Progress In Electromagnetics Research B, Vol. 29, 409–429, 2011

REDESIGN AND OPTIMIZATION OF THE PAVING ALGORITHM APPLIED TO ELECTROMAGNETIC TOOLS (INVITED PAPER)

J. Moreno, M. J. Algar, I. González, and F. Cátedra

Electromagnetic Computing Group
Computer Sciences Department
University of Alcalá, Alcalá de Henares, Madrid E-28871, Spain

Abstract—To study any electromagnetic system, the geometry model must be discretized into elements with an appropriate size for the working frequency. The discretization of a system must be transparent to the user of electromagnetic computing tools. A mesher is presented based on the paving algorithm. The algorithm has been modified to allow triangular elements and has been accelerated by distributing the load on multiple processors simultaneously. Also, a multilevel mode has been implemented. With this tool, any geometry defined by NURBS (Non Uniform Rational B-Spline) surfaces can be decomposed into triangular and quadrangular curved elements.

1. INTRODUCTION

One of the most important issues in electromagnetic field prediction problems is the geometrical model, which affects the convergence and accuracy of the results. Most popular methods for 3D solid modeling can be classified into one of six groups: parameterized forms, cell decomposition models, sweep models, constructive solid geometry models, boundary models and wire models. In the last decade new methods have been developed by combining some of the above methods, e.g., combining boundary models with curved surfaces. For example, NURBS surfaces (Non Uniform Rational B-Spline) can describe any geometry efficiently using less information than other methods.

The accurate analysis of complex and/or large antenna or scattering problems using computer tools requires continuous conformed and

Received 23 February 2011, Accepted 22 April 2011, Scheduled 28 April 2011

Corresponding author: Manuel Felipe Cátedra (felipe.catedra@uah.es).

fitted meshes of the problem geometries. The quality of the meshes is critical for these tools to provide accurate results and convergence. In addition the memory resources and CPU-time required to obtain the meshes is an important issue.

Due to the importance of the meshes in electromagnetic results, a new mesh generator of curved elements has been developed for geometries built with NURBS surfaces. Generated elements are curved to follow the geometry details closely. This mesh generator is based on the paving algorithm [1–3] and generates quadrilateral curved elements, but the possibility of inserting triangular elements has been considered in specific cases in which the insertion of these can improve the quality of the final mesh. An accurate electromagnetic simulation requires a quality mesh, the size of the edges of the mesh elements fitted for the simulation frequency and the shape of the elements must be as simple as possible (quads or triangles provide good results).

Currently, computers have developed to the point that it is difficult to increase the speed of processors. The most popular solution has been to provide the computers with several processors. To accelerate the new algorithm as much as possible, this mesh generator has been parallelized using the MPI (Message Passing Interface) paradigm [4] to allow very large problems to be meshed in a few minutes.

The multilevel mode is another feature that can be used to optimize the paving algorithm. The meshes are obtained in several steps using coarse meshes for the first level. This mode can be very useful for cases in which the geometries are composed of many surfaces of very different sizes and shapes.

This mesher has been included in module MONURBS of the NewFasant computer tool [5]. This module is based on parallelized MoM (Method of Moments) with MLFMM (Multilevel Fast Multipole Method) [6–9] and CBFM (Characteristics Basis Function Method) [10] to reduce the CPU-time and memory requirements for problems of a single excitation (antennas) or multiple excitations (monostatic RCS) [11]. To solve very large problems, a new algorithm based on "domain decomposition" has been successfully implemented for previous versions of the code. The mesher is also useful for FASANT a GTD (Geometrical Theory of Diffraction) module [12, 13] that was recently incorporated in NewFasant. This module includes a new ray-tracing technique based on heuristic techniques such as the A* algorithm and the "first deep search" approach.

Several meshes of real geometries are shown at different frequencies. To validate the developed mesher, MONURBS was applied to the meshes obtained with the proposed mesher and with other commercial meshers for several test cases. The comparisons between the results obtained with these meshes indicate that this mesher has better accuracy. Also several examples of how the parallelization and the multilevel mode reduce the time required to mesh complex geometries are included. It is shown that the proposer mesher is capable of meshing any geometry composed of curved surfaces and ensures a fast convergence to electromagnetic calculation tools.

The paper is organized as follows. A description of the developed mesh generator is presented in Part 2; the new features added to the original paving algorithm are described. In Part 3 several mesh examples for difficult and special geometries are presented. Part 4 shows the performances of the proposed meshing algorithm by first testing the quality of the obtained meshes and then comparing the accuracy of the results obtained from different meshes and the CPU-time reduction using the parallelized and multilevel version. The paper ends with a summary of the main features of the new meshing algorithm.

2. THE NEW PAVING ALGORITHM ADAPTED FOR ELECTROMAGNETIC COMPUTATIONS

The purpose is divide the parametric surfaces that compose a geometric object into a finite number of quadrilateral curved elements (or "quads") that have edge sizes that depend on the frequency and the number of divisions per wavelength. In some parts of the geometry triangular curved elements can also be included if it is too difficult to obtain nearly regular quads. The mesh obtained must faithfully respect the true shapes of the original geometry.

The mesh generator operation is based on the paving algorithm, specifically a version that includes a factory facet that is used to generate elements during the meshing, combined with other pre-processing techniques, such as the previous calculation of mesh points in the boundary surfaces to ensure the continuity of the elements with common edges. The mesher also has a post-processing clean-up stage that improves the quality of the mesh by forcing elements to be as close to regular quads as possible. One example is the insertion of an element formed by four corners but with a triangular shape (three nodes aligned). Including triangular elements resolves this problem.

