Journal of Southeast University (English Edition) Mar. 2002 Vol.18 No.1 ISSN 1003—7985

Mesh Generation for Finite Element Analysis of Electric Machines

Wan Shui[1 *] Wang Desheng[2]

([1] College of Traffic and Transportation Engineering, Southeast University, Nanjing 210096, China)
([2] Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100080, China)

Abstract: This paper describes two modified methods for triangular and quadrilateral meshing for finite element analysis of 2D electric machines. One is coupling the classic Delaunay method and advancing front method to generate optimal triangulation; the other is coupling the classic paving and Delaunay triangulation for optimal quadrilateral meshing. Various electric machine models are meshed successfully to demonstrate the robustness and effectiveness of the methods.

Key words: advancing front method, automatic mesh generation, Delaunay triangulation, paving

Automatic mesh generation for finite element analysis of electromagnetic field has been widely researched, and has been developing to maturity in many areas. The most popular mesh generation methods are Delaunay-based methods and advancing front technique (AFT). These two methods have been widely used to generate high-quality meshes for arbitrary domains. But for electric machines, both 2D and 3D mesh generation still needs further research due to electric machines' complex geometries and the necessity of large size-ratio mesh in air gap and its neighboring areas. In this paper, two modifications to the classic Delaunay triangulation and advancing front quadrilateral meshing (or paving) are proposed to deal with the 2D cases respectively.

In the procedure of the Delaunay triangulation, one important step is to generate inner field points. Various inner points generation methods have been used, such as placing nodes on the gravity centers of triangles, or along edges, or generating nodes in the octree method[1-7]. But these methods cannot generate inner points with optimal positions, which will influence the mesh quality greatly. In this paper, the classic Delaunay method is coupled with the advancing front technique. It generates the inner points in AFT and inserts them in Delaunay method front by front. This couplement combines AFT's forte of generating inner points with optimal positions and Delaunay method's rapidity of points insertion to generate optimal triangulation efficiently. Even for meshing with very large size-transition (500 : 1), this modified triangulation performs very well.

Several automatic quadrilateral meshing schemes have been developed and their applications have increased rapidly because of their superiority in FE analysis to the triangular elements[8-17]. These schemes can be classified into two categories: direct and indirect quadrilateral meshing. Indirect methods usually use a generated triangulation to generate quadrilateral elements by conversion or other techniques[12-14,16], while direct methods not[8-11,15]. Paving is in the direct category and has three characteristics desirable for FE analysis: ① boundary sensitive; ② orientation insensitive; ③ few irregular nodes[8,13]. However, even though it has been modified by several authors[12,13,15,16], the paving method still has two shortcomings: it needs expensive computation for frontal edges' intersection checking; it cannot generate elements in a global light with respect to sizing hence it cannot treat cases needing very large size-transitions. In this paper, we first use Delaunay method to obtain the boundary triangulation, which helps to construct the sizing space and neighboring grid. Then we use the sizing space and neighboring grid to generate quadrilaterals globally with respect to metric consideration and locally with respect to frontal edges' intersection checking. This modification makes possible high-quality quadrilateral meshing for FE analysis of electric machines.

This paper is organized as follows. Section 1 discusses the Delaunay triangulation coupled with AFT; section 2 describes the modified paving method coupled with the Delaunay triangulation; section 3 presents two meshing examples for an electric machine,

Received 2001 – 06 – 29.
* Born in 1960, male, associate professor.

triangular and quadri-lateral respectively. Finally, we make our concluding remarks.

1 Delaunay Triangulation Coupled with AFT

Usually, the procedure of a Delaunay-based triangulation includes the following three successive steps: ① boundary triangulation; ② interior refinement; ③ mesh improvement. ① and ③ was discussed in detail by several authors[2-7,18,19]. Here, we just recall the outlines of the procedures. Our focus is on the interior refinement.

1.1 Boundary triangulation

The outline of the procedure of boundary triangulation goes as follows:

1) Create the set of the initial points which include the boundary and the specified points;

2) Create the four points defining a rectangle enclosing the above set;

3) Triangulate the rectangle by two triangles;

4) Insert the initial points and get an initial mesh including these points;

5) Regenerate the boundary edges and define the different domains;

6) Remove the exterior triangles to obtain the boundary triangulation.

After the boundary triangulation, we use interpolation method to derive a control space or sizing space from it. Then for a point p in the domain, we have $h(p)$ as the element size or the sizing value for it. With this sizing distribution, we introduce the notion of edge normal length. Let A and B be the two end points of edge AB, then the normal length of AB is $L(AB) = 2 \cdot \mathrm{dis}(AB)/(h(A) + h(B))$ where $\mathrm{dis}(AB)$ is the length of AB. And we say a triangle is acceptable if all its three edges' normal edge length is less than a given criterion (we set it to be $\sqrt{2}$), otherwise unacceptable.

