

Original Research Paper

Exploring Mesh Generation and Quality Enhancement with Open Source Codes

¹Leandro Alves Neves, ¹José Márcio Machado, ¹Julio Cesar Momente,
¹Geraldo Francisco Donegá Zafalon, ¹Álvaro Magri Nogueira da Cruz,
²Marcelo Zanchetta do Nascimento and ¹Carlos Roberto Valêncio

¹Department of Computer Science and Statistics (DCCE), São Paulo State University (UNESP),
 Institute of Biosciences, Humanities and Exact Sciences (IBILCE), São José do Rio Preto, São Paulo, Brazil
²Faculty of Computing (FACOM), Federal University of Uberlândia (UFU), Uberlândia, MG, Brazil

Article history

Received: 14-03-2018

Revised: 01-06-2018

Accepted: 16-07-2018

Corresponding Author:

Leandro Alves Neves
 Department of Computer
 Science and Statistics (DCCE),
 São Paulo State University
 (UNESP), Institute of
 Biosciences, Humanities and
 Exact Sciences (IBILCE), São
 José do Rio Preto, São Paulo,
 Brazil
 Email: neves.leandro@gmail.com

Abstract: The finite element method is a very reliable and precise technique for solving partial differential equations in three-dimensional domains, with relevant applications in several areas. However, 3D simulations by FEM require computer programs for solid modeling and automatic mesh generation and there are few examples of open source codes available and dedicated to these tasks. Unfortunately, these open source codes are not usually conceived to operate together in an integrated fashion, showing distinct life cycles and different origins, which may result in contradictory specifications. In this study, a method is proposed to integrate solid modeling and automatic mesh generation with focus on open source codes and how the quality of FEM simulations can be improved by the improvement of the mesh. The method was structured in desired features for the solid construction and in integration strategies for an automatic mesh generation. The approach was tested in nontrivial domains and with known relevance for studies focused on computational electromagnetics. Meshes were generated with millions of tetrahedral elements and the results were compared to the quality values commonly discussed in literature focused on FEM. Complex geometries were meshed in a few seconds, with consistent values of aspect ratios (more than 90% of the tetrahedral elements were constructed with values at most 5) and dihedral angles (the values were bounded between 5.9 to 166.7°C, with a peak value around 90°C). Finally, in order to show the relation among highly refined meshes and quality criteria which can be explored by proposed method, the Laplace's equation was simulated by FEM in order to analyze the equipotential lines of a parallel-plate capacitor. The results show how the quality of a simulation can be improved, especially concerning the increasing number of tetrahedra in the mesh with proper aspect ratio.

Keywords: Delaunay Tetrahedralization, Solid Modeling, Open Source Codes, Integration Strategies, FEM Applications

Introduction

The Finite Element Method (FEM) has become, in the last five decades, a well consolidated technique for the treatment of partial differential equations with extensive applications in computational electromagnetics, structural analysis, heat transfer, fluid mechanics of compressible and incompressible flows. The whole potential of the FEM was not entirely explored by

researchers yet. One of the reasons for this, is that the more complex is the geometry of the problem domain, the more sophisticated are the software tools for geometric modeling and mesh discretization required by the FEM. In this context, several papers have been addressed the topic of mesh generation, for both 2 and 3-dimensional elements, in different areas (Cacace and Blöcher, 2015; Leng *et al.*, 2013; Wang and Yu, 2012; Chen and Biro, 2012; Wall *et al.*, 2012; Zhao *et al.*, 2012;

Bracken *et al.*, 2012; Ecabert *et al.*, 2011; Zhang and Kumar, 2011; Ho *et al.*, 2011; Krebs *et al.*, 2010; Sun and Chaichana, 2010; Milasinovic *et al.*, 2008; Barauskas *et al.*, 2007). In these studies, strategies of algorithms and software tools for domain construction and mesh generation, well as the performance and the availability of the tools were present issues.

In many FEM applications, almost all problem domains can be described in terms of geometric design parameters as dimensions, embrace ratios, among other, being adequate to be reproduced by applying a Computer-Aided Design (CAD) tool. The necessary commitment to represent the complex geometries of the involved domains is fundamental, which requires sophisticated resources being sometimes under development in specific tools for geometric modeling and mesh discretization (OpenFOAM Foundation, [Online]; EnGrid, [Online]). Therefore, a CAD tool and a reliable automatic mesh generator, both integrated in an user friendly environment, are essential in a FEM simulation process. Nowadays, a typical integrated software tool for three-dimensional domain construction (solid modeling) and finite element meshing may have a time development cycle of more than 10 years (Schneider *et al.*, 2016; Geuzaine and Remacle, 2009) and frequently there are exceptions to allow linking with other mesh generators.

The Delaunay algorithm, which is also applied into our approach, is able to generate meshes with high degree of topological complexity, even for problems in which the geometric domain is not regular and can be properly described only by surface smoothing techniques, which is common in biomedical modeling (Wagner *et al.*, 2016; Wang and Yu, 2012; Wall *et al.*, 2012; Ecabert *et al.*, 2011; Sun and Chaichana, 2010; Milasinovic *et al.*, 2008; Barauskas *et al.*, 2007). However, in many application contexts these difficulties might be avoided, since the geometric domains are easily reproduced by the classical methods of solid modeling, as in problems of computational electromagnetics, for instance (Specogna, 2015). The open source codes developed for these tasks are commonly conceived as separated products for the geometric modeling and mesh generation steps. These open source codes can apply different requirements which make difficult their integration in a single package. Moreover, this task can discourage users involved in problems for FEM applications due to the time required to associate software packages developed with different standards. Therefore, the conclusion is that strategies and tools for integration of open source solid modelers and mesh generators are relevant in the context of FEM applications. For instance, some preliminary approaches were described by Pavarino *et al.* (2013), Pavarino *et al.* (2014) and Neves *et al.* (2015), considering tridimensional elements for applications in medicine. However, in these

strategies the computational simulations were not performed with the tetrahedral meshes.

In this study, we described a proposal based on strategies for the 3D solid construction and script capable of providing the desired integration with an automatic mesh generator. The main contributions concern in the following two aspects: (1) The whole access to an integrated environment of open source codes, providing averages to perform the entire pre-processing step (solid modeling and mesh generation) of a FEM simulation problem; (2) FEM simulations using the proposed method to show the improvement of the results, accordingly to the improvement of the mesh, considering the increase number of tetrahedra and the quality of the aspect ratio. In the next sections, arguments are presented to justify the choices, as well as a discussion concerning the proposal for software integration, performance tests in realistic application domains and some performed simulations.

Mesh Generation Techniques: An Overview

In order to apply the FEM, the problem domain should be continuously divided into sub-domains with simple geometry (triangles, squares, cubes, tetrahedral and others), named finite elements, which must be connected without overlapping regions or failures to compose a valid mesh (Lo, 2012; Nunes *et al.*, 2011; Owen, 1998; Ho-Le, 1988). Thus, the automatic mesh generation may be classified into two main groups, structured mesh generation and unstructured mesh generation (Li *et al.*, 2016; George *et al.*, 2004; Mavriplis, 1995). Automatic methods for structured meshes employ a construction technique which follows an interconnection of parametrization curves, with these ones describing the boundary domain. In this group are the methods of algebraic grid generation, elliptic grid generation and grid marching, based on algebraic constructions or in differential mappings (Shaw and Weatherill, 1992; Cook, 1974). This approach is usually limited by domains which are not much complex. In the second group, automatic methods for unstructured meshes do not depend on the overall domain geometries, being able to control the local meshing density, as well as the transition smoothness between regions with distinct mesh densities (Guo *et al.*, 2015; George *et al.*, 2004; Mavriplis, 1995).

Since the methods in the second group are also more easily applied to the domains with more complex geometries, this group was chosen. The method was defined among the following techniques: hierarchical spatial decomposition using quadtree and octree based methods, advancing front with front propagation method and the Delaunay algorithm (Ghisi *et al.*, 2014; Ray and Adviser-Dey, 2006; Marcum and Weatherill, 1995; Weatherill and Hassam, 1994; Lo, 1985; Yerry and Shephard, 1984; Nguyen-Van-Phai, 1982). Each choice

has some advantages but also drawbacks. The hierarchical spatial decomposition is a technique that partitions a domain into variable-sized blocks based on quadtree or octree data structures. This approach is unusual for modeling domains with sharp angles (Yerry and Shephard, 1984; 1983). The advancing front algorithms work with the insertion of new elements into a mesh by following geometric constraints, which are given as input of the process. The advantage in this case is a better control of dimensions and classification of the elements in a tetrahedral mesh, but at the same time the algorithm complexity is increased, due to the verification tests for optimal point insertions (Lo, 1985; Nguyen-Van-Phai, 1982). The Delaunay algorithm is related to a geometric structure introduced in 1934 by Delaunay (Si and Shewchuk, 2014; Ray and Adviser-Dey, 2006; Marcum and Weatherill, 1995; Weatherill and Hassam, 1994), which is the Delaunay triangulation of a vertex set. The Delaunay algorithm brings many advantages for the automatic mesh generation, since it needs only the coordinates of the vertex set to work properly, not requiring orientations or detailed information about the solid structure. However, this algorithm has also some drawbacks, especially in 3D domains. For instance, when there are holes in the solids, the Delaunay tetrahedralization of a given vertex set may not conform to the original boundaries of the volume that vertex set is intended to describe. In some circumstances, zero volume elements may be generated and the software should include tests to detect these and other possible mesh integrity violations.

The decision for the Delaunay algorithm, which is the method implemented in the TetGen package, was conditioned not only by its general characteristics, especially the non-local properties of the tetrahedralization, but also by the availability of open source packages implemented on it, what is the main factor in the context of the proposed method.

Methodology

The proposed method was organized as follows: in subsection *Open source packages* are some definitions concerning the packages and the arguments to justify the corresponding choices; in subsection *Solid construction* are the required features for solid construction in order to guarantee the integration of the automatic mesh generator with the Delaunay algorithm, independently of the considered geometry complexity; in subsection *Application of the integration strategy* are the integration strategies and the script developed to obtain the tetrahedral meshes; in subsection *Application context* are the solids in order to validate the proposed method, considering the complex geometries with known relevance in studies focused on computational electromagnetics; in subsection *Performance evaluation measures* are the metrics used to evaluate the quality of

the resultant meshes, such as aspect ratio, meshing time and dihedral angles; and, in subsection *Simulation and result analysis* the Laplace's equation was simulated by FEM in order to show a practice application of the proposed method and how the quality of a simulation can be improved by the improvement of the mesh. Each step was described in detail in the next subsections, considering an applied research methodology.

Open Source Packages

There are few examples of open source packages available and dedicated to the tasks of solid modeling and 3D mesh generation. The package Blender (Blender Foundation, [Online]) was selected to perform the solid modeling step in the proposed method. This package is an integrated system of software tools with resources for modeling and animation of solids with high quality and complexity. Capabilities are also provided to export and import objects in different formats. The Blender package is available under a double License (BL/GNU general public license) and some parts of the code also support the Python as the programming language for scripts, under a Python Software Foundation License. Thus, in the proposed method, the solid structures were modeled by applying the package Blender and Python was employed to export data about boundaries and surfaces, in a convenient format, which were given as input to the mesh generation package.

The TetGen package (Si, 2013) was selected for automatic generation of tetrahedral meshes. This package uses the Delaunay tetrahedralization for the 3D mesh generation. Although there are different methods for generating three-dimensional meshes, the Delaunay algorithm (Si, 2013; Ho-Le, 1988) is one of the most popular and efficient methods (Lizier *et al.*, 2008; Buell and Bush, 1973), being applied to automatic mesh generation of three-dimensional complex domains (Buell and Bush, 1973; Lizier *et al.*, 2008; Ho-Le, 1988; Si, 2013). Thus, just as Blender, TetGen is an open source software and was considered for the integration with the Blender solid modeler. TetGen is available under MIT License and is maintained by the research group called Numerical Mathematics and Scientific Computing, Weierstrass Institute for Applied Analysis and Stochastics (WIAS), Berlin, Germany.

The approaches were implemented using the software Blender 2.59 and the mesh generator TetGen 1.4. In addition, Python language was used for writing the proposed scripts.

Solid Construction

The proper representation of the geometry of the problem domain can define the success of the mesh generation step. Thus, the method was based on three general recommendations to represent a solid with the package Blender. The first two recommendations ensure

a proper construction of the solids to be meshed, avoid possible interruptions or warnings and assure appropriate construction of tetrahedral elements by Delaunay algorithm. The third recommendation can be applied to minimize the formation of flatten elements.

First Recommendation

Each solid can be constructed with quadrilateral faces in order to reduce the number of geometrical elements (vertices and edges) required for representing a solid. The internal angles of a face comprise values in the interval from 30 to 160 degrees approximately (Fig. 1), in order to represent acute angles in boundaries. The range was defined considering the values which provided highly refined meshes with the desired quality during the mesh generation and similar conditions presented by Watson (1981), George *et al.* (1991), Caendish *et al.* (1985), Schroeder and Shephard (1988) and Baker (1989). It is important to remark that, sometimes, boundary angular deviations of great magnitude may be necessary in the solid face, where the second and third recommendations should be applied.

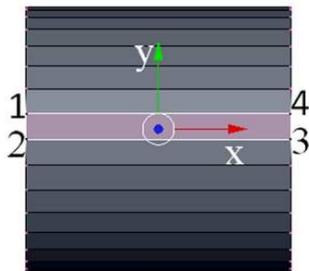


Fig. 1: Structured faces used in the solid representation. In highlighted face is shown the nodes and the internal angles, limited to values in the interval from 30 to 160°C

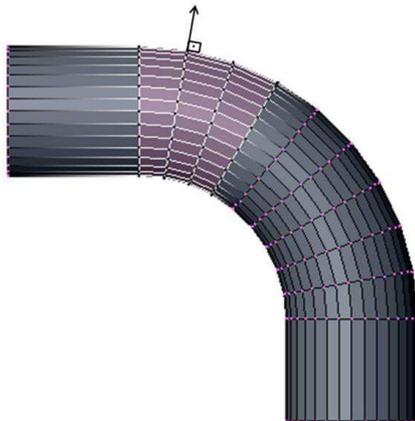


Fig. 2: Solid used to show the smoothness in the transition between different boundary regions: this recommendation was obtained by increasing the node and face densities along the curve region

Second Recommendation

The vertex number can be increased in order to discretize borders and transition of direction of an element (Fig. 2). This strategy allows a better representation and smoothness of borders, nontrivial regions and transition of direction. This strategy helps to construct consistent elements in the mesh generation step.

Third Recommendation

This is a property, which complements the second recommendation and can be applied when the internal angles of faces do not comprise values from 30 to 160 degrees. Thus, a region can be successively refined, dividing existing faces and insert new vertices, to accomplish this interval. Each “subdivide surface” is applied on selected edges and faces by cutting them in half, adding the necessary vertices and subdividing accordingly the faces involved. For instance, when four edges of a face (a quadrilateral face) are selected, the face is subdivided into four smaller quadrilaterals (Fig. 3). Therefore, a new distribution is obtained and can assure more consistent tetrahedral elements. This approach may be justified for instance in Watson (1981), George *et al.* (1991), Caendish *et al.* (1985), Schroeder and Shephard (1988), Baker (1989).

Notice, however, that a limitation is imposed here by precision factors. The successive refinements may generate faces with very small areas, what can start a detection of superimposed vertices by the Delaunay algorithm. These regions with transition of direction can occur after some successive refinements. For instance, transition of direction after three successive refinements is shown in Fig. 4.

The proposed recommendations ensure a finite number of points of a domain Ω to represent in an approximate way, however acceptable, the solid boundary, its shell and its inner region. Each obtained point set was given as input for the Delaunay tetrahedralization, which was performed through the approach described by Si and Gartner (2005). The automatic mesh generation was possible considering a simplex K as a sorted vertex list P_i , where $1 \leq i \leq d + 1$ and d the dimension of Euclidean affine space E (Weatherill and Hassam, 1994; Schroeder and Shephard, 1988; Baker, 1989). Thus, the determinant $\det(K)$ of order $d + 1$, was given by Equation 1:

$$\det(K) = \begin{vmatrix} 1 & \dots & \dots & \dots & 1 \\ P_1^1 & \dots & \dots & \dots & P_{d+1}^1 \\ P_1^2 & \dots & \dots & \dots & P_{d+1}^2 \\ \vdots & \dots & \dots & \dots & \vdots \\ P_1^d & \dots & \dots & \dots & P_{d+1}^d \end{vmatrix} \quad (1)$$

where, P_i^j was the coordinate j of point P_i to $1 \leq i \leq d + 1$ and $1 \leq j \leq d$.

