A new constrained Delaunay tetrahedralization algorithm

Hang Si

Weierstrass Institute for Applied Analysis and Stochastics, Berlin, Germany
si@wias-berlin.de

1 Introduction

Constrained Delaunay tetrahedralizations (CDTs) have many optimal properties similar to those of Delaunay tetrahedralizations (DTs) [8, 9]. They are eligible structures for mesh generation.

A major task in constructing CDTs is how to recover the domain boundaries, i.e., edges and faces. At first hand, one might think it is an easy task since it has been addressed intensively in the literature. Unfortunately, this problem is far from being solved in 3D. Classical engineering algorithms [3, 12, 2] come with no performance guarantee. Their details (involving swaps and point insertions) are complicated to realize. On the other hand, CDT algorithms with theoretical guarantees are proposed [6, 7, 11]. The algorithm [11] is available in the program `TetGen` [10].

In this paper, we present a new algorithm for constructing CDTs for 3-dimensional polyhedral domains. This algorithm is based on our previous one [11]. A new facet recovery algorithm is proposed. It can handle facets which are not exactly planar – a problem ubiquitously exists in engineering data and the use of exact geometric predicates. We distinguish the inconsistencies between 2 and 3 dimensional constrained Delaunay simplices. These inconsistencies are removed by facet re-meshing and Steiner point insertions. This algorithm has no complex detail and is easy to implement. It can be extended to handle boundary consists of smoothly curved surfaces.

2 The Algorithm

The input is a 3D *piecewise linear complex* [4] (PLC) $\mathcal{X}$, i.e., $\mathcal{X}$ is a collection of polyhedra of dimensions up to 3, see Figure 1 left. We call 1- and 2-dimensional polyhedra of $\mathcal{X}$ *segments* and *facets*. The *boundary complex* of $\mathcal{X}$ is its 2-skeleton, which is the set of vertices, segments, and facets of $\mathcal{X}$.

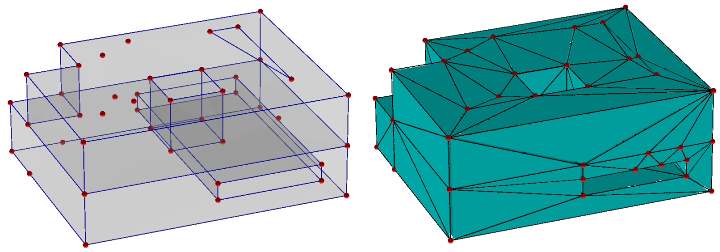

Fig. 1. Left: A 3D PLC. Right: The surface triangulation of the facets of the PLC with Steiner points inserted on its segments.

Our algorithm constructs a CDT $\mathcal{T}$ of $\mathcal{X}$. $\mathcal{T}$ may contain Steiner points, i.e., each segment and facet of $\mathcal{X}$ is represented by the union of a subcomplex of $\mathcal{T}$. We call 1- and 2-dimensional simplices of these subcomplexs *subsegments* and *subfaces* to distinguish them from other simplices of $\mathcal{T}$.

The proposed algorithm proceeds in the increasing order of the dimensions of the skeletons of $\mathcal{X}$. It is divided into three phases:

1. Create a DT $\mathcal{D}_0$ of the vertex set of $\mathcal{X}$.
2. Let $\mathcal{D}_1 = \mathcal{D}_0$. Recover segments of $\mathcal{X}$ in a DT $\mathcal{D}_1$, refine $\mathcal{X}$ into $\mathcal{X}'$.
3. Let $\mathcal{D}_2 = \mathcal{D}_1$. Recover facets of $\mathcal{X}'$ in a CDT $\mathcal{D}_2$.

The first and second phases of this algorithm are already discussed in [11]. Steiner points are inserted in the segments of $\mathcal{X}$ such that every subsegments are Delaunay in $\mathcal{D}_1$. A symbolic perturbation technique [6] is used to ensure that the vertex set of $\mathcal{D}_1$ is in *general position*, i.e., no 5 points share a common sphere. After the second phase, the original PLC $\mathcal{X}$ has been refined into a PLC $\mathcal{X}'$ including Steiner points. It has been proven by Shewchuk [5] that the CDT of $\mathcal{X}'$ exists. Facet can be recovered without Steiner points.