1.2 Interior refinement

Once the boundary triangulation has been obtained, we come to the next task: interior refinement which includes generating inner field points and inserting them into the existing mesh. For this, our method is using AFT to generate inner points with optimal placement and insert them in efficient Delaunay method. The procedure of the algorithm is presented as follows:

1) Initialize the front F_i ($i = 0$) to be the boundary edges of the boundary mesh, and mark all triangles to be unacceptable. Let A_i equal to all the acceptable elements, U_i equal to all the unacceptable elements. Obviously $A_0 = T_0$ (T_i is the triangle of the ith stage).

2) Find an optimal point for each element e^* of F_i. Once the front F_i is identified, an optimal point is determined for each element of F_i. This point lies on the same side of the edge e^* as the unacceptable element leaning on it, and is placed at a position which is determined to form an optimal triangle with e^*. Using the control space and the notion of normalize length, we carry out an iterative procedure to construct the optimal point. These points form the points cloud $N_i + 1$.

3) Filtering of the $N_i + 1$. As every point of the $N_i + 1$ points cloud is independently created, a filtration must be performed to remove from $N_i + 1$ a point which violates the size criterion (via the control space) when compared with a previous selected point from $N_i + 1$.

4) If the filtered point cloud $N_i + 1$ is empty, go to end.

5) Insertion of the retained points into T_i via the constrained Delaunay insertion procedure which is a restriction of the standard Delaunay insertion method to avoid the deletion of an existing constrained entity like constrained points, edges[4,20-22].

6) Update the front F_i and go to 2). Once the T_i is constructed, we classify the elements of T_i into A_i and U_i. Then: $F_i := (f : f = (k_1, k_2)$, where k_1 is in A_i and k_2 is in U_i), where (k_1, k_2) indicates a pair of neighboring elements, i.e. f is an edge between two elements, one is acceptable, the other is unacceptable.

1.3 Mesh improvement

The outline of the mesh improvement is as follows:

1) Improve the mesh connection structure by mesh relaxation process;

2) Apply optimal smoothing or Lapacian smoothing to smooth the obtained mesh to recover the mesh quality;

3) Use edges swapping to topologically optimize the quality of a few badly shaped triangles.

Now, we have completed the process of our Delaunay triangulation coupled with AFT.

2 Modified Paving Method

Since the introduction by Ted D. Blacker[8], paving method has been developed by many researchers[12,13,15,16]. Among these modified paving schemes, the typical procedure of the most developed and mature ones comes as follows:

1) Input the boundary node loops;

2) Find the best starting loop;

3) Find the best starting edge of the loop;

4) Create an element for the selected edge;

5) Check the intersection, if the intersection occurs, solve it and goto 2); else, continue;

6) Insert the element into the data structure;

7) Perform local smoothing for improved quality;

8) Update the loop; if the loops are not empty goto 2);

9) Globally optimize the mesh.

Two key steps of the above procedure are 4) and 5). Blacker's paving method create an element locally and perform intersection test globally, which results in creating meshes without smooth element-size transition and needing costly computation for intersection checking[8]. Owen and S. H. Lo have done good jobs for this problem[12,13,16]. They make use of a prior triangulation of the domain to govern the sizing distribution and perform the element creating globally, the intersection test globally. But this indirect treatment requires a very complicated data structure which makes the implementation difficult to manage. Also it needs additional mesh generation time for the triangulation. Furthermore, the conversion speed is comparatively slow.

For solving these problems, we have the following strategies:

1) Using a boundary triangulation to derive a control space or sizing space to govern the element sizing globally;

2) Using a neighbor grid and the above control space to govern the intersection test and perform the checking locally, thus inexpensively.

In the following, we describe the above two strategies.

2.1 Modification to element creation

Firstly, we make a Delaunay triangulation of the boundary and specified points. And as in section 1, using this triangulation we derive a control space (or sizing space) from it and also introduce the notion of normal edge length. Then for a point p in the domain, we have $h(p)$ as the element size for it; for an edge AB we have $L(AB)$ as its normal edge length. Also, in the triangulation, we introduce a neighboring grid G which is constituted by a set of uniform rectangles (we call these rectangles the cells of G) whose sides are parallel to the co-ordinates axis and which covers the whole domain.

Secondly, with the above preparation, we come to element edges' generation. In our starting edge's selection, we use the method introduced by Owen[13]. First we define a state dual for each edge of the starting loop. The possible states include (1,1), (1,0), (0,1), (0,0). Then we start from the edges with states (1,1), next (1,0), (0,1), (0,0) in order. Let e_i (as shown in Fig.1) be the starting edge and p, p_r are its end points. p_l is the left point of p on the loop. Without loss of generality, we consider the (0,0) case, i.e. we need to create three edges e_l (the left one), e_r (the right one), e_c (the above one). To this end, we first create $p*$ as follows:

1) Find p^* satisfing: the line p^*p bisects the inner angle ang 1 and $\mathrm{dis}(p^*p) = (\mathrm{dis}(p_lp) + \mathrm{dis}(pp_r))/2.0$.