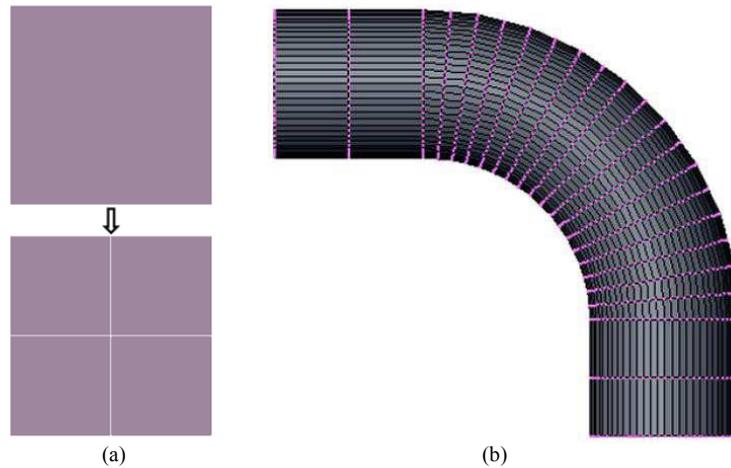


Fig. 3: Solids used to illustrate the third recommendation. The refinement process for a single face is shown in (a) and the corresponding overall result in the solid is presented in (b)

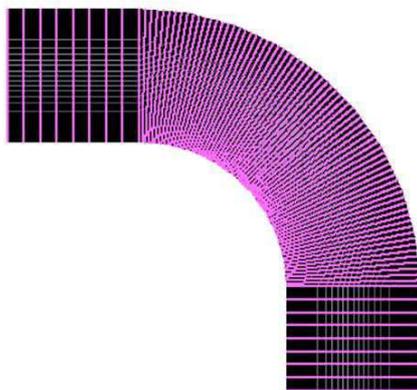


Fig. 4: Result after three successive refinements, which were applied in the original solid shown in Fig. 2. The refined region presents a higher density of separation lines

Therefore, if the $\det(K) > 0$, K is positively oriented and, if $\det(k) = 0$, K is degenerate and all its vertices belong to the same hyperplane (George and Hermeline, 1992). The proposed recommendations in this study ensure K positively oriented to obtain the desired tetrahedralization. The tetrahedralization was extended as a ζ set of simplexes which were achieved from a domain Ω used as input. This approach was possible considering: the intersection of two elements of ζ was empty or reduced to one face; the union of the elements of ζ was equal to Ω ; the elements of ζ were topologically regular, but not necessarily equal; and, the incidence of elements was greater in the regions of domain Ω where this was necessary. The optimal number of vertices or tetrahedral elements depends on some parameters, such as relation between the volumes from the domain and tetrahedra. For instance, considering a bound enforced to the maximum volume on every tetrahedron of the mesh and reducing this bound, the results were meshes involving

more elements (vertices and tetrahedra). This process is named as mesh refinement. Complementary details concerning the maximum tetrahedron volume constraint are also discussed by Shewchuk (1998) and Si (2013).

Application of the Integration Strategy

A simplex or complex solid constructed through Blender is stored in its native file format as *.blend*. The information concerning each constructed solid is stored in this kind of file container, designated as a *datablock*. *Datablocks* are classes of data structures of the type *object*, *mesh*, *material*, *camera*, *texture*, *lamp* and others. In a simple or complex solid, several *datablocks* may be involved. For instance, an *object datablock* contains a pointer to a *geometric datablock*, such as mesh, a matrix which stores the transformation properties, as location, rotation, scale and dimension and optionally pointers to other *datablocks*. *Mesh datablock* stores information about vertices, edges, faces, holes and vertex groups, which are achieved from the operations assigned by an *object datablock*. However, linear transformations applied over vertices are contained in *mesh datablocks*. A *material datablock* contains information about colors, specular and translucency, which can be associated to a *mesh datablock*. In this case, an *object datablock* is the parent data structure of another datablock, which in its turn can have some children data structures and so on, resulting in a hierarchy scheme or tree. In Fig. 5 are presented the structures described for material, mesh and object datablocks, based on Blender architecture.

The TetGen input was written in ASCII pattern, with four separate parts. The first part is an indexed list containing the point coordinates, which represent the solid geometry as nodes or solid vertices. The second part is a list of solid faces, each face being structured by a sequence of connected points, the face components.

The indexes in the first part of the input file allow the identification of the nodes. The third part contains a list of volume holes. These holes are identified by points inside them. The TetGen package initially forms the constrained Delaunay tetrahedralization and then creates inner spaces by removing properly the tetrahedral inside the mesh, following the boundaries of holes belonging to the third part list. The fourth part is not mandatory and may contain a list of attributes or constraints which can be applied to the mesh of a given domain sub-region. It is possible for the user to specify the maximum tetrahedral volume, which is desirable for a given region, in this optional fourth part. In Fig. 6 is shown a typical TetGen file input.

The desired integration was accomplished by Python script, which was developed to integrate the file structures previously described, such as the 3D domains generated by Blender with the input data structures of the mesh generator TetGen. This script comprises code

segments developed to access the *material*, *mesh* and *object datablocks*. The Python script actions into these *datablocks* were controlled by four iterative structures. These structures read and translate conveniently the information contained in datablocks, which describes vertices, faces, holes and attributes of the domain. The processed information was written in an ASCII file (*.poly*), according to the format required by TetGen (Fig. 6). It is important to notice that a new contribution is presented here, an algorithm capable of providing the meshing of mixed solids: the proposed method automatically executes the process for each member in the group of existing solids, updating the lists and generating a single final ASCII file (*.poly*). In Fig. 7 is shown the integration strategies written in Python language. Also, in Fig. 8 is shown the hierarchy scheme (*material*, *mesh* and *object datablocks*) using a simple cube as example. The *.poly* file format is given from the integration strategy application.

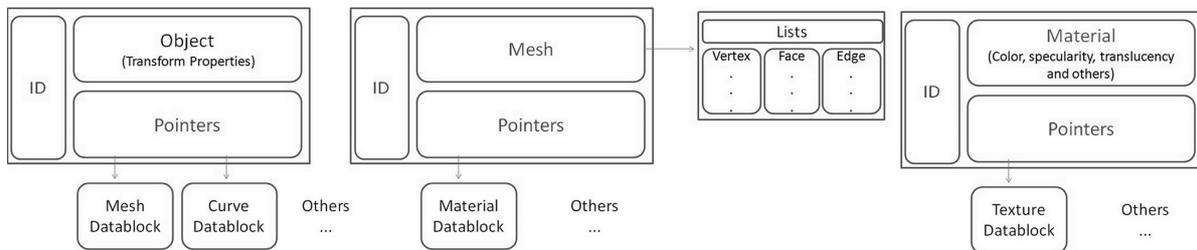


Fig. 5: A representation of the structures contained in material, mesh and object datablocks

Part 1 - node list

```
First line: <# of points> <dimension (3)> <# of attributes>
            <boundary markers (0 or 1)>
Remaining lines list # of points:
    <point #> <x> <y> <z> [attributes] [boundary marker]
...

```

Part 2 - facet list

```
One line: <# of facets> <boundary markers (0 or 1)>
Following lines list # of facets:
    <facet #>
...

```

Part 3 - (volume) hole list

```
One line: <# of holes>
Following lines list # of holes:
    <hole #> <x> <y> <z>
...

```

Part 4 - region attributes list

```
One line: <# of region>
Following lines list # of region attributes:
    <region #> <x> <y> <z> <region number> <region attribute>
...

```

Fig. 6: A Typical file structure required as input by the TetGen package (Si, 2013)

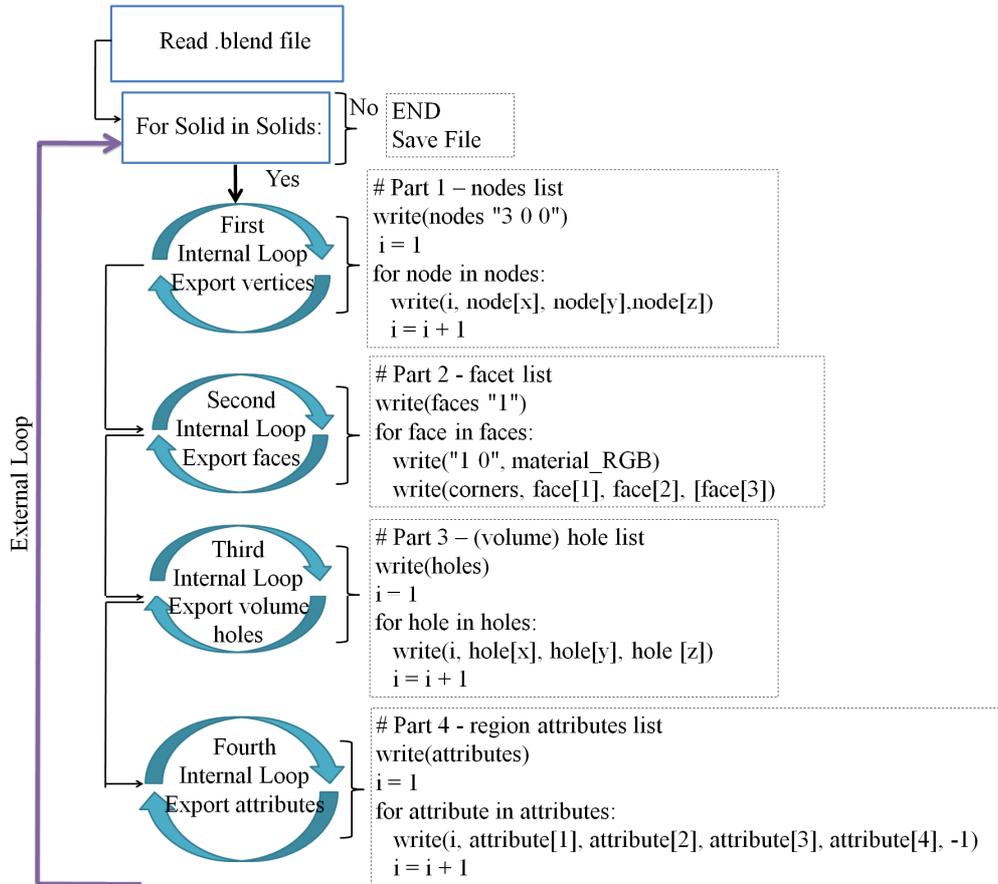
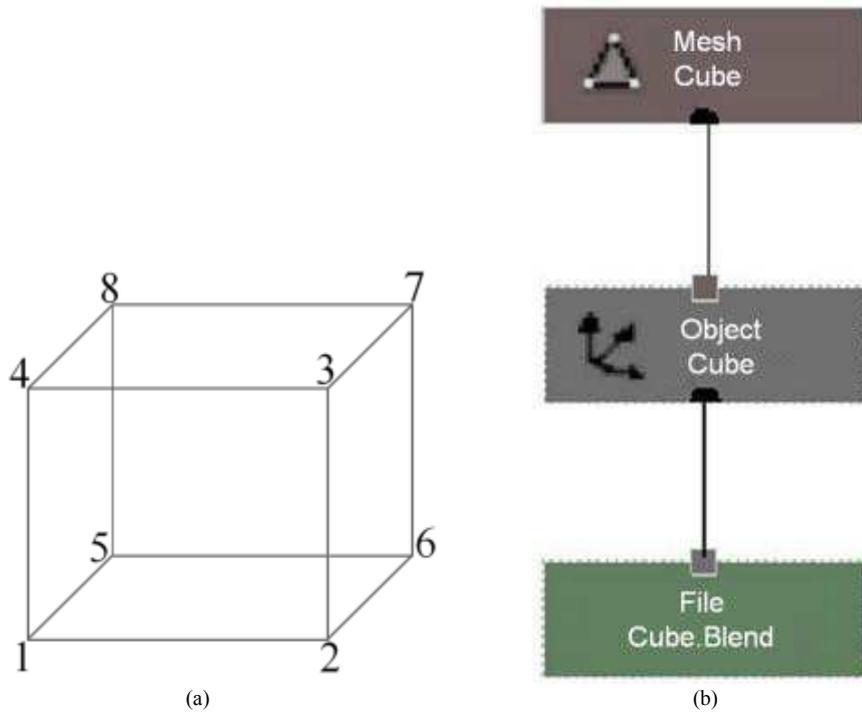


Fig. 7: Proposed method used to export a group of existing solids and generates a .poly file format



```
# Part 1 - node list
# node count, 3 dim, no attribute, no boundary marker
8 3 0 0
# Node index, node coordinates
1 2.0 2.0 0.0
2 2.0 0.0 0.0
3 0.0 0.0 0.0
4 0.0 2.0 0.0
5 2.0 2.0 2.0
6 2.0 0.0 2.0
7 0.0 0.0 2.0
8 0.0 2.0 2.0

# Part 2 - facet list
# Facet count, no boundary marker
6 0
# facets
1 0 0 #1 polygon, no hole, no boundary marker
4 1 2 3 4
1 0 0
4 5 8 7 6
1 0 0
4 1 5 6 2
1 0 0
4 2 6 7 3
1 0 0
4 3 7 8 4
1 0 0
4 5 1 4 8

# Part 3 - hole list
0 # no hole

# Part 4 - region list
0 # no region
```

(c)

Fig. 8: Cube ($2 \times 2 \times 2$) without material is shown in (a), the corresponding hierarchy scheme for datablocks is shown in (b) and the *.poly* file format obtained with the proposed method is shown in (c)

Application Context

The proposed method was tested considering solids with complex geometries: “C-type” magnet, turbo-compressor device, Klein bottle and an electrical motor with supplementary permanent magnets. These choices were motivated by the several studies focused on computational electromagnetics considering similar geometries (Li *et al.*, 2016; Chen and Biro, 2012; Wall *et al.*, 2012; Zhao *et al.*, 2012; Bracken *et al.*, 2012; Zhang and Kumar, 2011; Ho *et al.*, 2011; Chang *et al.*, 2010; Jang *et al.*, 2007; Cho *et al.*, 2006).

Performance Evaluation Measures

Several quality metrics have been used to measure the quality of tetrahedral meshes (Chen and Chen, 2016; Paille *et al.*, 2015; Leng *et al.*, 2013; Misztal *et al.*, 2009; Si, 2013), such as the dihedral angles, aspect ratio, number of tetrahedral elements and meshing time. These features can significantly affect the convergence and

stability of numerical algorithms such as the FEM (Shewchuk, 2002; Babuvska and Aziz, 1976). Therefore, the obtained meshes with the proposed method were tested against these quality metrics.

The dihedral angles are the angles between two faces of a tetrahedron, defining its shape. Elements with very small and large dihedral angles should be avoided since they usually downgrade the accuracy and performance of numerical methods. The optimal desired value should be close to the dihedral angle of a regular tetrahedron, as discussed by Labelle and Shewchuk (2007). Therefore, histograms were constructed to show the total of dihedral angles inside some prescribed intervals, considering the smallest and the largest dihedral angle (Wang and Yu, 2012; Labelle and Shewchuk, 2007). In addition to the dihedral angles, another measure was the aspect ratio Q of an element, which can be defined as the ratio of the radius of its circumsphere to the radius of its inscribed

The proposed method allows automatic mesh generation and successive refinements of nontrivial domains. These relevant features are required in studies focused on computational electromagnetics. Therefore, all the mesh examples discussed in this study were accomplished from a coarse initial mesh, which was refined in repeated steps of node inclusions and mesh reconstruction. This approach allowed calculating the processing time growth as a function of the increasing number of tetrahedral elements. The result was the meshing time considered as another quality metric.

The solid construction and the mesh generation were both performed in a computer with a 1.87 GHz quad core processor, 4 GB of RAM memory and an operating system with 32 bit architecture. Although the considered system have a multi-core processor, the open source mesh generator applied in the tests is not available to use automatically the multiple cores. Therefore, the performance measures were calculated using one CPU core. In Fig. 9 is shown a summary of the proposed method with all steps previously described.

