A parallel mesh subdivision based on the vertex label assignment

Kenjiro T. MIURA and Fuhua CHENG†*
**Shape Modeling Laboratory, University of Aizu, Aizu-Wakamatus, 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.

Abstract

This paper introduces a new mesh subdivision algorithm based on the vertex label assignment scheme which makes the mesh generation processed in parallel. By subdividing a quadrilateral into at most nine subquadrilaterals in one step instead of four, any combination of labels at the vertices in permitted. It is not necessary to find an admissible extension of a given label assignment.

Keywords

mesh subdivision, vertex label assignment, parallel mesh generation algorithm

1 INTRODUCTION

Mesh generation is the process of generating finite element models for simulated structural analysis. As pointed out by Cheng et al.(1989) and Luo(1993), Since the accuracy of the finite element solution is dependent on the mesh layout, and the cost of the analysis becomes prohibitively expensive if the number of elements inthe 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 with the conformity of the mesh.

There are several different ways for mesh generation 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 in that the algorithms often need to have some special checking steps(if not impossible) in order to ensure the conformity of the meshes, to allow variable density and independent local refinement in the surface.

In this paper, we shall present an algorithm and to generate 2D meshes of triangles based on vertex label assignment. The algorithm is using Divide-and-conquer technique.

The beauty of this algorithm is shown on the following characteristics,

- Automatic conformity of the resultant meshes;
- Subdivision independence.

There is no need for special care on the conformity of the resultant meshes. Local mesh refinement can be carried out without changing the remaining part of the surface so that parallel processing can be implemented.

2 DEFINITION OF THE PROBLEM

We obey almost the same definitions and notations used in Cheng et al.[?]. However, the problem we would like to solve is a little bit modified as follows. Given a regular quadrilateral network **P** and a subdivision mesh $\mathbf{P}^*$ 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 $\mathbf{P}^*$ is regular, i.e. faces of $\mathbf{P}^*$ are not too long or too narrow.

(R3) The resultant subdivision mesh $\mathbf{P}^*$ 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 $\mathbf{P}^*$ is minimal over all subdivision meshes of **P** satisfying the goal(R1).

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

3 PARALLEL MESH SUBDIVISION ALGORITHM

The algorithm is based on five types of elementary subdivision procedures. These procedures will subdivide a given quadrilateral $f = v_1v_2v_3v_4$ into several subquadrilaterals.

1) one_v:only one vertex is labeled 1 or more

This procedure generates four subquadrilaterals $f_1 = q_1q_2q_3q_4$, $f_2 = r_1r_2r_3r_4$, $f_3 = s_1s_2s_3s_4$ and $f_4 = t_1t_2t_3t_4$, and assigns a label to each of their vertices (see Fig. 1).

The new vertices are defined as follows.

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

Labels assigned to the new vertices are defined as follows.

$$\begin{aligned}
Lab(q_1) &= Lab(v_1) - 1, \\
Lab(q_2) &= Lab(r_1) = Lab(q_1), \\
Lab(q_3) &= Lab(r_4) = Lab(s_2) = Lab(q_1), \\
Lab(q_4) &= Lab(s_1) = Lab(q_1), \\
Lab(r_2) &= Lab(r_3) = Lab(s_3) = Lab(s_4) = 0.
\end{aligned}$$

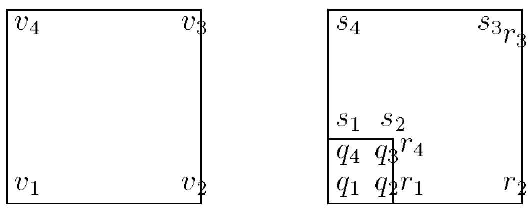

Figure 1: Only one vertex≥1

2) two_va:two adjacent vertices are labeled 1 or more

This procedure generates seven subquadrilaterals $f_1 = q_1q_2q_3q_4$, ..., and $f_7 = x_1x_2x_3x_4$ as shown in Fig.2. The new vertices and their labels are defined as follows.

