13

Parallel mesh generation based on a new label-driven subdivision technique

K. T. Miura and F. Cheng†*

**Shape Modeling Laboratory, University of Aizu,*
Aizu-Wakamatsu, Fukushima, 965-80, Japan
Phone:+81-242-37-2605, Fax:+81-242-37-2728, E-mail:ktmiura@u-aizu.ac.jp
†Department of Computer Science, University of Kentucky,
Lexington, KY 40506, U.S.A.
Phone:+1-606-257-6760, Fax:+1-606-323-1971, E-mail:cheng@ms.uky.edu

Abstract

A mesh generation algorithm based on a new label-driven subdivision technique is presented. The algorithm generates 2D mesh of quadrilaterals/triangles on a regular quadrilateral network such as the parameter space of a piecewise surface. Each face (patch) of the quadrilateral network has to be assigned a subdivision level (mesh density) first, which is in tern used to assign labels to the vertices of the quadrilateral network. The mesh is generated on the basis of individual faces by performing label-driven subdivision on each face separately. Parallel processing is achieved by performing label-driven subdivision on all the faces simultaneously. The new label-driven subdivision technique improves a previous label-driven subdivision technique in that it does not require the labels of the vertices to satisfy certain requirement and, consequently, the generation of an admissible extension of the given label assignment is not necessary.

Keywords

mesh generation, vertex label assignment, label-driven subdivision, parallel mesh generation

1 INTRODUCTION

Mesh generation is the process of generating finite element models for simulated structural analysis. Since the accuracy of the finite element solution depends directly on the mesh layout, and the cost of the analysis becomes prohibitively

Q. Sun et al. (eds.), *Computer Applications in Production Engineering*

expensive if the number of elements in the mesh is too large, a good mesh generating method should let the user generate a mesh that just fine enough to give an adequate solution accuracy and satisfies the mesh conformity requirement(Cheng et al.,1989).

Several different mesh generation methods are available in the literature. These methods can be categorized as follows:

1. interpolation mesh generation(Gordon and Hall,1973, Harber et al.,1981),
2. automatic triangulation(Rivara,1987, Sadek,1980),
3. quadtree/octree approach(Baehmann et al.,1987),
4. mesh generation based on constructive solid geometry(CSG)(Lee,1984).

Among others, one of the problems in many of the algorithms is that they all require special checking steps to ensure the conformity of the meshes, to allow variable densities and independent local refinement in the surface.

A mesh generation technique which does not require the confirmity checking process has been published recently (Cheng et al., 1989). This mesh generation technique generates 2D mesh of quadrilaterals/triangles on a regular quadrilateral network (such as the parameter space of a piecewise surface) by performing a special subdivision technique, called label-driven subdivision, on the basis of individual faces (patches). Each vertex of the neweork has to be assigned a label first. The label is determined by the subdivision levels (mesh densities) of the adjacent faces. Confirmity requirement is achieved by ensuring that the vertices generated on the common edge of adjacent faces depend on the labels of its labels only. Parallel processing is achieved by performing label-driven subdivision on the faces of the quadrilateral network simultaneously. Unfortunately, the label-driven subdivision process can be performed for faces whose labels satisfy certain requirement only. Otherwise, an admissible extension of the label assignment has to be constructed first.

In this paper, we will present a parallel mesh generation algorithm based on a new label-driven subdivision technique. The new subdivision technique does not require the vertex labels to satisfy any special requirement. Hence, an admissible extension is not required for any given label assignment. Nevertheless, the label-driven subdivision technique has most of the advantages of the previous label-driven subdivision technique, i.e., it is performed on the basis of individual faces and, hence, making parallel processing of the mesh generation process possible; it automatically ensure the conformity of the resultant mesh, no back tracking is required; it allows selective local refinement of the mesh layout. One potential disadvantage of the new subdivision technique is that it would generate more elements in the final mesh than the previous subdivision technique.

2 DEFINITIONS AND NOTATIONS

In this section basic definitions and notations will be summerized, following the same ones used in Cheng et al(1989). We will start with a standard definition of polyhedral networks.

Consider a surface in 3 dimensional space. A *network* on the surface is a finite set of points(called nodes) and curve segments such that

1. each node is an endpoint of a curve segment,
2. each endpoint of a curve segment is a node,
3. two curve segments intersect only at their endpoints.