Figure 1 compares the meshes obtained with the developed hybrid mesh generator and the commercial mesher GiD [14, 15]. It is the common example in which the insertion of a triangular element in a mesh composed of quads simplifies the final mesh. The reason why

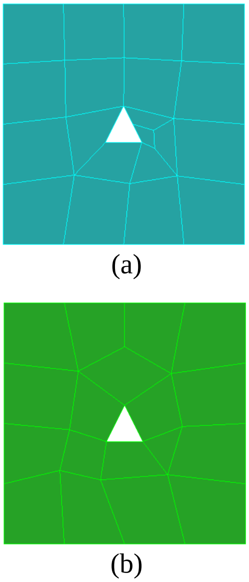

Figure 1. Triangular hole meshed with (a) GiD and (b) developed mesher.

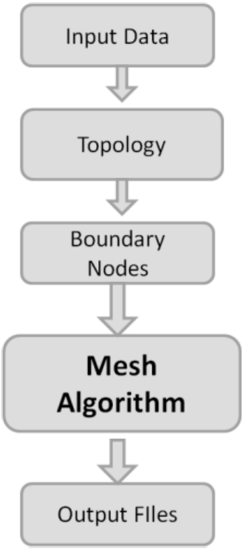

Figure 2. General scheme of the mesh generator.

the adopted solution is a hybrid mesh generator instead a triangular is that the number of subdomains to be solved is reduced almost to a half. As the generated elements are conformed to the surfaces, the accuracy of results is guaranteed although there are less elements than in a triangular (conformed or not) mesh.

The new feature for generating occasionally triangular elements provides highquality meshes in which all elements are close to perfect quads with the size edge desired, and triangular elements are only inserted when it is impossible to generate an adequate new quadrilateral element.

2.1. Architecture of the Mesher

The general scheme of the mesher is shown in Figure 2. The reading and writing of the input and output files are not included. The three important stages in the mesh generator are the calculation of the topology, boundary nodes, and obviously, the meshing algorithm.

Prior to any operation, the topologies between all the surfaces that compose the geometry must be checked (or calculated if they are not included in the input files). Correct topology ensures continuity in the mesh (it is essential that topologies be well-defined in the original geometries).

Then, all the contours of the surfaces are divided according to the edge size for meshing. Once the boundary nodes have been obtained, the meshing must begin with them. Surfaces that share contours also share the boundary nodes of their common contours.

In the following, a short description of the implementation of this mesh is presented.

2.2. Mesh Algorithm

The paving algorithm generates boundary elements using the nodes in the edges that compose the surfaces. The insertion of elements starts from the contours of the surfaces and the boundaries are updated as new elements are inserted until the surfaces are completely meshed. An example of how a plate with a hole can be meshed with the paving algorithm is shown in Figure 3. The mesh generator is composed of different modules each of which has a specific purpose, as shown in Figure 4.

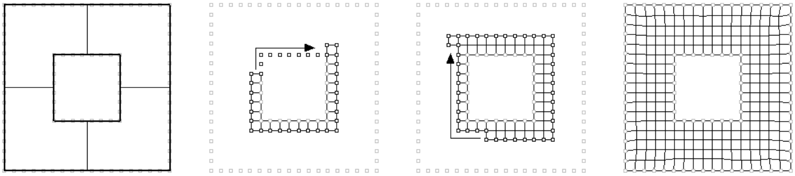

Figure 3. Process of meshing a plate with an inner hole.

First, the best starting node must be selected for inserting elements. The selection of the starting node can strongly influence the quality of the final mesh, which is particularly important in surfaces with loops with small edges compared to the rest of the edges. For example, Figure 5 shows a plate with several small holes meshed with the proposed mesh generator; the holes have been successfully meshed despite their relatively small size edges.

Once the starting node has been chosen, the most important stage of the mesh generator is the "factory facet". The factory facet generates new nodes to form new elements. The elements are generated according to the type of the starting node; the boundary nodes are qualified as a function of their inner angles in the boundary. For example, if the starting node has an inner angle of 90 degrees, a single node must be inserted to generate the new quadrilateral element; however, if the node has an angle of 270 degrees, three nodes must be inserted to generate the new element. The edge size of the generated elements must be close to the desired size; thus, it is controlled in the factory facet.

The last stage of the mesh generator improves the quality of the final mesh when the factory facet has finished the complete surface or geometry. The "*Clean Up*" [16] algorithm evaluates the sizes and shapes of all the generated elements to determine those that could be improved either by moving the nodes that form it or combining the element with its neighboring elements.

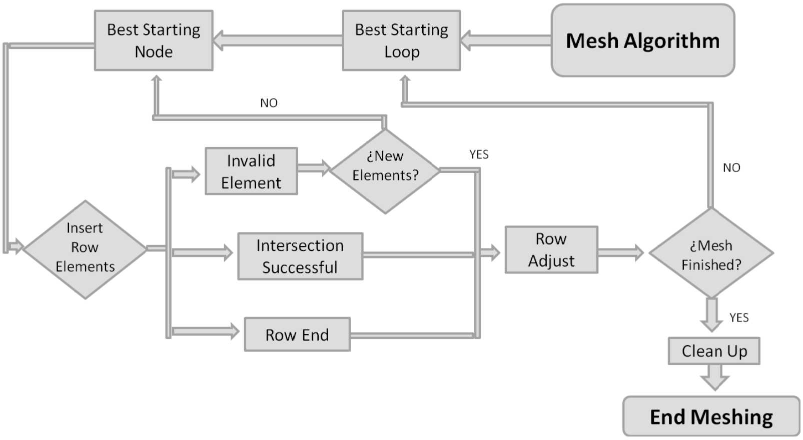

Figure 4. Detailed description of the paving algorithm.

