Solving The Cutting Flow Problem for Prismatic Mesh Subdivision

Xiaotian Yin* Wei Han† Xianfeng Gu‡ Shing-Tung Yau§

November 5, 2012

Abstract

This paper is motivated by the problem of subdividing a prismatic mesh to a tetrahedral mesh (without inserting Steiner points) so as to not only match arbitrarily prescribed boundary conditions but also allow arbitrary topologies in the base mesh. We explore all possible combinations of these two factors, and propose a complete solution to this 3D problem by converting it to an equivalent 2D graph problem, called "cutting flow problem". For each case, we not only prove the sufficient and necessary condition for the existence of solutions, but also provide linear and provable algorithms to compute a solution whenever there is one.

1 Motivations

A **prismatic mesh** consists of a set of triangular prisms, where each prism is a volumetric element bounded by two triangular faces and three quadrilateral faces, and different prisms are glued together along same type of faces (i.e. triangle to triangle, quadrangle to quadrangle). It in general comes in layers, where each layer is an extrusion of a triangular mesh (i.e. **base mesh**) along a line interval (i.e. fiber).

prismatic meshes are often required to be converted to tetrahedral meshes, especially for the purpose of computation and simulation. In finite element methods, many solvers are designed for tetrahedral meshes and do not support prismatic elements. In computer graphics, many efficient algorithms for volume rendering, iso-contouring and particle advection only work for meshes of tetrahedra. Therefore how to triangulate a prismatic mesh becomes a desirable task.

Splitting a single prism into three tetrahedra is an easy task, but cutting a set of prisms consistently is much more challenging. Here we only consider conversions without inserting additional points (i.e. Steiner points). Under certain circumstances user may wish to have control on the boundary triangulation, i.e. the subdivision of the quadrilateral faces on the boundary of the prismatic mesh, and the internal subdivision must conform to such boundary conditions. In addition, the underlying base mesh may have various topologies, which could bring another level of difficulty to the problem of extending the boundary triangulation into the inside.

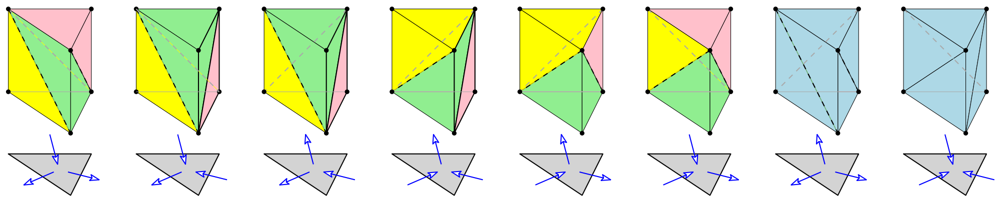
There has been a rich literature on triangulating non-tetrahedral volumetric meshes. As an example, [1] proposed an algorithm to subdivide a volumetric mesh consisting of mixed elements (pyramids, prisms, and hexahedra) into tetrahedra by comparing and ordering vertex indices. However, most of these works do not discuss fixed boundary conditions, which makes a completely different problem. In one of our earlier work [2], we studied this problem with prescribed boundary conditions, but only for topological disks. To our best knowledge, the result presented here is the first complete solution to prismatic mesh subdivision for all possible boundary conditions and base topologies.

2 Problem Statement

We subdivide each layer in a prismatic mesh separately, and formulate this problem as an equivalent 2D graph flow problem in the underlying base mesh. The work is based on the following intuition.

For every individual prism in the mesh, as shown in Figure 1, each quadrilateral face should be split into two triangles, either through the *diagonal* (lower-left to upper-right) or *anti-diagonal* (lower-right to upper-left). We model such a splitting process by assigning directed flows across edges of the base triangular face. If a quadrilateral face is split along diagonal, we put a flow into the base triangle across the corresponding

*Mathematics Department, Harvard University, xyin@math.harvard.edu.
†Mathematics Department, Harvard University, weihan@math.harvard.edu.
‡Computer Science Department, Stony Brook University, gu@cs.sunysb.edu.
§Mathematics Department, Harvard University, yau@math.harvard.edu.



(a1) (a2) (a3) (a4) (a5) (a6) (a7) (a8)

Figure 1: The intuition: subdividing each individual prism by cutting flow. (a1) to (a6) are valid cuttings, (a7) and (a8) are invalid.

base edge; otherwise, put a flow out of the base triangle. As shown in Figure 1, a splitting over a prism is valid if and only if there are both inflow and outflow in the base triangle. In addition, two adjacent prisms should have a consistent flow on their common quadrilateral face.

To formalize this problem, we need some notations here. Given a triangular mesh G (i.e. a primal graph), denote its *augmented dual graph* (or *dual graph* for short) as $\widetilde{G^*} = (V^* \cup \widetilde{V}, E^* \cup \widetilde{E})$, where E^* and $\widetilde{E}$ are sets of dual edges corresponding to the inner and boundary primal edges in G respectively, V^* is a set of dual vertices corresponding to the primal faces in G, $\widetilde{V}$ is a set of virtual dual vertices placed off the boundary of G to bound the edges in $\widetilde{E}$.

Now we can formally define the original 3D problem of prismatic mesh subdivision as an equivalent 2D flow problem in the augmented dual graph.

Problem 1. (*The Cutting Flow Problem*) Given a triangular mesh M with augmented dual graph $\widetilde{G^*} = (V^* \cup \widetilde{V}, E^* \cup \widetilde{E})$, find a flow (called *cutting flow*) on the edge set of $\widetilde{G^*}$, such that:

- *Fixed Boundary*: The flow on $\widetilde{E}$ is given as input and cannot be changed.
- *No Source/Sink*: Every vertex in V^* must have both inflow and outflow.

3 Results

In this work we provide a complete solution to the cutting flow problem (and therefore the original problem of prismatic mesh subdivision). We consider all possible boundary conditions (fixed or free) and all possible base mesh topologies (simply-connected or multiply-connected, planar or non-planar). For each combination of these two factors, we not only prove the sufficient and necessary condition for existence of solutions, but also provide efficient algorithms to find a solution if there is one.

The results are summarized in table 1, case by case. Note that this problem is solvable in most cases, except for a special case on simply-connected planar domain. In fact, the problem is not solvable if and only if the dual graph is a tree and the boundary condition is uniform. In any other case, the problem can be solved by a linear algorithm.

Topo / *Bnd*	*Simply-Connected & Planar*	*Multiply-Connected & Planar*	*Multiply-Connected & Non-Planar*	*Simply-Connected & Non-Planar*
Fixed	**Conditionally Solvable**	**Always Solvable**	**Always Solvable**	**Always Solvable**
Free	**Always Solvable**			

Table 1: Results for all possible cases of different boundary conditions ("*Bnd*") and base topologies ("*Topo*").

References

[1] J. Dompierre, P. Labb, M.-G. Vallet, and R. Camarero. How to subdivide pyramids, prisms, and hexahedra into tetrahedra. In *8th International Meshing Roundtable*, pages 195–204, 1999.

[2] X. Yin, W. Han, X. Gu, and S.-T. Yau. The cutting pattern problem for tetrahedral mesh generation. In *20th International Meshing Roundtable*, pages 217–236, 2011.