ADAPTIVE FINITE ELEMENT MESH GENERATION USING THE DELAUNAY ALGORITHM

by

Z.J. Cendes, D. Shenton i H. Shahnasser

December, 19B2

DRC-18-60-32

ADAPTIVE FINITE ELEMENT MESH GENERATION USING THE DELAUNAY ALGORITHM

Z.J. Cendes D. Shenton H. Shahnasser
Electrical Engineering Department, Carnegie-Mellon University, Pittsburgh, PA 15213

ABSTRACT

A two-dimensional generator is described which automatically creates optimal finite element meshes using the Delaunay triangulation algorithm. The mesh generator is adaptive in the sense that elements containing the largest normalized errors are automatically refined, providing meshes with a uniform error density.

The system runs on a PERQ computer made by Three Rivers Computer Company. It is menu oriented and utilizes mutipte command and display windows to create and edit the object description interactively. Mesh generation from the object data base is automatic, although it may be modified interactively by the user if desired.

Application of the mesh generator to electric machine design and to magnetic bubble simulation shows it to be one of the most powerful and easy to use systems yet devised.

INTRODUCTION

Finite element mesh generation systems fall into two broad categories:

1. Object-oriented systems in which the user specifies the geometry of the problem and the computer generates the finite element mesh [1,2]; and
2. Element-oriented systems in which the user creates the object and mesh descriptions simultaneously. [3]

The advantage of the grid generators in the first category is their ease of application; their disadvantage is the occasional creation of a poor finite element mesh with either too few elements or badly distorted elements in some regions. The advantage of second category mesh generators is mesh location control; disadvantages include greater input specification times and increased user understanding of the mesh generation process.

This paper describes a Category I finite element mesh generator in which the problems of bad mesh generation are eliminated. First, by using the Delaunay triangulation algorithm described below, the finite element meshes produced by the system are optimized in the sense that the sum of the smallest angles in the triangles in the mesh is maximized. Second, using the difference between the variational bounds for the upper and lower energy limits as an estimate of the error in each element the grid is successively refined until all elements contribute very nearly the same error to the overall solution.

DESCRIPTION

Objects

The basic or primary quantity used in the mesh generator is the object An object is a set of points which describe an enclosed region, and are entered into the data base by executing the object command. In operation, the finite element mesh generator is menu oriented. As seen in Figure 1, the PERO's display is divided into three basic regions. The upper left region is the *command window,* the upper right is the *information window* and the lower region is the *data window.*

After selecting the object command, the user places the object into the data base by

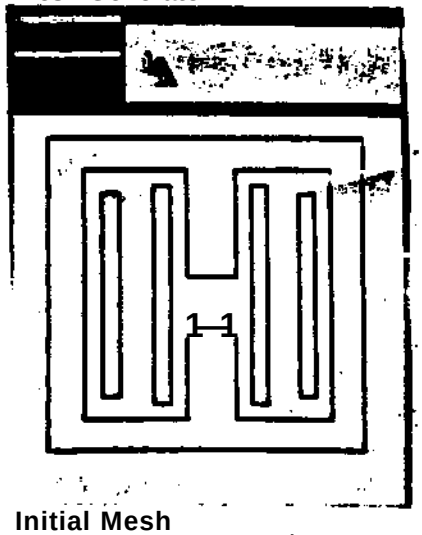
moving the cursor within the *data window* and pressing the tablet's button whenever a vertex is to be entered. There is no limit to the number of objects which may be placed into the object data base.

The object data base consists of a linked list of objects; each object is defined by a list of coordinates and a link. The link contains the address of the next object in the data base; the last object is linked to nothing *(nil).* In this way, the object data space is allocated during run time instead of being predefined.

Editing Objects

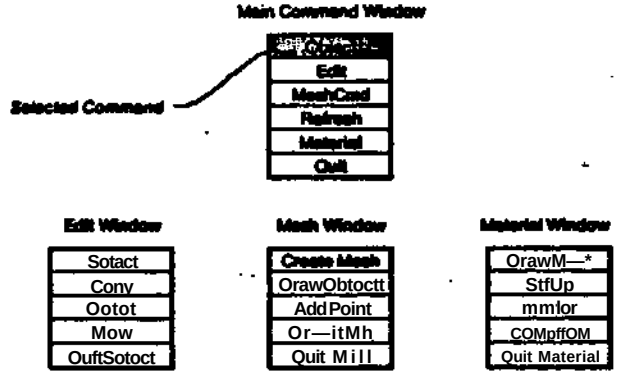
In many instances, it is desireable to edit objects. By using the *edit* command window, it is possible to select an object or group of objects and perform one of several functions on them. Some of the more common functions are copying, moving and deleting. In many instances the selected command accesses another command window. Using a heirarchy of command windows, it is possible to offer a large selection of features while keeping the number of commands offered at any one time at a minimum. A representation of the command window structure is provided in Figure 2.

