

Meshing Volumes Bounded by Smooth Surfaces*

Steve Oudot[†] Laurent Rineau[‡] Mariette Yvinec[§]

Abstract

This paper introduces a three-dimensional mesh generation algorithm for domains bounded by smooth surfaces. The algorithm combines a Delaunay-based surface mesher with a Ruppert-like volume mesher, to get a greedy algorithm that samples the interior and the boundary of the domain at once. The algorithm constructs provably-good meshes, it gives control on the size of the mesh elements through a user-defined sizing field, and it guarantees the accuracy of the approximation of the domain boundary. A noticeable feature is that the domain boundary has to be known only through an oracle that can tell whether a given point lies inside the object and whether a given line segment intersects the boundary. This makes the algorithm generic enough to be applied to a wide variety of objects, ranging from domains defined by implicit surfaces to domains defined by level-sets in 3D grey-scaled images or by point-set surfaces.

1 Introduction

Simplicial meshes are one of the most popular representations for surfaces, volumes, scalar fields and vector fields, in applications such as Geographic Information Systems (GIS), computer graphics, virtual reality, medical imaging and finite element analysis. However, constructing discrete representations of continuous objects can be time-consuming, especially when the geometry of the object is complex. In this case, mesh generation becomes the pacing phase in the computational simulation cycle. Roughly speaking, the more the user is involved in the mesh generation process, the longer the latter is. An appealing example is given in [Mav00], where the mesh generation time is shown to be 45 times that required to compute the solution. This motivates the search for

*Work partially supported by the IST Programme of the EU as a Shared-cost RTD (FET Open) Project under Contract No IST-006413 (ACS - Algorithms for Complex Shapes) and by the European Network of Excellence AIM@shape (FP6 IST NoE 506766).

[†]INRIA, BP 93 06902 Sophia Antipolis, France

[‡]ENS, 45 rue d'Ulm, 75005 Paris, France

[§]INRIA, BP 93 06902 Sophia Antipolis, France

fully-automated mesh-generation methods, which inherently require the use of guaranteed-quality meshing algorithms.

Delaunay refinement is recognized as one of the most powerful techniques for generating meshes with guaranteed quality. It allows the user to get an easy control on the sizes of the mesh elements, for instance through a (possibly non-uniform) sizing field. Moreover, it constructs meshes with a good grading, able to conform to quickly varying sizing fields. The pioneer work on Delaunay refinement is due to Ruppert [Rup95], who proposed a two-dimensional mesh generator for domains with piecewise linear boundaries and constraints. Provided that the boundaries and constraints do not form angles smaller than $\frac{\pi}{3}$, Ruppert's algorithm guarantees a lower bound on the smallest angle in the mesh. Furthermore, this bound is achieved by adding an asymptotically optimal number of Steiner vertices. Later on, Shewchuk improved the handling of small angles in two dimensions [She02] and generalized the method to the meshing of three-dimensional domains with piecewise linear boundaries [She98]. The handling of small angles is more puzzling in three dimensions, where dihedral angles and facet angles come into play. Using the idea of protecting spheres around sharp edges, first proposed by Cohen-Steiner et al. [CCY04], Cheng and Poon [CP03] provided a thoroughful handling of small input angles formed by boundaries and constraints. Cheng et al. [CDRR04] turned the same idea into a simpler and practical meshing algorithm.

In three-dimensional space, Delaunay refinement is able to produce tetrahedral meshes with an upper bound on the radius-edge ratios of the tetrahedra, where the radius-edge ratio of a tetrahedron is the ratio between its circumradius and the length of its shortest edge. This eliminates from the mesh all kinds of degenerate tetrahedra, except the ones called *slivers*. A sliver can be described as a tetrahedron formed by four vertices close to the equatorial circle of a sphere and roughly equally spaced on this circle. Cheng et al. [CDE⁺00], and later on Cheng and Dey [CD02], proposed to exude slivers from the mesh by turning the Delaunay triangulation into a weighted Delaunay triangulation with carefully-chosen small weights applied to the vertices. Li and Teng [LT01] proposed to avoid slivers by relaxing the choice of refinement vertices inside small areas around the circumcenters of the elements to be refined.

The main drawback of the above techniques is that they deal exclusively with domains with piecewise linear boundaries, whereas in many applications, objects have curved boundaries. In such applications, time is spent discretizing the boundary B of the object into a polyhedron P , before the interior of the object can be sampled. Then, the original boundary B is dropped away and replaced by its discretized version, P . On one hand, mesh generation algorithms based on advancing front methods [FBG96], as well as some Delaunay refinement techniques, like the unit edge mesher of [GHS90, GHS91], construct meshes that conform strictly to the discretized boundary P . On the other hand, Ruppert-like methods [She98] refine the boundary mesh: whenever a point should be inserted on B , it is in fact inserted on P . However, in both cases, the quality of the resulting mesh and the accuracy of the boundary approximation depend highly on the initial surface mesh P .

Several methods have been proposed for meshing two-dimensional or three-dimensional domains with curved boundaries. Most of them deal only with specific types of boundaries (parametric, implicit etc.) [SU01], or they simply come with no guarantee regarding the topology of the output mesh, or the quality of its elements, or even the termination of the process [DOS99, LBG96, ACSYD05]. One noticeable exception is [BOG02], where the algorithm is able to handle any two-dimensional domain bounded by piecewise smooth curves, of any type, provided that a small number of geometric quantities can be estimated, such as the curvature of a given curve at a given point or the total variation of the unit tangent vector between two points on a given curve. The problem with this method is that it is designed exclusively for the two-dimensional case. Moreover, estimating the required geometric quantities can be time-consuming on certain types of curves.

In this paper, we take advantage of recent results on the front of smooth surface meshing and approximation using Delaunay refinement [BO05], to build a fully-automated algorithm that can mesh three-dimensional domains bounded by smooth surfaces. Specifically, we combine the surface mesher of [BO05] with a Ruppert-like volume mesher, to get a greedy Delaunay-based algorithm that samples the interior and the boundary of the domain at the same time. A noticeable feature of this algorithm is that the boundary of the object has to be known only through an oracle that can answer two simple geometric questions: whether a given point lies inside the object, and whether a given line segment intersects the boundary. This makes the algorithm generic enough to be applied to objects with a wide variety of boundary types, such as implicit surfaces, level-sets in 3D grey-scaled images, point-set surfaces, etc. Concerning guarantees, our algorithm terminates and constructs good-quality meshes for domains whose boundaries are (not necessarily connected) smooth surfaces. The sizes of the mesh elements are controlled through a user-defined sizing field. Moreover, the accuracy of the approximation of the original boundary is guaranteed, and the size of the output mesh is bounded.

The paper is organized as follows. Section 2 recalls a few known facts about restricted Delaunay triangulations and surface approximation. Section 3 describes the main algorithm. Section 4 deals with the accuracy of the approximation of the object by the output mesh. In Section 5, we prove that the meshing algorithm terminates, and we bound the number of vertices of the output mesh. Section 6 addresses the practicality of the algorithm: it gives some details about the choice of the sizing field and it explains how to remove slivers. Finally, Section 7 provides a few examples and experimental results.

