Automatic Quadrilateral Mesh Generation for FEM Using Dynamic Bubble System

Satoshi Nagakura, So Noguchi, Kazufumi Kaneda, Hideo Yamashita, and Vlatko Cingoski

***Abstract*—In this paper, a method for automatic quadrilateral (Quad) mesh generation using dynamic bubble system is presented. The proposed method can be separated into three separate computational processes: the first one generates nodes inside the analysis domain using physically-based system of bubbles, the second one for automatic generation of Quad finite elements using previously generated set of nodes, and the last process where the generated elements are smoothed using *Laplacian* smoothing operator. The proposed method is suitable and easy extendable for automatic meshing of a complex 3-D domains.**

***Index Terms*—Bubble system, FEM, Laplacian, quad elements, vector field.**

I. Introduction

RECENTLY, a large interest for Quad and Hex meshes for finite element electromagnetic field analysis was observed mainly due to higher accuracy and better convergence rates that these finite elements provide. However, the generation of such meshes is more difficult than the generation of triangular and tetrahedral meshes, because, 1) lack of suitable numerical method such as Delaunay Algorithm, and 2) inconveniences for meshing complex domains, especially curvilinear ones. It is well known that the necessary conditions for a good automatic mesh generation system are:

a) Correct meshing with small amount of input data
b) Easy mesh density control
c) Generation of finite elements with better shapes
d) Easy applicability to complicated analysis domain

Moreover, for generation of Quad and Hex meshes, one additional conditions should be considered, such as

e) Correlation between elements.

Recently, physically based method for automatic mesh generation that satisfies above-mentioned conditions has been proposed [1]. In this paper, we extend this method. First, we generate nodes inside the analysis domain using proposed node generation method like in reference [1], followed by newly proposed method for direct generation of the Quad elements using these nodes. Next, the elements shape is smoothed and improved using the well-known *Laplacian* smoothing operator. The proposed method is suitable for extension into 3-D, thus for the generation of Hex meshes. In this paper, in order to prove its usefulness, we present only its 2-D application.

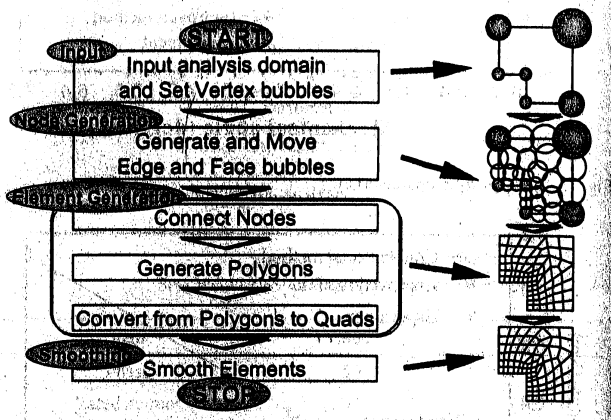


Fig. 1. Simplified algorithm for the automatic mesh generation system.

II. The Flow of Processes

Fig. 1. Simplified algorithm for the automatic mesh generation system. The proposed method has three routines: the first one in which we generate and move nodes inside the analysis domain using physically-based system of bubbles, the second one for automatic generation of Quad finite elements, and the last one which smooths elements (see Fig. 1). In the proposed method, a limited amount on input data is used; only the parameters of the objects and the radii of bubbles that are set on vertices which determine the object shape.

After setting vertex bubbles, radii and center of the edge and face bubbles that depend on radii of the vertex bubbles are generated [1]. Edge bubbles can move only along the edges, while face bubbles can move on the face. After each bubble is generated at its initial position, the movement simulation is performed until well-balanced dynamical system is achieved. Each bubble center of such generated system becomes a node for the mesh generation, where the finite elements are generated by suitable connection of the available nodes into Quads.

The above mentioned division process usually results in Quad finite elements with poor geometrical shapes. Therefore, a method for shape improvement must be used. In this paper, we implemented a smoothing procedure for shape improvements. The main feature of this smoothing method is that it keeps the topology of the elements unchanged. It only places nodes at the optimal position in accordance with the coordinates of all nodes that surround one particular node. This method is usually called the Laplacian smoothing method.

Manuscript received June 6, 2000.

S. Nagakura, S. Noguchi, K. Kaneda and H. Yamashita are with Hiroshima University, Japan (e- mail: yama@eml.hiroshima-u.ac.jp).

V. Cingoski is with the Electric Power Company of Macedonia, Skopje, Macedonia (e-mail: vlatko@esmak.com.mk).

Publisher Item Identifier S 0018-9464(01)07860-8.