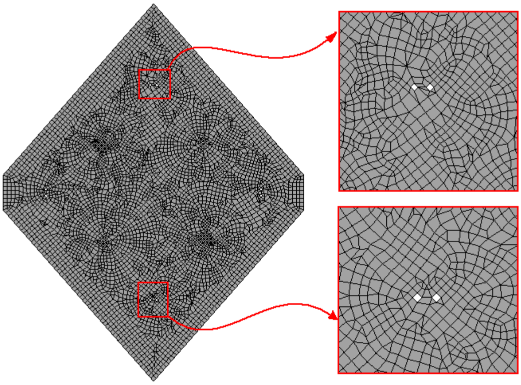

Figure 5. Plate with several small inner holes.

2.3. The Adapted Factory Facet

The factory facet generates new quadrilateral elements in the row. New nodes are inserted to create new elements and form new boundaries. These nodes can be generated in different ways depending on the type of the node from which they are created.

The new factory facet has the ability to create triangular elements, but most elements that form the mesh are quads. As mentioned above, the factory facet inserts new nodes that form new quadrilateral elements to complete the rows to be meshed. All new nodes are inserted to create quadrilateral elements; thus, triangular elements are only formed when the boundaries to be meshed are composed of three nodes. This new feature can significantly simplify the mesh because it is not necessary to insert an additional boundary node (when meshing with only quadrilateral nodes all the surfaces must have an even number of boundary nodes). It is important to remember that all the elements generated in the mesh are curved elements and thus the curvature of the original geometry to be meshed must be faithfully respected.

Figure 6 shows a meshed geometry at the lowest frequency with which it can be meshed; i.e., the frequency is small enough that the desired size edge of the elements is bigger than the edges that compose the surfaces of the geometry. As a result, the output file should have as many elements as surfaces of the original geometry (because these

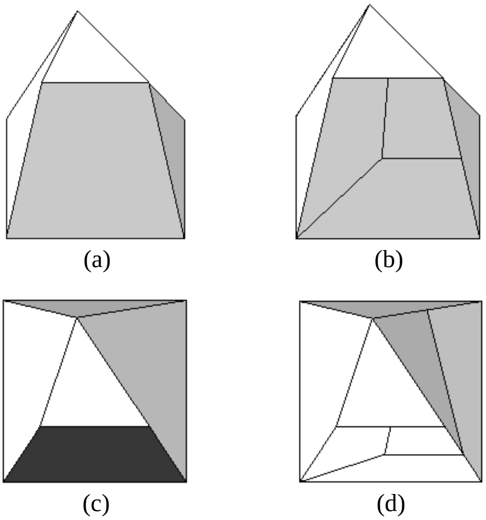

Figure 6. Pyramidal geometry meshed. (a) Front view developed. (b) Front view GiD. (c) Top view developed. (d) Top view GiD.

surfaces are rectangular or triangular). The developed mesh generator creates the expected result, while the quadrilateral mesh generator GiD creates smaller elements than desired because an extra node has been inserted in one of the edges of the triangular surface. The extra node is a result of the restriction that only quadrilateral elements be used, and the neighbor surfaces are affected by the extra node.

2.4. Parallelization

The proposed mesher was developed with the ability to use several processors, which is one of the most important features of this mesh generator because it significantly reduces the computation time as the number of processors involved increases When the mesh generator works with several processors, all of them must share some critical information. As shown in Figure 7, all processors share a full description of the geometry and topology before starting to generate the mesh. Each processor meshes the surfaces that have been assigned to it. When all processors have finished, a single output file is written.

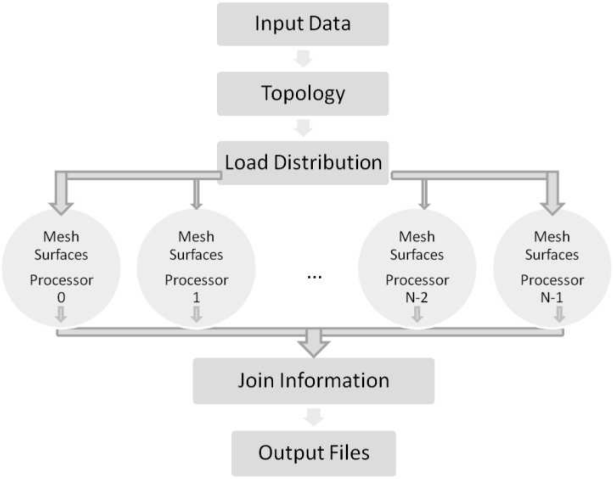
Input Data
Topology
Load Distribution
Mesh Surfaces Processor 0
Mesh Surfaces Processor 1
...
Mesh Surfaces Processor N-2
Mesh Surfaces Processor N-1
Join Information
Output Files

Figure 7. Behavior of the parallelized mesh generator.

2.5. Multilevel Mode

The multilevel mode is a useful feature added to the mesh generator because it significantly reduces the time required for meshing in very large problems or when the geometries are composed of surfaces with very different sizes. This mode is implemented according to the scheme shown in Figure 8.

