A novel 3D automatic mesh generation and refinement package

C. Boccaletti, E. Nisticò & P. Sordi
Department of Electrical Engineering, University of Rome "La Sapienza", Italy.

Abstract

This paper describes an automatic mesh generator usually suitable for finite element simulations. The mesh generation algorithms that allow to create a mesh respecting some prescribed constraints and boundaries are illustrated. The aim of the authors has been to build a user-friendly software that is able to solve practical 3D finite elements problems. The refinement process of the mesh allows to calculate the most important quantities for a general electromagnetic configuration. The software has been developed on the Windows platform. The 3D visualisation provided into the package is obtained by the OpenGL programming language. This allows to follow the evolution of the calculation and to check the results.

1 Introduction

The availability of automatic mesh generators for 3D domains with complex geometry is more and more a crucial necessity for the solution of numerical simulations based upon Finite Elements. In the context of unstructured meshes, several methods have been addressed in the twenty last years including Advancing-front, octree and Delaunay based methods [1,2,3]. A great attention has been paid to Delaunay algorithms both by the computational geometry people and by the P.D.E. groups. These methods implement the construction of the mesh in different phases. The first task consists of introducing all the nodes of the structure into the tesselation. This can be achieved by creation of a mesh consisting of 5 tetrahedrons of a "box" including all the given nodes of the structure. During the construction the
nodes can be inside an element, located on a face or along an edge. The temporary mesh must be modified to satisfy the imposed constraints of the geometric structure. In fact the temporary mesh, in general, does not contain all the structural edges and faces of the geometry. In the literature a number of methods for mesh adaptations are reported [5]. To solve this crucial problem we can modify the mesh by using a sequence of local swaps to suppress elements crossing the boundary. In order to disentangle some complicated domains we are forced to abandon the global Delaunay criteria and accept triangulations that are not strictly Delaunay. Hazlewood [6] has shown that it is possible to produce constrained triangulations by just retriangulating the tetrahedra whose interiors are intersected by a missing, constraining, convex polygon. First, all points on the intersection of the missing polygon and the edges and faces of tetrahedra are found. Then, every intersected tetrahedron is retriangulated locally using the Delaunay criteria.

Some methods claim that if you want to tetrahedrize a geometrical structure involving vertices, edges and triangles, you need to add Steiner points in the general case. And their number can be very large, resulting in a by far more complex tetrahedralization.

When studying all these efforts the question is simple: Why constructing a temporary mesh and trying to alter it in order to respect the constraints, instead of just constructing a mesh that at any moment respects all the boundaries and constraints of the geometry?

The adopted automatic mesh generation method is a variant of the Delaunay—Voronoi’s tessellation with a control of the structure boundary performed during the evolution of the mesh.

When the insertion of all the nodes is finished it is not necessary to modify the structure any more.

2 General principle of the method

The geometrical structure shown in figure 1 is the sketch of an electrical pole of an axial flux machine. A number of 3D sectors can be located in the structure. In figure 2 the dark area represents the iron material used in the rotor disc and in the stator of the machine. The permanent magnet joined to the rotor disc is located between the two ferromagnetic elements.

In order to apply a finite elements solution each sector of the structure must be defined in the material property, boundary conditions, and forces. Each domain must be bordered by a triangular surface mesh. Each triangle of the surface is defined by three edges. The vertexes of the triangle are located by three primary nodes of the geometry.

The nodes and the edges of the structure are drawn in the wireframe of Figure 3. At the end of the mesh generation every node of the structure has to belong at least to a vertex of a tetrahedron. This condition is necessary but not sufficient if a correct mesh is desired.
Figure 1: Structure of an electrical pole of an axial flux machine. The structure is composed by 17 three dimensional sectors.

Figure 2: Solid sectors and wireframe of the geometrical structure. The structure is composed by: 132 primary nodes, 538 edges, 668 triangular faces.
Another important condition is the following: each edge of the structure has to belong at least to an edge of a tetrahedron. If the node and the edge conditions are respected the mesh is correct and the generation is finished.

In the bibliography the insertion and elimination faces problem is broadly discussed. Otherwise if the Steiner points are not inserted into the tessellation this case can be neglected. In order to demonstrate this assertion let us consider a material of the structure that is bordered by a superficial mesh made up by triangular elements. If each edge of the surface is inserted into the mesh and each edge of any tetrahedron does not intersect the surface then all the triangular faces are inserted.

3 The method proposed by the authors

The mesh generation algorithm the authors propose in this paper is of the Delaunay type. In Figure 4 a two-dimensional case is investigated. The node of Figure 4 is not still inserted and is outside the provisional mesh bordered by a thick border line. The edges 1 2 3 and 4 of the figure separate the node from the temporary mesh. The new triangles can be formed joining the node to the four edges of the border. This general explanation can be easily extended to the three-dimensional case.