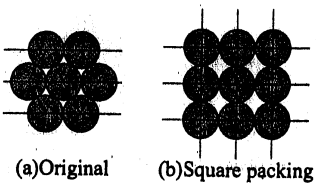


Fig. 2. Stable bubble positions. (a) Original. (b) Square packing.

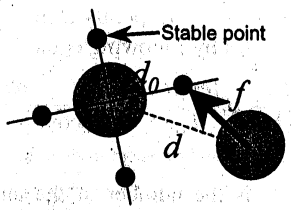


Fig. 3. Force with 2 bubbles.

III. Node Generation

A. Square Packing Bubble System

In the traditional bubble system, when many bubbles with the same size exist, they stand in a hexagonal pattern as shown in Fig. 2(a). Connecting their centers generates high quality triangular element [2]. However, if we generate Quad elements using this method, elements are similar to a parallelogram and a lozenge. In [1], the authors propose the node generation method to obtain the so-called *squire packing bubbles* which with their suitable position enable generation of high quality Quad meshes based on a metallic combination [see Fig. 2(b)].

Next, we briefly explain the *squire packing method* for two bubbles. In the traditional bubble system the movement simulation is performed according to the forces that act between bubbles, attractive and repulsive forces in particular. In the *square packing method*, forces do not act between centers of bubbles. As shown in Fig. 3, first we determine four points as candidates where the attractive forces can act. We call the point *stable point*. Later we select only one of them, and the attractive forces act between this chosen point and the bubble. By means of this method, we can obtain position of the bubble that stood in the direction of stable points. If the bubble approaches the stable point that is the position of d_0, the force is weaken, and if it reaches the position d_0, the acting force becomes zero.

In what follows, we explain the way of calculating four points that are used as stable points where the attractive forces can act. First, a vector field is defined inside the analysis domain. Four stable points are generated along the perpendicular and parallel direction of its central coordinate. In our approach we determined the vector field as follows: the direction of the vector field in a point is the same as the direction of the nearest edge. Then, the acting point of the attracting force is the closest stable point toward the neighboring bubble.

B. Definition of the Initial Bubble Position

For fast convergence of the bubble system dynamics, the initial positions have to be carefully determined. In other words, it is great importance to determine the proper number and density of the generated bubbles. If the number of bubbles is too low and they are set too coarse, low quality mesh will be generated.

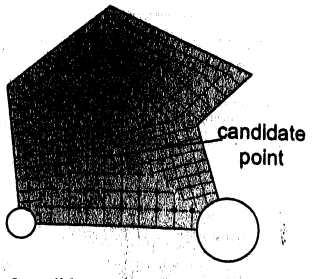


Fig. 4. Definition of candidate points for face bubbles.

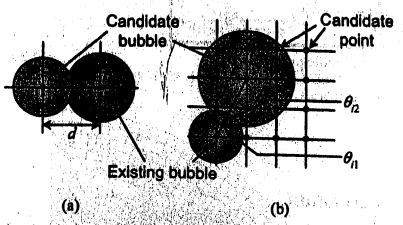


Fig. 5. Definition if a candidate point becomes a new face bubble.

On the other side, if the number of bubbles is too big, and they are set too dense, large computation time is needed to achieve the system's stability. In our method, setting of the number and position bubble is executed in two steps: 1) Generating candidate points inside each face, and 2) Positioning new bubbles that are selected from candidate points.

For fast convergence of the bubble system dynamics, we generate the candidate points using vector field as shown in Fig. 4. A candidate points is each point of the grid as shown in Fig. 4. The size of the grid g_{width} which is in a certain part on the face is given by following equations:

$$r_{av} = Ae^{B(l/2)} \tag{1}$$

$$g_{width} = \alpha r_{av} \tag{2}$$

where

- $A = r_1$,
- $B = \log(r_2/r_1)/l$,
- l represents the length of the edge,
- α is constant, and
- r_1, r_2 are radii of two bubbles that are end points on the nearest edge.