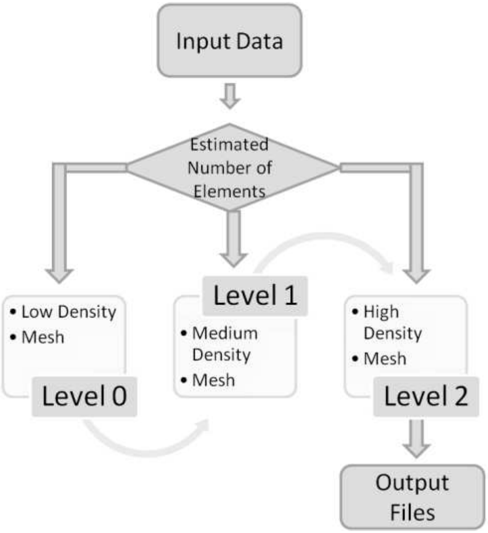

Figure 8. Flow chart of the multilevel meshing mode.

Once the input data and the complete geometry have been defined, an estimate of the number of elements required for meshing is calculated. If the estimated number of elements in the problem at the desired frequency is too high, then the final mesh is generated in several steps In the first step an intermediate mesh at a lower frequency (it is traduced on a low number of elements) than desired is made and its output mesh is used as input geometry to a second step that uses a higher frequency (higher density of elements). This process is repeated until a final mesh with the desired frequency (desired mesh density) is obtained.

The multilevel mode provides more regular meshes than a direct mesh to the desired frequency when the estimated number of elements is high. The number of levels required for a mesh is calculated automatically, and a balance between the number of levels and the frequency intervals between adjacent levels must be maintained. If the mesh is generated with many levels and the frequencies are close, too much time is required for meshing and the multilevel mode is not efficient.

One of the examples in which the multilevel mode is useful is meshing microstrip circuits. In Figure 9 and Figure 10, a reflectarray and a microstrip circuit meshed with the mesh generator are shown, respectively.

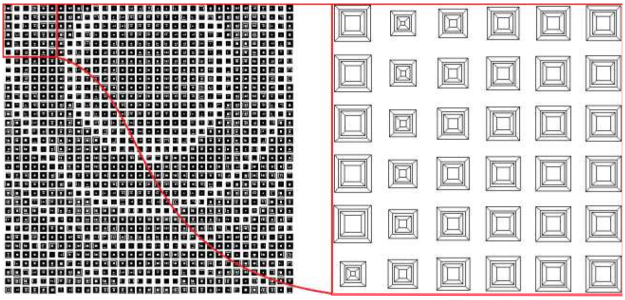

Figure 9. Reflectarray meshed.

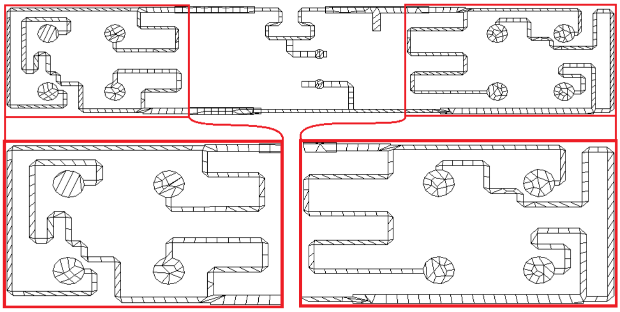

Figure 10. Microstrip circuit meshed.

3. EXAMPLES OF MESH CASES

In this section a set of meshed geometries is shown from the simplest to the most complex geometry. All the cases considered have been meshed using different edge sizes; thus, a variety of resolutions in the mesh can be observed.

3.1. Plate with a Rectangular Hole

In Figure 11 the original geometry and the mesh obtained with a low frequency are shown. Because the geometry is simple, it has been meshed using only quadrilateral elements.

3.2. Plate with a Circular Hole

To complicate the previous case, the rectangular hole has been changed to a circular hole. In Figure 12, the object has been meshed at a

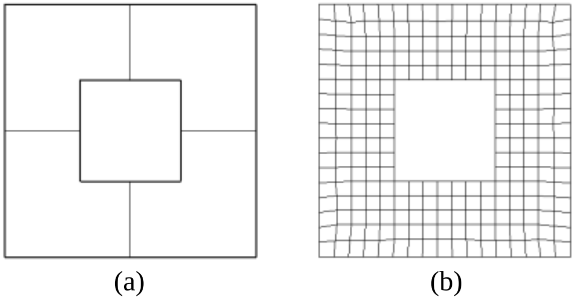

Figure 11. Plate with a rectangular hole. (a) Original and (b) meshed.

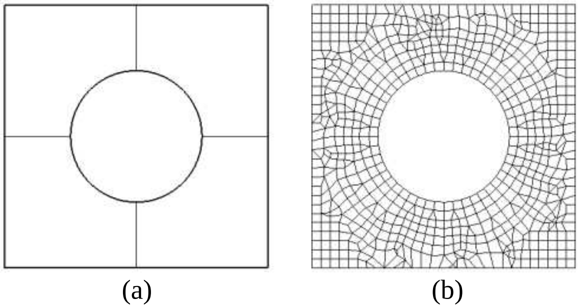

Figure 12. Plate with a circular hole. (a) Original and (b) meshed.

slightly higher frequency; thus, a higher resolution and the insertion of triangular elements can be observed.

3.3. Disk with an Small Hole

A very high frequency has been used to mesh a disk with a tiny hole, as shown in Figure 13. A homogeneous mesh can be observed with triangular and quadrilateral elements with similar edge sizes.

3.4. Truncated-cone Sphere

In Figure 14, a truncated cone-sphere composed of several surfaces is shown; thus, topologies must be taken into account. Moreover, this geometry is formed by surfaces with very different sizes.