$$\begin{aligned}
q_1 &= v_1,\ r_2 = u_2 = v_2,\ u_3 = x_3 = v_3,\ w_4 = v_4, \\
q_2 &= r_1 = \frac{2v_1 + v_2}{3},\ w_3 = x_4 = \frac{v_3 + 2v_4}{3}, \\
s_4 &= w_1 = \frac{v_1 + 2v_4}{3},\ q_4 = s_1 = \frac{2v_1 + v_4}{3}, \\
q_3 &= r_4 = s_2 = t_1 = \frac{3v_1 + v_2 + v_3 + v_4}{6}, \\
r_3 &= t_2 = u_1 = \frac{v_1 + 3v_2 + v_3 + v_4}{6}, \\
t_3 &= u_4 = x_2 = \frac{v_1 + v_2 + 3v_3 + v_4}{6}, \\
s_3 &= t_4 = w_2 = x_1 = \frac{v_1 + v_2 + v_3 + 3v_4}{6},
\end{aligned}$$

$$
\begin{aligned}
Lab(q_1) &= Lab(v_1) - 1,\ Lab(w_4) = Lab(v_4) - 1,\\
Lab(q_2) &= Lab(r_1) = Lab(q_1),\\
Lab(q_3) &= Lab(r_4) = Lab(s_2) = Lab(t_1) = Lab(q_1),\\
Lab(q_4) &= Lab(s_1) = Lab(q_1),\\
Lab(s_4) &= Lab(w_1) = Lab(w_4),\\
Lab(s_3) &= Lab(t_4) = Lab(w_2) = Lab(x_1) = Lab(w_4),\\
Lab(w_3) &= Lab(x_4) = Lab(w_4),
\end{aligned}
$$

other new vertices are labeled 0.

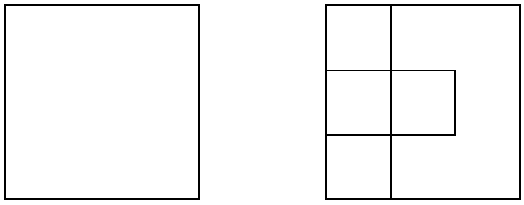

Figure 2: Two vertices$\geq$1 (adjacent type)

3) two_vd:two diagonal vertices are labeled 1 or more

This procedure generates seven subquadrilaterals shown in Fig.3. Similarly,

$$
\begin{aligned}
q_1 &= v_1,\ r_2 = u_2 = v_2,\ x_3 = v_3,\ s_4 = w_4 = v_4,\\
q_2 &= r_2 = \frac{2v_1 + v_2}{3},\ u_3 = x_2 = \frac{v_1 + 2v_3}{3},\\
w_3 &= x_4 = \frac{2v_3 + v_4}{3},\ q_4 = s_1 = \frac{2v_1 + v_4}{3},\\
q_3 &= r_4 = s_2 = t_1 = \frac{3v_1 + v_2 + v_3 + v_4}{6},\\
r_3 &= t_2 = u_1 = \frac{v_1 + 3v_2 + v_3 + v_4}{6},\\
t_3 &= u_4 = w_2 = x_1 = \frac{v_1 + v_2 + 3v_3 + v_4}{6},\\
s_3 &= t_4 = w_1 = \frac{v_1 + v_2 + v_3 + 3v_4}{6},
\end{aligned}
$$

$$
\begin{aligned}
Lab(q_1) &= Lab(v_1) - 1,\ Lab(x_3) = Lab(v_3) - 1,\\
Lab(q_2) &= Lab(r_1) = Lab(q_1),\\
Lab(q_3) &= Lab(r_4) = Lab(s_2) = Lab(t_1) = Lab(q_1),\\
Lab(q_4) &= Lab(s_1) = Lab(q_1),\\
Lab(t_3) &= Lab(u_4) = Lab(w_2) = Lab(x_1) = Lab(x_3),\\
Lab(u_3) &= Lab(x_2) = Lab(x_3),\\
Lab(w_3) &= Lab(x_4) = Lab(x_4),
\end{aligned}
$$

other new vertices are labeled 0.