2 Preliminary definitions

In the sequel, \mathcal{O} denotes a bounded open subset of the Euclidean space \mathbb{R}^3 , and $\bar{\mathcal{O}}$ denotes the topological closure of \mathcal{O} . We call $\partial\mathcal{O}$ the boundary of \mathcal{O} , and we assume that $\partial\mathcal{O}$ is $C^{1,1}$, *i.e.* that its normal vector field is well-defined and

1-Lipschitz.

Definition 2.1.

- The medial axis M of $\partial\mathcal{O}$ is the topological closure of the set of points of \mathbb{R}^3 that have at least two nearest neighbors on $\partial\mathcal{O}$. Every point of M is the center of an open ball that is maximal w.r.t. inclusion among the set of open balls included in $\mathbb{R}^3 \setminus \partial\mathcal{O}$. Such a ball is called a medial ball.

- Given a point $x \in \mathbb{R}^3$, we call distance to the medial axis at x , or $d_M(x)$, the Euclidean distance from x to M .

It is well-known [Fed70] that, since $\partial\mathcal{O}$ is $C^{1,1}$, the infimum of d_M over $\partial\mathcal{O}$ is positive. This infimum is called the *reach* of $\partial\mathcal{O}$. The class of surfaces with positive reach has been intensively studied in the recent years, and the distance to the medial axis was used to define a notion of *good* sample, called ε -sample [AB99]:

Definition 2.2. Let \mathcal{P} be a finite set of points. \mathcal{P} is a ε -sample of $\partial\mathcal{O}$ if it is included in $\partial\mathcal{O}$ and if $\forall x \in \partial\mathcal{O}$, $d(x, \mathcal{P}) \leq \varepsilon d_M(x)$.

Definition 2.3. Let \mathcal{P} be a finite set of points.

- The Voronoi cell of $p \in \mathcal{P}$ is the set of all points of \mathbb{R}^3 that are closer to p than to any other $p' \in \mathcal{P}$.

- The Voronoi diagram of \mathcal{P} , $\mathcal{V}(\mathcal{P})$, is the cellular complex formed by the Voronoi cells of the points of \mathcal{P} .

It is well known that, if the points of \mathcal{P} are in general position, then the dual complex of $\mathcal{V}(\mathcal{P})$ is a tetrahedrization of the convex hull of \mathcal{P} , called the *Delaunay triangulation* ($\mathcal{D}(\mathcal{P})$ for short). The meshing strategy described in this paper relies on a subcomplex of $\mathcal{D}(\mathcal{P})$, defined below.

Definition 2.4. Let \mathcal{P} be a finite point set.

- The Delaunay triangulation of \mathcal{P} restricted to \mathcal{O} , or $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ for short, is the subcomplex of $\mathcal{D}(\mathcal{P})$ formed by the tetrahedra whose dual Voronoi vertices lie in \mathcal{O} .

- The Delaunay triangulation of \mathcal{P} restricted to $\partial\mathcal{O}$, or $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ for short, is the subcomplex of $\mathcal{D}(\mathcal{P})$ formed by the triangles whose dual Voronoi edges intersect $\partial\mathcal{O}$.

Given a facet f of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ and its dual Voronoi edge e , every point of $e \cap \partial\mathcal{O}$ is the center of an open ball containing no point of \mathcal{P} , and whose bounding sphere passes through the vertices of f . This ball is called a *surface Delaunay ball* of \mathcal{P} .

The main idea of our algorithm is to sample \mathcal{O} and $\partial\mathcal{O}$ greedily and simultaneously, using $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ and $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ to drive the choice of the next point to insert. The output is a point set whose restriction to $\partial\mathcal{O}$ is a *loose ε -sample* of $\partial\mathcal{O}$ [BO05]:

Definition 2.5. Let \mathcal{P} be a finite point set, and ε be a positive value. \mathcal{P} is a loose ε -sample of $\partial\mathcal{O}$ if the following conditions hold:

L1 $\mathcal{P} \subset \partial\mathcal{O}$;

L2 $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ has vertices on every connected component of $\partial\mathcal{O}$;

L3 the center c of any surface Delaunay ball of \mathcal{P} is closer to \mathcal{P} than $\varepsilon d_M(c)$.

Notice that ε -samples verify Assertions L1 and L3. Moreover, if $\varepsilon < 0.1$, then L2 is verified as well, by Theorem 2 of [AB99]. It follows that any ε -sample is a loose ε -sample, for $\varepsilon < 0.1$. Loose ε -samples enjoy many properties [BO05], which we summarize below:

Theorem 2.6. *If \mathcal{P} is a loose ε -sample of $\partial\mathcal{O}$, with $\varepsilon \leq 0.09$, then $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ is a closed 2-manifold ambient isotopic to $\partial\mathcal{O}$, at Hausdorff distance $O(\varepsilon^2)$ from $\partial\mathcal{O}$, and its normals approximate the normals of $\partial\mathcal{O}$ within an error of $O(\varepsilon)$. Moreover, $\partial\mathcal{O}$ is covered by the surface Delaunay balls of \mathcal{P} , and \mathcal{P} is a $\varepsilon(1 + 8.5\varepsilon)$ -sample of $\partial\mathcal{O}$.*

Ambient isotopy and Hausdorff approximation are most interesting for our problem. As for normal and curvature approximations, they are useful in all applications where the user wants to estimate differential quantities on surfaces.

3 Main algorithm

The algorithm takes as input the domain \mathcal{O} to be meshed, a sizing field σ , and two parameter values α and B . The domain is known through an oracle that can tell whether a given point lies inside \mathcal{O} or outside. The oracle can also detect whether a given segment intersects $\partial\mathcal{O}$ and, in the affirmative, return all the points of intersection (which are finitely many, generically). The sizing field is a positive function $\sigma : \bar{\mathcal{O}} \rightarrow \mathbb{R}^+$ defined over $\bar{\mathcal{O}}$ and assumed to be 1-Lipschitz. The algorithm first constructs an initial point set $\mathcal{P}_i \subset \partial\mathcal{O}$ that is a $\frac{1}{3}$ -sparse 0.09-sample of $\partial\mathcal{O}$, that is:

- $\forall x \in \partial\mathcal{O}, d(x, \mathcal{P}_i) \leq 0.09 d_M(x)$;
- $\forall p \in \mathcal{P}_i, d(p, \mathcal{P}_i \setminus \{p\}) \geq 0.03 d_M(p)$.

The construction of such a point set is described extensively in [BO05], thus we skip it here. Once \mathcal{P}_i is built, the algorithm constructs \mathcal{P} iteratively, starting with $\mathcal{P} = \mathcal{P}_i$ and inserting one point in \mathcal{P} per iteration. In the meantime, the restricted Delaunay triangulations $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ and $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ are maintained, using the oracle.