Simulation and Result Analysis

A FEM simulation was performed in order to analyze and verify the usefulness of a highly refined mesh, as well as to show a practice application of the proposed method and how the quality of a simulation can be improved by the improvement of the mesh. The Laplace's equation, Equation 3, where V is the electric potential, was simulated in order to analyze the equipotential lines of a parallel-plate capacitor. Laplace's equation is suitable for this simulation since there is no free charges in the analyzed region.

FEM is one of the most appropriate methods to perform this analysis due to its accuracy and it also make possible the comparison between the meshes, from less refined to highly refined ones.

$$\nabla^2 V = 0 \quad (3)$$

The results from the simulations were analyzed by cutting a plane in the model and calculating the electric potential in each element the cross or touch that plane.

Results and Discussion

The solid construction and automatic mesh generation were performed using the proposed method. To clarify the understanding and discussion of the results, in subsection *Tetrahedralization and mesh refinements of nontrivial domains* are presented the raw and highly refined meshes which were constructed with the proposed method; and, in subsection *Mesh quality* are discussed the quality values used to evaluate the tetrahedral meshes, such as meshing time, aspect ratio and dihedral angles.

Tetrahedralization and Mesh Refinements of Nontrivial Domains

A "C-type" magnet was used to demonstrate the obtained results with the proposed method and validate the integration strategy. The obtained solid with application of the first two recommendations (subsection "Solid construction") is shown in Fig. 10a. The smoothness in the transition boundaries is clearly illustrated. Also, the required hierarchical data structures by "C-type" magnet are presented in Fig. 10b in order to show the 6 datablocks which were accessed by the proposed script. Each part of the "C-type" magnet requires 3 datablock types. Thus, in Fig. 11a is shown the test solid with the corresponding raw mesh, used to validate the concept of iteration loop described in subsection "Application of the integration strategy". The "C-type" magnet was meshed with 951 vertices and 2,698 tetrahedral elements. An application proof for the third recommendation is illustrated in Fig. 12a, in which a new and refined, set of vertices was defined, when compared to raw mesh shown in Fig. 11. The result was an updated redistribution of the original solid volume and more tetrahedra were generated, resulting in 3,063 vertices and 8,810 tetrahedral elements (Fig. 12b and 12c).

The topology domain may be easily modified and enlarged to reproduce classical geometries of power transformers and electromagnets (Duan *et al.*, 2015; Chen and Biro, 2012; Zhao *et al.*, 2012; Krebs *et al.*, 2010; Ho *et al.*, 2011; Zhang and Kumar, 2011; Chang *et al.*, 2010). For instance, in Fig. 13 to 15 are shown a kind of turbo-compressor device, a Klein bottle and an electrical motor with permanent magnet supplementary, respectively. The turbo-compressor was meshed in 1.60 seconds, involving 13,830 nodes and 42,821 tetrahedral elements. In this example, a combustion chamber and a turbine were connected through manifolds. Similar

structures, based in the original concept of centrifugal engines, are also used in jet propulsion applications (Jang *et al.*, 2007; Cho *et al.*, 2006). The Klein bottle was meshed with 8,420 nodes and 25,676 tetrahedral elements, in 1.01 sec. Similar topology was used in studies focused on symmetry in meta-materials (Chang *et al.*, 2010). The electrical motor with permanent magnet supplementary was constructed with 5,584 and 16,176 tetrahedral elements. The meshing time was of 0.75 seconds. This topology was explored by Zhao *et al.* (2012) in order to analyze electromagnetic devices, considering a parameterized mesh generation and refinement method for finite elements.

Examples of *.poly* files, are partially shown in Fig. 16, since is not feasible to give them in their integrity due to

huge file sizes. These data were automatically written by the proposed script. For instance, the *.poly* file for the turbo compressor case comprises a set of 6,638 vertices, 6,676 faces and 7 color markers. The markers are numerical codes, which can be applied to assign different colors to some vertices in order to better distinguish the domain boundaries in adaptive refinement of the mesh. For instance, the numerical code -417148 was used to assign the red color to the vertices of the device part designated as combustion chamber. The Klein bottle comprises 3,460 vertices, 3,462 faces and only the color marker -454880 (green). The hierarchical data structure of the turbo-compressor device comprises 21 datablocks and the Klein bottle considers three datablocks. These hierarchical data structures are shown in Fig. 17 and 18, respectively.

All the mesh examples previously presented were successively refined. For instance, highly refined meshes are shown in Fig. 19 to 22, in which are illustrated the “C-type” magnet, turbo-compressor device, Klein bottle and electrical motor with supplementary permanent, respectively. The “C-type” magnet was generated with 307,327 vertices and 1,788,145 tetrahedra. The turbo-compressor device was generated with 349,421 vertices and 1,838,948 tetrahedral elements. The Klein bottle was constructed with 344,534 vertices and 1,774,489 tetrahedra. The electrical motor with supplementary permanent was meshed with 453,806 vertices and 2,753,985 tetrahedral elements. Also, in Fig. 23 is shown the running time as a function of the increasing number of tetrahedra for each example previously presented.

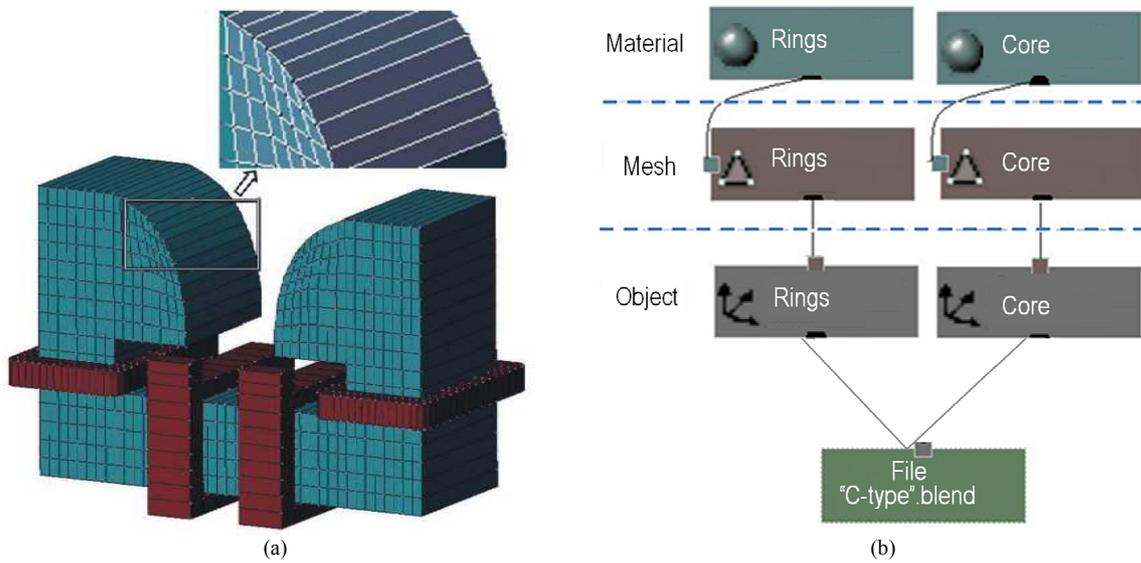


Fig. 10: “C-type” magnet used to validate the first two recommendations is shown in (a) and the required hierarchical datablocks (material, mesh and object) are presented in (b)

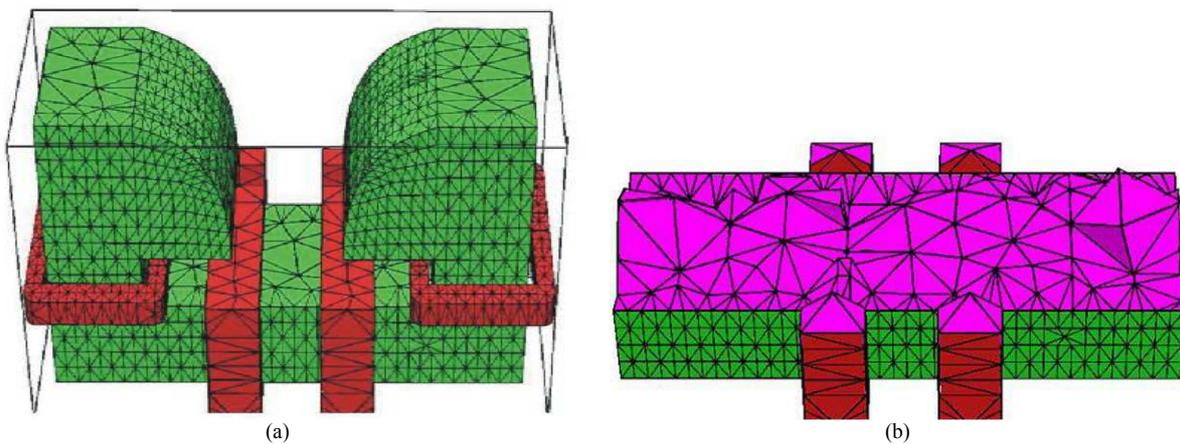


Fig. 11: The meshed “C-type” magnet is shown in (a) and a plane cut is illustrated in (b)

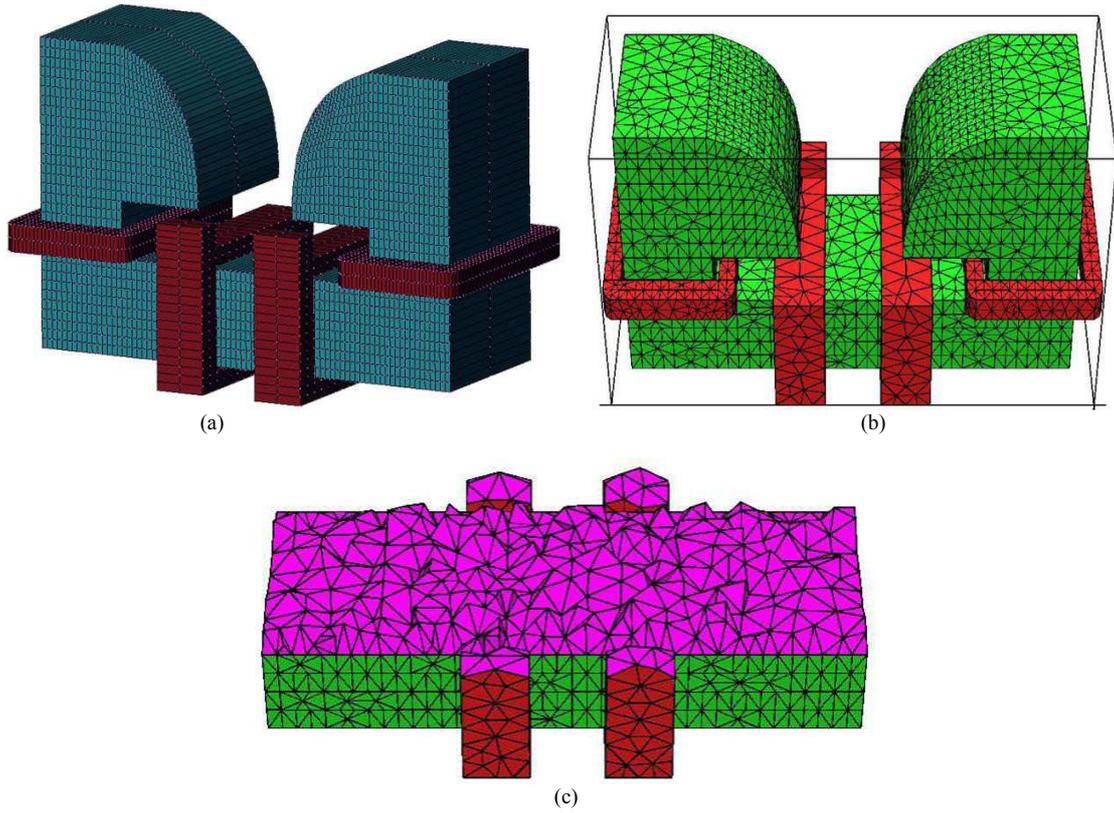


Fig. 12: “C-type” magnet constructed with the third recommendation (subsection “Solid construction”) is shown in (a), the corresponding finite element mesh is presented in (b) and a plane cut is illustrated in (c)

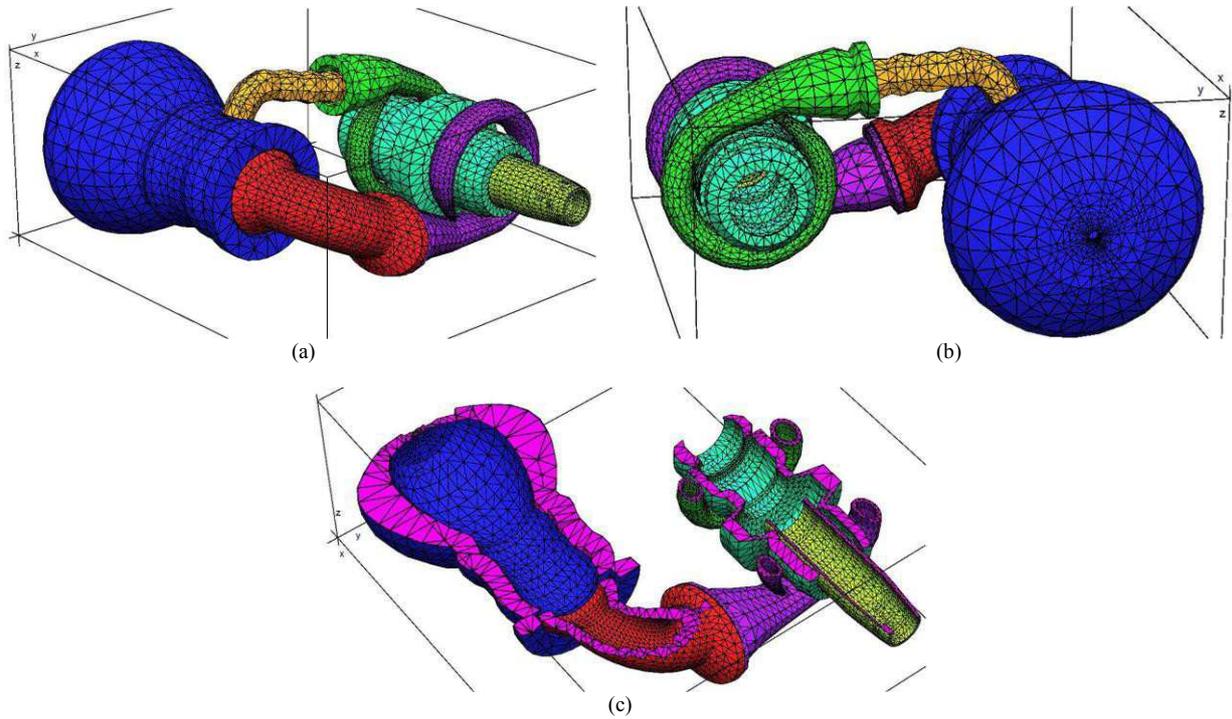


Fig. 13: Turbo-compressor meshed in 1.60 sec is shown with different views in (a) and (b) and details of the tetrahedral elements after a plane cut is presented (c)

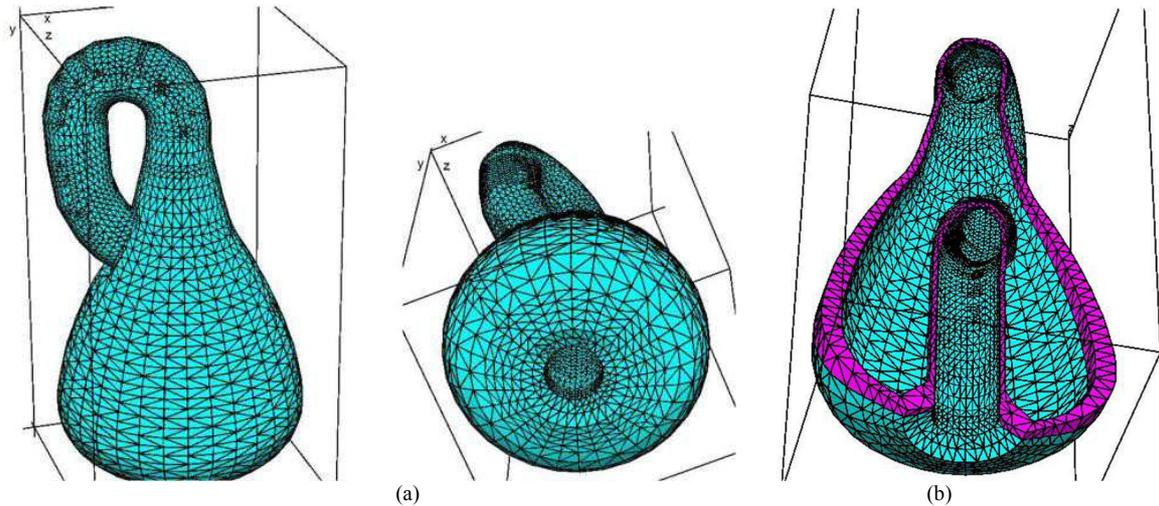


Fig. 14: Klein bottle meshed in 1.01 sec is shown with different views in (a) and details of the tetrahedral elements after a plane cut is presented (b)