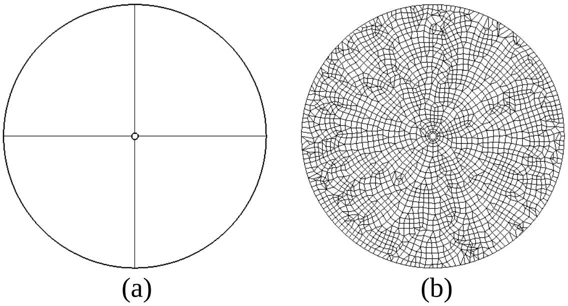

Figure 13. Disk with a small hole. (a) Original and (b) meshed.

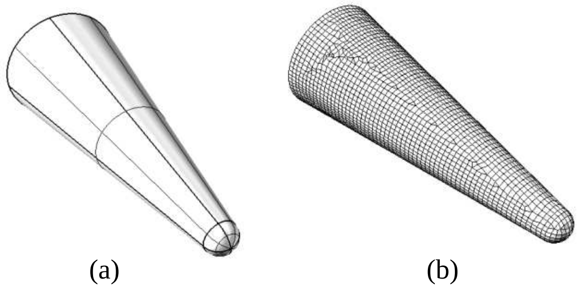

Figure 14. Truncated cone-sphere. (a) Original and (b) meshed.

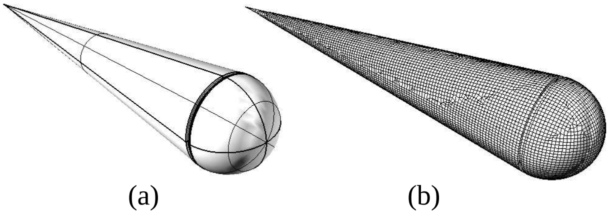

Figure 15. Cone-sphere. (a) Original and (b) meshed.

3.5. Cone-sphere

In this case a cone-sphere with a small slot has been meshed at a low frequency as shown in Figure 15; as a result, there are only a few elements.

3.6. Jason Satellite

In Figure 16 the original geometry and the corresponding mesh of part of a Jason satellite at a medium frequency are shown.

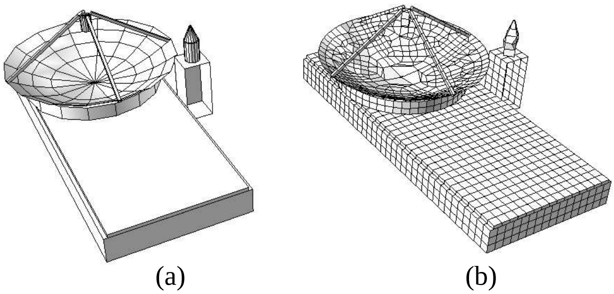

Figure 16. Jason Satellite. (a) Original and (b) meshed.

3.7. Wind Generator

A wind generator meshed at a low frequency is presented in Figure 17. In this case the number of generated elements is very low but the continuity of the mesh can be observed easily.

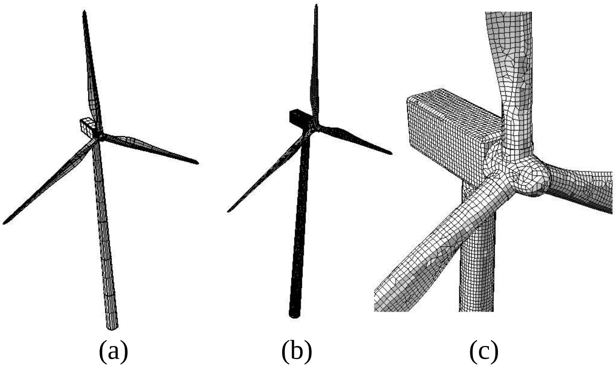

Figure 17. Wind generator. (a) Original, (b) meshed and (c) details of the mesh.

3.8. RG-31 Armored Vehicle

Figure 18 shows the original and meshed geometry of a real armored vehicle: a RG-31 model. The model has been meshed with few elements to illustrate the mesh details, like the continuity.

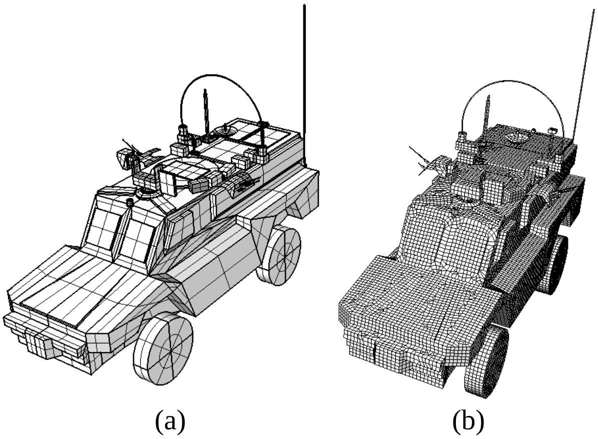

Figure 18. RG-31 armored vehicle. (a) Original, (b) meshed and details of the mesh.

3.9. Airplane

In Figure 19, the original geometry of an airplane and its corresponding mesh are shown. Some details of this mesh have been included.

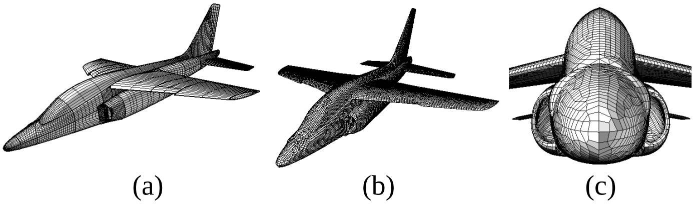

Figure 19. Airplane. (a) Original, (b) meshed and (c) details of the mesh.