At each iteration, one element of the mesh (a facet of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ or a tetrahedron of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$) is refined. To refine a tetrahedron, the algorithm inserts its circumcenter in \mathcal{P} . A facet f of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ may be circumscribed by several surface Delaunay balls. Thus, to refine f , the algorithm inserts in \mathcal{P} the center of the surface Delaunay ball $B(c, r)$ circumscribing f with largest ratio $r/\sigma(c)$. The choice of the next element to be refined is driven by the following rules, considered in this order:

- R1** if a facet f of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ does not have its three vertices on $\partial\mathcal{O}$, then refine f ;
- R2** if a facet f of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ has a surface Delaunay ball $B(c, r)$ with ratio $r/\sigma(c) > \alpha$, then refine f ;
- R3** if a tetrahedron t of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ has a circumradius greater than $\sigma(c)$, where c is the circumcenter of t , or if t has a radius-edge ratio greater than B , then consider the circumcenter c of t :
 - R3.1** if c is not included in any surface Delaunay ball, then insert c in \mathcal{P} ;
 - R3.2** else, insert in \mathcal{P} the center of one surface Delaunay ball containing c .

The algorithm terminates when the triggering conditions of Rules R1, R2 and R3 are no longer met. Upon termination, every facet of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ has its three vertices on $\partial\mathcal{O}$ (Rule R1) and every surface Delaunay ball $B(c, r)$ has a radius $r \leq \alpha \sigma(c)$ (Rule R2). Moreover (Rule R3), every tetrahedron t of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ has a circumradius $r \leq \min\{\sigma(c), B l_{min}\}$, where c is the circumcenter of t and l_{min} is the length of the shortest edge of t .

4 Approximation accuracy

In this section, we assume that the algorithm terminates. Termination is discussed in Section 5, which uses several results stated here. From now on, \mathcal{P}_i denotes the initial point set and \mathcal{P} the output point set. Let $\mathcal{P}_{|\partial\mathcal{O}} = \mathcal{P} \cap \partial\mathcal{O}$.

Since \mathcal{P}_i is a 0.09-sample of $\partial\mathcal{O}$, $\mathcal{P}_{|\partial\mathcal{O}}$ is also a 0.09-sample of $\partial\mathcal{O}$, since no point is deleted during the course of the algorithm. Thus, $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$ is a closed 2-manifold with the same topology type as $\partial\mathcal{O}$, by Theorem 2.6. Therefore, to have topological guarantees on the output of the algorithm, it suffices to prove that the boundary of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ is equal to $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$.

There exists a strong relationship between the boundary of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ and $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$:

Lemma 4.1. *The boundary of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ is a subcomplex of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$. Moreover, if every edge of the Voronoi diagram $\mathcal{V}(\mathcal{P})$ intersects $\partial\mathcal{O}$ at most once, and transversally, then the boundary of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ is equal to $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$.*

Proof. Since $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ is a union of Delaunay tetrahedra, its boundary is a union of Delaunay facets. Let f be a facet of the boundary of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$. By definition, it belongs to two Delaunay tetrahedra, one of which has its dual Voronoi vertex inside \mathcal{O} , whereas the other one has its dual Voronoi vertex outside \mathcal{O} . It follows that the Voronoi edge dual to f intersects $\partial\mathcal{O}$, which means that $f \in \mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$.

Let us now assume that every edge of $\mathcal{V}(\mathcal{P})$ intersects $\partial\mathcal{O}$ at most once. Let f be a facet of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$. By definition, the Voronoi edge dual to f intersects $\partial\mathcal{O}$. Since this edge intersects $\partial\mathcal{O}$ only once, one of its vertices lies inside \mathcal{O} whereas the other one (which may be at infinity) lies outside \mathcal{O} . It follows, by definition of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$, that one of the Delaunay tetrahedra incident to f belongs to $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$, while the other one does not. Hence, f belongs to the boundary of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$. □ □

In our case, $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ is precisely the boundary of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$, due to the following result:

Lemma 4.2. *Every edge of $\mathcal{V}(\mathcal{P})$ intersects $\partial\mathcal{O}$ at most once, and transversally.*

Proof. Among the edges of $\mathcal{V}(\mathcal{P})$, only those whose dual Delaunay facets have their three vertices on $\partial\mathcal{O}$ can intersect $\partial\mathcal{O}$, thanks to Rule R1. Let e be such an edge. It is included in an edge e' of $\mathcal{V}(\mathcal{P}_{|\partial\mathcal{O}})$. Since $\mathcal{P}_{|\partial\mathcal{O}}$ is a 0.09-sample of $\partial\mathcal{O}$, Lemma 3.6 of [BO05] tells that e' intersects $\partial\mathcal{O}$ at most once, and transversally, which yields the lemma. \square \square

Corollary 4.3. *The boundary of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ is $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$.*

It follows from Corollary 4.3 that, if we can prove that $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}) = \mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$, then the boundary of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ will be equal to $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$, which approximates $\partial\mathcal{O}$ topologically. We need an intermediate result.

Lemma 4.4. *$\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ has vertices on all the connected components of $\partial\mathcal{O}$.*

Proof. By Rule R1, every edge e of $\mathcal{V}(\mathcal{P})$ that intersects $\partial\mathcal{O}$ has a dual Delaunay facet f whose three vertices are in $\mathcal{P}_{|\partial\mathcal{O}}$. Since $\mathcal{P}_{|\partial\mathcal{O}}$ is a 0.09-sample of $\partial\mathcal{O}$, the point $c = e \cap \partial\mathcal{O}$ lies at distance at most $0.09 d_M(c)$ from the vertices of f . It follows, by Lemma 8 of [AB99], that c and the vertices of f lie on the same connected component of $\partial\mathcal{O}$. As a consequence, to prove the lemma, it suffices to show that every connected component of $\partial\mathcal{O}$ is intersected by at least one Voronoi edge.

Notice that every connected component \mathcal{C} of $\partial\mathcal{O}$ is the frontier between two connected components Ω_1 and Ω_2 of $\mathbb{R}^3 \setminus \partial\mathcal{O}$, so that every connected path from Ω_1 to Ω_2 crosses \mathcal{C} . Therefore, to prove that \mathcal{C} is intersected by a Voronoi edge, it suffices to prove that Ω_1 and Ω_2 both contain Voronoi vertices, since the graph made of the Voronoi vertices and edges is connected.