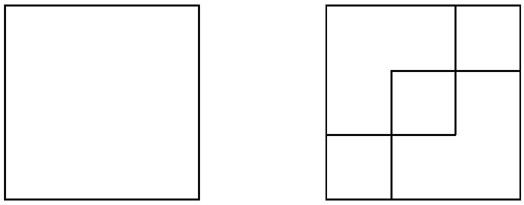

Figure 3: Two vertices$\geq$1 (diagonal type)

4) three_v:three vertices are labeled 1 or more

This procedure generates eight subquadrilaterals shown in Fig.4. Similarly,

$$
\begin{aligned}
q_1 &= v_1,\ s_2 = v_2,\ w_3 = y_3 = v_3,\ x_4 = v_4, \\
q_2 &= r_1 = \frac{2v_1 + v_2}{3},\ r_2 = s_1 = \frac{v_1 + 2v_2}{3}, \\
s_3 &= w_2 = \frac{2v_2 + v_3}{3},\ x_3 = y_4 = \frac{v_1 + 2v_4}{3}, \\
t_4 &= x_1 = \frac{v_1 + 2v_4}{3},\ q_4 = t_1 = \frac{2v_1 + v_4}{3}, \\
q_3 &= r_4 = t_2 = u_1 = \frac{3v_1 + v_2 + v_3 + v_4}{6}, \\
r_3 &= s_4 = u_2 = w_1 = \frac{v_1 + 3v_2 + v_3 + v_4}{6}, \\
u_3 &= w_4 = y_2 = \frac{v_1 + v_2 + 3v_3 + v_4}{6}, \\
t_3 &= u_4 = x_2 = y_1 = \frac{v_1 + v_2 + v_3 + 3v_4}{6},
\end{aligned}
$$

$$
\begin{aligned}
Lab(q_1) &= Lab(v_1) - 1,\ Lab(s_2) = Lab(v_2) - 1, \\
Lab(x_4) &= Lab(v_4) - 1, \\
Lab(q_2) &= Lab(r_1) = Lab(q_1), \\
Lab(q_3) &= Lab(r_4) = Lab(t_2) = Lab(u_1) = Lab(q_1), \\
Lab(q_4) &= Lab(t_1) = Lab(q_1), \\
Lab(r_2) &= Lab(s_1) = Lab(s_2), \\
Lab(r_3) &= Lab(s_4) = Lab(u_2) = Lab(w_1) = Lab(s_1), \\
Lab(s_3) &= Lab(w_2) = Lab(s_2), \\
Lab(t_4) &= Lab(x_1) = Lab(x_4), \\
Lab(t_3) &= Lab(u_4) = Lab(x_2) = Lab(y_1) = Lab(x_4), \\
Lab(x_3) &= Lab(y_4) = Lab(x_4),
\end{aligned}
$$

other new vertices are labeled 0.

5) four_v:all vertices are labeled 1 or more

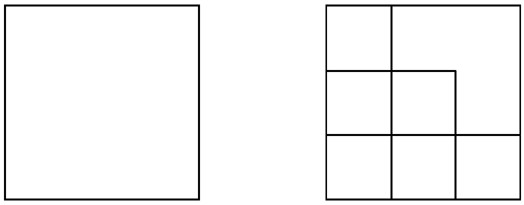

Figure 4: Three vertices$\geq$1

This procedure generates nine subquadrilaterals shown in Fig.5. Similarly,

$$
\begin{aligned}
q_1 &= v_1,\ s_2 = v_2,\ z_3 = v_3,\ x_4 = v_4,\\
q_2 &= r_1 = \frac{2v_1 + v_2}{3},\ r_2 = s_1 = \frac{v_1 + 2v_2}{3},\\
s_3 &= w_2 = \frac{2v_2 + v_3}{3},\ w_3 = z_2 = \frac{v_1 + 2v_3}{3},\\
t_4 &= x_1 = \frac{v_1 + 2v_4}{3},\ q_4 = t_1 = \frac{2v_1 + v_4}{3},\\
q_3 &= r_4 = t_2 = u_1 = \frac{3v_1 + v_2 + v_3 + v_4}{6},\\
r_3 &= s_4 = u_2 = w_1 = \frac{v_1 + 3v_2 + v_3 + v_4}{6},\\
u_3 &= w_4 = y_2 = z_1 = \frac{v_1 + v_2 + 3v_3 + v_4}{6},\\
t_3 &= u_4 = x_2 = y_1 = \frac{v_1 + v_2 + v_3 + 3v_4}{6},
\end{aligned}
$$