During the construction of the mesh the boundary containing the tetrahedrons must be convex as much as possible.
In order to match this condition the initial phase of the process consists of ordering the list of the edges of the structure. The ordering of the nodes attends two criteria. The first type of ordering follows a distance criterion. The list of the edges is ordered from the shortest to the longest distance of the middle point of the edges to the initial node. A sorting order of a triangular mesh in 2D is shown in Figure 5a.

The second type of edge ordering is performed by maximising the number of linked edges. The effects of the second ordering criterion are shown in Figure 5b. The arrangement of a three-dimensional structure follows the same criterion.

At this point the mesh can be built. The first tetrahedron of the mesh can be created. This tetrahedron has to contain the first edge of the ordered list.
The first tetrahedron shown in Figure 6 has the following properties: it does not contain primary nodes; it does not intersect any edge; it does not intersect any face of the structure.

Figure 6: Wireframe of the structure and view of the first tetrahedron.

The whole mesh will be built around this first tetrahedron. The algorithm of the insertion of the edges can be schematized in the flow chart of Figure 7. The process is now described in detail.

1) The list of the edges is scanned looking for the first edge that has not been yet inserted.

2) Since the edge list has been ordered, one of the two nodes of the edge is already inserted in the provisional mesh. The node not yet inserted must be located.

3) The node not inserted is connected to the mesh. In order to create a new tetrahedron, the node is linked with a triangular face of the external mesh surface (third Delaunay-Voronoy case).

4) The program checks if a new tetrahedron can be joined to the mesh. The new tetrahedron is dropped if it intersects some faces or edges of the structure. The new tetrahedron is also dropped if it contains some primary nodes.

5) Each triangular face of the surface boundary is checked by means of a loop that repeats the instructions described in points 3 and 4.

6) At the end of the loop some new tetrahedrons are joined to the mesh. These elements have a common vertex: the node not inserted into the mesh. However, the new configuration can not block the insertion of some missing edges or faces of the structure. For instance, an edge can have its two vertex nodes inserted into the mesh while the edge is not still contained. In this case the new configuration is not correct; the tetrahedrons must be dropped and the algorithm starts to check the insertion of new edges.
4 Application

The evolution of the mesh generation described up to now can be observed in Figures 8-10. The wireframe of the obtained mesh is shown in Figure 11a. The mesh produced at this step is a correct tetrahedral partition of the domain. Usually this mesh is not adequate for FEM computations, so it is necessary to modify it to obtain a very good refinement. The refinement process is achieved by means of the insertion of additional secondary nodes [7]. The placement of these nodes is chosen in order to enforce boundary conformity and to improve the quality of the mesh.

The orthogonal projection of the refinement mesh are drawn into a dialog window shown in Figure 12. In the software a number of orthogonal projection views of the 3D structure can be selected in order to make both the input and the output processes easier.
Figure 8: Mesh generation evolution: 77 edges out of 538 have been inserted; 52 tetrahedrons have been created.

Figure 9: Mesh generation evolution: 226 out of 538 edges have been inserted; 209 tetrahedrons have been created.

Figure 10: Mesh generation evolution: 441 out of 538 edges have been inserted; 443 tetrahedrons have been created.
Figure 11a,b: View of the mesh wireframe. a) Mesh before the refinement: 508 tetrahedrons have been created. b) Mesh after one of the phases of the refinement: 4478 elements have been created by inserting 870 secondary nodes.

The dialog window reports some information about the project. It is reported the CPU time necessary to develop the primary mesh formed by 508 tetrahedrons (13.8 s by means of a 1300 MHz processor). The quality factors specify the degree at which the regularity of the tetrahedron mesh is achieved. In Figure 12 three important parameters measuring the quality of the created mesh are shown. The first parameter $Q_m$ can be defined as follows:

$$Q_m = 3 \frac{R_{insc}}{R_{circ}}$$

(1)

$R_{insc}$ being the radius of the inscribed sphere; $R_{circ}$ being the radius of that circumscribing the tetrahedron. When $Q_m = 1$ the tetrahedron is regular.

Figure 12: Dialog window containing some information about the project.
The Glassmeier quality factor is defined as:

$$Q_g = \frac{\text{True Vol}}{\text{Ideal Vol}} + \frac{\text{True Surf}}{\text{Ideal Surf}} + 1$$

(2)

The ideal volume and surface are calculated for a regular tetrahedron. If $Q_g = 1$ the four vertexes are colinear; $Q_g = 2$ when the vertexes all lie in a plane; $Q_g = 3$ when a regular tetrahedron is formed. The Robert Roux parameter is defined as

$$Q_r = \sqrt[3]{\frac{\text{True Vol.}}{\text{SphereVol.}}}$$

(3)

where the sphere is that circumscribing the tetrahedron. $Q_r = 1$ for a regular tetrahedron. The range of values is $0 - 1$.

5 Conclusions

In this paper the authors summarise their experience with the implementation of a robust 3D mesh generator using a minimalist approach to generate constrained Delaunay triangulations. The authors set up a user-friendly software that is able to solve practical 3D finite elements problems. This mesh generator can be a useful tool for mesh generation and engineering applications based on the Finite Element Method.

References