However, not every candidate point becomes a new bubble. For each candidate point to become a new bubble, two conditions must be satisfied. First:

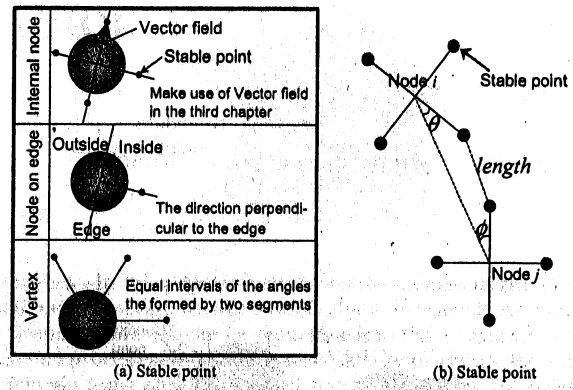
$$\beta(r_{can} + r_{old}) < d \tag{3}$$

where

- β is constant,
- r_{can} and r_{old} represent the radii of the candidate bubble and the already existing bubble in the neighborhood of the candidate point, respectively, and
- d represents the distance between bubbles' centers as shown in Fig. 5(a).

Second condition is

$$\eta < \frac{\sum_i^n \left(|\cos\theta_i| \times \frac{1}{\gamma_i} \right)}{\sum_i^n \frac{1}{\gamma_i}} \tag{4}$$



(a) Stable point (b) Stable point

Fig. 6. Make up segment.

where

η is constant,

n represents the number of existing bubble around the candidate bubble,

γ_i represents the overlap between the candidate bubble and the already existing bubble, and

$|\cos\theta_i|$ represents the biggest of $|\cos\theta_{i1}|$ and $|\cos\theta_{i2}|$, as shown in Fig. 5(b).

Utilizing this condition, initial bubbles will already stand in the direction of the vector field.

If a candidate point satisfies the above conditions, then a new bubble is positioned at that point. Otherwise, the procedure continues with the next available candidate point until all candidate points are tested.

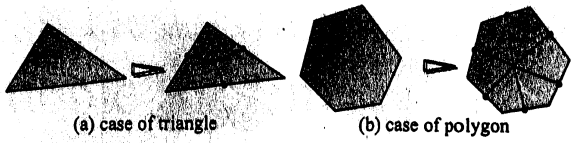
IV. Element Generation

In [1], first the triangular elements are generated and later they are connected into Quad elements. In this paper we present a different approach. We directly generate Quad elements from the available set of nodes. This new approach enables easy extension of the method into 3-D space and generation of Hex finite elements.

A. Making Segment Data and Polygons

In our method, first a set of polygons is generated using generated set of nodes by means of dynamic bubble system. Next, each polygon is subdivided into several Quad finite elements.

For this subdivision, initially a set of neighboring nodes is defined for all nodes. Since it is difficult to define this set of nodes by means of the degrees of freedom, we use a method presented in Fig. 6(a). The nodes can be sorted by their degrees of freedom, in other words, internal node, node on edge and vertex. First, for nodes with higher degree of freedom (internal nodes) four neighboring points are defined as described previously. For edge nodes one neighboring point is generated in the direction perpendicular to that edge. Finally, at the vertex that decides the object shape, zero or three neighboring points are generated on equal intervals of the angles formed by two segments. These neighboring points are used to search the connecting nodes. Now, we call a neighboring point "stable point."



(a) case of triangle (b) case of polygon

Fig. 7. Change from a Polygon to Quad.

The search should be performed at all stable points of all nodes. We search other stable points from a stable point. We select connecting nodes by following equation:

$$F = \alpha \frac{T}{D} + \beta \frac{length}{l_{av}} + \gamma \frac{2}{|\cos\theta| + |\cos\phi|} \tag{5}$$

where

T is the number of the already connecting nodes which is the node of search opponent,

D is the number of the designed connecting nodes,

l_{av} is the average length to a stable point,

α, β and γ are weights, and

angles θ and ϕ are shown in Fig. 6(b).

This operation is performed about all neighboring stable points, we connect nodes of minimum function F.

After segment data are decided, polygons are generated. But if one polygon is concave, it is divided again, to avoid its concavity. If we generate polygons this way, many Quad elements that are stood in vector direction are generated.

B. Subdivision of Polygons Into Quadrilateral Elements

Subdivision of polygons into Quad elements is done with node inclusions. New nodes are generated in the center of gravity of all polygons, and at mid-points of all segments. Using these nodes as shown in Fig. 7, each n-polygon can be divided into n Quad elements.

V. Applications

A. Mesh Quality

The mesh quality is a very important factor in order to obtain an accurate finite element analysis. Therefore it is extremely important to evaluate element quality of the generated finite elements, too. We evaluate Quad meshes' topological irregularity and geometrical irregularity.