3.10. MQ-1 Predator

To conclude, the mesh obtained of an UAV (Unmanned Aerial Vehicle), a MQ-1 Predator, is shown in Figure 20. In this case it can be observed that the mesh is homogeneous but continuity does not exist between some parts of the geometry because the original geometry has not been modeled to represent the topologies correctly.

4. RESULTS

In this section the results of the validation of the mesher are presented. The validation includes comparisons of the results of electromagnetic simulations performed with meshes obtained with proposed mesher

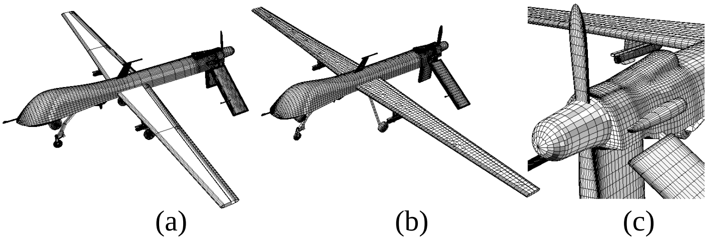

Figure 20. MQ-1 Predator. (a) Original, (b) meshed and (c) details of the mesh.

and with commercial mesh generators. Also several examples have been considered to demonstrate the improved computational efficiency of the proposed mesher.

4.1. Electromagnetic Results

To validate the mesh generator, this section shows the results of two electromagnetic simulations that apply the module tool MONURBS from the NewFasant tool to the meshes of two test cases obtained using the developed mesh generator and commercial meshers GiD and CUBIT.

First, the monostatic scattering results of the cone-sphere shown in Figure 15 were analyzed at frequencies 869 MHz and 9 GHz for both horizontal polarization and vertical polarizations at the $\theta = 90^\circ$ cut and a sweep from $\Phi = 0^\circ$ to $\Phi = 180^\circ$. A comparison of the monostatic scattering at 869 MHz using meshes obtained with developed mesher, Cubit and GiD is shown in Figures 21 and 22. A considerable difference can be observed between the results of both meshes, especially for the horizontal polarization case. After studying the results, it is observed that the cleft that appears in the original cone-sphere has some irregular elements in the mesh generated with GiD that distort the results. Figure 23 and 24 show the results obtained at 9 GHz using the meshes of new mesher, Cubit and GiD. Now both results are more similar. The time required for computing the monostatic scattering is similar with the three meshes: about 32 seconds at 869 MHz and an hour at 9 GHz for each mesh (using 4 processors).

Secondly, a truncated-cone sphere has been simulated at 10 GHz for horizontal polarization. Figure 25 compares the monostatic scattering results for the $\Phi = 0^\circ$ cut and a sweep from $\theta = 0^\circ$ to $\theta = 91^\circ$. Results obtained using meshes generated by GiD Cubit [17] and our approach are compared The results obtained from the three different meshers are similar; thus, the mesh obtained with the developed mesh generator is valid.

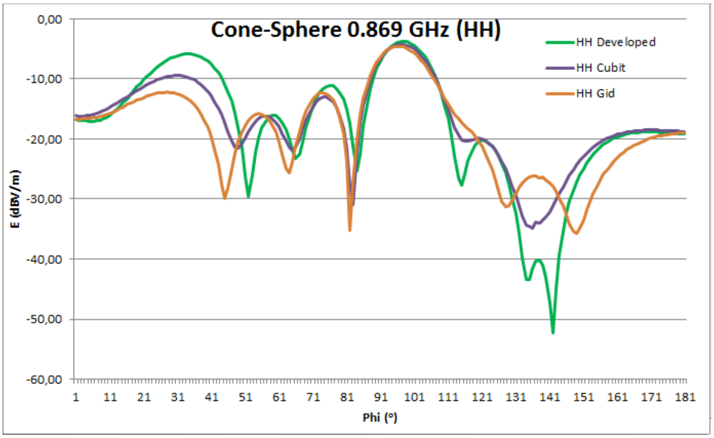

Figure 21. Monostatic scattering of the cone-sphere at 869 MHz (HH) for $\theta = 90^\circ$ and Φ ranging from 0° to 180°.

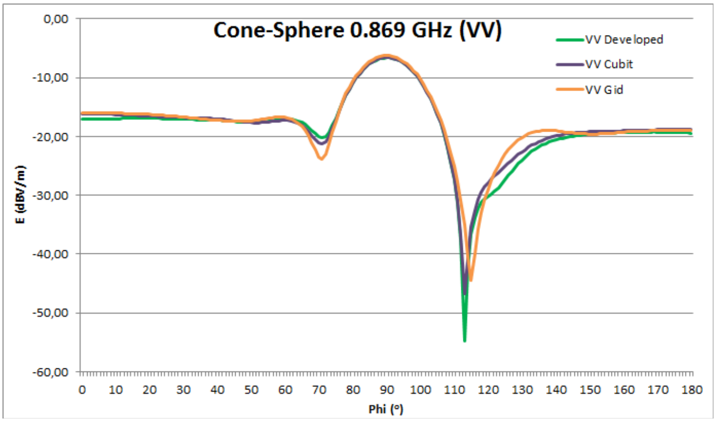

Figure 22. Monostatic scattering of the cone-sphere at 869 MHz (VV) for $\theta = 90^\circ$ and Φ ranging from 0° to 180°.

