Applied Mathematical Modelling 51 (2017) 764–790

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Improved boundary constrained tetrahedral mesh generation by shell transformation

Jianjun Chen[a,b,*], Jianjing Zheng[a], Yao Zheng[a], Hang Si[c], Oubay Hassan[b], Kenneth Morgan[b]

[a] Center for Engineering and Scientific Computation, and School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China
[b] Zienkiewicz Centre for Computational Engineering, College of Engineering, Swansea University, Swansea SA2 8PP, Wales, UK
[c] Weierstrass Institute for Applied Analysis and Stochastics, Mohrenstrasse 39, 10117 Berlin, Germany

ARTICLE INFO

Article history:
Received 10 December 2015
Revised 21 June 2017
Accepted 3 July 2017
Available online 20 July 2017

Keywords:
Mesh generation
Boundary recovery
Shell transformation
Delaunay triangulation
Steiner points
Tetrahedral meshes

ABSTRACT

An excessive number of Steiner points may be inserted during the process of boundary recovery for constrained tetrahedral mesh generation, and these Steiner points are harmful in some circumstances. In this study, a new flip named *shell transformation* is proposed to reduce the usage of Steiner points in boundary recovery and thus to improve the performance of boundary recovery in terms of robustness, efficiency and element quality. Shell transformation searches for a local optimal mesh among multiple choices. Meanwhile, its recursive callings can perform flips on a much larger element set than a single flip, thereby leading the way to a better local optimum solution. By employing shell transformation properly, a mesh that intersects predefined constraints intensively can be transformed to another one with much fewer intersections, thus remarkably reducing the occasions of Steiner point insertion. Besides, shell transformation can be used to remove existing Steiner points by flipping the mesh aggressively. Meshing examples for various industrial applications and surface inputs mainly composed of stretched triangles are presented to illustrate how the improved algorithm works on difficult boundary constrained meshing tasks.

1. Introduction

1.1. Reducing the usage of Steiner points in boundary recovery: why?

The Delaunay criterion provides a reasonable method to triangulate a given point set. However, boundary constraints may be lost in the resulting mesh, and either *conforming* or *constrained* methods are required to recover the lost constraints. For the conforming recovery method, Steiner points are inserted on the constraints and are not removed in the resulting meshes; thus, some of the lost constraints are recovered as concatenations of sub-constraints. For the constrained recovery method, the constraints are the same as the prescribed ones, and no Steiner points can remain on the constraints.

* Corresponding author at: Center for Engineering and Scientific Computation, and School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China.
E-mail address: chenjj@zju.edu.cn (J. Chen).

http://dx.doi.org/10.1016/j.apm.2017.07.011
0307-904X/

There is no guarantee to recover an edge or face in a tetrahedral mesh without adding Steiner points [1]. The typical failing examples are Schönhardt polyhedron [2] and Chazelle polyhedron [3], which can only be tetrahedralized by adding Steiner points. Therefore, a robust three-dimensional boundary recovery algorithm must contain a *main procedure* that considers how to insert Steiner points [4–19]. For instance, Weatherill and Hassan [4] presented an algorithm for constructing 3D conforming triangulations, with Steiner points inserted on surface boundaries. Later, George et al. [5] and Du and Wang [6] presented a very similar point-splitting idea to attempt to remove all Steiner points from surface boundaries, which is successful in a large percentage (but <100%) of application instances.

In [15], we presented a boundary recovery algorithm that first inserts Steiner points at intersection positions between lost boundary constraints and the tetrahedral mesh to achieve a conforming recovery, and then removes these points from the surface to achieve the final constrained recovery. In the appendix of this paper, we provide the *theoretical* proofs to explain why this algorithm could output a constrained recovery result by calling a finite number of local operations on the tetrahedral mesh. Nevertheless, these proofs do not consider the round-off errors due to floating point numbers. Thus, the robustness of the algorithm presented in [15] could be challenged in the real world. It was observed that this algorithm likely fails when an excessive number of Steiner points are required during the boundary recovery procedure. This undesirable result occurs when the input surface contains a certain number of elements having high aspect ratios. In this circumstance, Steiner points are harmful to robustness and efficiency of boundary recovery and element quality.