Fig. 1: ThePERQ Mesh Generator



Initial Mesh

Fig. 2: Representation of Command Window Structure



MESH CREATION

Using the object data base description, an initial finite element mesh is created using the points provided by the objects. Each point is added to the mesh individually, taking the boundary of the entire data region as the boundary of the mesh area.

The finite element shape employed for mesh generation is the triangle. The data structure for a triangle contains the triangle vertices numbered counterclockwise, the three adjacent triangle numbers, and a triangle type parameter.

The Delaunay Algorithm

In all stages of the finite element mesh generation process, an optimal mesh is created in the sense that the sum of the smallest angles of all triangles is maximized. This optimal grid is obtained by using the Delaunay algorithm as described in the literature. [4]

To indicate briefly the nature of the Delaunay algorithm, consider a triangle in which a

point is to be added. The added point provides a subdivision of the triangle into three smaller triangles, as seen in Figure 3. Each of these new triangles forms a quadrilateral with its neighbor.

The heart of the Delaunay algorithm is the swapping rule illustrated in Figure 4. In this figure, we conSkJer one of the three quadrilaterals formed from the new triangles and their neighbors and draw the circumcircle defined by the vertices of the new triangle. One of the properties of a Delaunay triangulation is that the circumcircle of a Delaunay triangle may not contain another Delaunay vertex in its interior. Hence, in the event that the fourth point of the quadrilateral is located within the circle, the diagonal is swapped; otherwise no change in the data occurs. In the event that the quadrilateral diagonal is swapped, it is necessary to check the new triangles again for additional swaps. This procedure repeats until no more swaps are necessary.

Fig. 3: Addition of a new point

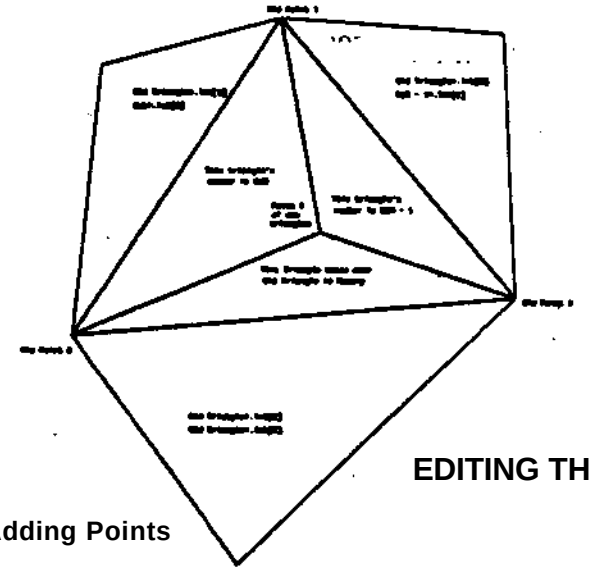
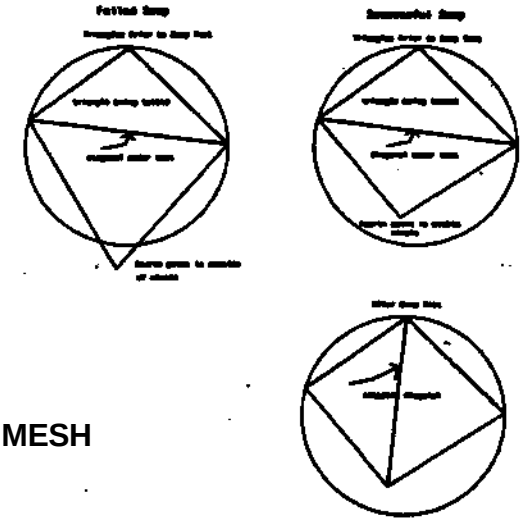


Fig. 4: Check for Diagonal Swap within a triangle



EDITING THE MESH

Adding Points

Once the mesh has been created using the data base, it is sometimes desireable to add points to the mesh. This can be done in one of the three following ways:

- Manually using the tablet
- Automatically placing a point at the centroid of all the triangles within an object.
- Automatically placing a point at the midpoints of the sides of all the triangles within an object

In all cases, the Deiaunay algorithm produces the necessary triangulation.