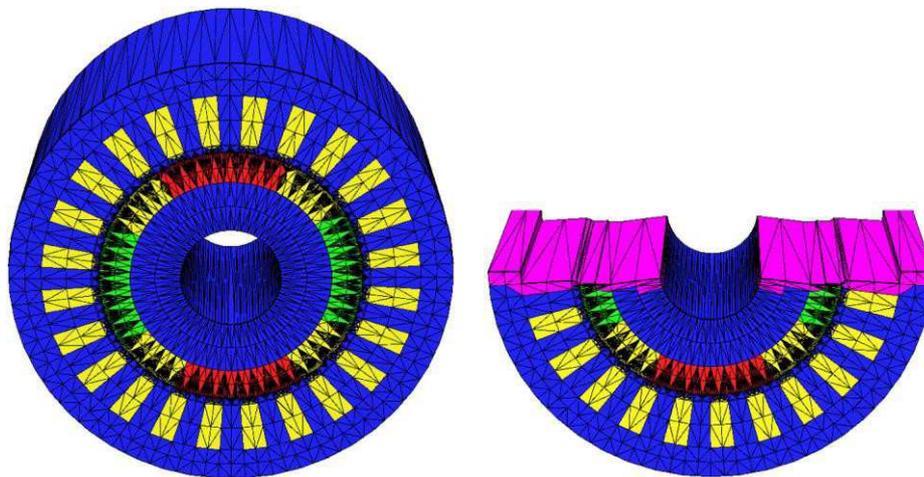


Fig. 15: Electrical motor with permanent magnet supplementary poles meshed in 0.75 sec is shown in (a) and details of the tetrahedral elements after a plane cut is presented in (b)

Mesh quality

Several mesh generation approaches have been explored on studies considering computational electromagnetics (Moreno *et al.*, 2015; Leng *et al.*, 2013; Wang and Yu, 2012), but they are not based on integration of open source packages yet. This can be a limitation for comparisons of the results presented here. However, the proposed method was evaluated considering different quality metrics, such as aspect ratio and dihedral angles of the tetrahedra. Moreover, the improvement in the quality has a direct relation to the number of elements and, therefore, in the meshing time.

A well shaped tetrahedron has aspect ratio defined as 1.00 (Si, 2013), but a bounded aspect ratio at most five also was discussed in detail by Ruppert (1995).

Therefore, to ensure accurate results, a well shaped tetrahedron has a low aspect ratio (Gerritsen *et al.*, 2015; Cheng, 2006). Considering these values and each mesh explored here, histograms are shown for both the raw and highly refined meshes: “C-type” magnet (Fig. 24), turbo-compressor device (Fig. 25), Klein bottle (Fig. 26) and electrical motor with supplementary permanent (Fig. 27). This strategy allows verifying aspect ratios more appropriate in refined meshes than raw meshes. For instance, more than 90% of the tetrahedral elements were constructed with values at most five and mainly closer to ideal condition. The best numerical accuracy is achieved by a mesh with uniform perfect tetrahedral elements. However, considering nontrivial domains as explored here, it is not possible to generate meshes only with perfect tetrahedra, due to both multiply connected

sub-domains and thin and curved regions (Field, 1986). Thus, the proposed method is able to provide consistent values, mainly when compared to those presented in

different studies (Cheng, 2006; Si, 2013; Shewchuk, 1998; Ruppert, 1995) and with the advantage of using open source packages.

```

# Part 1 - node list
# node count, 3 dim, no attribute, no boundary marker
6638 3 0 0
# Node index, node coordinates
1 0.361 -0.433 0.324
2 0.361 -0.4 0.301
3 0.361 -0.364 0.284
.
.
6637 -0.731 -0.529 0.362
6638 -0.731 -0.502 0.319

# Part 2 - facet list
# Facet count
6676 1
# facets
1 0 -926353 #1 polygon, no hole, boundary marker (-926353)
4 3 62 63 2
1 0 -926353
4 4 61 62 3
1 0 -926353
4 5 60 61 4
.
.
1 0 -820000
4 6515 6314 6315 6514
1 0 -820000
4 6514 6315 6316 6513

# Part 3 - hole list
0 # no hole

# Part 4 - region list
0 # no region

# Part 1 - node list
# node count, 3 dim, no attribute, no boundary marker
3460 3 0 0
# Node index, node coordinates
1 0.825 0.604 -2.183
2 0.777 0.702 -2.183
3 0.712 0.788 -2.183
.
.
3459 0.286 0.092 -2.057
3460 0.346 0.096 -2.057

# Part 2 - facet list
# Facet count
3462 1
# facets
1 0 -454880 #1 polygon, no hole, boundary marker (-454880)
4 1 61 33 2
1 0 -454880
4 61 62 34 33
1 0 -454880
4 62 63 35 34
.
.
1 0 -454880
4 3458 3459 23 22
1 0 -454880
4 3459 3460 24 23

# Part 3 - hole list
0 # no hole

# Part 4 - region list
0 # no region
    
```

(a)

(b)

Fig. 16: Structures of .poly files for a part of the turbo-compressor are shown in (a) and the structure of .poly files for the entire Klein bottle is presented in (b)

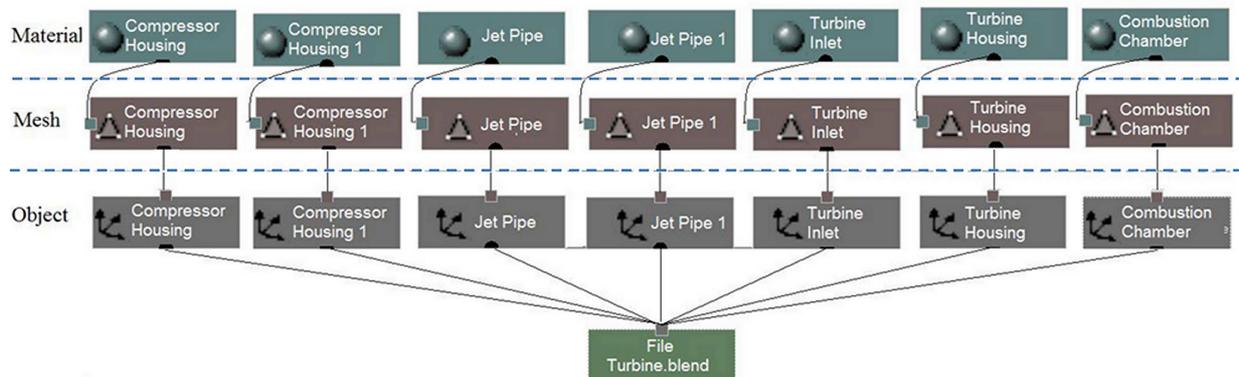


Fig. 17: Hierarchical datablocks obtained from the turbo-compressor representation, in which each part of the turbo-compressor required three datablock types: Material, mesh and object

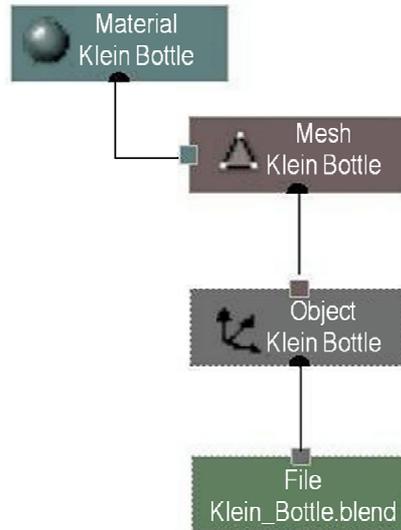


Fig. 18: Hierarchical datablocks obtained from the Klein bottle representation, which comprises the material, mesh and object datablocks

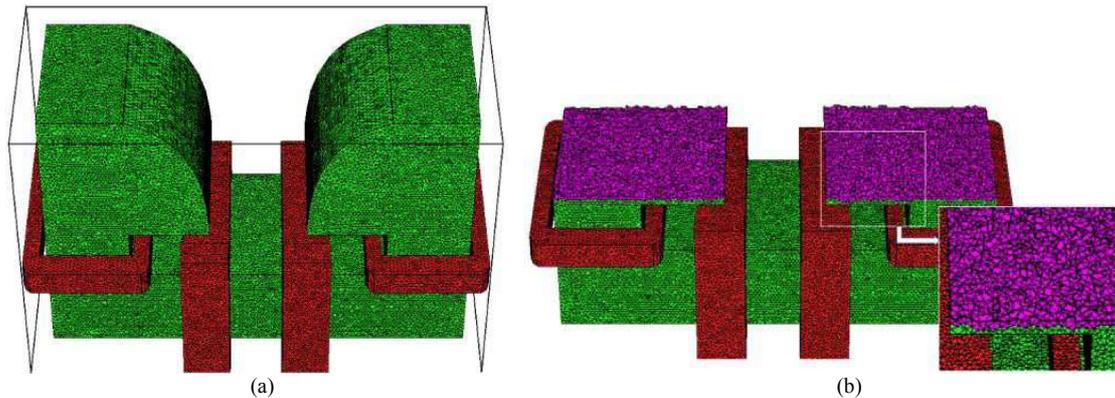


Fig. 19: “C-type” magnet with high degree of refinement, which was meshed in 39.60 sec, is shown in (a) and a “zoom view” detail is given in (b), according to the area enclosed by the highlighted rectangle

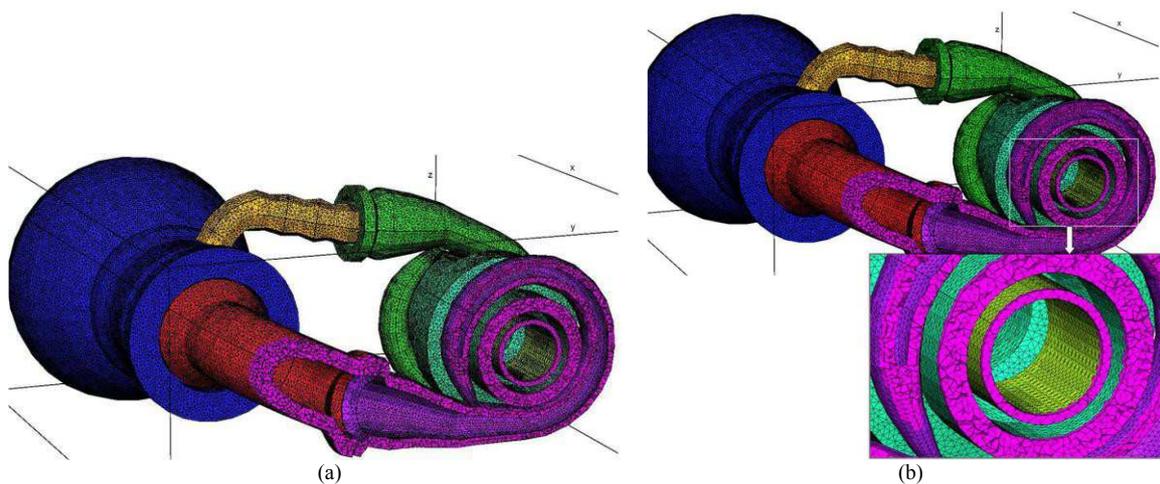


Fig. 20: Turbo-compressor device with high degree of refinement, which was meshed in 44.60 sec, is shown in (a) and a cross sectional cut is presented in (b)

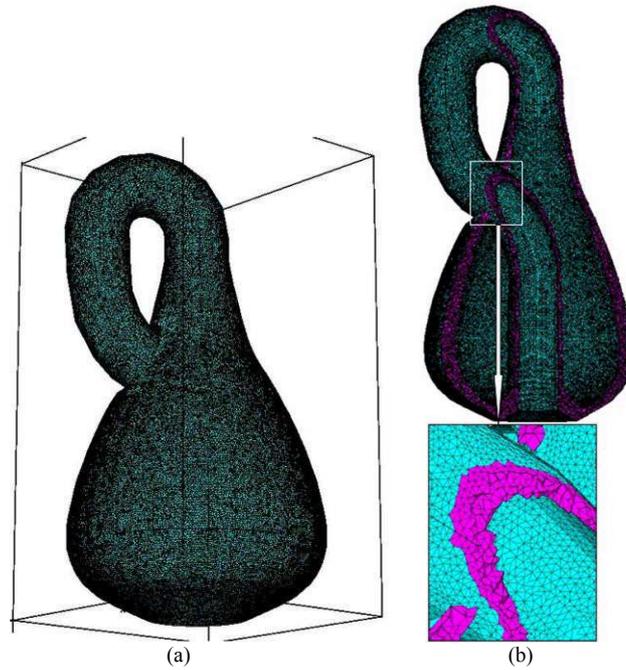


Fig. 21: Klein bottle with high degree of refinement, which was meshed in 45.30 sec, is shown in (a) and a cross sectional cut is given in (b)

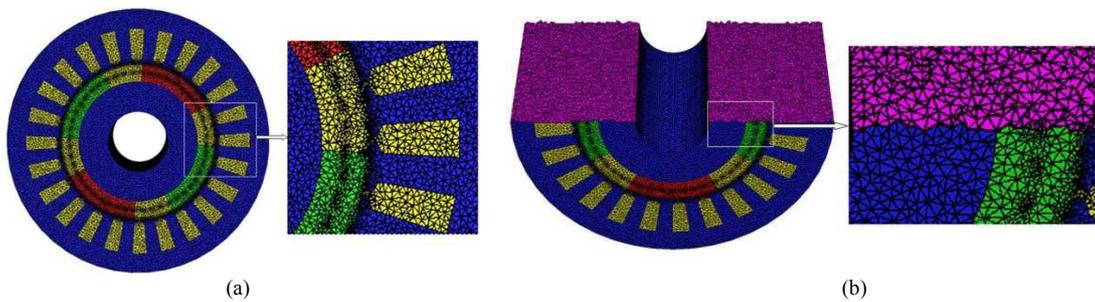
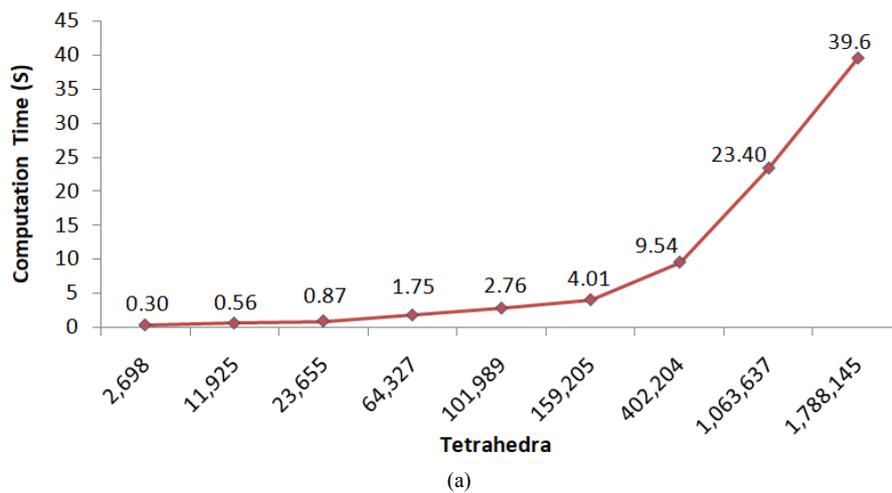


Fig. 22: Electrical motor with supplementary permanent magnets, which was meshed in 60.60 sec, is shown in (a) and a cross sectional cut is illustrated in (b)