4.2. Performance Using Parallelization

The wind generator shown in Figure 17 was meshed at 500 MHz while the number of processors used was varied. The meshes were performed on a SUN X4000 QUAD OPTERON 32-core machine at 2.4 GHz with 256 GB of RAM. Figure 26 shows the time required to perform a complete meshing versus the number of cores used. The number of elements generated in this mesh is 441409. As observed

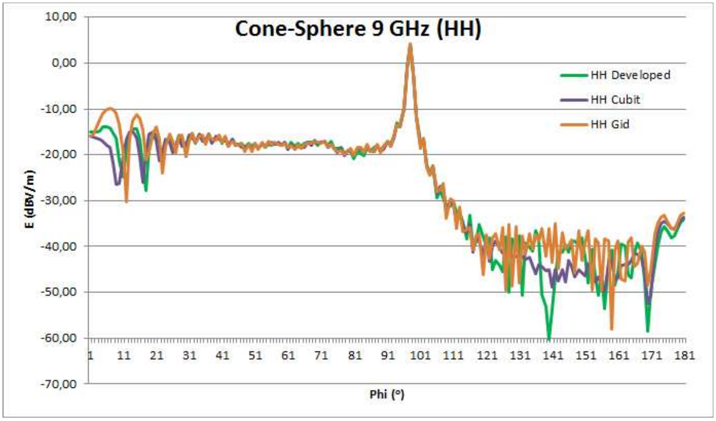

Figure 23. Monostatic scattering of the cone-sphere at 9 GHz (HH) for $\theta = 90^\circ$ and Φ ranging from 0° to 180°.

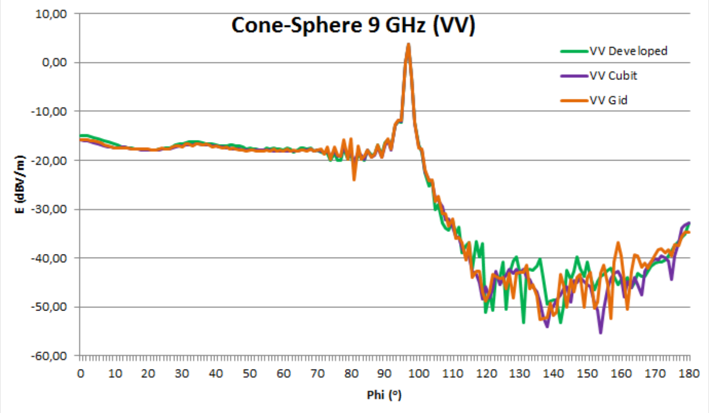

Figure 24. Monostatic scattering of the cone-sphere at 9 GHz (VV) for $\theta = 90^\circ$ and Φ ranging from 0° to 180°.

in the figure, the time is reduced from 2.908 to 126 seconds using 32 processors, which is a reduction factor of about 25. The shape of the time versus number of processors graph has the form of a decreasing exponential because of the parts common to all processors that cannot be parallelized, as shown in Figure 7. Because the load distribution is a function of the surfaces, the parallel mode is more efficient if the shapes are simple and have similar sizes.

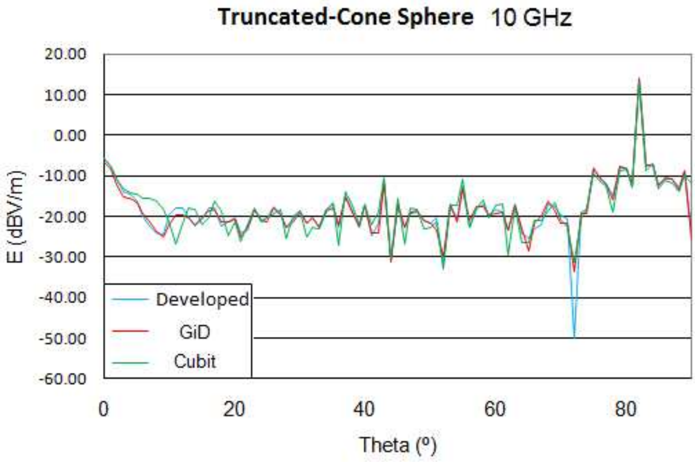

Figure 25. Monostatic scattering of the truncated cone-sphere at 10 GHz (HH) for $\Phi = 0°$ and θ ranging from 0° to 91°.

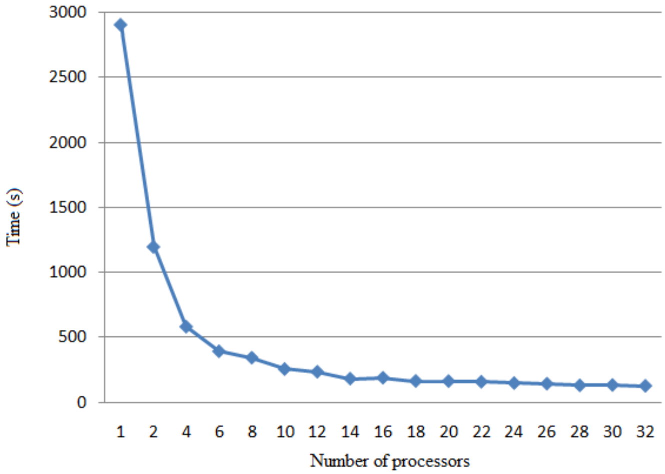

Figure 26. Influence of the number of processors used in the meshing of the wind generator at 500 MHz (441.409 elements).

4.3. Multilevel Mode Results

To validate the efficiency of the multilevel mode, the cone-sphere shown in Figure 15 was meshed at 50 GHz It was composed of four big surfaces, four medium surfaces and sixteen small surfaces. Table 1 compares the times required to mesh the cone-sphere with and without

Table 1. Direct mode vs. multilevel mode.

Mode	Number of elements	Time required (s)
Direct	525.635	2.963
Multilevel	536.021	393

the multilevel mode The meshes were performed using 6 processors in a SUN X4000 QUAD OPTERON 32-core machine at 2.4 GHz with 256 GB of RAM.