Shewchuk's theorem [5] is based on an assumption that all vertices of the facets are exactly co-planar. Unfortunately, it is commonly not the case in the real world data and by the use of finite precision of computer's floating point numbers. Therefore, a new facet recovery algorithm is developed to incorporate this situation. It is detailed in the following section.

2.1 Facet recovery

Every facet $F \in \mathcal{X}'$ is first triangulated into a 2-dimensional CDT $\mathcal{T}_F$, see Fig. 1 right. F is recovered in a 3-dimensional CDT $\mathcal{D}_2$ when all its subfaces are also faces of $\mathcal{D}_2$. The algorithm is summarized in Fig. 2.

From each missing subface σ one can form a *missing region* Ω which is a set of subfaces such that: *(i)* the edges on the boundary of Ω are edges of $\mathcal{D}_2$, and *(ii)* the edges in the interior of Ω are missing in $\mathcal{D}_2$.

```
FacetRecovery (X', D_1)
D_2 := D_1;
Initialize a queue Q of all subfaces of X';
while Q ≠ ∅ do
    pop a subface σ from Q;
    if σ is missing in D_2, then
        form a missing region Ω containing σ;
        search a set C of crossing tetrahedra of Ω;
        if C ≠ ∅ then
            RemeshCavity(C, Ω);
            if Ω is not recovered, then
                RefineRegion(X', D_2, Ω);
            endif
        else
            RemeshRegion(D_2, Ω);
        endif
    endif
endwhile
return D_2;
```

Fig. 2. The facet recovery algorithm.

Facets with exactly co-planar vertices. In this case, the 2D CDT of a facet $F \in \mathcal{X}'$ is unique viewed by all vertices of $\mathcal{X}'$. If a subface σ is missing in the current tetrahedralization $\mathcal{D}_2$, then there must exist at least one tetrahedron (called *crossing tetrahedra*) in $\mathcal{D}_2$ whose interior intersect with σ. The set of crossing tetrahedra of Ω forms a cavity C inside the space of $\mathcal{D}_2$. The boundaries of C are faces in $\mathcal{D}_2$. The sub-routine RemeshCavity presented in [11] replaces the set of crossing tetrahedra by a set of new tetrahedra conforming to both the boundaries of C and Ω. This process needs no Steiner points and $\mathcal{D}_2$ is updated into a new CDT conforming to Ω.

Facets with not exactly co-planar vertices. In this case, there are two types of inconsistencies: *(1)* the vertices of $\mathcal{X}'$ may not agree on a unique 2D CDT of a facet F, and *(2)* even they agree on a unique 2D CDT, it may not be the current triangulation of F. Type-*(2)* was due to the pre-computation of the facet CDT assumes that the vertices of F are co-planar. To avoid these inconsistencies in advance would need expensive computations. The proposed algorithm try to handle them during the execution of the algorithm.

A type-*(2)* inconsistency is detected when we found no crossing tetrahedron for a missing region Ω. It implies that Ω can be re-meshed by a set of faces in $\mathcal{D}_2$. The RemeshRegion sub-routine searches the set of faces in $\mathcal{D}_2$ conforming to Ω and replaces the old set of subfaces in Ω by the new set. As a result, the facet triangulation has been corrected by the constrained Delaunay faces of $\mathcal{D}_2$. This process needs no Steiner points.

The type-*(1)* inconsistency is difficult to be detected earlier. It could be signaled when the RemeshCavity is failed to recover Ω. A typical failure caused by this inconsistency is that one found that a quadrilateral *abcd* in Ω can not be recovered since the two triangulations of *abcd* in the two halfspaces of Ω are not the same. This is due to the non-coplanarity of the quadrilateral *abcd*. To resolve this case, Steiner points are added into Ω. We choose to split an edge of one of the not recovered subfaces in Ω. This is done in the RefineRegion sub-routine.

Note that by adding only one Steiner point may not be sufficient to achieve a new CDT. For an example, if this point is inserted on a subsegment of $\mathcal{X}'$, it may encroaches upon other subsegments of $\mathcal{X}'$. Therefore, some tetrahedra containing these subsegments may not be constrained Delaunay anymore. A solution to solve this problem is to continue the insertion of Steiner points until no subsegment of $\mathcal{X}'$ is encroached. The RefineRegion sub-routine may insert several Steiner points into $\mathcal{D}_2$, and update $\mathcal{D}_2$ into a new CDT including these Steiner points.