Let us assume for a contradiction that some component Ω of $\mathbb{R}^3 \setminus \partial\mathcal{O}$ contains no Voronoi vertex. Since the Delaunay balls centered at the Voronoi vertices (including the ones at infinity) cover \mathbb{R}^3 , at least one such ball (say $B(c, r)$) contains a point x of $M \cap \Omega$. Since c lies outside Ω while x lies inside, the line segment $[c, x]$ intersects the boundary of Ω (which is part of $\partial\mathcal{O}$). Let y be a point of intersection. The ball centered at y , of radius $d(x, y)$, is contained in the interior of $B(c, r)$. Therefore, it contains no point of \mathcal{P} . Now, its radius is $d(x, y)$, which is at least the distance from y to M since $x \in M \cap \Omega$. It follows that y is farther from \mathcal{P} than $d_M(y)$, which contradicts the fact that $\mathcal{P}_{|\partial\mathcal{O}}$ is a 0.09-sample of $\partial\mathcal{O}$. It follows that Ω contains at least one Voronoi vertex, which ends the proof of Lemma 4.4. \square \square

We can now prove that $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}) = \mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$, by using the fact that $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ is the boundary of a three-dimensional object, namely $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ (Corollary 4.3).

Lemma 4.5. *$\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}) = \mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$.*

Proof. Thanks to Rule R1, all the facets of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ have their three vertices in $\mathcal{P}_{|\partial\mathcal{O}}$, hence their dual Voronoi edges are included in edges of $\mathcal{V}(\mathcal{P}_{|\partial\mathcal{O}})$. It follows that $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ is a subcomplex of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$.

To prove the lemma, it suffices then to show that every facet of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$ is also a facet of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$. Let us assume for a contradiction that there exists a facet f of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$ that is not a facet of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$. Let C be the connected component of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$ to which f belongs. By Lemma 8 of [AB99], the vertices of C belong to a single component \mathcal{C} of $\partial\mathcal{O}$. By Lemma 4.4, at least one vertex v of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ lies on \mathcal{C} . Let C' be the connected component of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ that contains v . Since $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ is a subset of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$, C' is included in C (which is a connected 2-manifold without boundary). Moreover, since f is not included in C' while v is, C' has a boundary. Now, by Corollary 4.3, C' is a connected component of the boundary of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$. Thus, C' cannot have a boundary, which raises a contradiction. \square \square

It follows from the previous results that the boundary of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ is equal to $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$, which is ambient isotopic to $\partial\mathcal{O}$, by Theorem 2.6. In addition to this topological result, we would like to give a bound on the Hausdorff distance between $\partial\mathcal{O}$ and the boundary of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$, depending on the input sizing field σ . Let $\varepsilon = \min\{0.09, \sup_{x \in \partial\mathcal{O}} \frac{\alpha \sigma(x)}{d_M(x)}\}$. Our bound will depend on ε . So far, we know that $\mathcal{P}_{|\partial\mathcal{O}}$ is a 0.09-sample of $\partial\mathcal{O}$.

Lemma 4.6. *The surface Delaunay balls of \mathcal{P} and those of $\mathcal{P}_{|\partial\mathcal{O}}$ are the same.*

Proof. Since every edge of $\mathcal{V}(\mathcal{P})$ that intersects $\partial\mathcal{O}$ is included in an edge of $\mathcal{V}(\mathcal{P}_{|\partial\mathcal{O}})$, the surface Delaunay balls of \mathcal{P} are also surface Delaunay balls of $\mathcal{P}_{|\partial\mathcal{O}}$. Let us show that the converse is true. Let e be an edge of $\mathcal{V}(\mathcal{P}_{|\partial\mathcal{O}})$. If $e \cap \partial\mathcal{O} \neq \emptyset$, then $|e \cap \partial\mathcal{O}| = 1$, by Lemma 3.6 of [BO05]. Moreover, the Delaunay facet dual to e belongs to $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$, by Lemma 4.5. This means that e contains an edge e' of $\mathcal{V}(\mathcal{P})$, such that $|e' \cap \partial\mathcal{O}| \geq 1$. Hence $e \cap \partial\mathcal{O} = e' \cap \partial\mathcal{O}$. \square \square

Thanks to Lemma 4.6, Rule R2 controls the radii of all the surface Delaunay balls of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$, which implies that, upon termination of the algorithm, $\mathcal{P}_{|\partial\mathcal{O}}$ is a loose ε -sample of $\partial\mathcal{O}$. Hence, $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$ (and therefore the boundary of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$) approximates $\partial\mathcal{O}$ both topologically and geometrically.

Theorem 4.7. *$\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ is a 3-manifold ambient isotopic to $\bar{\mathcal{O}}$, at Hausdorff distance $O(\varepsilon^2)$ from $\bar{\mathcal{O}}$, where $\varepsilon = \min\{0.09, \sup_{x \in \partial\mathcal{O}} \frac{\alpha \sigma(x)}{d_M(x)}\}$. Moreover, the surface Delaunay balls of \mathcal{P} cover $\partial\mathcal{O}$.*

Proof. By Corollary 4.3 and Lemma 4.5, the boundary of $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ is equal to $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$. Since $\mathcal{P}_{|\partial\mathcal{O}}$ is a loose ε -sample of $\partial\mathcal{O}$, we know by Theorem 2.6 that there exists an ambient isotopy $h : [0, 1] \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that maps $\partial\mathcal{O}$ to $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$. The map $h(1, \cdot) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is an ambient homeomorphism that maps the compact 3-manifold $\bar{\mathcal{O}}$ to a compact 3-manifold bounded by $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$. Now, the only compact 3-manifold bounded by $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}})$ is

$\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ itself¹. Thus, we have $h(1, \bar{\mathcal{O}}) = \mathcal{D}_{|\mathcal{O}}(\mathcal{P})$, which means that $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ is ambient isotopic to $\bar{\mathcal{O}}$.

Since $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ and $\bar{\mathcal{O}}$ are both compact, their Hausdorff distance is achieved by a pair of points lying on their boundaries. Hence, we have $d_H(\mathcal{D}_{|\mathcal{O}}(\mathcal{P}), \bar{\mathcal{O}}) = d_H(\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_{|\partial\mathcal{O}}), \partial\mathcal{O})$, which is $O(\varepsilon^2)$ since $\mathcal{P}_{|\partial\mathcal{O}}$ is a loose ε -sample of $\partial\mathcal{O}$. As for the fact that the surface Delaunay balls of \mathcal{P} cover $\partial\mathcal{O}$, it is a direct consequence of Theorem 2.6 and Lemma 4.6. \square \square

Observe that the results of this section do not rely on Rule R3. Hence, they hold not only upon termination, but also during the course of the algorithm, each time neither Rule R1 nor Rule R2 can be applied. In particular, Theorem 4.7 holds every time Rule R3 is triggered. This observation will be instrumental in proving Lemma 5.2 of Section 5.

5 Termination and size of the output

In this section, we provide conditions on parameters α and B to ensure that the algorithm terminates. We assume that the sizing field σ is 1-Lipschitz over $\bar{\mathcal{O}}$.

Our strategy is to prove an upper bound on the size of the point sample constructed by the algorithm. The termination of the algorithm results from this bound.

