COMMUNICATIONS IN NUMERICAL METHODS IN ENGINEERING, Vol. 12, 343–349 (1996)

A COMBINED OCTANT/DELAUNAY METHOD FOR FULLY AUTOMATIC 3D MESH GENERATION WITH MULTIPLE LEVEL OCTANT DIFFERENCES

ROGER MARCHAND, SEDKI RIAD AND A. ELSHABINI-RIAD

The Bradley Department of Electrical Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24060–0111, U.S.A

SUMMARY

Automated mesh generation using a combined octree/Delaunay approach typically requires that rectangular elements be bounded by other rectangular elements which are not more that one division smaller (i.e. 1/8 the volume in 3D). This limitation is sometimes referred to as the 2 : 1 rule. This paper presents a modification which allows for any level of difference.

KEY WORDS mesh generation; Delaunay mesh; Watson's method

1. INTRODUCTION

Several articles have appeared in the literature using a combined octree/Delaunay method for automated 3D mesh generation.[1-3] This technique involves the subdivision of a domain into rectangular regions which are then individually divided into tetrahedrons. The rectangular regions are formed by successive division of the domain.

A shortcoming of this approach is that neighbouring rectangular regions are not allowed to differ by more than one level (i.e. 1/8 the volume). This limitation is sometimes referred to as the 2 : 1 rule. This paper presents a modification which allows for any level of difference.

Mesh generation is an important first step in the solution of partial differential equations using finite element methods. A good review paper on the subject is Reference 4. Using the octree/Delaunay method, mesh generation is completed in a four-step process:

(1) generating a rectangular mesh using octant subdivision
(2) converting the rectangular mesh to a tetrahedral Delaunay mesh
(3) fitting the Delaunay mesh to the geometry
(4) smoothing interior nodes (optional).

The following four Sections briefly describe each of the above steps. The overall method is similar to that described in Reference 1, except that (a) the initial mesh placed in each rectangular element varies with its position and (b) the nodes added to each region are carefully ordered.

CCC 1069–8299/96/060343–07

Received 9 January 1993
Revised August 1995

2. GENERATING A RECTANGULAR MESH USING OCTANT SUBDIVISION

The goal of this step is to fill the region of interest with a rectangular mesh. The density of this mesh should vary with the given geometry. Boundary regions should be heavily packed while empty regions should be lightly packed. The technique works as follows:

(1) A given area of interest is first enclosed in a square.
(2) The region is then examined for a geometric boundary.
(3) If a boundary is found and the rectangular region has not reached some predetermined minimum size, then subdivide the region:

   (a) Divide into four smaller regions (called quadrants) in two dimensions.
   (b) Divide into eight smaller regions (called octants) in three dimensions.

(4) If no boundary is found OR the rectangular region has reached the minimum size, then:

   (a) Add the corners of the region to a list of points which comprise the mesh. The list of points is called the *node list*. No entry in the list should be repeated.
   (b) Add the rectangular region to a list of rectangular regions.

(5) Repeat the above steps for each subdivided region until all rectangular regions either contain no boundaries or have reached the minimum size.

Note that the above algorithm did not make explicit use of a datatree structure. Because the mesh generator does not require a 2 : 1 rule, neighbouring rectangular regions do not need to be examined and use of a data tree may be avoided.
Implementation note: As one can observe from the above algorithm, this approach lends itself nicely to a recursive code. Further rectangles can be stored in a dynamic linked list which allows memory used to store these elements to be released as soon as the region has been triangulated.

3. CONVERTING THE RECTANGULAR MESH TO A TETRAHEDRONAL DELAUNAY MESH

Watson's method,[5] summarized in Figure 1, shows how a node can be added to a mesh which is initially Delaunay, such that the resultant mesh is also Delaunay. This process, however, increases as more than N^2, where N is the number of nodes in the mesh. The advantage of the combined octant/Delaunay method is that the tetrahedronal mesh is created in small rectangular regions with a small number of nodes.

The difficulty arising from this approach is that the mesh, independently created in neighbouring octants, must have elements which align (or are compatible), as shown in Figure 2. In two dimensions the regions will automatically align. However, in three or higher dimensions compatibility is not automatically ensured.