If the curve segments are segments of straight lines then the corresponding network is called a *polyhedral network.* Polyhedral networks can be represented in a plane by planar graphs. A standard graph terminology (such as vertices, faces, edges) will be used to refer to objects of polyhedral networks. If all of the bounded faces of a polyhedral network are convex quadrilaterals then the network is refered as a *quadrilateral network.* In addition, such a network is called a *regular quadrilateral networks* if the degree of each vertex (in the corresponding planar graph) belonging only to bounded regions is equal to four.

Let F be the set of all faces of a regular quadrilateral network P. A *subdivision level assignment* S of P is a function defined on F, $S : F \rightarrow N \cup \{0\}$, where N is the set of all positive integers. $S(f)$ is called the *subdivision level* of f for $f \in F$.

The problem we consider here is defined as follows which is slightly different from the one dealt with by Cheng at al(1987). Given a regular quadrilateral network P and a subdivision level assignment S on P, our task is to construct a *subdivision mesh* P^* of P such that

(R1) Each specified face f of P is subdivided into at least $9^{S(f)}$ subquadrilaterals.

(R2) The shape of faces generated in P^* is regular, i.e. faces of P^* are not too long or too narrow.

(R3) The resultant subdivision mesh P^* is amenable to local modification, i.e. changing the size or shape of some of the faces without affecting the remainder.

(R4) The number of faces generated in P^* is minimal over all subdivision meshes of P satisfying the goal(R1).

In the next section, we will present a solution to this problem that meets the goals (R1)-(R4).

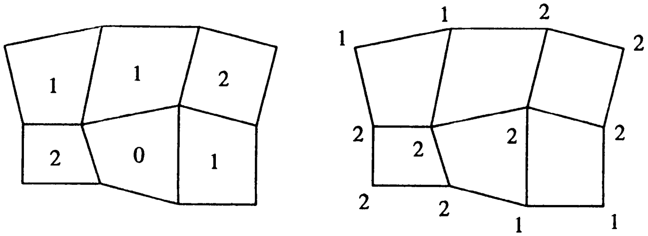
3 PARALLEL MESH SUBDIVISION ALGORITHM

The algorithm consists of the following two phases:

1. vertex label assignment based on subdivision levels,
2. mesh subdivision based on vertex labels.

3.1 Vertex label assignment

Let P be a regular quadrilateral mesh, and V and F be the sets of vertices and faces, respectively. S is a subdivion level assignment of P. A vertex label assignment L of P with respect to S is a function $L : F \rightarrow N \cup \{0\}$ such that $L(v) = \max\{S(f)|f \in F$ and v is a vertex of $f\}$. Figure 1 shows a subdivision level assignment and the corresponding vertex label assignment.



(a) subdivision level (b) vertex label assignment

Figure 1 Vetex label assignment from subdivision level

3.2 Mesh subdivision based on vertex labels

The algorithm is based on five types of elementary subdivision procedures and they are applied repeatedly according to the relative positions of the vertices labeled 0 of a given quadrilateral until the label of every vertex becomes 0.

The types are classidied into five cases except that all labels are 0:

(1) only one label is non-zero,

(2) two adjacent labels are non-zero,

(3) two diagonal labels are non-zero,

(4) three labels are non-zero,

(5) all labels are non-zero.

For each case, one of the elementatary procedues shown in Figures 2-6 will be applied. For example, in case (1):only one label is non-zero, a given quadrilateral $f = v_1v_2v_3v_4$ shown in Figure 2(a) is subdivide into three smaller subquadrilaterals $f_1 = q_1q_2q_3q_4$, $f_2 = r_1r_2r_3r_4$, and $f_3 = s_1s_2s_3s_4$. New labels are assigned as shown in Figure 2(b). For instance, the label of vertex v_1 is originally i. After the subdivision, it becomes $i - 1$. The coordinates of vertices are calculated as the following expressions:

$$\begin{aligned} q_1 &= v_1,\ r_2 = v_2,\ r_3 = s_3 = v_3,\ s_4 = v_4, \\ q_2 &= r_1 = \frac{2v_1 + v_2}{3},\ q_4 = s_1 = \frac{2v_1 + v_4}{3}, \\ q_3 &= r_4 = s_2 = \frac{3v_1 + v_2 + v_3 + v_4}{6}. \end{aligned}$$

In Figures 3-6, original labels and newly assigned ones are shown in (a) and (b), respectively.