Definition 5.1. *Given a point p inserted in \mathcal{P} by the algorithm, the insertion radius of p , or $r(p)$ for short, is the Euclidean distance from p to \mathcal{P} right before its insertion². The insertion radius of a point p of the initial point set \mathcal{P}_i is the Euclidean distance from p to $\mathcal{P}_i \setminus \{p\}$.*

Our first task is to provide a lower bound on the insertion radius of every point of \mathcal{P} . In fact, we will prove a stronger result, stated as Lemma 5.2. We define a sizing field σ_0 which can be considered as an extension of d_M over $\bar{\mathcal{O}}$:

$$\forall x \in \bar{\mathcal{O}}, \sigma_0(x) = \inf \{d(x, x') + d_M(x') \mid x' \in \partial\mathcal{O}\}$$

As proved in [MTT99, TW00, ACSYD05], σ_0 is a 1-Lipschitz function, equal to $d_M(x)$ on $\partial\mathcal{O}$. In fact, σ_0 is the pointwise maximal 1-Lipschitz function which is at most d_M on $\partial\mathcal{O}$. Let $\sigma'(p) = \min\{\alpha \sigma(p), 0.03 \sigma_0(p)\}$, $\forall p \in \bar{\mathcal{O}}$. Notice that, since σ and σ_0 are 1-Lipschitz, σ' is γ -Lipschitz, where $\gamma = \max\{\alpha, 0.03\}$.

Lemma 5.2. *If $\alpha < \frac{1}{5}$ and $B \geq \frac{4}{1-5\gamma}$, then the following conditions are verified:*

- C1 $\forall p \in \mathcal{P}, r(p) \geq \sigma'(p)$;
- C2 $\forall p \in \mathcal{P} \setminus \mathcal{P}_{|\partial\mathcal{O}}, \delta(p) \geq \frac{1}{1-\gamma} \sigma'(p)$, where $\delta(p)$ is the Euclidean distance from p to $\partial\mathcal{O}$.

¹ $\mathcal{D}_{|\mathcal{O}}(\mathcal{P})$ is compact because it is a finite union of finite tetrahedra.

²Notice that it is also the length of the smallest Delaunay edge created when p is inserted.

Proof. We prove the lemma by induction. Initially, we have $\mathcal{P} = \mathcal{P}_i$, and every point of \mathcal{P}_i verifies C1, since \mathcal{P}_i is a $\frac{1}{3}$ -sparse 0.09-sample. Moreover, the points of \mathcal{P}_i belong to $\partial\mathcal{O}$, thus C2 is also verified. Let us now assume that C1 and C2 are verified by every point of \mathcal{P} , up to a certain step where point c is inserted in \mathcal{P} . We will prove that c also verifies C1 and C2.

- If Rule R1 is being applied, then c is the center of a surface Delaunay ball of \mathcal{P} whose bounding sphere passes through a point $p \in \mathcal{P} \setminus \mathcal{P}_{|\partial\mathcal{O}}$. The insertion radius of c is the radius of the surface Delaunay ball, *i.e.* $r(c) = d(c, p)$. Moreover, $d(c, p)$ is at least the distance $\delta(p)$ from p to $\partial\mathcal{O}$, which is at least $\frac{1}{1-\gamma} \sigma'(p)$ by C2. Since σ' is γ -Lipschitz, we have $\sigma'(p) \geq (\sigma'(c) - \gamma d(c, p))$, hence:

$$d(c, p) \geq \frac{1}{1-\gamma} (\sigma'(c) - \gamma d(c, p)) \Rightarrow r(c) = d(c, p) \geq \sigma'(c)$$

It follows that C1 is verified for c . Moreover, C2 is also verified, since c belongs to $\partial\mathcal{O}$.

- If Rule R2 is applied, then c is the center of a surface Delaunay ball of radius greater than $\alpha \sigma(c) \geq \sigma'(c)$, thus the insertion radius of c is at least $\sigma'(c)$, which satisfies C1. Moreover, C2 is satisfied since c belongs to $\partial\mathcal{O}$.

- If Rule R3.1 is applied, then c is the center of a tetrahedron t , and the insertion radius $r(c)$ is the circumradius r of t . According to Rule R3.1, r is either greater than $\sigma(c)$ or greater than $B l_{min}$, where l_{min} is the length of the shortest edge of t . In the first case, we have $r > \sigma(c) > \alpha \sigma(c) \geq \sigma'(c)$, since $\alpha < 1$. In the second case, we have $r > B l_{min}$. Among the vertices of the shortest edge of t , let p be the one inserted last. We have $r(p) \leq l_{min}$, thus $r > B r(p)$. Moreover, by C1, we have $r(p) \geq \sigma'(p)$. Hence, $r \geq B \sigma'(p)$. Since σ' is γ -Lipschitz, $B \sigma'(p)$ is at least $B (\sigma'(c) - \gamma d(c, p)) \geq B (\sigma'(c) - \gamma r)$. It follows that $r \geq \frac{B}{1+B\gamma} \sigma'(c)$, which means that C1 is verified for c if B satisfies:

$$B \geq \frac{1}{1-\gamma} \tag{1}$$

To check C2, we notice that, in both cases ($r > \sigma(c)$ and $r > B l_{min}$), $r(c)$ is bounded from below by $\frac{B}{1+B\gamma} \sigma'(c)$. Let q be the point of $\partial\mathcal{O}$ that is closest³ to c . We have $\delta(c) = d(c, q) \geq d(c, \mathcal{P}_{|\partial\mathcal{O}}) - d(q, \mathcal{P}_{|\partial\mathcal{O}})$, where $d(c, \mathcal{P}_{|\partial\mathcal{O}}) \geq r(c) \geq \frac{B}{1+B\gamma} \sigma'(c)$.

Since Rule R3 is applied only when R1 and R2 are fulfilled, Theorem 4.7 holds right before c is inserted. Hence, the surface Delaunay balls of \mathcal{P} cover $\partial\mathcal{O}$, and q belongs to a surface Delaunay ball B'' , of center c'' and radius r'' . Let $p \in \mathcal{P}$ be a vertex of the facet of $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P})$ circumscribed by B'' . $d(q, p)$ is at most $2r''$ because p and q both belong to B'' . Due to Rule R2, r'' is at most $\alpha \sigma(c'')$, which is at most $\alpha(\sigma(q) + r'')$ since σ is 1-Lipschitz. It follows

³if there are more than one such points, then choose any of them

that $d(q, \mathcal{P}_{|\partial\mathcal{O}}) \leq d(q, p) \leq \frac{2\alpha}{1-\alpha} \sigma(q)$, which is less than $3\alpha \sigma(q)$ because $\alpha < \frac{1}{3}$. Moreover, since $\mathcal{P}_i \subseteq \mathcal{P}_{|\partial\mathcal{O}}$ is a 0.09-sample of $\partial\mathcal{O}$, $d(q, \mathcal{P}_{|\partial\mathcal{O}})$ is also bounded by $0.09 d_M(q) = 3(0.03 \sigma_0(q))$. Thus, $d(q, \mathcal{P}_{|\partial\mathcal{O}}) \leq 3 \sigma'(q)$.