As observed in Table 1, the number of elements generated between the two modes is slightly different, but there is a time reduction of 86.8% using the multilevel mode compared to the direct mode. This result is logical because in the multilevel mode a mesh is generated at a low frequency and all its elements have the same size edge. The next levels start from different boundaries and surfaces than the first (or direct mode); thus, the new load distribution is more accurate and the final mesh does not have to be identical to the mesh obtained with the direct mode. However, the number of generated elements must be similar because the final results are the meshes obtained from the same geometry at the same frequency.

5. CONCLUSIONS

This work has covered several topics such as the need for a good mesh generator that ensures the convergence of electromagnetic computation tools and the importance of parallelization on any computer tool.

To implement a mesh generator, it is essential that the mesh strictly follows the features of the original geometry. It is important for the mesh generator to be able to insert triangular elements because it offers more flexibility to the algorithm and ensures that the edge sizes of all the generated elements are very close, which is one of the most important requirements for electromagnetic tools.

In the results obtained with the tool MONURBS from different meshes the similarity between the curves plotted with the development mesh generator and the commercial mesh generators is evident. Therefore, the mesh generator provides high quality meshes.

Comparing the result from the developed mesh generator with other commercial meshers that provide meshes in which most of the elements are perfect quads, it can be concluded that the proximity of the elements to perfect quads is not as important as homogeneity in the edge sizes of these elements for electromagnetic results.

Two methods were used to optimize the speed of the mesh generator:

- First, parallelization is a common solution to accelerate any computer tool. To parallelize the mesh generator, all the processors must share some critical information Then the load must be distributed, which reduces the operation time; thus, all the processors require the same time to mesh their surfaces. Parallelization ensures time reduction, but the reduction is not linear; the time decreases as the number of used processors increases, but some common parts like the reading of input data, load distribution or writing a single output file cannot be parallelized and always require the same time, independent of the number of processors used.
- A multilevel approach improves the results when it is combined with the use of multiple processors. The multilevel mode simplifies the geometry to be meshed between adjacent levels; thus, the paving algorithm can converge faster provided that the number of levels is adequate.

Therefore, the proposed mesher is powerful because it provides convergence, quality and speed.

ACKNOWLEDGMENT

This work has been supported in part by the Comunidad de Madrid Project S-2009/TIC1485 the Castilla-La Mancha Project PPII10-0192-0083 and the Spanish Department of Science, Technology Projects TEC2010-15706 and CONSOLIDER-INGENIO No. CSD-2008-0068.

REFERENCES

1. Blacker, T. D. and M. B. Stephenson. "Paving: A new approach to automated quadrilateral mesh generation," *International Journal for Numerical Methods in Engineering*, Vol. 32, 811–847, 1991.
2. White, D. R. and P. Kinney, "Redesign of the paving algorithm: Robustness enhancements through element by element meshing," *Proceedings 6th International Meshing Roundtable*, 323–335, Oct. 1997.
3. Cass, R. J., S. E. Benzley, R. J. Meyers, and T. D. Blacker, "Generalized 3D paving: An automated quadrilateral surface mesh generation algorithm," *IJNME*, Vol. 39, No. 9, 1475–1490, May 1996.
4. http://www.mcs.anl.gov/research/projects/mpi/.

5. www.fasant.com.
6. Pan, X. M. and X.-Q. Sheng, "A highly efficient parallel approach of multi-level fast multipole algorithm," *Journal of Electromagnetic Waves and Applications*, Vol. 20, No. 8, 1081–1092, 2006.
7. Wang, P. and Y. Xie, "Scattering and radiation problem of surface/surface junction structure with multilevel fast multipole algorithm," *Journal of Electromagnetic Waves and Applications*, Vol. 20, No. 15, 2189–2200, 2006.
8. Zhao, X.-W., C.-H. Liang, and L. Liang, "Multilevel fast multipole algorithm for radiation characteristics of shipborne antennas above seawater," *Progress In Electromagnetics Research*, Vol. 81, 291–302, 2008.
9. Zhao, X.-W., X.-J. Dang, Y. Zhang, and C.-H. Liang, "The multilevel fast multipole algorithm for EMC analysis of multiple antennas on electrically large platforms," *Progress In Electromagnetics Research*, Vol. 69, 161–176, 2007.
10. Mittra, R. and K. Du, "Characteristic basis function method for iteration-free solution of large method of moments problems," *Progress In Electromagnetics Research B*, Vol. 6, 307–336, 2008.
11. González, I., J. Gómez, A. Tayebi, and F. Cátedra, "MONURBS: A parallelized moment method code that combines FMLMP, CBF and MPI," *Third European Conference on Antennas and Propagation*, Berlin, Germany, Mar. 23–27, 2009.
12. González, I., L. Lozano, S. Cejudo, F. Sáez de Adana, and F. Cátedra, "New version of fasant code," *IEEE Antennas and Propagation Society International Symposium, AP-S 2008*, Jul. 5–11, 2008.
13. Cátedra, F., L. Lozano, and I. González, "Fast ray-tracing for computing N-bounces between curved surfaces," *IEEE Antennas and Propagation Society International Symposium, AP-S 2008*, Jul. 5–11, 2008.
14. GiD, developments by CIMNE (The International Center for Numerical Methods in Engineering), 2005.
15. http://gid.cimne.upc.es/index.html.
16. Kinney, P., "Clean up: Improving quadrilateral finite elements meshes," *Proceedings 6th International Meshing Roundtable*, 437–447, 1997.
17. http://cubit.sandia.gov/.