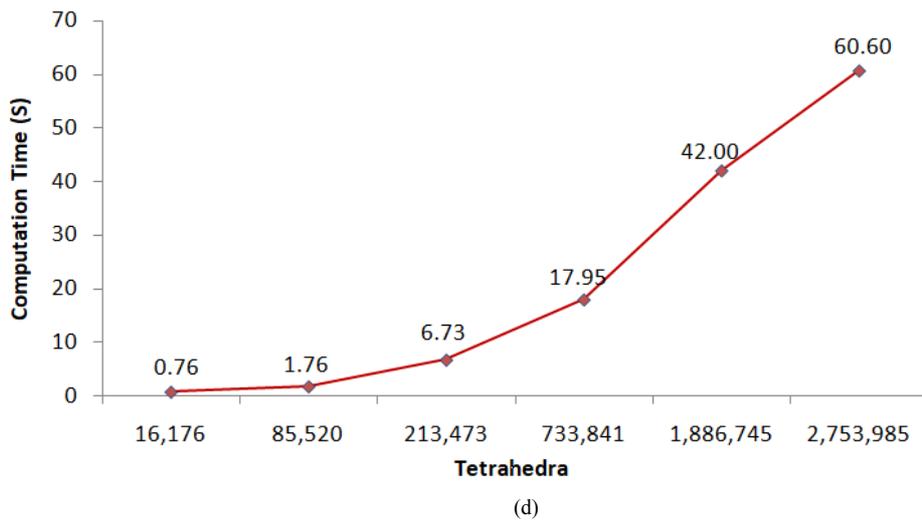
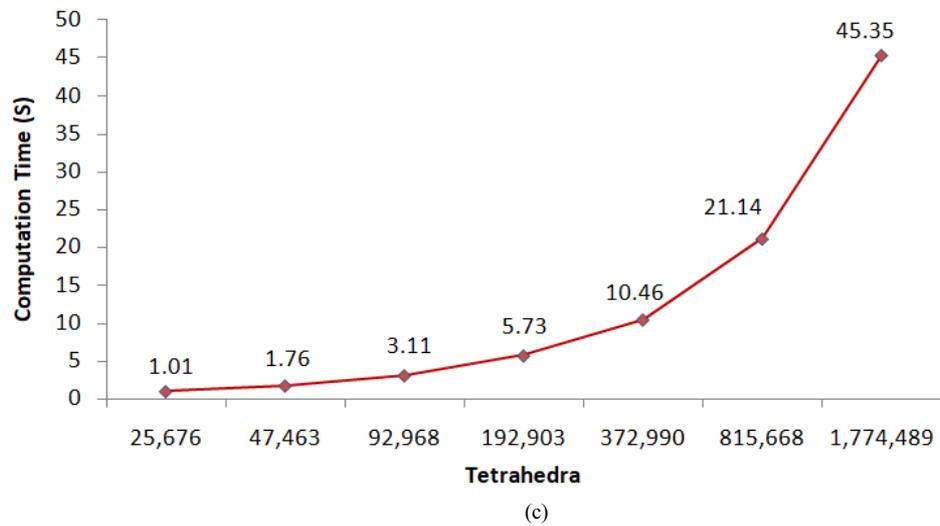
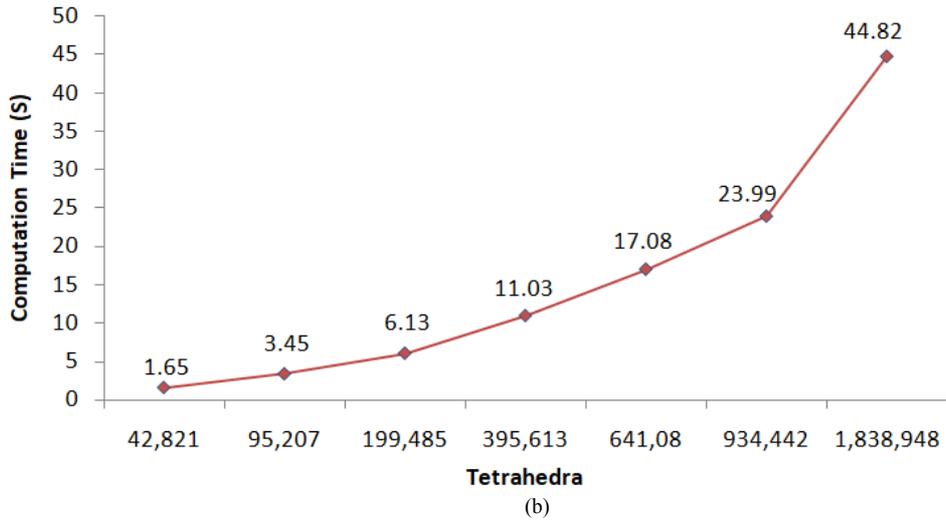
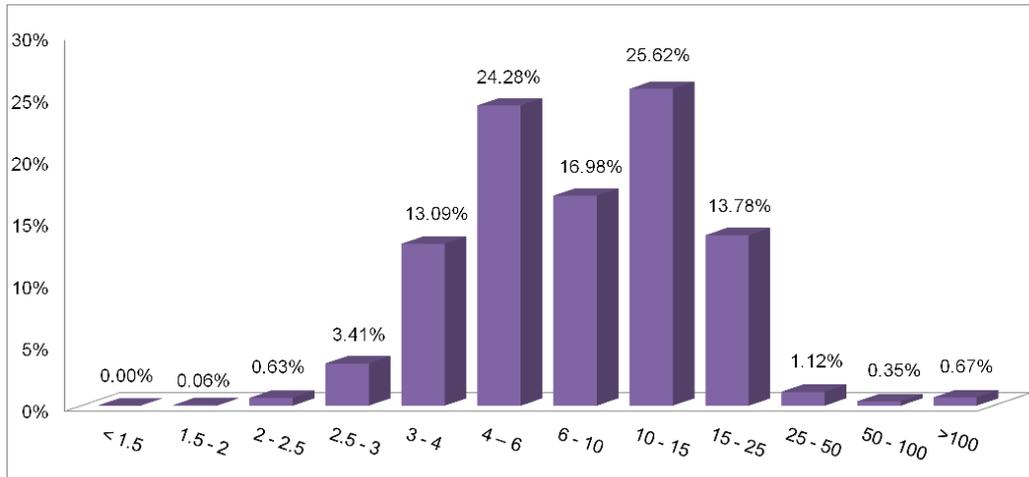
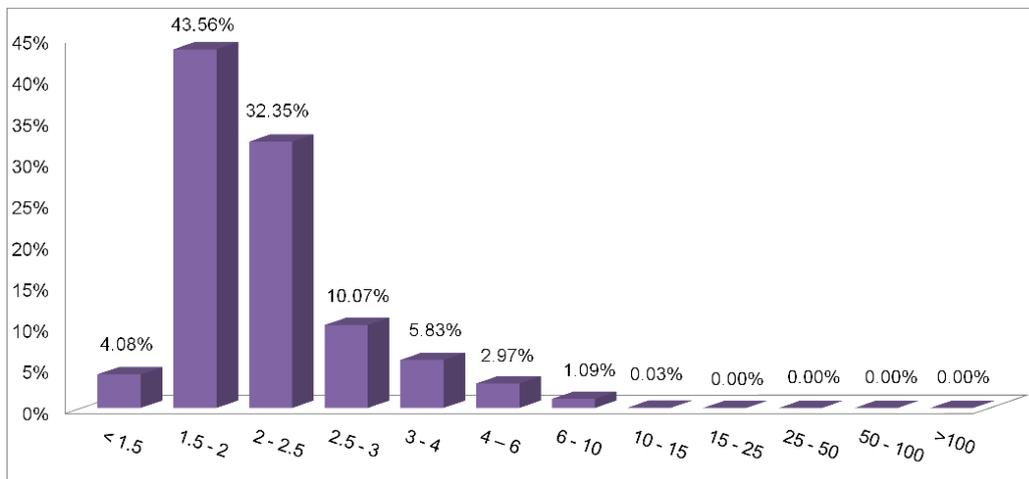


Fig. 23: Histograms considering meshing time in seconds (y axis) as a function of the number of elements in the tetrahedral mesh (x axis) are shown for the “C-type” magnet (a), turbo-compressor device (b), Klein bottle (c) and electrical motor (d)

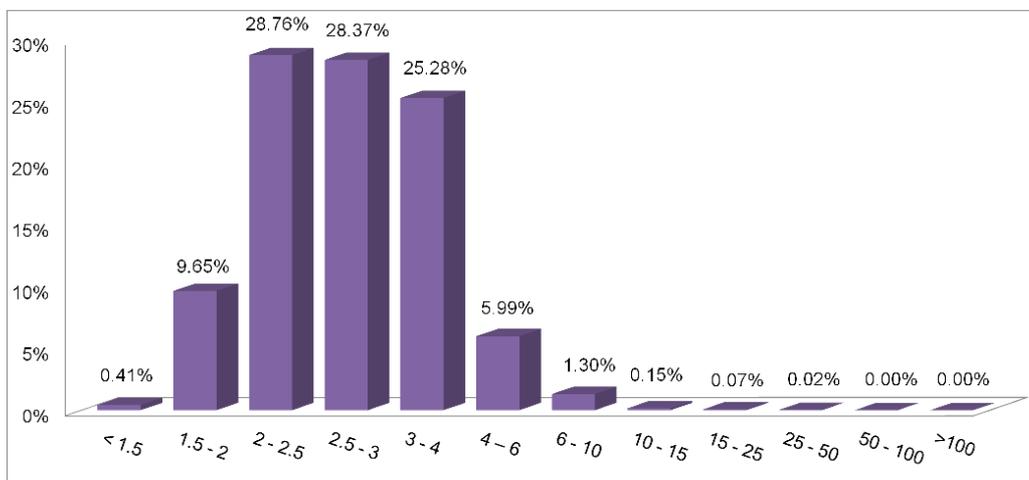


(a)

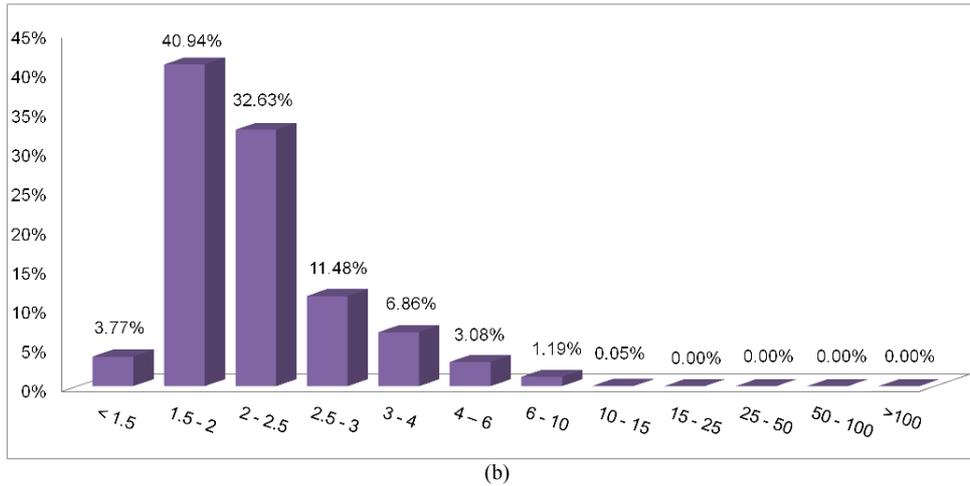


(b)

Fig. 24: Aspect ratio histograms for the “C-type” magnet, considering the total of tetrahedra (y axis) inside of prescribed aspect ratio intervals (x axis). In (a) is presented the histogram calculated from the raw mesh (2,698 tetrahedral elements - Fig. 11) and in (b) is shown the obtained histogram from the mesh with high degree of refinement (1,788,145 tetrahedral elements - Fig. 19)

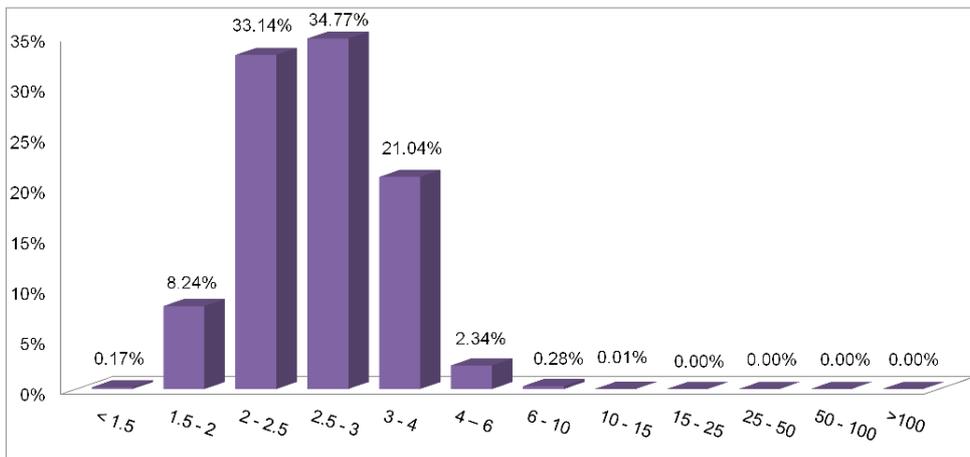


(a)

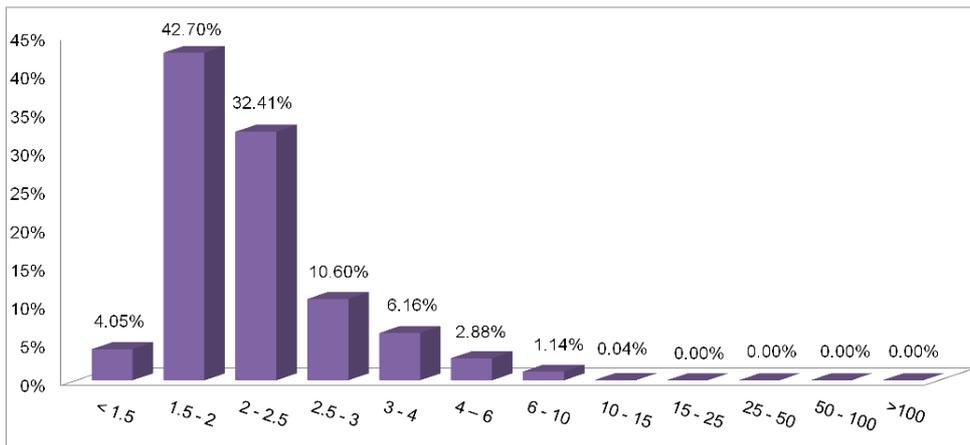


(b)

Fig. 25: Aspect ratio histograms for the turbo-compressor device, with the total of tetrahedra (y axis) inside of prescribed aspect ratio intervals (x axis). In (a) is shown the histogram defined from the raw mesh (42,821 tetrahedral elements - Fig. 13) and in (b) is illustrated the obtained histogram from the mesh with high degree of refinement (1,838,948 tetrahedra - Fig. 20)



(a)



(b)

Fig. 26: Aspect ratio histograms for the Klein bottle, illustrating the total of tetrahedra (y axis) inside of prescribed aspect ratio intervals (x axis). In (a) is presented the obtained histogram from the raw mesh (25,676 tetrahedral elements Fig. 14) and in (b) is shown the calculated histogram from the mesh with high degree of refinement (1,774,489 tetrahedra - Fig. 21)