Hence, $\delta(c) = d(c, q) \geq \frac{B}{1+B\gamma} \sigma'(c) - 3 \sigma'(q)$. Since σ' is γ -Lipschitz, we have $\sigma'(q) \leq \sigma'(c) + \gamma d(c, q)$, thus $\delta(c) = d(c, q) \geq \frac{1}{1+3\gamma} \left(\frac{B}{1+B\gamma} - 3 \right) \sigma'(c)$. It follows that C2 is verified for c if B satisfies:

$$\frac{1}{1+3\gamma} \left(\frac{B}{1+B\gamma} - 3 \right) \geq \frac{1}{1-\gamma}, \quad i.e. \quad B \geq \frac{4}{1-5\gamma} \quad (2)$$

• If Rule R3.2 is applied, then c is the center of a surface Delaunay ball B , of radius $r = r(c)$, containing the circumcenter c' of a tetrahedron t' of circumradius $r' \geq \frac{B}{1+B\gamma} \sigma'(c')$ (see case R3.1). Since σ' is γ -Lipschitz, we have $\sigma'(c') \geq \sigma'(c) - \gamma r(c)$. Moreover, the circumsphere of t' is empty, thus $r' \leq d(c', p)$, for any point p of \mathcal{P} lying on the bounding sphere of B . Since B contains both p and c' , $d(c', p)$ is at most $2r(c)$. Hence,

$$2r(c) \geq d(c', p) \geq r' \geq \frac{B}{1+B\gamma} (\sigma'(c) - \gamma r(c)), \quad i.e. \quad r(c) \geq \frac{B}{2+3B\gamma} \sigma'(c)$$

Therefore, C1 is verified for c if B satisfies:

$$B \geq \frac{2}{1-3\gamma} \quad (3)$$

Moreover, C2 is verified because $c \in \partial\mathcal{O}$.

To conclude, Conditions C1 and C2 are verified if B and γ satisfy Eqs. (1), (2) and (3), which is granted if we choose $\gamma < \frac{1}{5}$ (and hence $\alpha < \frac{1}{5}$) and $B \geq \frac{4}{1-5\gamma}$. \square \square

From now on, we assume that $\alpha < \frac{1}{5}$ and that $B \geq \frac{4}{1-5\gamma}$, where $\gamma = \max\{\alpha, 0.03\}$. Given $p \in \mathcal{P}$, we define $B(p)$ as the open ball centered at p , of radius $\varrho(p) = \frac{1}{2(1+\gamma)} \sigma'(p)$.

Lemma 5.3. *The balls $(B(p))_{p \in \mathcal{P}}$ are pairwise disjoint.*

Proof. Given two points p, q of \mathcal{P} , we assume without loss of generality that q was inserted in \mathcal{P} before p . The distance from p to q is then at least $r(p)$. By Lemma 5.2 (Condition C1), we have $r(p) \geq \sigma'(p)$, which is at least $\frac{1}{1+\gamma} \sigma'(q)$ since σ' is γ -Lipschitz. Thus, $d(p, q) \geq \frac{1}{1+\gamma} \max\{\sigma'(p), \sigma'(q)\}$. It follows that $\frac{1}{2(1+\gamma)} \sigma'(p) + \frac{1}{2(1+\gamma)} \sigma'(q) \leq d(p, q)$, which means that $B(p)$ and $B(q)$ are disjoint. \square \square

To compute an upper bound on the size of the output point sample, we need another result, which states that every ball $B(p)$ lies partly inside \mathcal{O} , and that the volume of the part included in \mathcal{O} can be lower-bounded.

Lemma 5.4. *For any $p \in \mathcal{P}$, $B(p) \cap \mathcal{O}$ contains a ball of radius $\frac{1}{2} \varrho(p)$.*

Proof. We distinguish between two cases:

- If p lies inside \mathcal{O} , then, according to Lemma 5.2 (Condition C2), the distance $\delta(p)$ from p to $\partial\mathcal{O}$ is at least $\frac{1}{1-\gamma} \sigma'(p)$, which is greater than $\frac{1}{2} \varrho(p)$. Hence, the ball centered at p , of radius $\frac{1}{2} \varrho(p)$, is included in \mathcal{O} .
- Otherwise, p lies on $\partial\mathcal{O}$. There are two medial balls B_i and B_o tangent to $\partial\mathcal{O}$ at p . One of them (say B_i) is included in \mathcal{O} , whereas the other one is included in $\mathbb{R}^3 \setminus \mathcal{O}$. Since B_i is a medial ball, its radius is at least $d_M(p) > \sigma'(p)$. Moreover, the radius of $B(p)$ is $\varrho(p) < \sigma'(p)$. It follows that the intersection of $B(p)$ with B_i contains a ball of radius $\frac{1}{2} \varrho(p)$. \square \square

Theorem 5.5. *If $\alpha < \frac{1}{5}$ and $B \geq \frac{4}{1-5\gamma}$ (where $\gamma = \min\{\alpha, 0.03\}$), then the output point sample \mathcal{P} verifies:*

$$|\mathcal{P}| = O\left(\iiint_{\mathcal{O}} \frac{dx}{\sigma_0^3(x)} + \frac{1}{\alpha^3} \iiint_{\mathcal{O}} \frac{dx}{\sigma^3(x)}\right)$$

where σ_0 depends only on \mathcal{O} (not on σ).

Proof. We use a standard scheme [BO05]. We will bound the integral of $1/\sigma'^3$ over \mathcal{O} , where σ' is the minimum of $\alpha \sigma$ and of $0.03 \sigma_0$. Since $B(p) \cap \mathcal{O} \subseteq \mathcal{O}$ for any $p \in \mathcal{P}$, we have $\iiint_{\mathcal{O}} \frac{dx}{\sigma'^3(x)} \geq \iiint_{\bigcup_{p \in \mathcal{P}} (B(p) \cap \mathcal{O})} \frac{dx}{\sigma'^3(x)}$. Moreover, the balls $B(p)$ are pairwise disjoint, by Lemma 5.3, thus $\iiint_{\bigcup_{p \in \mathcal{P}} (B(p) \cap \mathcal{O})} \frac{dx}{\sigma'^3(x)} = \sum_{p \in \mathcal{P}} \iiint_{(B(p) \cap \mathcal{O})} \frac{dx}{\sigma'^3(x)}$. In addition, since σ' is γ -Lipschitz, we have:

$$\forall x \in B(p), \sigma'(x) \leq \sigma'(p) + \gamma d(x, p) \leq \sigma'(p) + \gamma \varrho(p) = \left(1 + \frac{\gamma}{2(1+\gamma)}\right) \sigma'(p)$$