In the remainder of this Section, an 'alternating initial mesh' scheme is described which allows for any difference level between neighbouring octants. The only limitation is that *the rectangular regions must be square*. To create a tetrahedronal mesh, the following approach is taken for each square element:

(1) Create the position-dependent initial tetrahedral mesh which uses all of the cube's corners and completely fills the region.
(2) Search the node list for a set of points which are located on the cube (except for those used to create the initial mesh).

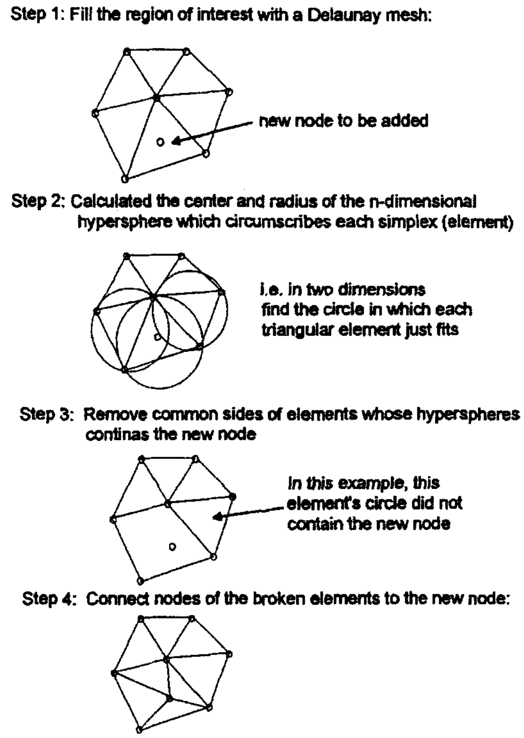


Figure 1. Summary of Watson's algorithm

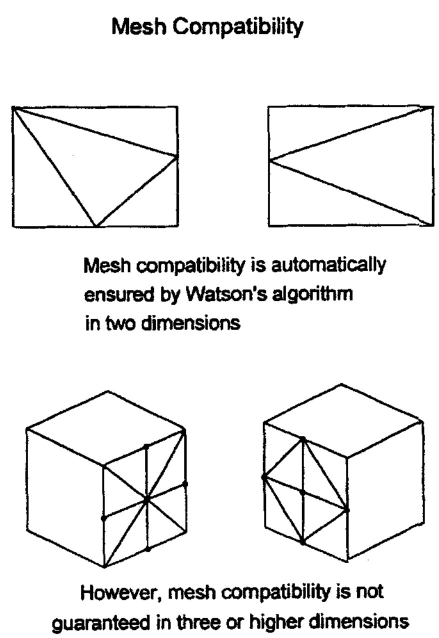


Figure 2. Mesh compatibility

(3) Order the points to be added to the mesh as described below to ensure mesh compatibility.
(4) Use Watson's algorithm to add the above set of points to the tetrahedronal mesh.

The scheme is described as 'alternating' because the initial mesh varies with the position of the square element. Figure 3 shows one possible alternating initial mesh. This example requires four different initial tetrahedral mesh configurations. When combined, they form a kite-like pattern on each face. As the rectangular mesh is being generated, the rectangular regions are subdivided into eight octants. The final octant to which each cube belongs must be stored with that element so that the correct initial mesh can be chosen.

Essentially, the technique works by using the initial mesh configuration to provide a pattern which can then be reproduced on larger neighbouring cube surfaces. That is to say that Watson's method is not random in any sense—if a set of points is added to the initial mesh in a specific order, the resultant mesh is always the same. Further, if a set of points is added to the surface of the mesh in a specific order, it always produces the same pattern on the surface. By using square elements, the pattern produced by Watson's method becomes symmetric.