For estimation of the topological irregularity, we use the following expression

$$\varepsilon_t = \frac{1}{n} \sum_{i=1}^{n} |\delta_i - D| \tag{6}$$

where

n represents the total number of nodes,

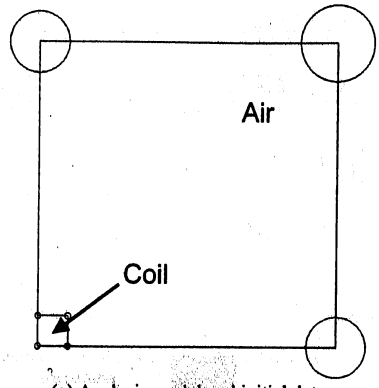
δ_i represents the number of nodes that are connected with node i, and

D = 4 if the ith node is an internal node;

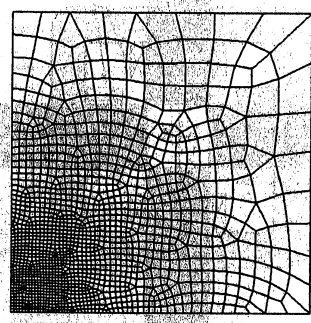
D = 3 if the ith node is a boundary node; and

D = 2~5 if the ith node is vertex.

If the mesh topology is nearly good, ε_t is near zero. However, when the mesh topology is bad,

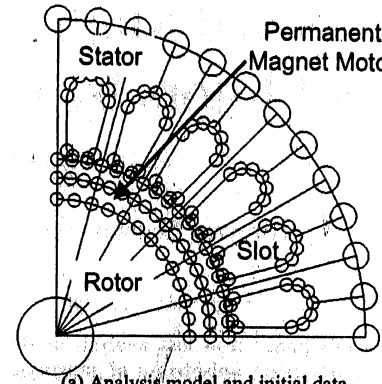


(a) Analysis model and initial data

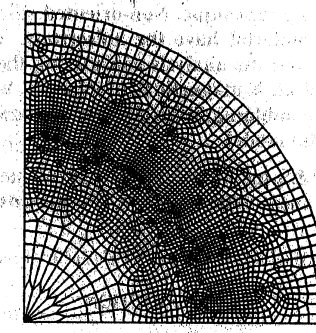


(b) Subdivision map

Fig. 8. Model for magnetic field analysis.



(a) Analysis model and initial data



(b) Subdivision map

Fig. 9. Permanent magnet motor.

For estimation of the geometrical irregularity, we use following expression:

$$\varepsilon_g = \frac{1}{m}\sum_{i=1}^{m} g_i \tag{7}$$

$$g_i = \frac{1}{\sqrt{2}} - \frac{r_i}{R_i} \tag{8}$$

where

m represents the total number of elements,

r_i represents the minimum inscribed circle radius of the ith element, and

R_i represents the maximum circumscribed circle radius of the ith element.

If the element has a good shape that is near square, ε_g is near zero.

B. Applications

In order to verify the usefulness of the proposed method, it was applied for automatic mesh generation of two models. The first one is a model for magnetic field analysis shown in Fig. 8. Fig. 8(a) shows the analysis model and the initial data, while Fig. 8(b) shows the generated subdivision map. The second model is a permanent magnet motor shown in Fig. 9.

From the generated subdivision map, we could confirm that a suitable Quad mesh with controlled mesh density was generated using only radii of several vertex bubbles as shown in Figs. 8(b) and 9(b).

Table I shows the numbers of nodes, finite elements, ε_t, ε_g, and a computation time for generated finite element meshes using an SGI O2 Workstation with a R12000/270 MHz CPU. We could confirm that elements with good quality are generated for a very short computation time.

TABLE I
GENERATED FINITE ELEMENT MESH DATA

Model	*Model1*	*Motor*
Nodes	1,499	2,992
Elements	1,434	2,933
ε_t	0.2168	0.1344
ε_g	0.0645	0.1352
Computation time	3.14(s)	27.42(s)

VI. CONCLUSIONS

In this paper, a new method for automatic Quad mesh generation using dynamic bubble system is presented. The main features of the proposed method are: small amount of input data, ability to easily control mesh density over the entire analysis domain and to directly generate the quadrilateral mesh. As a future work, we would like to improve the mesh quality, and make a 3-D extension of the proposed method.

REFERENCES

[1] K. Shimada *et al.*, "Quadrilateral meshing with directionality control through the packing of square cells," in *Proceedings of the 7th International Meshing Roundtable*, October 1998, pp. 61–76.

[2] V. Cingoski, R. Murakawa, K. Kaneda, and H. Yamashita, "Automatic mesh generation in finite element analysis using dynamic bubble system," *Journal of Applied Physics*, vol. 81, no. 8, pp. 4085–4087, April 15, 1997.