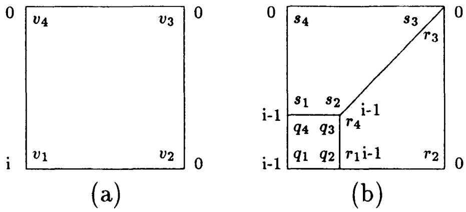


Figure 2 Only one label≥1 (i:vertex label).

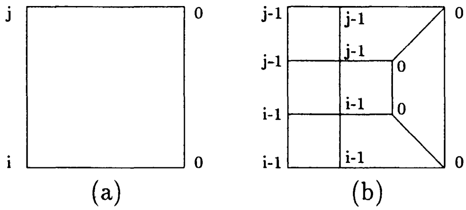


Figure 3 Two labels≥1 (adjacent type).

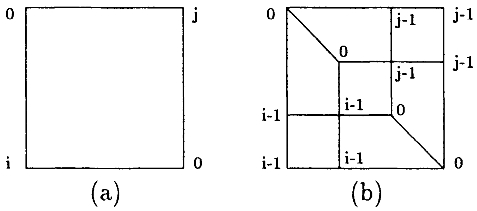


Figure 4 Two labels≥1 (diagonal type).

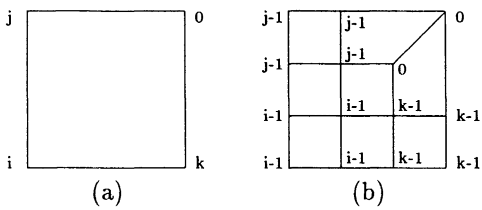


Figure 5 Three labels≥1.

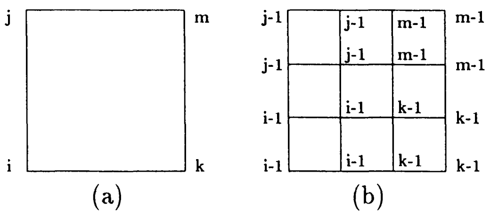


Figure 6 All labels≥1.

3.3 Structure of the algorithm

The overall structure of the algorithm is given in Figure 7. The algorithm has two phases, vertex label assignment and mesh subdivision algorithm explained in the previous sections. We use parallel processing in each phase.

A procedure, subdivide() in Phase 2 is defined in Figure 8. Procedure one_v() is called to subdivide a quadrilateral face one level down if the face has only one non-zero label. It corresponds to the mesh subdivision illustrated in Figure 2.

Similar to one_v(), procedures adjacent_v(), diagonal_v(), ... are all subdivision procedures conducting subdivision in one level deeper, corresponding to different subdivision types illustrated in Figures 3-6.

Algorithm PMS:Parallel Mesh Subdivision
{input:a regular quadrilateral network P and a subdivision level assignment S on P}
{output:a subdivision mesh P^* of P}

Phase 1:[Construct the vertex label assignment L of P with respect to S.]
    PARDO for each vertex v of P **do**
        $L(v)$:=max$\{S(f)|f\in F, v$ is a vertex of $f\}$
    DOPAR

Phase 2:[Subdivide the faces of P in parallel.]
    PARDO for each face f of P **do**
        subdivide(f);
    DOPAR

Figure 7 Parallel mesh subdivision.

subdivide(f:quadrilateral);
    begin
        if(only one label of $f > 0$)**then**
            begin
                one_v(f, g_1, g_2, g_3);
                subdivide(g_1); *subdivide*(g_2); *subdivide*(g_3);
            end
        else if(two adjacent labels of $f > 0$)**then**
            begin
                adjacent_v($f, g_1, g_2, g_3, g_4, g_5, g_6, g_7$); ...
            end
        else if(two diagonal labels of $f > 0$)**then**
                ...
    end;{*subdivide*}

Figure 8 Procedure *subdivide*().

Figures 9-12 illustrate the mesh generation for 2-D grids with different number of faces and subdivision levels. Each resultant quadrilateral is divided into two triangles by one of the diagonals.

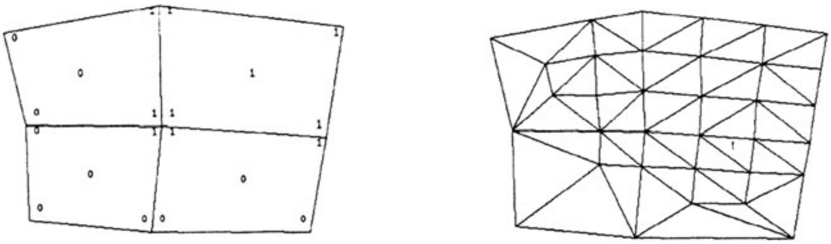


Figure 9 Example no.1.