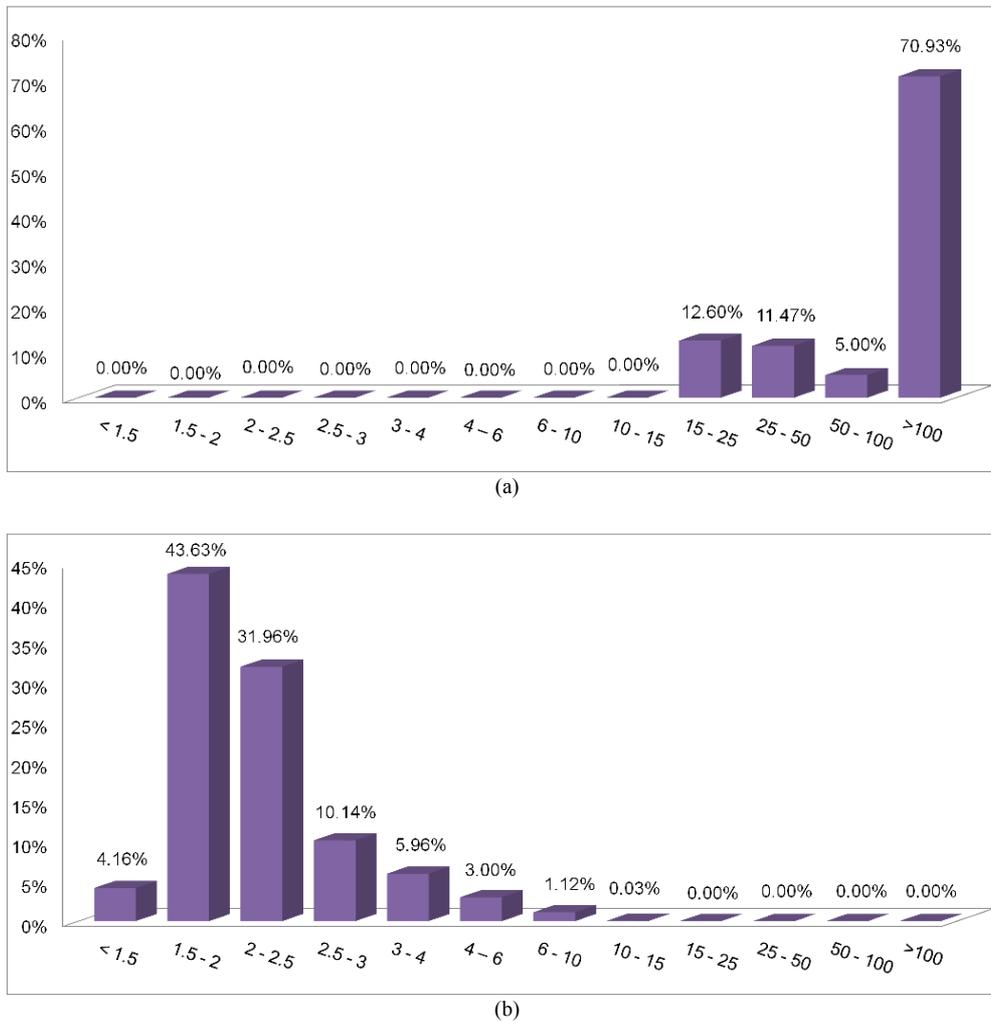
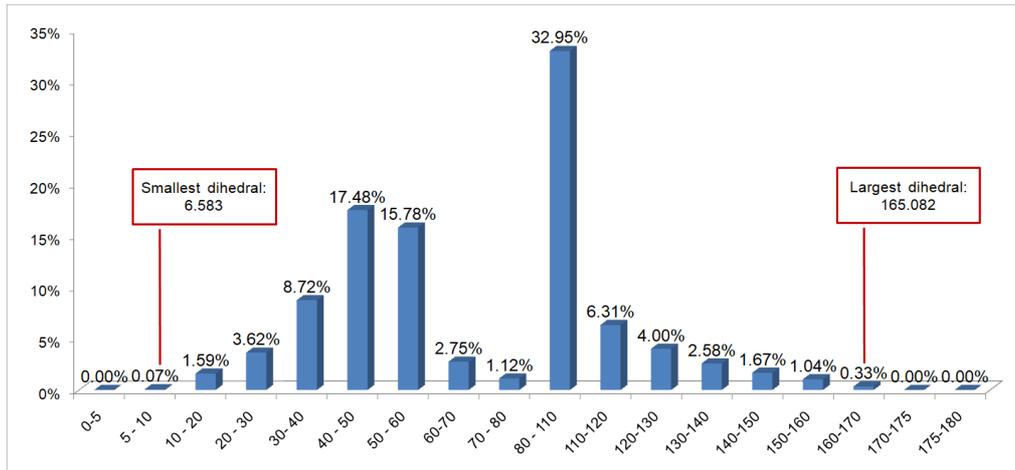


Fig. 27: Aspect ratio histograms for the electrical motor with permanent magnet supplementary poles magnet, considering the total of tetrahedra (y axis) inside of prescribed aspect ratio intervals (x axis). In (a) is presented the calculated histogram from the raw mesh (16,176 tetrahedral elements - Fig. 15) and in (b) is shown the obtained histogram from the mesh with high degree of refinement (2,753,985 tetrahedra - Fig. 22).

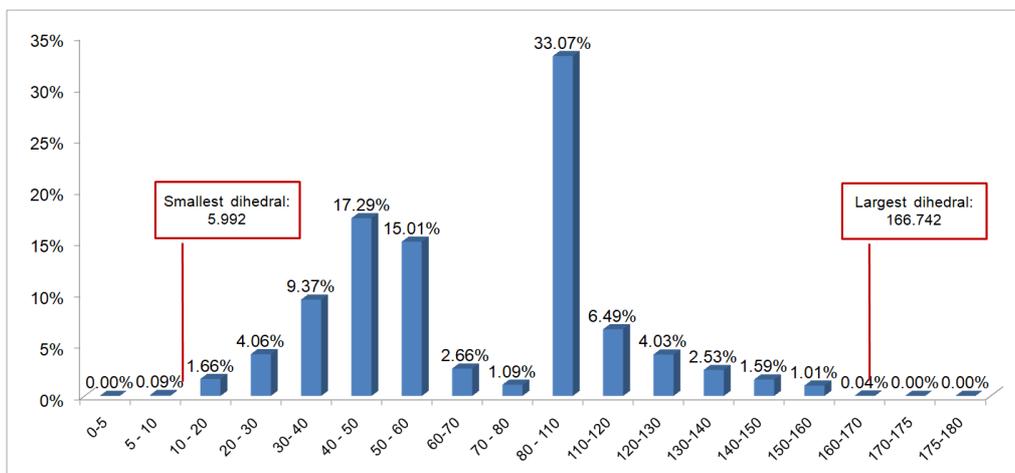
Regarding the dihedral angles, the histograms are shown in Fig. 28 for all highly refined meshes previously discussed, namely the “C-type” magnet with 1,788,145 tetrahedral elements, turbo-compressor device meshed with 1,838,948 elements, the Klein Bottle meshed with 1,774,489 elements and the electrical motor with 2,753,985 tetrahedral elements. Notice that the histograms were constructed for all elements in the mesh and not only for those elements belonging to the domain borders as in Wang and Yu (2012). From these histograms, it is possible to verify dihedral angles whose values belong to the interval from 5.90 to 166.70 degrees, with a peak value around the condition for a regular tetrahedron (90 degrees) (Ruppert, 1995). The proposed algorithm by Labelle and Shewchuk (2007) provided tetrahedral meshes with dihedral angles between 8.90 and 164.80°C. Also, the dihedral angles were bounded between 1.66 and

174.72°C when non-uniform tetrahedra on the surface boundary were chosen. In a study presented by Wang and Yu (2012), tetrahedral meshes were constructed with the minimal dihedral angle being guaranteed to be greater than or equal to 5.71 degrees. Therefore, the proposed method in this study is able to generate meshes with consistent dihedral angles, mainly when compared to the results provided by other approaches (Wang and Yu, 2012; Labelle and Shewchuk, 2007).

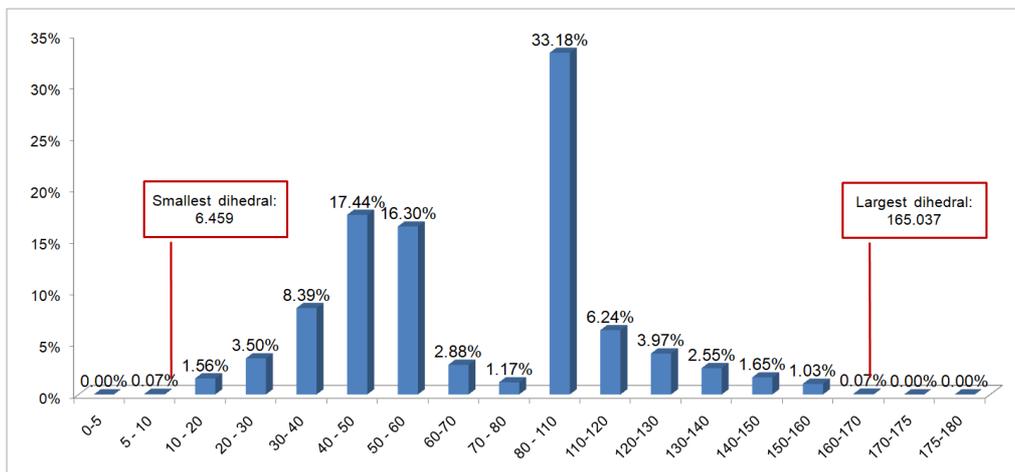
An efficient method can mesh complex geometries with high degree of refinement and with a minimum of running time, without applying parallel computing resources. This is particularly important for three-dimensional simulations of nontrivial domains, as the presented here and commonly explored in studies focused on computational electromagnetics. These two features are presented in the proposed method.



(a)



(b)



(c)

Fig. 28: Total of dihedral angles (y axis) inside of prescribed angle intervals in degrees (x axis), considering also the angle bounds: in (a) is shown the histogram from the “C-type” magnet - the corresponding mesh was illustrated in Fig. 19; in (b) is presented the histogram from the turbo-compressor device - the corresponding finite element mesh was illustrated in Fig. 20; in (c) is defined the histogram from the Klein bottle - the corresponding mesh was presented in Fig. 21; and, in (d) is shown the histogram from the electrical motor - the corresponding tetrahedral mesh was illustrated in Fig. 22

Thus, the running time as a function of the increasing number of tetrahedra was presented in Fig. 23 for each explored context in this study, such as the “C-type” magnet, turbo-compressor device, Klein bottle and electrical motor with supplementary permanent. From these histograms (Fig. 23), it is possible to verify that highly refined meshes were obtained in acceptable running times, considering the used computational resources and accessible to many users. Millions of tetrahedral elements were constructed in a few seconds. The minimum running time was of 39.60 sec, the maximum running time was of 60.60 sec and the mean value was of 47.59 sec, which was calculated computing only the most refined mesh from each domain.

It is important to notice that the running time required to construct mesh with the desired quality is related to the complexity of the geometry. For instance, the electrical motor with permanent magnet supplementary poles presented by Zhao *et al.* (2012) was reproduced here. The 3D domain related to the same motor topology was meshed into tetrahedral elements (Fig. 22). The model described by Zhao *et al.* (2012) was meshed in triangles (2-dimensional elements) through a parameterized mesh generation technique. The refinement procedure was performed in an adaptive manner, with 6,930 times FEM computations in five hours to obtain all solutions. The Delaunay algorithm was also employed and, for each FEM computation, the overall mesh generation and refinement processes required 30% of time, which represented a value of 0.80 sec in a single mesh generation Zhao *et al.* (2012). The proposed method provided the electrical motor meshed with 2,753,985 tetrahedral elements in approximately 60 sec (Fig. 22). Since the domain in this case is bigger (3D) with a high number of 3D elements, this performance is very encouraging, for instance, considering future developments of 3D adaptive calculations.

All meshes included in this study do not represent the refinement limits, which could be reached, even using one core of the processor. In this context, several methods have been developed to study mesh generation and FEM simulations (Bracken *et al.*, 2012; Li *et al.*, 2013; Chen and Biro, 2012; Zhao *et al.*, 2012; Ho *et al.*, 2011; Zhang and Kumar, 2011; Krebs *et al.*, 2010). However, none of them has used our proposed combination. Most of these methods lead to an “almost” ideal result, considering different strategies, models and systems. Therefore, compare our proposal in order to define the best one would be a difficult task, not to say senseless. In fact, our method is rather complementary than ratable, capable of providing relevant and reliable results in any other similar FEM simulations. Moreover, we contribute to future works, in which one could profit from valuable information contained in each of the explored domains.

Simulation Analysis by FEM

It is not useful a highly refined mesh involving tetrahedral elements without the required quality by FEM simulations. The equation solutions are strongly dependent on the shape and number of elements. In a valid mesh constructed under proper conditions, such as meshes defined with well shaped tetrahedra, the FEM approximations converge to the problem solution when the number of elements is increased (mesh refinements) (Shewchuk, 2002; Silvester and Ferrari, 1996; Babuvska and Aziz, 1976). Therefore, in addition to the total values of tetrahedra (highly refined meshes) which were previously discussed, the proposed method provided also well shaped tetrahedra. This conclusion was possible exploring comparisons of aspect ratio, which were verified through FEM simulations.

The quality of the mesh is an important part of the simulation process. The FEM can be used to solve many kind of physical problems and mathematical equations and sets of equations, including non-trivial ones. The setup phase, in which the geometry of the problem is defined, the boundary conditions are applied and the mesh is crucial to the quality of the solution. The FEM simulations can be performed by two strategies. One is increasing the degree of interpolation functions that describe the behavior of the solution inside an element. This strategy can be useful, but it represents an additional complexity in the solving process and turn the matrix assembly process more complex and computationally expensive. The other strategy is the increase of the number of elements given as input to the simulation process. This strategy hinders the solution process using FEM by increasing the size of the matrices. More elements implies bigger matrices (Silvester and Ferrari, 1996). Although the increase in the size of matrices, these are sparse matrices that do not add much complexity neither increases too much the computational costs of the solution process.

One fact needs to be considered about the amount of elements and the quality of the solution using FEM. The number of elements has an impact in the quality of the solution depending on what is being analyzed. For instance, when evaluating the electric field between the plates of a parallel-plate capacitor, a great number of elements will not improve the solution, since the mentioned electric field is uniform. This way, a great number of elements will be useless to this analysis. On the other hand, if the edge effects needs to be analyzed, a better precision is required and this can be achieved increasing the number of elements in the interest area. Therefore, depending on the complexity of the effect to be analyzed, a better precision is required and the improvement of the precision can be achieved by the uplifting quality of the mesh used in the simulation process.

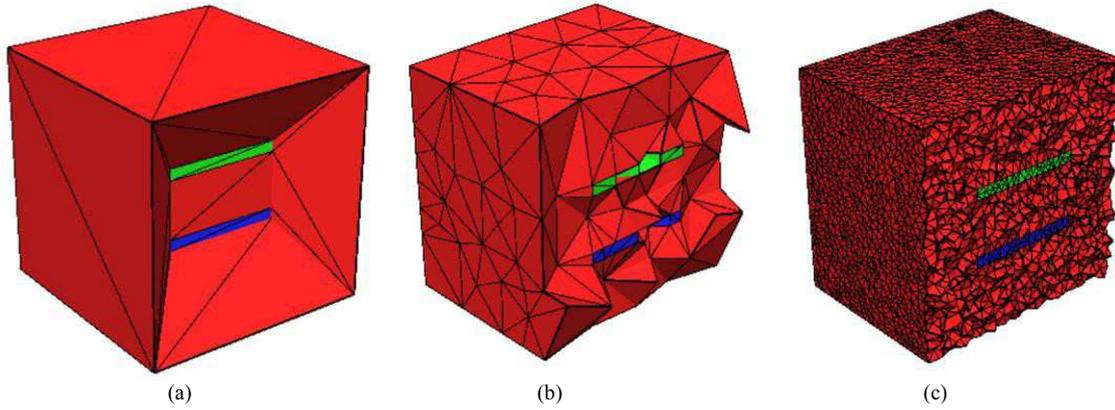
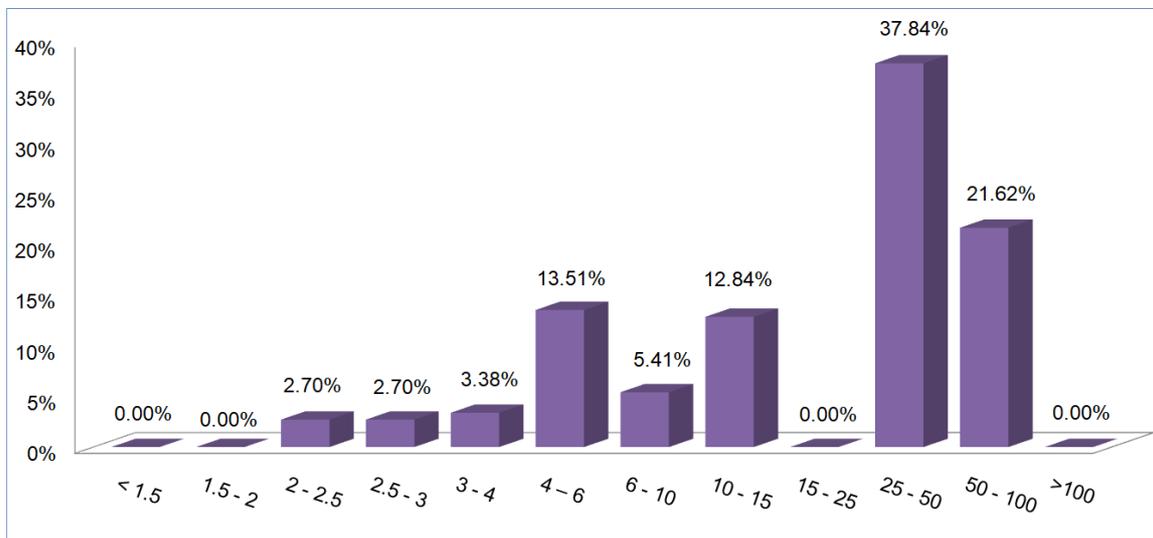
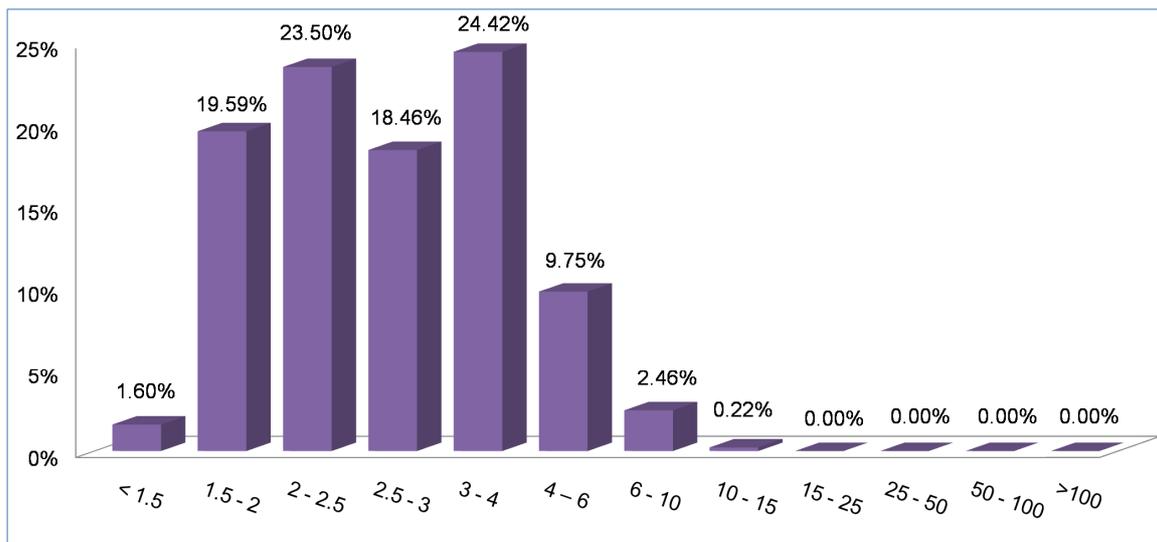


