An Efficient Paving Method of Pure Quad Mesh Generation

Yongcai Liu, Wenliang Chen and Yidong Bao

ABSTRACT

Triangle mesh and quadrilateral mesh are the most common representations in many applications such as geometric modelling, computer graphics and mechanical engineering. Although triangle mesh are much more popular in applications such as computer graphics, the quadrilateral mesh – which is made only by pure quadrilateral elements – is superior to triangular mesh for finite element method are also well discovered[1] in geometric modelling and simulation applications.

The quadrilateral mesh generation can be classified into two categories: direct method and indirect method. The indirect method is relying on ground triangle mesh [2]. The main idea of this method is to form quadrilateral mesh by merging and recombining triangle mesh[3-4]. The direct method constructs quadrilateral using quadrilateral trees or advancing front method[5-6].The former one may often produce many low-quality quadrilateral elements around the boundaries of surface. The advancing front method for quad mesh generation is considered to be non-robust and computationally expensive.

The method to pure quad mesh generation presented in this paper is a direct approach based on paving method. There are some differences between the traditional paving method, the method has two clear advantages: (i) An alternating
digital tree (ADT) structure[7] are applied to reduce the cost of intersection check of
segments. (ii) Some parallelograms are formed to overcome the disadvantage of non-
robust. The robustness of this method is demonstrated and verified on many complex surfaces.

Method Outline

Some steps of the method in this paper are implemented to ensure correctness
and improve element quality. Because the method is based on paving method, it is
necessary to briefly outline it in the following:

(1) Meshing on surface boundary. Nodes are generated according to element size
defined by the user and curvature of surface boundary.

(2) Initializing set of front edges. A front edge is combined with two successive
adjacent nodes and a state which is defined by two angles, as shown in Fig.1.

(3) Generating quad element and updating front edges. A front edge is selected,
and some new optional nodes are found, more details will be seen in section 3.1.

(4) Checking closure. If there is no front edge on the current paving loop, the
paving is completed. Otherwise, go to step 3.

(5) Topology clean-up and smoothing. When quad mesh generated by paving
algorithm is completed, some operations are necessary to modify topological
structure and geometry information to improve element quality.

We will focus on a comprehensive study ADT data structure for efficient edge
intersection check when traditional box-intersection algorithm is inefficient, and
some operations after a triangle element formed in paving method in the next
section.

Implementation Details

GENERATING QUAD MESH ON PLANER DOMAIN

A front edge selected for constructing an element is called active front edge. The
state of a front edge is classified to four types, as shown in Figure 1. As shown in
Fig.2(a), two nodes on front edge loop will be connected to form a new front edge
and a new quadrilateral element is constructed.
One new optional point is generated if the mode of active front edge is 0-1 or 1-0, as shown in Fig.2(b) and Fig.2(c). The x-y coordinate values of three nodes S, E and D are \((x_1, y_1), (x_2, y_2)\) and \((x, y)\) respectively. Then the coordinate values of the new optional node are:

\[
\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{4} \begin{pmatrix} a_1 + a_2 + a_3 \\ \sin \theta \\ \cos \theta \end{pmatrix} \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix} + \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}
\]  

Equation (1)

Two new optional nodes are generated if the mode is 1-1, which each optional node is achieved by above formula analogously, as shown in Fig.2(d).

If a triangle element is formed in the case of state of 1-0 or 0-1 in traditional paving method, but this is not suitable in pure quad generation in direct method. In order to generate pure quad elements on the planer domain, a parallelogram will be formed according to the active front edge and one of its neighbor nodes.

**ADT DATA STRUCTURE FOR EDGE INTERSECTION CHECKING**

Alternating digital tree (ADT) is an efficient solution for the geometric searching problem and geometric intersection problem [7].

First, the coordinates of a point in space \(\mathbb{R}^N\) are normalized to the range of \([0,1]\), then any coordinate of a point is satisfied:

\[0 \leq x^j \leq 1, \quad j = 1, 2, ..., N.\]

The intersection of any two bounding boxes which contain each an object is the sufficient condition of intersection of two objects. In general, one bounding box is presented by \([x_{0,\text{min}}, x_{0,\text{max}}]\), another bounding box is presented by \([x_{1,\text{min}}, x_{1,\text{max}}]\), two bounding box are intersect if the following condition is satisfied:

\[a \leq x_1 \leq b\]  

Equation (2)
where \( a = [0, \ldots, 0, x_{0, \min}^1, \ldots, x_{0, \min}^N]^T \), \( b = [x_{0, \max}^1, \ldots, x_{0, \max}^N, 1, \ldots, 1]^T \in \mathbb{R}^{2N} \) and \( x_1 = [x_{1, \min}^1, \ldots, x_{1, \min}^N, x_{1, \max}^1, \ldots, x_{1, \max}^N] \in \mathbb{R}^{2N} \).