$$
\begin{aligned}
Lab(q_1) &= Lab(v_1) - 1,\ Lab(s_2) = Lab(v_2) - 1,\\
Lab(z_3) &= Lab(v_3) - 1,\ Lab(x_4) = Lab(v_4) - 1,\\
Lab(q_2) &= Lab(r_1) = Lab(q_1),\\
Lab(q_3) &= Lab(r_4) = Lab(t_2) = Lab(u_1) = Lab(q_1),\\
Lab(q_4) &= Lab(t_1) = Lab(q_1),\\
Lab(r_2) &= Lab(s_1) = Lab(s_2),\\
Lab(r_3) &= Lab(s_4) = Lab(u_2) = Lab(w_1) = Lab(s_1),\\
Lab(s_3) &= Lab(w_2) = Lab(s_2),\\
Lab(y_3) &= Lab(z_4) = Lab(z_3),\\
Lab(u_3) &= Lab(w_4) = Lab(y_2) = Lab(z_1) = Lab(z_3),\\
Lab(w_3) &= Lab(z_2) = Lab(z_3),\\
Lab(t_4) &= Lab(x_1) = Lab(x_4),\\
Lab(t_3) &= Lab(u_4) = Lab(x_2) = Lab(y_1) = Lab(x_4),\\
Lab(x_3) &= Lab(y_4) = Lab(x_4),
\end{aligned}
$$

other new vertices are labeled 0. The overall structure of the algorithm is given in Fig.6. A procedure, subdivide(g) in Phase 2 is defined in Fig.7.

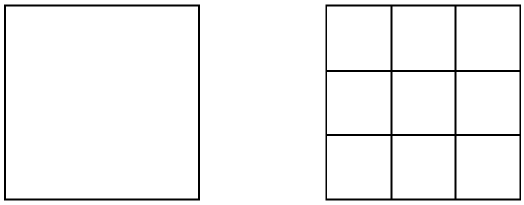

Figure 5: All vertices≥1

Algorithm PMS:Parallel Mesh Subdivision
{input:a regular quadrilateral network **P** and a subdivision level assignment **S** on **P**}
{output:a subdivision mesh $\mathbf{P}^*$ 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 g of **P** **do**
$subdivide(g)$;
DOPAR

Figure 10: Parallel mesh subdivision

$subdivide(g$:quadrilateral);
begin
if($Lab(v) > 0$ for only 1-v v of g)**then**
begin
$one\_v(g, g_1, g_2, g_3, g_4)$;
$subdivide(g_1)$;
$subdivide(g_2)$;
$subdivide(g_3)$;
$subdivide(g_4)$;
end
else if($Lab(v) > 0$ for 2-ad-v v of g)**then**
...
end;{$subdivide$}

Figure 11: Procedure *subdivide*

See Fig.8-9 that illustrate the mesh generation for 2-D grids with different number of faces and subdivision levels.

Figure 8: Example No.1

Figure 9: Example No.2

4 CONCLUSIONS

This paper introduces a new mesh subdivision algorithm based on the vertex label assignment scheme which makes the mesh generation processed in parallel. By subdividing a quadrilateral into at most nine subquadrilaterals in one step instead of four, any combination of labels at the vertices in permitted. It is not necessary to find an admissible extension of a given label assignment. We have implemented the algorithm on a parallel computer and verified its efficiency.

5 REFERENCES

Baehmann,P.L., Wittchen,S.L., Shephard,M.A., Grice,K.R. and Yerry,M.A.(1987) Robust, geometrically based, automatic two-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.

Luo F.(1993) A parallel mesh subdivision generator based on the vertex label assignment. Technical Report, University of Kentucky

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.

6 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, Columbus, Ohio, 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.