2.2 Termination

The main challenge in the proof of the termination is the additions of Steiner points in the facet recovery phase. A key lemma we need is: The RefineRegion sub-routine returns a new CDT (with added Steiner points) of the set of recovered subfaces. Note that this sub-routine repairs the edges of the facet triangulations which are not constrained Delaunay in 3 dimensions. Once such edges are removed (by Steiner points), the facet triangulations can be recovered by either RemeshCavity or RemeshRegion. A complete proof of this lemma is our future work.

3 Examples and Discussions

This algorithm has been implemented in the latest version of the program `TetGen` [10] (version 1.4.3). An example is shown in Fig. 3. Comparing with the old algorithm, this algorithm outperforms it in both of its robustness and speed. Moreover, the new algorithm adds less Steiner points. One reason is due to the number of segments of the input is reduced when adjacent not exactly co-planar facets are merged into one facet.

In the future, the termination and correctness of this algorithm need to be proven. The analysis of this algorithm is not complete yet. A number of questions arise: How to incrementally update a CDT? How many Steiner points are needed in this process? A very interesting extension of this algorithm is to consider a piecewise smooth complex [1] as input.

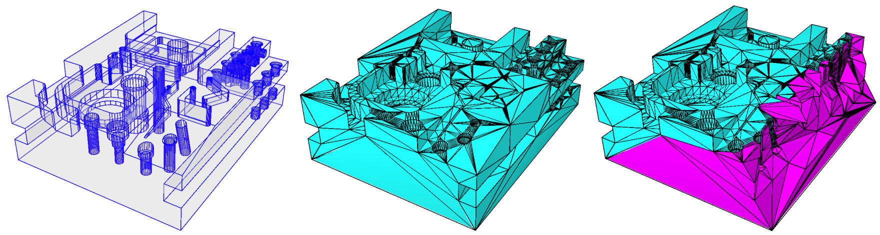

Fig. 3. Example. Left: a input PLC containing non-coplanar facets. Middle: the surface mesh of the generated CDT. Right: a cut view of the CDT.

References

1. S.-W. Cheng, T. K. Dey, and E. A. Ramos. Delaunay refinement for piecewise smooth complexes. In *Proc. 18th annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1096–1105, 2007.
2. P. L. George, H. Borouchaki, and E. Saltel. Ultimate robustness in meshing an arbitrary polyhedron. *Int. J. Numer. Meth. Engng*, 58:1061–1089, 2003.
3. P. L. George, F. Hecht, and E. Saltel. Automatic mesh generator with specified boundary. *Computer Methods in Applied Mechanics and Engineering*, 92:269–288, 1991.
4. G. L. Miller, D. Talmor, S.-H. Teng, N. J. Walkington, and H. Wang. Control volume meshes using sphere packing: Generation, refinement and coarsening. In *Proc. 5th International Meshing Roundtable*, 1996.
5. J. R. Shewchuk. A condition guaranteeing the existence of higher-dimensional constrained Delaunay triangulations. In *Proc. 14th Annual Symposium on Computational Geometry*, pages 76–85, June 1998.
6. J. R. Shewchuk. Constrained Delaunay tetrahedralization and provably good boundary recovery. In *Proc. 11th International Meshing Roundtable*, pages 193–204, September 2002.
7. J. R. Shewchuk. Updating and constructing constrained Delaunay and constrained regular triangulations by flips. In *Proc. 19th Annual Symposium on Computational Geometry*, pages 181–190, June 2003.
8. J. R. Shewchuk. General-dimensional constrained Delaunay and constrained regular triangulations I: Combinatorial properties. *Discrete and Computational Geometry*, 39:580–637, 2008.
9. H. Si. *Three dimensional boundary conforming Delaunay mesh generation.* PhD thesis, Institute of Mathematics, Technische Universität Berlin, 2008. `http://opus.kobv.de/tuberlin/volltexte/2008/1966/`.
10. H. Si. TetGen. `http://tetgen.berlios.de`, 2009.
11. H. Si and K. Gärtner. Meshing piecewise linear complexes by constrained Delaunay tetrahedralizations. In *Proc. 14th International Meshing Roundtable*, pages 147–163, 2005.
12. N. P. Weatherill and O. Hassan. Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. *Int. J. Numer. Meth. Engng*, 37:2005–2039, 1994.