2) Compute the normal length $L(p^*p)$. If $L(p^*p)$ is above 1.0, move p^* in the direction from p^* to p until $L(p^*p)$ is near 1.0; else, move p^* in the direction from p to p^*. Then, in the same manner, we create p^{**} for p_r. Finally, we simultaneously move p^* and p^{**} in proper direction (beginning from the midpoint p_m of the edge p^*p^{**}) along the line p^*p^{**} until $L(p^*p^{**})$ is in the range$[1-a, 1+a]$. Here, we let a equal to 0.25.

Thus, we form an element for e_i with the above created three edges.

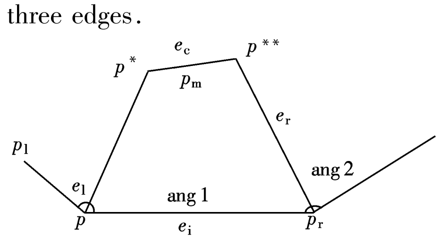

Fig.1 Element edges' generation

2.2 Modification to intersection check

Because the neighboring grid G covers the entire domain, we can associate with each cell of G the boundary points and newly generated field points. For a point p, we find the cell which p is in and associate the cell with it. Thus, using G we have the following intersection check.

For a newly created edge e.g. e_l (see Fig.1), we first find the smallest cells set g^* which covers the edge e_l; then we use the control space to derive the maximum element size h_m and with h_m, we define another cells set g^{**} which includes g^{*} and has a radius larger than that of g^{*} by h_m. That is to say, if a point p lies outside g^{**}, it has no edges (connecting p) intersecting the edge e_1. From g^{**}, we find the set A constituted by the points which are in g^{**}. Finally, for each point q of A, we perform the intersection test of the edges (connecting q) with the edge e_1, and store the intersection edges for later intersection treatments if the intersection occurs. If no point of A has edges which intersect e_1, we say e_1 has no intersection with the current loop. In this manner, we can perform the intersection checking locally, not globally, thus more efficiently than before.

With the above two modifications, we can generate quadrilaterals with elements globally generated and intersection check locally performed. Hence, in this modified paving method, we can generate high-quality quadrilateral meshes with large element size ratios robustly and efficiently.

3 Application Examples

In this section, we present two meshing examples for electric machines. One is a triangular mesh generated in the above discussed method of Delaunay triangulation coupled with AFT. Fig.2 shows a quarter of the mesh. The other is a quadrilateral mesh generated in the modified paving method. Fig.3 shows a part of the mesh. These two examples obviously demonstrate the robustness and effectiveness of our two modified mesh generation methods.

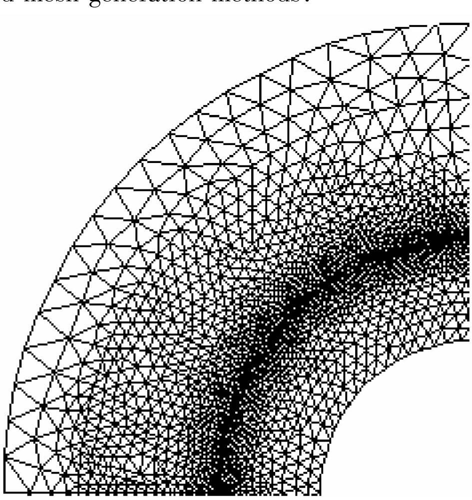

Fig.2 A quarter of a triangular mesh of an electric machine

4 Concluding Remarks

This paper proposes two modified 2D mesh generation methods for FE analysis of electric machines.

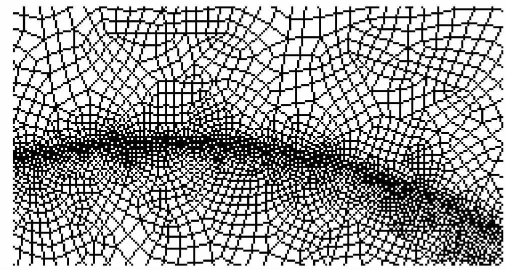

Fig.3 A part of a quadrilateral mesh of an electric machine

The first one couples the Delaunay efficient points insertion with AFT's optimal points placement for high-quality triangulation of arbitrary 2D domains. The second one modifies the classic paving method using two auxiliary tools: the control space and neighboring grid. The modified paving can generate elements globally and perform intersection checks locally, due to which we can generate quadrilateral meshes robustly and efficiently. Complex application examples have been shown to illustrate the capability and robustness of the modified meshing schemes.