It follows that $\sum_{p \in \mathcal{P}} \iiint_{(B(p) \cap \mathcal{O})} \frac{dx}{\sigma'^3(x)} \geq \sum_{p \in \mathcal{P}} \frac{\text{Vol}(B(p) \cap \mathcal{O})}{\left(1 + \frac{\gamma}{2(1+\gamma)}\right)^3 \sigma'^3(p)}$. Now, by Lemma 5.4, the volume of $B(p) \cap \mathcal{O}$ is at least $\frac{4}{3} \pi \frac{1}{64(1+\gamma)^3} \sigma'^3(p)$, which yields:

$$\sum_{p \in \mathcal{P}} \frac{\text{Vol}(B(p) \cap \mathcal{O})}{\left(1 + \frac{\gamma}{2(1+\gamma)}\right)^3 \sigma'^3(p)} \geq \sum_{p \in \mathcal{P}} \frac{\frac{\pi}{48(1+\gamma)^3} \sigma'^3(p)}{\left(1 + \frac{\gamma}{2(1+\gamma)}\right)^3 \sigma'^3(p)} = \frac{\pi}{6(2+3\gamma)^3} |\mathcal{P}|$$

which is at least $\frac{1}{34} |\mathcal{P}|$ since $\gamma < \frac{1}{5}$. Hence, $|\mathcal{P}|$ is at most $34 \iiint_{\mathcal{O}} \frac{dx}{\sigma'^3(x)}$. Now, $\sigma'(x)$ is defined as the minimum of $\alpha \sigma(p)$ and of $0.03 \sigma_0(p)$, which are positive functions. It follows that $\iiint_{\mathcal{O}} \frac{dx}{\sigma'^3(x)}$ is at most $\iiint_{\mathcal{O}} \frac{dx}{\alpha^3 \sigma^3(x)} + \iiint_{\mathcal{O}} \frac{dx}{0.03^3 \sigma_0^3(x)}$, which ends the proof of the theorem. \square \square

Since σ_0 and σ are both positive and continuous over $\bar{\mathcal{O}}$, which is compact, the bound given in Theorem 5.5 is finite. The next result follows, since the algorithm inserts one point in \mathcal{P} per iteration and never removes points from \mathcal{P} .

Corollary 5.6. *The algorithm terminates.*

6 Practicality of the algorithm

6.1 Sizing field

The meshing algorithm presented in the previous sections takes as input a sizing field $\sigma : \bar{\mathcal{O}} \rightarrow \mathbb{R}^+$ which, for the purpose of the analysis in Section 5, is assumed to be 1-Lipschitz. This section explains how to deal with user-defined sizing fields that are not Lipschitz or not defined everywhere in $\bar{\mathcal{O}}$.

Let us assume that the user wants a mesh whose grading conforms to a sizing field σ_u that is not 1-Lipschitz. Then we can use the technique of Miller, Talmor and Teng [MTT99] to derive from σ_u a new sizing field σ'_u that is 1-Lipschitz:

$$\forall x \in \bar{\mathcal{O}}, \sigma'_u(x) = \inf \{d(x, x') + \sigma_u(x') \mid x' \in \bar{\mathcal{O}}\}$$

Notice that $\sigma'_u(p) \leq \sigma_u(p)$, $\forall p \in \bar{\mathcal{O}}$. The field σ'_u is the best 1-Lipschitz approximation of σ_u [ACSYD05], because any 1-Lipschitz function that is pointwise at most σ_u is also pointwise at most σ'_u .

The meshing algorithm can be run using the sizing field σ'_u , however it is not necessary to compute σ'_u inside \mathcal{O} . Indeed, the algorithm requires an evaluation of the sizing field at internal points only in Rule R3, in order to trigger the refinement of a tetrahedron. The refinement of a tetrahedron t is triggered either for a size reason (*i.e.* $r \geq \sigma(c)$, where c and r are respectively the circumcenter and the circumradius of t) or for a shape reason (*i.e.* the radius-edge ratio of t is greater than B). A careful look at the proof of termination shows that a 1-Lipschitz lower bound on the circumradius of tetrahedra refined for size reason is sufficient for the proof. Then, since $\sigma'_u(p) \leq \sigma_u(p)$ for any $p \in \mathcal{P}$, the proofs still hold if rule R3 is triggered by the condition $r \geq \sigma_u(p)$. Besides saving some sizing field evaluations, this variant of the algorithm constructs a sparser mesh whose density conforms to the user-defined sizing field, with a grading bounded only by the bound B on the radius-edge ratio.

In the case where the user has no particular sizing requirements, the 1-Lipschitz sizing field used in the analysis is the field σ_0 introduced at the beginning of Section 5. Here again, the algorithm does not need to evaluate σ_0 inside \mathcal{O} . It may simply skip the size test for tetrahedra and consider for refinement only the badly-shaped tetrahedra (*i.e.* those with a radius-edge ratio greater than B). Since the occasions of refining tetrahedra are fewer in this variant than in the original version of the algorithm, it is clear that this variant also terminates. Its output is a mesh whose sizing is εd_M on $\partial\mathcal{O}$ and grows as fast as possible (regarding the bound on the radius-edge ratio) when moving towards the medial axis.

In any case, the algorithm needs to compute $d_M(x)$ at some points on $\partial\mathcal{O}$ in order to test whether the precondition of Rule R2 is met. This point has been addressed in [BO05], where the authors propose an approximation of $d_M(x)$ based on the notion of λ -medial axis introduced by [CL04].

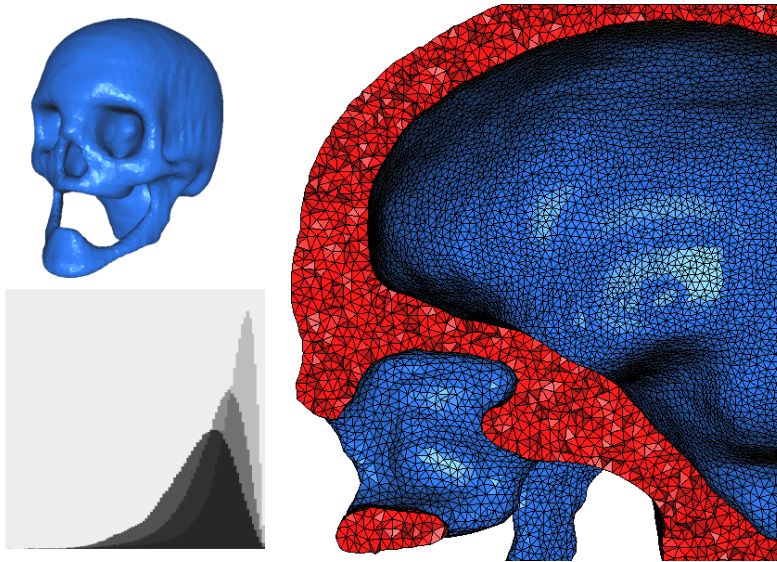


Figure 1: Skull model: 89,245 vertices and 442,542 tetrahedra.