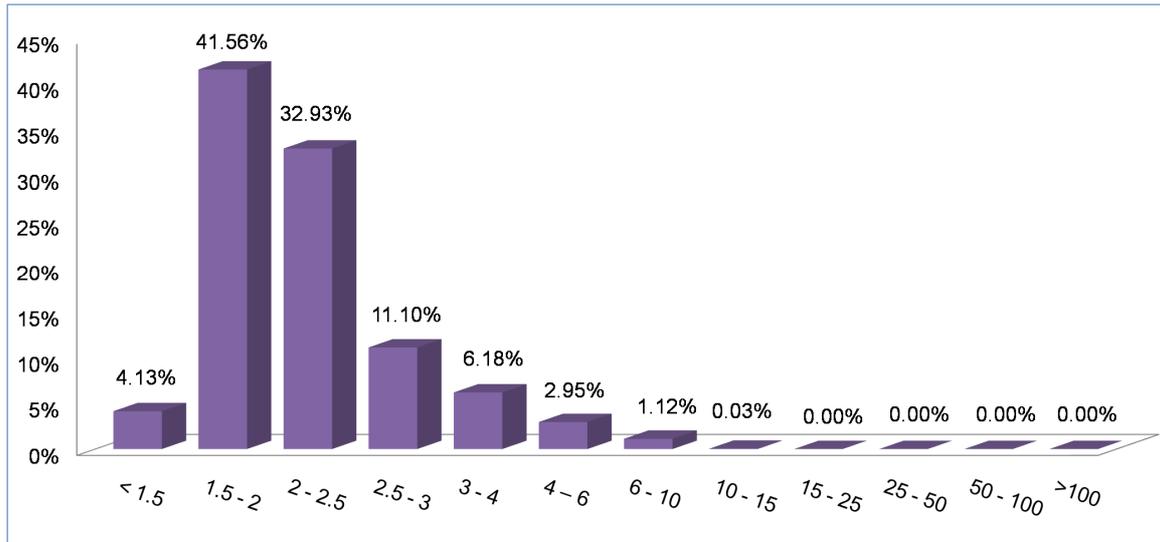
Fig. 29: Models with different degrees of refinements applied in FEM simulations. Mesh with 148 elements (a), mesh with 3,251 elements (b) and mesh with 122,842 elements (c)



(a)



(b)



(c)

Fig. 30: Aspect ratio histograms for the capacitor, considering the total of tetrahedra (y axis) inside of prescribed aspect ratio intervals (x axis): in (a) is presented the calculated histogram from the raw mesh (148 tetrahedral elements - Fig. 29a); in (b) is shown the obtained histogram from the mesh with high degree of refinement (3,251 tetrahedra - Fig. 29b) and in (c) is shown the obtained histogram from the mesh with high degree of refinement (122,84 tetrahedra - Fig. 29c)

Three simulations of a parallel-plate capacitor were executed with three different degrees of refinement, whose models are shown in Fig. 29. It is important to notice that aspect ratio presents similar behavior to the ones previously discussed and it can be verified in Fig. 30. The results of these simulations are shown in Fig. 31.

The simulations were performed in order to analyze the electric potential immediately after the end of the capacitor. This problem is governed by the Laplace's equation, Equation 3 and it was chosen because its results are well-known and can be easily verified in its correctness.

Also, simulating a well-known problem allows to focus on the influence of the refinement of the mesh in the final result more than the resultant values themselves.

All the simulations were performed by applying the same model and setup. The only variation is the refinement of the mesh used to achieve the results. For simulations, the upper plate was assigned with 3,500V of electric potential, while the lower plate was assigned with 0V of electric potential. The results were analyzed within a plane with x-axis coordinate fixed at 0.011 cm, 0.001 cm away from the end of the plates. We intended to show the equipotential lines according to the improvement of the mesh.

The first simulation was performed using a mesh with 41 nodes and 148 elements. This mesh led to the result shown in Fig. 31a, in which there is 48 elements crossing or touching the plane. The second simulation was performed using a mesh with 614 nodes and

3,251 elements. This mesh led to the result shown in Fig. 31b, in which there is 275 elements crossing or touching the plane. Finally, the third simulation was performed using a mesh with 21,486 nodes and 122,842 elements. This mesh led to the result shown in Fig. 31c, in which there is 4,891 elements crossing or touching the plane. It is important to notice the relation among the colors, where the red represents the highest value of electric potential (3,500V) and the deep blue represents the lowest value of electric potential (0V).

Analyzing the results, it is possible to verify the enhancement of the simulations, using only two steps of refinements. While in the first simulation result, Fig. 31a, the plates of the capacitor were unrecognizable. In the second simulation (Fig. 31b) there was a slightly improvement in the identification of equipotential lines. In the last simulation, Fig. 31c, the plates and the equipotential lines are perfectly clear. These results show how the quality of a simulation was improved through the adjustment of the mesh. The improvement of the mesh and consequently in the result of FEM simulation, is related with the increasing number of tetrahedra in the mesh and with the better accuracy of the aspect ratio, as previously presented in this study. It is extremely important to reinforce that, a mesh in which the elements do not have proper aspect ratio is not suitable for using in FEM. Moreover, tetrahedra with the proper characteristics turn a FEM simulation possible and correct.

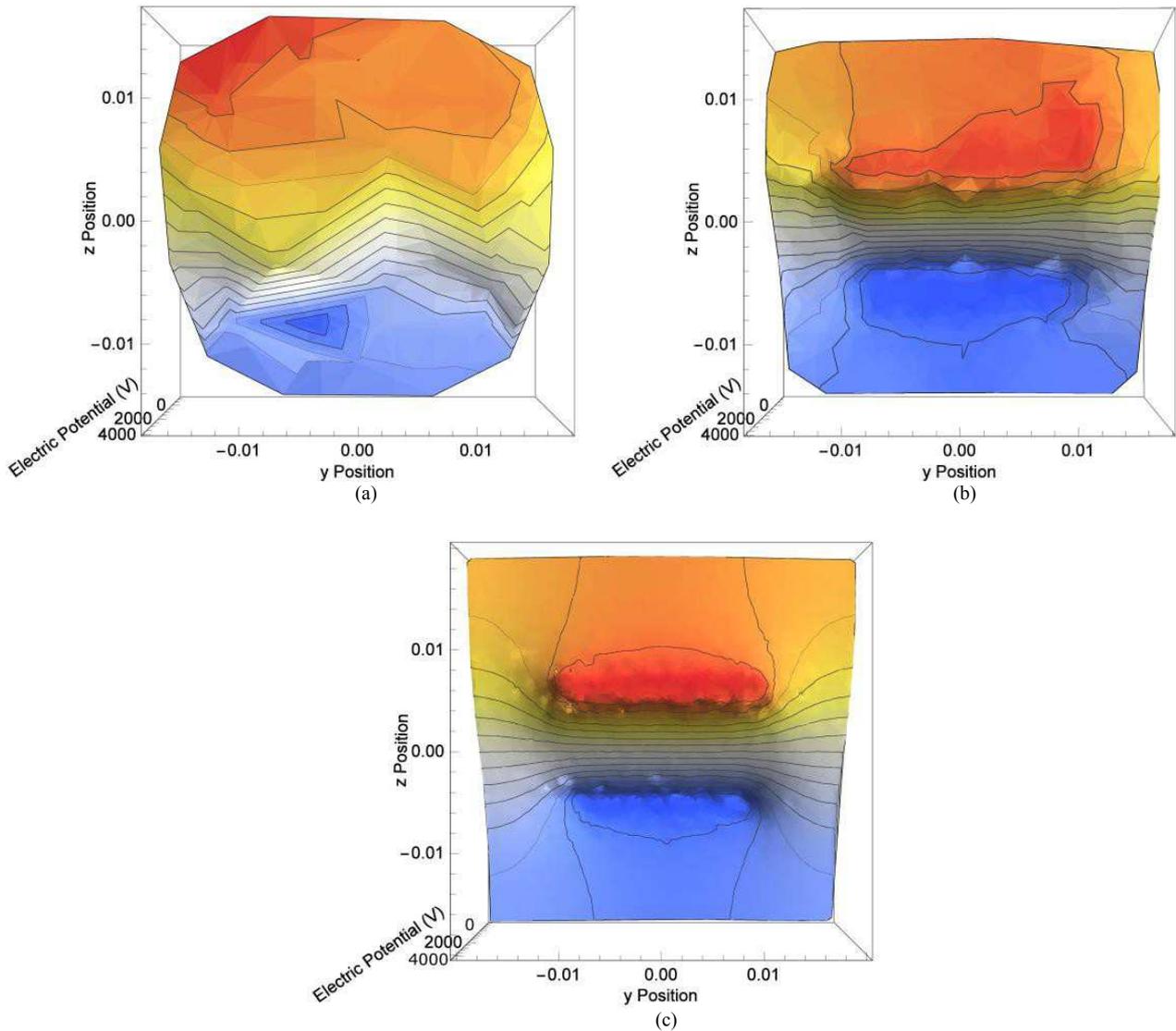


Fig. 31: Results of simulation of the Laplace's equation in order to analyze the electric potential of a parallel-plate capacitor using three different degrees of refinement. Mesh with 148 elements (a), mesh with 3,251 elements (b) and mesh with 122,842 elements (c)

Conclusion

In this study was described a method to construct and mesh 3D nontrivial domains, in a continuous and integrated process, totally controlled by script code segments written in Python language. The strategies were developed from the Blender and TetGen packages. The method was tested in solids considering relevant geometries for computational electromagnetics problems, such as “C-type” magnet, turbo-compressor, Klein bottle and an electrical motor with supplementary permanent magnets.

The open source packages were integrated with success and the results were highly refined meshes, with guaranteed quality. Millions of tetrahedral elements were

constructed in a few seconds, with a minimum running time of 39.60 sec, a maximum running time of 60.60 sec and a mean value of 47.59 sec. The highest number of mesh elements was of 2,800,000 tetrahedra approximately, but it is not a limit in the degree of mesh refinement. These values are important, mainly considering the 3D nontrivial domains, which were chosen to test our proposal. Therefore, unless the parallel computing resources are used (Bracken *et al.*, 2012), neither the topological domain complexity, nor the mesh refinements and time processing performance of the examples discussed in relevant studies (Li *et al.*, 2013; Chen and Biro, 2012; Zhao *et al.*, 2012; Ho *et al.*, 2011; Zhang and Kumar, 2011; Krebs *et al.*, 2010) surpass the equivalent features of the test cases presented in this

study, which were explored using one CPU core. Moreover, the approach described is able to provide highly quality meshes with common and very accessible hardware resources.

Considering the mesh quality, the dihedral angles were bounded between 5.90 to 166.70°C, with a peak value around the condition for a regular tetrahedron. These values are relevant when compared to the values presented in other studies (Wang and Yu, 2012; Labelle and Shewchuk, 2007). Also, in the refined meshes, more than 90% of the tetrahedral elements were constructed with aspect ratios at most 5 and closer to ideal condition (aspect ratio defined as 1). These values are important in comparison to the discussed in other studies (Cheng, 2006; Si, 2013; Shewchuk, 1998; Ruppert, 1995), as well as, still considering the explored nontrivial domains (multiply connected sub-domains and many curved regions). As FEM solutions depend on the shape and number of tetrahedral elements, the proposed method as able to generate highly refined meshes with well shaped tetrahedra. These characteristics are guarantees for convergence and stability of the numerical solution, as discussed by Shewchuk (2002), Silvester and Ferrari (1996) and Babuvska and Aziz (1976). Finally, FEM simulations were performed to show how the quality of a simulation can be more suitable by the improvement of the mesh. Meshes with 41 nodes and 148 elements, 614 nodes and 3,251 elements and 21,486 nodes and 122,842 elements were used and we are able to notice that the improvement of the mesh were related to the increasing number of tetrahedra. This feature implies directly in the quality of the results, where the elements do not have proper aspect ratio is not suitable for using in FEM. Thus, proper characteristics of tetrahedra turn FEM simulation feasible and correct. Regarding our proposal, we could imply that was possible to integrate the environment of open source codes to perform FEM simulations. Then, we are able to show better results, accordingly to the improvement of the mesh, considering the increase number of tetrahedra and the quality of the aspect ratio.

In this context, the proposed method not only is able to decompose nontrivial domains with multiply connected sub-domains into meshes of tetrahedral elements, but also performs successive mesh refinements in acceptable running times and with the quality required by FEM simulations. Thus, these results encourage new applications of the FEM in problems of different areas, especially in computational electromagnetics.

Acknowledgement

This work was financially supported by PROPE/UNESP (Pró-Reitoria de Pesquisa/UNESP) and FAPERP (Fundação de Apoio à Pesquisa e Extensão de São José do Rio Preto). The authors gratefully

acknowledge the continuous guidance and support of G. Krein (IFT-UNESP) in the preparation of this paper.

Authors Contributions

Leandro Alves Neves: Developed and implemented the strategies, analyzed the results and wrote the paper.

José Márcio Machado: Helped in the development of the model proposed, analyzed the mathematical approach used.

Julio Cesar Momente: Implemented the strategies, analyzed the results and wrote the paper.

Geraldo Francisco Donegá Zafalon: Helped in the improvement of the model proposed, analyzed the results and wrote the paper.

Álvaro Magri Nogueira da Cruz: Analyzed the results and wrote the paper.

Marcelo Zanchetta do Nascimento: Analyzed the integration of the developed strategies and the results.

Carlos Roberto Valêncio: Analyzed the integration of the developed strategies and the results

Ethics

All authors have approved the manuscript and agree with the publication. This manuscript is an original article and has not been published elsewhere. The authors have no conflicts of interest to declare.