References

[1] George P L, Hecht F, Vallet M G. Creation of internal points in voronoi type method[J]. *Control Adaptation*, *Adv Eng Software*, 1991, **13**: 303 - 312.

[2] Sapidis N, Perucchio K. Delaunay triangulation of arbitrarily shaped domains[J]. *Comput Aided Geom Design*, 1991, **8**: 421 - 437.

[3] Weatherill N P. The integrity of geometrical boundaries in the 2-dimensional delaunay triangulation[J]. *Commun Appl Numer Methods*, 1990, **6**: 101 - 109.

[4] Borouchaki H, George P L. Aspects of 2-D delaunay mesh generation[J]. *Int J Numer Meth Engng*, 1997, **40**: 1957 - 1975.

[5] Marcum D L, Weatherill N P. Unstructured grid generation using iterative point insertion and local reconnection [J]. *AIAA*, 1995, **33**(9): 1619 - 1625.

[6] Schroeder W J, Shepard M S. Geometry-based fully automatic mesh generation and the delaunay triangulation [J]. *Int J Numer Meth Engng*, 1988, **26**: 2503 - 2515.

[7] Chew L P. Guranteed quality mesh generation for curved surfaces[A]. In: *Proc 9th Annual Comput Geom*[C]. 1993. 274 - 280.

[8] Blacker T D, Stephenson M B. Paving: a new approach to automated quadrilateral mesh generation[J]. *Int J Numer Meth Engng*, 1991, **32**: 811 - 847.

[9] Joe B. Quadrilateral mesh generation in polygonal regions[J]. *Comput Aid Des*, 1995, **27**: 209 - 222.

[10] Zhu J Z, Zienkiewicz O C, Hinton E, Wu J. A new approach to the development of automatic quadrilateral mesh generation[J]. *Int J Numer Meth Engng*, 1991, **32**: 849 - 866.

[11] Talbert J A, Parkinson A R. Development of an automatic two dimensional finite element mesh generator using quadri-

lateral elements and bezier curve boundary definition[J]. *Int J Numer Meth Engng*, 1991, **29**: 1551 - 1567.

[12] Lo S H. Generating quadrilateral elements on plane and over curved surfaces[J]. *Comput Struct*, 1989, **31**: 421 - 426.

[13] Owen S J, Staten M L, Canann S A, Saigal S. Q-morph: an indirect approach to advancing front quad meshing [J]. *Int J Numer Meth Engng*, 1999, **44**: 1317 - 1340.

[14] Johnston B P, Sullivan J M, Kwasnik A. Automatic conversion of triangular finite element meshes to quadrilateral elements[J]. *Int J Numer Meth Engng*, 1991, **31**: 67 - 84.

[15] White D R, Kinney P. Redesign of the paving algorithm: robustness enhancements through element by element meshing [A]. In: *Proc 6th Int Meshing Roundtable*[C], 1995. 441 - 449.

[16] Lee C K, Lo S H. A new scheme for the generation of a graded quadrilateral mesh[J]. *Comput Struct*, 1994, **52**: 847 - 857.

[17] Kinney P. Cleanup: improving quadrilateral finite element meshes[A]. In: *Proc 6th Int Meshing Roundtable*, 1995. 449 - 461.

[18] Cherfils C, Hermeline F. Diagonal swap procedures and characterization of 2D-delaunay triangulations[J]. *M^2 AN*, 1990, **24**(5): 613 - 625.

[19] Frey W H, Field D A. Mesh relaxation: a new technique for improving triangulation[J]. *Int J Numer Meth Engng*, 1991, **31**: 1121 - 1131.

[20] Borouchaki H, George P L, Lo S H. Optimal delaunay point insertion[J]. *Int J Numer Meth Engng*, 1996, **39**: 3407 - 3437.

[21] Borouchaki H, Lo S H. Fast delaunay triangulation in three dimensions[J]. *Comput J Numer Methods Eng*, 1995, **128**: 153 - 167.

[22] Watson D F. Computing the delaunay tesselation with application to voronoi polytope[J]. *Comput J*, 1981, **24**: 167 - 172.

电机有限元分析的网格生成

万　水　　　　王德生

（东南大学交通学院，南京 210096）　（中国科学院数学与系统科学研究院数学研究所，北京 100080）

摘　要　该文描述了电机有限元分析中的2种修正的三角形和四边形网格的生成方法.一种是结合经典的 Delaunary 方法和波前法产生优化的三角形网格，另一种是结合传统的铺砖法和 Delaunary 三角形方法来生成优化的四边形网格.这2种网格生成方法已成功地用于多种电机模型的网格划分，证实了该方法的可靠性与有效性.

关键词　波前法，自动网格生成，Delaunary 三角形生成，铺砖法

中图分类号　O242.21