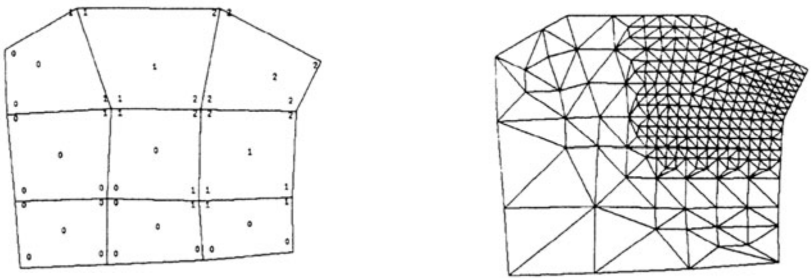


Figure 10 Example no.2.

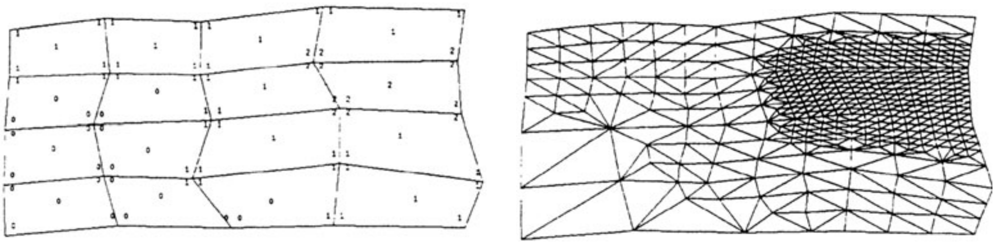


Figure 11 Example no.3.

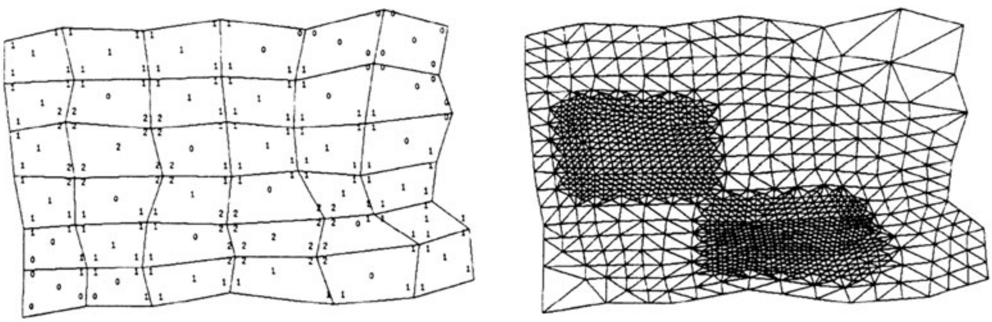


Figure 12 Example no.4.

4 PERFORMANCE ANALYSIS

The algorithm has been implemented in sequential and parallel on Sequent Balance 8000/21000 computer with 26 processors. The performance data for parallel and sequential mesh subdivision recorded from the test are shown in Table 1. The comparison graph bewteen the parallel and sequential version is shown in Figure 13.

In our algorithm, each face can be processed independently. Since the computer has 26 processors, we can process 26 different faces in parallel. As we see from Table 1, when the number of face reaches 25, the ratio of parallel vs sequential is the lowest. The data here is consistent with our expectation.

In Figure 13, the execution time collected for the parallel and the sequential versions is shown. For a regular quadrilateral network of m faces with randomly assigned subdivision levels(maximam subdivision level = 3), the sequential version is implemented using one processor only, the parallel version is implemented using m processors (when the number of processors is greater than the number of patches); one processor per face. According to the data we have collected, parallel version appears well suited to the mesh generation process.