References

- Blender Foundation.
<https://www.blender.org/about/license/>
- Engrid - open-source mesh generation.
<http://engits.eu/en/engrid>
- Openfoam Foundation. <http://www.openfoam.org/>
- Babuvska, I. and A.K. Aziz, 1976. On the angle condition in the finite element method. *SIAM J. Numerical Anal.*, 13: 214-226.
DOI: 10.1137/071302
- Baker, T.J., 1989. Automatic mesh generation for complex three-dimensional regions using a constrained delaunay triangulation. *Eng. Comput.*, 5: 161-175. DOI: 10.1007/BF02274210
- Barauskas, R., A. Gulbinas and G. Barauskas, 2007. Investigation of radiofrequency ablation process in liver tissue by finite element modeling and experiment. *Medicina (Kaunas)*, 43: 310-325.
- Bracken, E., S. Polstyanko, N. Lambert, R. Suda and D. Giannacopoulos, 2012. Power aware parallel 3-d finite element mesh refinement performance modeling and analysis with cuda/mpi on gpu and multi-core architecture. *IEEE Trans. Magnet.*, 48: 335-338. DOI: 10.1109/TMAG.2011.2177814
- Buell, W. and B. Bush, 1973. Mesh generation a survey. *J. Eng. Industry*, 95: 332-338.
DOI: 10.1115/1.3438132

- Cacace, M. and G. Blocher, 2015. Meshita software for three-dimensional volumetric meshing of complex faulted reservoirs. *Environ. Earth Sci.*, 74: 5191-5209. DOI: 10.1007/s1266
- Caendish, J.C., D.A. Field and W.H. Frey, 1985. An approach to automatic three-dimensional finite element mesh generation. *Int. J. Numerical Meth. Eng.*, 21: 329-347. DOI: 10.1002/nme.1620210210
- Chang, C.W., M. Liu, S. Nam, S. Hang and Y. Liu *et al.*, 2010. Optical mobius symmetry in metamaterials. *Phys. Rev. Lett.*, 105: 235501. DOI: 10.1103/PhysRevLett.105.235501
- Chen, C. and O. Biro, 2012. Geometric multigrid with plane smoothing for thin elements in 3-d magnetic fields calculation. *IEEE Trans. Magnet.*, 48: 443-446. DOI: 10.1109/TMAG.2011.2175203
- Chen, J. and Z. Chen, 2016. Three-dimensional superconvergent gradient recovery on tetrahedral meshes. *Int. J. Numerical Meth. Eng.*, 108: 819-838. DOI: 10.1002/nme.5229
- Cheng, S.W., 2006. On the sizes of delaunay meshes. *Comput. Geometry*, 33: 130-138. DOI: 10.1016/j.comgeo.2005.08.002
- Cho, H.W., S.M. Jang and S.K. Choi, 2006. A design approach to reduce rotor losses in high-speed permanent magnet machine for turbo compressor. *IEEE Trans. Magnet.*, 42: 3521-3523. DOI: 10.1109/TMAG.2006.879086
- Cook, W.A., 1974. Body oriented (natural) co-ordinates for generating three-dimensional meshes. *Int. J. Numerical Meth. Eng.*, 8: 27-43. DOI: 10.1002/nme.1620080104
- Duan, N.N., W.J. Xu, S.H. Wang, H.L. Li and Y.G. Guo *et al.*, 2015. Extended finite element method for electromagnetic fields. *Proceedings of the IEEE International Conference on Applied Superconductivity and Electromagnetic Devices*, Nov. 20-23, IEEE Xplore Press, Shanghai, China pp: 364-365. DOI: 10.1109/ASEMD.2015.7453614
- Ecabert, O., J. Peters, M.J. Walker, T. Ivanc and C. Lorenz *et al.*, 2011. Segmentation of the heart and great vessels in ct images using a model-based adaptation framework. *Med. Image Anal.*, 15: 863-876. DOI: 10.1016/j.media.2011.06.004
- Field, D.A., 1986. Implementing Watson's algorithm in three dimensions. *Proceedings of the 2nd Annual Symposium on Computational Geometry*, Jun. 2-4, ACM, Yorktown Heights, USA, pp: 246-259. DOI: 10.1145/10515.10542
- George, P. and F. Hermeline, 1992. Delaunay's mesh of a convex polyhedron in dimension d. application to arbitrary polyhedra. *Int. J. Numerical Meth. Eng.*, 33: 975-995. DOI: 10.1002/nme.1620330507
- George, P.L., H. Borouchaki, P.J. Frey, P. Laug and E. Saltel, 2004. *Mesh generation and mesh adaptivity: Theory and techniques*. Wiley Online Library.
- George, P.L., F. Hecht and E. Saltel, 1991. Automatic mesh generator with specified boundary. *Comput. Meth. Applied Mechan. Eng.*, 92: 269-288. DOI: 10.1016/0045-7825(91)90017-Z
- Geuzaine, C. and J.F. Remacle, 2009. Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. *Int. J. Numerical Meth. Eng.*, 79: 1309-1331. DOI: 10.1002/nme.2579
- Ghisi, I.T., J.J. Camata and A.L. Coutinho, 2014. Impact of Tetrahedralization on parallel conforming octree mesh generation. *Int. J. Numerical Meth. Fluids*, 75: 800-814. DOI: 10.1002/flid.3918
- Guo, X., M. Lange, G. Gorman, L. Mitchell and M. Weiland, 2015. Developing a scalable hybrid mpi/openmp unstructured finite element model. *Comput. Fluids*, 110: 227-234. DOI: 10.1016/j.compfluid.2014.09.007
- Gerritsen, B., P. Imre Horvath, R. Lou, J.P. Pernot and F. Giannini *et al.*, 2015. Filletting sharp edges of multi-partitioned volume finite element meshes. *Eng. Comput.*, 32: 129-154. DOI: 10.1108/EC-03-2013-0074
- Ho, S., N. Chen and W. Fu, 2011. A moving mesh embedded algorithm in finite element method for optimal design of electromagnetic devices. *IEEE Trans. Magnet.*, 47: 2947-2950. DOI: 10.1109/TMAG.2011.2148107
- Ho-Le, K., 1988. Finite element mesh generation methods: A review and classification. *Comput.-Aided Design*, 20: 27-38. DOI: 10.1016/0010-4485(88)90138-8
- Jang, S.M., H.W. Cho and S.K. Choi, 2007. Design and analysis of a high-speed brushless dc motor for centrifugal compressor. *IEEE Trans. Magnet.*, 43: 2573-2575. DOI: 10.1109/TMAG.2007.892328
- Krebs, G., S. Clenet and I. Tsukerman, 2010. Overlapping finite elements for arbitrary surfaces in 3-d. *IEEE Trans. Magnet.*, 46: 3473-3476. DOI: 10.1109/TMAG.2010.2046146
- Labelle, F. and J.R. Shewchuk, 2007. Isosurface stuffing: Fast tetrahedral meshes with good dihedral angles. *ACM Trans. Graph.*, 26: 57-57. DOI: 10.1145/1239451.1239508
- Leng, J., Y. Zhang and G. Xu, 2013. A novel geometric flow approach for quality improvement of multi-component tetrahedral meshes. *Comput.-Aided Design*, 45: 1182-1197. DOI: 10.1016/j.cad.2013.05.004
- Li, J., Y. Guo, P. Liu, Q. Wang and J. Qin, 2013. A gpu-accelerated finite element solver for simulation of soft-body deformation. *Proceedings of the IEEE International Conference on Information and Automation*, Aug. 26-28, IEEE Xplore Press, Yinchuan, China, pp: 631-636. DOI: 10.1109/ICInfA.2013.6720373

- Li, Y., J. Yue, R. Niu and G. Liu, 2016. Automatic mesh generation for 3d smoothed finite element method (s-fem) based on the weaken-weak formulation. *Adv. Eng. Software*, 99: 111-120.
DOI: 10.1016/j.advengsoft.2016.05.012
- Lizier, M., J. Shepherd, L. Nonato, J. Comba and C. Silva, 2008. Comparing techniques for tetrahedral mesh generation. Proceedings of the Inaugural International Conference of the Engineering Mechanics Institute, (EMI' 08), pp: 1-8.
- Lo, S., 1985. A new mesh generation scheme for arbitrary planar domains. *Int. J. Numerical Meth. Eng.*, 21: 1403-1426.
DOI: 10.1002/nme.1620210805
- Lo, S., 2012. Automatic merging of hexahedral meshes. *Finite Elements Anal. Design*, 55: 7-22.
DOI: 10.1016/j.finel.2012.02.003
- Marcum, D.L. and N.P. Weatherill, 1995. Unstructured grid generation using iterative point insertion and local reconnection. *AIAA J.*, 33: 1619-1625.
DOI: 10.2514/3.12701
- Mavriplis, D.J., 1995. Unstructured mesh generation and adaptivity. Technical Report, DTIC Document.
- Milasinovic, D., M. Ivanovic, H. Tengg-Kobligk, D. Bockler and N. Filipovic, 2008. Software tools for generating cfd simulation models of blood flow from ct images and for postprocessing. *J. Serbian Society Comput. Mechan.*, 2: 51-58.
- Misztal, M.K., J.A. Barentzen, F. Anton and K., Erleben, 2009. Tetrahedral mesh improvement using multi-face retriangulation. Proceedings of the 18th International Meshing Roundtable, (IMR' 09), Springer, Berlin, pp: 539-555.
DOI: 10.1007/978-3-642-04319-2_3
- Moreno, J., M.J. Algar, I. Gonz'alez and F. C'atedra, 2015. Design and evaluation of the multilevel mesh generation mode for computational electromagnetics. *Applied Comput. Electromagnet. Society J.*
- Neves, L.A., E. Pavarino, J.M. Machado, M.P. Souza and C.R. Val'encio *et al.*, 2015. Multiscale tetrahedral meshes for fem simulations of esophageal injury. Proceedings of the IEEE 28th International Symposium on Computer-Based Medical Systems, Jun. 22-25, IEEE Xplore Press, Sao Carlos, Brazil pp: 103-108.
DOI: 10.1109/CBMS.2015.76
- Nguyen-Van-Phai, 1982. Automatic mesh generation with tetrahedron elements. *Int. J. Numerical Meth. Eng.*, 18: 273-289. DOI: 10.1002/nme.1620180209
- Nunes, C.R., P.C. Mayrink, R.C. Mesquita and D.A. Lowther, 2011. A parallel remeshing method. *IEEE Trans. Magnet.*, 47: 1202-1205.
DOI: 10.1109/TMAG.2010.2090944
- Owen, S.J., 1998. A survey of unstructured mesh generation technology. Proceedings of the 7th International Meshing Roundtable, Oct. 26-28, Sandia National Lab, Dearborn, Michigan, USA, pp: 239-267.
- Paille, G.P., N. Ray, P. Poulin, A. Sheffer and B. L'evy, 2015. Dihedral angle-based maps of tetrahedral meshes. *ACM Trans. Graph.*, 34: 54-54.
DOI: 10.1145/2766900
- Pavarino, E., L.A. Neves, J.M. Machado, M.F. Godoy and Y. Shiyou *et al.*, 2013. Free tools and strategies for the generation of 3d finite element meshes: Modeling of the cardiac structures. *J. Biomed. Imag.*
DOI: 10.1155/2013/540571
- Pavarino, E., L.A. Neves, J.M. Machado, M.F. Godoy and Y. Shiyou *et al.*, 2014. Delaunay tetrahedralization of the heart based on integration of open source codes. *J. Phys.*
DOI: 10.1088/1742-6596/490/1/012131
- Ray, T. and T.K. Adviser-Dey, 2006. Quality Delaunay meshing of polyhedral volumes and surfaces. Ohio State University.
- Ruppert, J., 1995. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms*, 18: 548-585.
DOI: 10.1006/jagm.1995.1021
- Schneider, K., B. Klusemann and S. Bargmann, 2016. Automatic three-dimensional geometry and mesh generation of periodic representative volume elements for matrix-inclusion composites. *Adv. Eng. Software*, 99: 177-188.
DOI: 10.1016/j.advengsoft.2016.06.001
- Schroeder, W.J. and M.S. Shephard, 1988. Geometry-based fully automatic mesh generation and the Delaunay triangulation. *Int. J. Numerical Meth. Eng.*, 26: 2503-2515.
DOI: 10.1002/nme.1620261109
- Shaw, J. and N. Weatherill, 1992. Automatic topology generation for multiblock grids. *Applied Math. Comput.*, 52: 355-388.
DOI: 10.1016/0096-3003(92)90088-I
- Shewchuk, J.R., 1998. Tetrahedral mesh generation by Delaunay refinement. Proceedings of the 14th Annual Symposium on Computational Geometry, Jun. 07-10, ACM, Minneapolis, Minnesota, USA, pp: 86-95. DOI: 10.1145/276884.276894
- Shewchuk, J.R., 2002. Delaunay refinement algorithms for triangular mesh generation. *Comput. Geometry*, 22: 21-74. DOI: 10.1016/S0925-7721(01)00047-5
- Si, H. and K. Gartner, 2005. Meshing piecewise linear complexes by constrained Delaunay tetrahedralizations. Proceedings of the 14th International Meshing Roundtable, (IMR' 05), Springer, Berlin, pp: 147-163.
DOI: 10.1007/3-540-29090-7_9

- Si, H. and J.R. Shewchuk, 2014. Incrementally constructing and updating constrained Delaunay tetrahedralizations with finite-precision coordinates. *Eng. Comput.*, 30: 253-269.
DOI: 10.1007/s00366-013-0331-0
- Si, H., 2013. A quality tetrahedral mesh generator and three-dimensional Delaunay triangulator. Weierstrass Institute for Applied Analysis and Stochastic, Berlin, Germany.
- Silvester, P.P. and R.L. Ferrari, 1996. Finite elements for electrical engineers. Cambridge University Press.
- Specogna, R., 2015. Diagonal discrete hodge operators for simplicial meshes using the signed dual complex. *IEEE Trans. Magnet.*, 51: 1-4.
DOI: 10.1109/TMAG.2014.2350515
- Sun, Z. and T. Chaichana, 2010. Fenestrated stent graft repair of abdominal aortic aneurysm: Hemodynamic analysis of the effect of fenestrated stents on the renal arteries. *Korean J. Radiol.*, 11: 95-106.
DOI: 10.3348/kjr.2010.11.1.95
- Wagner, S., F. Lucka, J. Vorwerk, C. Herrmann and G. Nolte *et al.*, 2016. Using reciprocity for relating the simulation of transcranial current stimulation to the EEG forward problem. *NeuroImage*.
DOI: 10.1016/j.neuroimage.2016.04.005
- Wall, S.T., J.M. Guccione, M.B. Ratcliffe and J.S. Sundnes, 2012. Electromechanical feedback with reduced cellular connectivity alters electrical activity in an infarct injured left ventricle: A finite element model study. *Am. J. Physiol.-Heart Circulatory Physiol.*, 302: H206-H206.
- Wang, J. and Z. Yu, 2012. Feature-sensitive tetrahedral mesh generation with guaranteed quality. *Comput.-Aided Design*, 44: 400-412.
DOI: 10.1016/j.cad.2012.01.002
- Watson, D.F., 1981. Computing the n-dimensional Delaunay tessellation with application to voronoi polytopes. *Comput. J.*, 24: 167-172.
DOI: 10.1093/comjnl/24.2.167
- Weatherill, N.P. and O. Hassan, 1994. Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. *Int. J. Numerical Meth. Eng.*, 37: 2005-2039.
DOI: 10.1002/nme.1620371203
- Yerry, M.A. and M.S. Shephard, 1983. A modified quadtree approach to finite element mesh generation. *IEEE Comput. Graph. Applic.*, 3: 39-46.
DOI: 10.1109/MCG.1983.262997
- Yerry, M.A. and M.S. Shephard, 1984. Automatic three-dimensional mesh generation by the modified-octree technique. *Int. J. Numerical Meth. Eng.*, 20: 1965-1990.
DOI: 10.1002/nme.1620201103
- Zhang, S.U. and A.V. Kumar, 2011. Three-dimensional magnetostatic analysis using structured mesh and nodal elements. *IEEE Trans. Magnet.*, 47: 198-206.
DOI: 10.1109/TMAG.2010.2086064
- Zhao, Y., S. Niu, S. Ho, W. Fu and J. Zhu, 2012. A parameterized mesh generation and refinement method for finite element parameter sweeping analysis of electromagnetic devices. *IEEE Trans. Magnet.*, 48: 239-242.
DOI: 10.1109/TMAG.2011.2173908