This kite-like pattern is reproduced on larger cubes using the following point ordering scheme (also depicted in Figures 4):

(1) cube corners (part of initial mesh)
(2) corners of neighbouring cubes in face centre (neighbour 1 division smaller)
(3) corners of neighbouring cubes in centre of each face quadrant (2 divisions smaller)
(4) corners of neighbouring cubes in centre of each face subquadrant (3 sizes smaller)
...
(5) corners of neighbouring cubes 1 division smaller not already included
(6) corners of neighbouring cubes 2 divisions smaller not already included

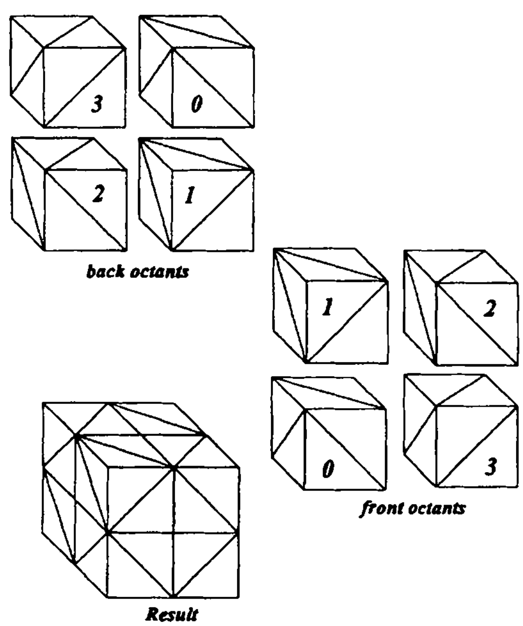


Figure 3. Initial mesh: 'Kite' pattern

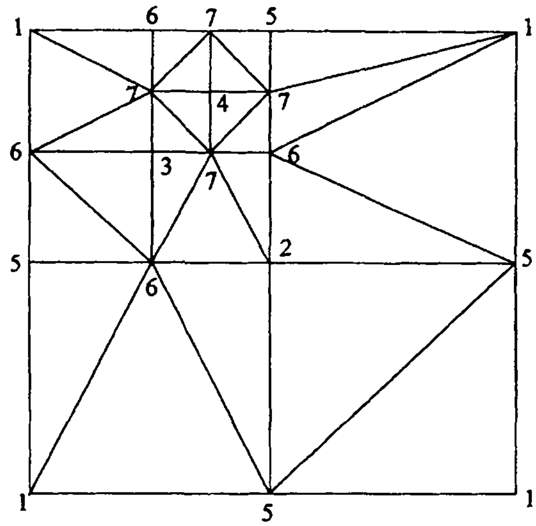
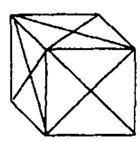


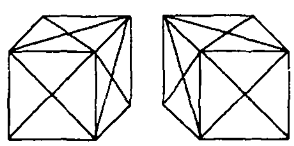
Figure 4. Point ordering scheme for 'Kite' Pattern

(7) corners of neighbouring cubes 3 divisions smaller not already included.
...

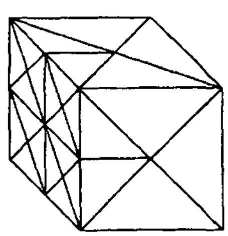
It is possible to create a pattern which does not require alternating the initial mesh. For example, an 'X' pattern can be used as shown in Figure 5. However, the alternating initial mesh method produces tetrahedronal meshes with the minimum number of nodes. Non-alternating patterns, such as the 'X', do so by introducing additional nodes on every cube surface.



Initial Mesh: "X" Pattern



Two Cube of Same Size Self Align



Point Ordering Reproduces X Pattern

Figure 5. Initial mesh: 'X' Pattern

4. FITTING THE DELAUNAY MESH TO THE GEOMETRY

Although the rectangular mesh was created with geometric boundaries in mind, the tetrahedral elements will not necessarily align with the boundary. As a result, additional nodes can be added to the mesh to prevent tetrahedronal elements from crossing these boundaries. This process is sometimes referred to as stitching. Reference 1 gives a good description of this process.

5. SMOOTHING INTERIOR NODES (OPTIONAL)

As a final step, the regularity of the mesh can be disrupted by moving the interior nodes. One such procedure is called lapacing smoothing. In this procedure, each interior node is moved to the averaged centre of all of its neighbouring nodes.[6]