(1) *Robustness*. Predicates such as those proposed by Shewchuk [20] can enhance the robustness of boundary recovery remarkably. However, the positions of Steiner points stored with floating-point numbers are essentially inaccurate due to round-off errors. These errors can accumulate if an excessive number of Steiner points are inserted. Predicates with these positions as inputs may return an undesirable value and collapse the entire boundary recovery procedure.
(2) *Efficiency*. For each Steiner point, massive time-consuming computations accompany with its creation, movement and suppression. Thus, the timing cost of a boundary recovery procedure is roughly proportional to the number of Steiner points.
(3) *Element quality*. Steiner points destroy local mesh size specifications and introduce elements having volumes close to zero. Various schemes have been proposed to improve the local mesh quality [21], but these schemes may fail when many bad elements cluster in a local region where many Steiner points are located. This case usually happens near bad surface triangles, and the situation becomes worse under the combined influence of Steiner points and bad boundaries.

In addition, the above issues have the *locality* nature: one stretched element or small angle in the surface may introduce many Steiner points in its neighborhood; if several stretched elements and/or small angles are adjacent to each other, an excessive number of Steiner points may be inserted locally. Therefore, although the input surfaces in practical applications are mainly composed of well-shaped triangles, the above issues may appear occasionally if undesirable geometry features are neighbored with each other. This sort of local imperfection may exist due to many reasons, for instance, when the geometry itself contains undesirable features, or when the mesh gradation is out of control locally. In parallel mesh generation [22–24], the domain decomposition approach may introduce undesirable artificial features on the inter-domain interfaces. In hybrid mesh generation for viscous simulations [25–27], the boundary layer mesher may introduce low-quality faces that are parts of the inputs for the tetrahedral mesher. In simulations of moving boundary problems [28–30], mesh faces were stretched in the mesh movement process, and some of them may appear in the boundary of the hole to be remeshed. To tackle the issue of minimizing the usage of Steiner points during boundary recovery is undoubtedly beneficial for these applications.

1.2. The role of mesh flip based schemes on boundary recovery

It is NP complete to predict whether a polyhedron can be tetrahedralized without adding Steiner points [1], and the lower bound of the number of Steiner points is shown to be quadratic [3]. Due to these theoretical difficulties, many heuristic schemes are employed to reduce the number of Steiner points [4–19]. These schemes can be classified into *the preprocessing scheme* and *the postprocessing scheme*. The preprocessing schemes improve mesh topologies to reduce intersections between lost boundary constraints and mesh entities before Steiner point insertion, while the postprocessing schemes suppress Steiner points directly after Steiner point insertion. For the same input, the numbers of Steiner points inserted by different boundary recovery algorithms may vary wildly.

Here, the preprocessing scheme must be highlighted. It produces a topologically improved mesh for the main procedure. The robustness and efficiency of the main procedure are mainly determined by the *quality* of this mesh. In previous studies, tremendous efforts have been made to tackle the main procedure. With respect to the preprocessing scheme, most of boundary recovery algorithms rely on a simple procedure that iteratively conducts the basic flips, i.e., 2-3, 3-2 and 4-4 flips (the numbers in these names denote the number of tetrahedra removed and created by the flips, respectively, see Fig. 1a and 1b) [31,32]. Nevertheless, one single flip calling may fail because it only involves a small number of elements. In the context of mesh improvement, Joe suggested improving the performance of flips by composing multiple basic flips [31]. In fact, Joe identified nine combinational operations; however, as pointed out by Shewchuk [33], the most elaborate combinational operations can be expressed as one or two *edge removal* operations [33,34]. Later, Liu and Baida [10] suggested adopting basic flips *recursively* for boundary recovery purposes. Although it was reported that their algorithm could perform