Deleting Regions

The mesh generator also has the ability to modify portions of the mesh by changing the material parameters of triangles inside objects. If an object is given a zero material parameter, this defines the absence of that area, and all triangles located within this region are eliminated.

ADAPTIVE PROCEDURE

Having created an initial mesh using the above procedures, the next step is to refine the

grid iteratively, subdividing only those elements having the largest errors. For magnetostatics, the procedure adopted is to first solve for the magnetic field over the initial.mesh and then to compute the upper and lower energy bounds in each element separately, according to established formulas. [5,6] Using the difference between the upper and lower energy bounds divided by the element area as an estimate of the error in each element, those elements having a greater than average error are then subdivided by placing a node at the corresponding centroids. This procedure is then repeated until all elements produce an error estimate within a specified limit

EXAMPLES

Two examples are presented here to illustrate the finite element meshes generated by the program. The first is the triangulation of the air-gap transformer, illustrated in Figure 5; the second is the discretizaton of dual conductor magnetic bubble device, illustrated in Figure 6. In the magnetic bubble simulation example, two conducting layers are shown perforated by holes, about which the magnetic field is to be computed.

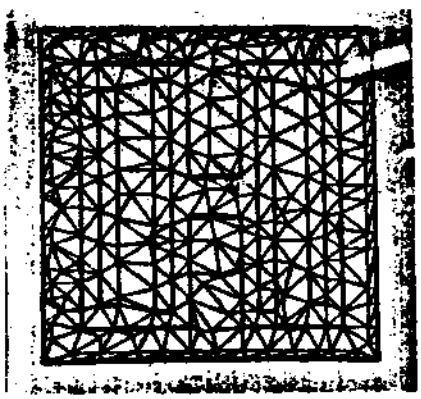
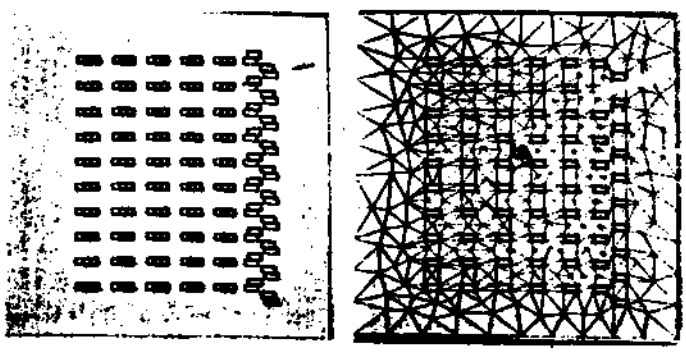


Fig. 5. Air-gap Transformer



Rg. 6. Dual Conductor Device and the Top Layer Mesh

1. J.M. Schneider K. Chaudhury S Salon, "The Use of Interactive Graphics in Electromagnetic Problems," *Winter Power Meeting, New York,* No. 82WM238-4,1982,.

2. C.S. Bibblecomb and J.Simkin, "PEZD User's Guide," Rutherford and Appleton Laboratories,RL-79-089/rev2,1981

3. Z J. Cendes, E.M. Freeman, D.A. Lowther, and P.P. Silvester, "Interactive Computer Graphics in Magnetic Field Analysis and Electric Machine Design," *IEEE Transactions on PAS,* Vol. PAS-100,1981, pp. 2862-2869.

4. D.T. Lee and B.J. Schachter, "Two Algorithms for Constucting a Delaunay Triangulation," *international Journal of Computer and Information Science,* Vol. 9, No. 3, 1980, pp. 219-242.

5. J. Penman and J.R. Fraser, "Complementary and Dual Energy Finite Element Principles in Magnetostatics," *IEEE Transactions on Magnetics,* Vol. MAG-17,1981,.

6. Ronald W.Thatcher, "Assessing the Error in a Finite Element Solution," *I.E.E.E. Transactions on Microwave Theory and Techniques,* Vol. MTT-30, No. 6, June 1982,.