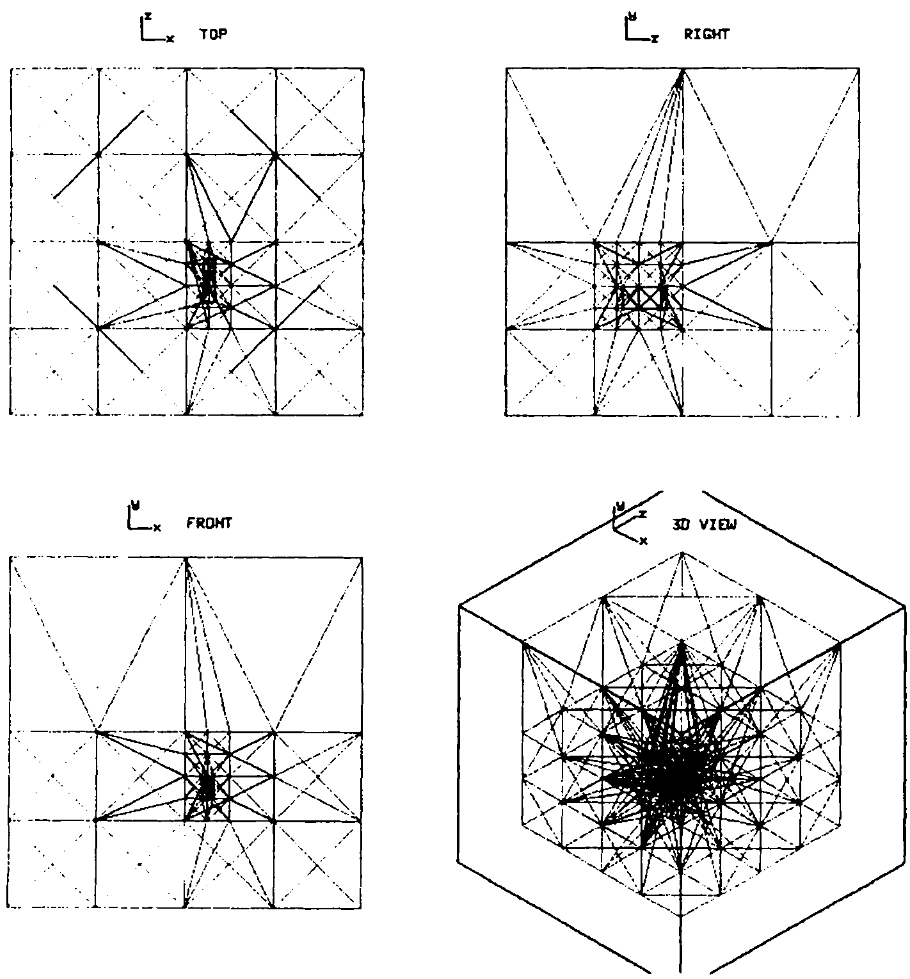


Figure 6. Example 3D mesh

6. FINAL REMARKS

This paper presented a modification to the combined octant/delaunay method which allows for any level of difference between neighbouring octants. However, it should be kept in mind that large difference levels are not always desirable as they often lead to elements with bad aspect ratios, as suggested in Figure 6. This problem is somewhat alleviated by smoothing, but difficulties will arise near boundaries where nodes cannot be moved. It may be possible to reduce the problem by adding nodes to the mesh, for example by dividing octants (in step 1) based on the presence of geometric boundaries in a volume greater than their actual volume or by stitching additional nodes into regions with bad aspect ratios and then smoothing the mesh.

REFERENCES

1. W. J. Schroeder, and M. S. Shepard, 'A combined octree/Delaunay method for fully automatic 3D mesh generation', *Int. j. numer. methods eng.*, **29**, 37–55 (1990).
2. Buratynski, Edward K. 'A fully automatic three-dimensional mesh generator for complex geometries', *Int. j. numer. methods eng.*, **30**, 931–952 (1990).
3. Yerry, Mark A. and Shepard, Mark S. 'Automatic three dimensional mesh generation by the modified-octree technique', *Int. j. numer. methods eng.*, **20**, 1965–1990 (1990).
4. Ho–Le K. 'Finite element mesh generation methods: a review and classification', *Computer Aided Des.*, **20**, 27–38 (1988).
5. Watson, D. F. 'Computing the N-dimensional Delaunay tessellation with application to Voronoi polytopes', *Comput. J.*, **24**, (2), (1981).
6. Yerry, Mark A. and Shepard, M.S. 'A modified quadtree approach to finite element mesh generation', *IEEE Trans. Comput. Graph. Appl.*, 36–46 (1983).