Table 1 Performance data for parallel and sequential mesh subdivision

No. of faces	Parallel(10^{-4}sec)	Sequential(10^{-4}sec)	ration(P/S)
4	3.35	8.93	1:2.67
9	3.40	9.81	1:2.89
16	3.43	22.01	1:6.42
25	4.20	33.94	1:8.08
36	5.11	39.37	1:7.70
49	4.61	24.07	1:5.22

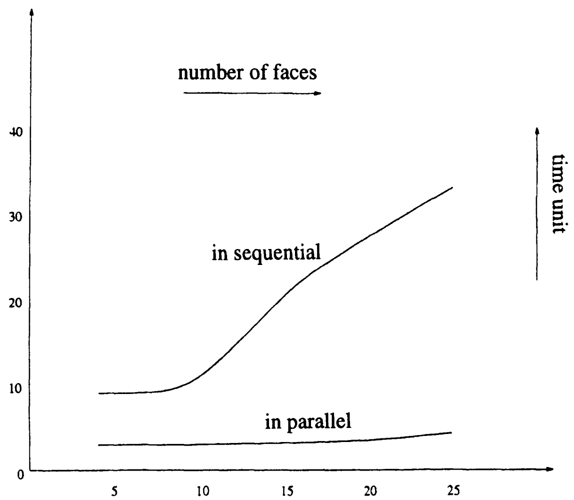


Figure 13 Comparison between the parallel and sequential versions.

5 CONCLUSIONS

A parallel mesh generation algorithm based on a new label-driven subdivision technique is presented. The new label-driven subdivision technique does not impose any restrictions on the labels of the vertices. Hence, it does not require the construction of an admissible extension of the vertex label assignment as does the previous approach. The new algorithm has been implemented in C on a Sequent Balance 21000 computer with 26 processors. The results are quite satisfactory and reliable. Some of the test cases are shown in Figures 9-12.

6 ACKNOWLEDGEMENTS

The authors would like to thank Feng Luo for the implementation of the algorithm presented in this paper. The work of the second author is partially supported by IBM and NSF grant DMI-94000823.

7 REFERENCES

Baehmann,P.L., Wittchen,S.L., Shephard,M.A., Grice,K.R. and Yerry,M.A.(1987) Robust, geometrically based, automatictwo-dimensional mesh generation. *International Journal for Numerical Methods in Engineering*, **24**, 1043-1078.

Cheng,F., Jaromczyk,J.W., Lin,J.R., Chang,S.S. and Lu,J.Y.(1989) A parallel mesh generation algorithm based on the vertex label assignment scheme. *International Journal for Numerical Methods in Engineering*, **28**, 1429-1448.

Gordon,W.F. and Hall,C.A.(1973) Construction of curvilinear coordinate systems and applications to mesh generation. *International Journal for Numerical Methods in Engineering*, **7**, 461-477.

Haber,R.H., Shephard,M.S., Abel,J.F., Gallagher,R.H. and Greenberg,D.P.(1981) A general two-dimensional finite element preprocessor utilizing discrete transfinite mappings. *International Journal for Numerical Methods in Engineering*, **17**, 1015-1044.

Lee,Y.T., De Pennington,A and Shaw,N.K.(1984) Automatic finite-element mesh generation from geometriuc models-A point based approach. *ACM Transactions on Graphics*, **3**, 287-311.

Rivara,M.-C.(1987) A grid generator based on 4-triangles confroming mesh-refinement algorithms. *International Journal for Numerical Methods in Engineering*, **24**, 1343-1354.

Sadek,E.A.(1980) A scheme for the automatic generation of triangular finite elements. *International Journal for Numerical Methods in Engineering*, **15**, 1813-1822.

8 BIBLIOGRAPHY

Kenjiro T. Miura is an associate professor of the Department of Computer Software at the University of Aizu, Japan. He worked for Canon Inc., Japan as a research engineer from 1984 to 1992. His research interests include computer-aided geometric design, computer graphics, and mesh generation. He received his BEng and MEng in precision machinery engineering from the University of Tokyo in 1982 and 1984, respectively and his PhD in mechanical engineering from Cornell University in 1991. He is a member of ACM, ASME, and SIAM.

Fuhua (Frank) Cheng is an associate professor of computer science at the University of Kentucky. He received a BS and an MS in mathematics from the National Tsing Hua University in Taiwan, and an MS in mathematics, an MS in computer science, and a Ph.D. in applied math from the Ohio State University, USA. His research interests include finite-element mesh generation, geometric /solid modeling, computer graphics, and parallel computing in geometric modeling and computer graphics. He is a member of ACM, SIGGRAPH, SIAM, SIAM Activity Group on Geometric Design, and IEEE Computer Society.