltan J. Cendes. *Improovements in Adaptive Mesh Refinement and Multilevel Methods in Electromagnetics*. In URSI GA 2002, 2002.

[3]. M.J. Aftomis, M.J. Berger, and J.E. Melton. *Robust and Eficient Cartesian Mesh Generation for Component-based Geometry*. In 35th AIAA Aerospace Sciences Meeting, Reno NV, January 1995.

[4]. Wim Schoenmaker, Wim Magnus, Peter Meuris, and Bert Maleszka. *Renormalization Group Meshes and the Discretization of TCAD Equations*. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 21:1425{1433, December 2002.

[5]. Marsha J. Berger and Joseph Oliger. Adaptive mesh re_nement for hyperbolic partial differential equations. Jurnal of Computational Physics, 53(2):484{512, March 1984.

[6]. Marsha Berger and James C. Brown. *On Partitioning Adaptive Grid Hierarchies*. In HICSS (1), pages 604{613, 1996.

[7]. Greg L. Bryan, Tom Abel, and Michael L. Norman. *Achieving Extreme Resolution in Numerical Cosmology Using Adaptive Mesh Refinement: Resolving Primordial Star Formation*. In Super Computing 2001, Denver, Colorado, November 2001.

[8]. M.J. Aftomis, M.J. Berger, and J.E. Melton. *Adaptive Cartesian Mesh Generation. In RC Handbook of Grid Generation*, 1998.

[9]. Georg Border and Ralf Deiterding. *A Distributed Memory Adaptive Mesh Refinement Package for inviscid Flow Simualtions*. In Proc. of Colloquium on Fluid Dynamics, Institute of Thermodynamics, Academy of Sciences of Czech Republic, Prague, pages 9{14, October 1999.

[10]. William L. Briggs, Van Emden Henson, Steve F. McCormick, *A Multigrid Tutorial (2nd Edition),* SIAM, 2000.

[11]. Manish Parshar and James C. Browne. Object-Oriented Programming Abstractions for Parallel Adaptive Mesh-Re_nement, 1996.

[12]. Tiankai Tu, David R. O'Hallaron, and Julia C. Lopez. *Etree - A database-oriented method for generating large octree meshes*. In 11th International Meshing Roundtalb, Ithaca, NY, pages 127{138, September 2002.

[13]. M. Clemens and T. Weiland. *Discrete electromagnetism with the finite integration technique*. Progress In Electromagnetics Research, PIER, 32(3):65-87, 2001.

[14]. *D. Ioan, M.Radulescu,G.Ciuprina, Fast Extraction of Static Electric Parameters with Accuracy Control*, in Scientific Computing in Electrical Engineering (W.H.A. Schilders et al Eds), Springer, 2004,pp.248-256.