As a consequence, the problem of intersection of two objects in space \( \mathbb{R}^N \) is transformed to the problem of geometric searching in space \( \mathbb{R}^{2N} \).

The ADT data structure is presented by the binary tree, the storage form of each node in ADT data structure is:

\[
\text{Struct ADT} = \{ \text{bounding box } b; \text{point } pt; \text{ADT* left_child}; \text{ADT* right_child}; \}
\]

where the point \( pt \) is contained in the bounding box \( b \).

A given node \( k \) on the \( l \) level in the ADT structure, and the bounding box is \((c^l_k, d^l_k)\), its left children \((c_{kl}^l, d_{kl}^l)\) and right children \((c_{kr}^l, d_{kr}^l)\) are defined as:

\[
\begin{align*}
    c_{kl}^l &= c^l_k, \\
    d_{kl}^l &= d^l_k & i \neq j, \\
    c_{kl}^l &= c_{kl}^l, \\
    d_{kl}^l &= (c^l_k + d^l_k)/2 & i = j \\
    c_{kr}^l &= c^l_k, \\
    d_{kr}^l &= d^l_k & i \neq j, \\
    c_{kl}^l &= (c^l_k + d^l_k)/2, \\
    d_{kr}^l &= d^l_k & i = j
\end{align*}
\]

(3)

where \( j = 1 + \text{mod}(l, N) \).

Intersecting operation and deleting operation are often used in the ADT when any new object is added or deleted, and the two operations are more efficient than traditional binary tree.

TOPOLOGICAL CLEAN-UP AND SMOOTHING

When the last active front edge has been vanished, it is inevitable that bad-shape quadrilateral elements are introduced, some topological clean-up operations are needed to eliminate irregular nodes.

There are three different operations of topological clean-up to construct well-quality elements. If any angle of a quadrilateral element is larger than 180° and connectivity number of the node is two, the isolated node will be eliminated and a new quadrilateral will be constructed, as shown in Fig.3. If the connectivity numbers of the opposite nodes of a quadrilateral element are both three, the two nodes should be merged to one then two irregular nodes will be vanished, as shown in Fig.4. If there is a quadrilateral element which has three irregular nodes, two connectivity numbers of them are five and another one is three, two nodes should be merged to one and only one irregular node remain in the new mesh while three irregular nodes in old mesh, as shown in Fig.5.

Figure 3. Deleting a node from mesh. Figure 4. Seaming a quadrilateral element with 3-3.
When the non-convex node is on the surface boundary curve and only one quadrilateral contains it, a new node should be added and three new elements are generated to improve the quality of elements, as shown in Fig. 6.

After topological clean-up process, some elements are still bad-quality, a constrained Laplacian smoothing is applied after topological clean-up step.

**Numerical Results**

Based on the paving method and ADT data structure, an automatic method of pure quad mesh generation is implemented in Visual Studio C++ 2013. We have tested this method on many CAD models. Only two examples are showed in this section.

For a quadrilateral element, the quality of it is defined as:

\[
J = \frac{\min(s_{\Delta ABC}, s_{\Delta BCD}, s_{\Delta CDA}, s_{\Delta DAB})}{\max(s_{\Delta ABC}, s_{\Delta BCD}, s_{\Delta CDA}, s_{\Delta DAB})}
\]  

(4)

where \(s_{\Delta ABC}\) is sign area of a triangle ABC. The value ranges from 0 (degenerate element) to 1 (ideal shape one).

As shown in Fig. 7, there is a domain composed of a complicated boundary. Most elements (>99%) whose quality values are larger than 0.5 are acceptable.
There is a multi-connected domain composed of a rectangle exterior boundary and two circular interior boundaries as shown in Fig.8. There are well-aligned elements around interior boundaries. The speed of our method is faster than variant of paving method[5] as shown in Table 1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#num quadrilaterals</th>
<th>Variant of Paving method[5]</th>
<th>Our method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enneagon</td>
<td>3790</td>
<td>12.55</td>
<td>2.52</td>
</tr>
<tr>
<td>Two-circle</td>
<td>700</td>
<td>0.23</td>
<td>0.11</td>
</tr>
</tbody>
</table>

CONCLUSIONS

The improved paving method in this paper is presented to generate pure quad elements on the planer domains and surfaces. There are some improvements including efficient resolution of geometric intersections by employing ADT data structure which speed up intersection checking, forming some parallelograms to overcome disadvantage of non-robust in traditional paving method. Overall quality value of result mesh is desirable, while the number of elements is reasonably small to save the cost of numerical calculations in FEM simulation.

REFERENCES

5. Yu Zhao, Bo Yu, Wenquan Tao. 2013, An Improved Paving Method of Automatic Quadrilateral Mesh Generation [J], Numerical Heat Transfer Fundamentals, 64(3)218-238.