6.2 Sliver removal

Optimizing radius-edge ratios prevents our meshes from containing any bad tetrahedra, except possibly slivers. Following the definition of Edelsbrunner and Guoy [EG02], we say that a tetrahedron t is bad if the ratio between the radius of its inscribed sphere and its circumradius is less than 0.15, which corresponds to a minimal dihedral angle less than 5 degrees. This ratio is called the *radius-radius ratio* of t . In order to remove slivers from our meshes, we use the pumping algorithm of [CDE⁺00] as a post-process. This algorithm consists in assigning carefully chosen weights to the vertices of the mesh, so that their weighted Delaunay triangulation contains as few slivers as possible. Although the guaranteed theoretical bound on radius-radius ratios is known to be miserably low [CDE⁺00], the method is efficient in practice and generates almost sliver-free meshes [EG02].

7 Implementation and results

The algorithm has been implemented in C++, using the geometric library CGAL [CGAL] which provided us with an efficient and flexible implementation of the three-dimensional Delaunay triangulation.

Figures 1 and 2 show two meshes generated by our algorithm coupled with the post-processing step described in Section 6.2. Each figure is composed of two views of the output mesh: one shows the boundary (top left), the other

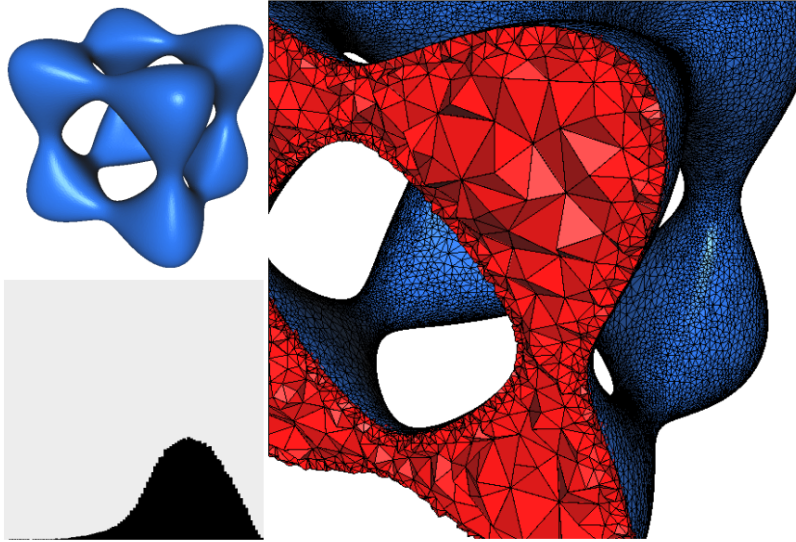


Figure 2: Tanglecube model: 57,293 vertices and 226,010 tetrahedra.

shows a zoom on the interior, cut by a plane⁴ (right). The bottom-left corner of each figure shows the distribution of the radius-radius ratios of the tetrahedra, represented on a linear scale ranging from 0 to $\frac{1}{3}$ (which corresponds to the radius-radius ratio of a regular tetrahedron). The histograms are normalized with respect to area, so that we can make fair comparisons between meshes of different sizes.

In Figure 1, the boundary of the domain is a level set in a 3D grey-scaled image. Its diameter is about 280 millimeters, and its reach approximately 1 millimeter. Although our theoretical results require strict conditions on σ , α and B , in practice the algorithm works well under weaker conditions. For instance, in this example we used a uniform sizing field of 2 millimeters, with $\alpha = 1$ and $B = 2$, which is far beyond the theoretical limits. Note that the topology of the domain has been captured, and that the boundary has been accurately approximated.

The radius-radius ratios distribution of our algorithm (in medium grey) has been superimposed with those obtained by two other algorithms: the unit edge mesher of [GHS90, GHS91] (in dark grey), which privileges the running time (approximately ten seconds for the skull model on a Pentium IV at 1.7 GHz), and the variational mesher of [ACSYD05] (in light grey), which privileges the quality of the output. These two programs, run with our initial surface mesh $\mathcal{D}_{|\partial\mathcal{O}}(\mathcal{P}_i)$ as input, generated approximately the same number of vertices as our mesher. It turns out that, in practice, our algorithm carries out a good compromise between running time and quality of the output.

⁴The screenshots were obtained using Medit [Med].

In Figure 2, the boundary of the domain is an algebraic surface of degree four and genus five, called *tanglecube*. We used no sizing field inside the domain and $\sigma_0 = 0.09 d_M$ on its boundary, as described in Section 6.1. The bound B on the radius-edge ratios was set to 2, which enforced the grading of the output mesh. Although the overall appearance of the radius-radius ratios distribution is deteriorated due to the non-uniformity of the sizing field, the quality of the output mesh remains quite acceptable.

8 Conclusion

We have introduced a new method for meshing three-dimensional domains bounded by smooth surfaces. This method is a combination of existing work on smooth surface meshing on the one hand, on piecewise linear volume meshing on the other hand. We have given theoretical guarantees on the output of the algorithm, regarding its size, the quality of its elements, and the accuracy of the approximation of the original object. We have also provided experimental evidence that the algorithm works well in practice.

The main advantage of our method is that it samples the object \mathcal{O} and its boundary $\partial\mathcal{O}$ at the same time, which lets the user free to decide which density he wants inside \mathcal{O} but also on $\partial\mathcal{O}$. Moreover, the algorithm takes as input the object itself, which makes it independent from any original discretization of the object's boundary and allows to approximate $\partial\mathcal{O}$ within any desired accuracy. In addition, the required a-priori knowledge of \mathcal{O} is minimal, since the algorithm needs only to know the object through an oracle capable of answering two basic geometric questions.

Notice that our algorithm is also able to mesh domains with smooth constraints. The difference between a constraint and a boundary is that both sides of the constraint have to be meshed, whereas only one side of the boundary has to. It turns out that our proofs hold for constraints as well.

Several possible extensions of the work presented in this paper should be addressed in a near future:

- The bound in Theorem 5.5 depends highly on α , whereas the latter influences the density of the mesh only in the vicinity of $\partial\mathcal{O}$. It would be nice to decompose the bound into two terms: one depending on α and on the integral of $1/\sigma^2$ over $\partial\mathcal{O}$, the other depending on the integral of $1/\sigma^3$ over \mathcal{O} .
- Our theoretical results assume that the input domain has a smooth boundary. However, the method has been tested with some success on domains with singularities. Further work for a systematic handling of singularities is in progress. Another related direction of research is to mesh domains with non-manifold constraints.

References

- [AB99] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete Comput. Geom.*, 22(4):481–504, 1999.
- [ACSYD05] P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun. Variational tetrahedral meshing. In *Proceedings SIGGRAPH*, 2005.
- [BO05] J.-D. Boissonnat and S. Oudot. Provably good sampling and meshing of surfaces. *Graphical Models, special issue of Solid Modeling '04*, 2005. Article in press.
- [BOG02] C. Boivin and C. Ollivier-Gooch. Guaranteed-quality triangular mesh generation for domains with curved boundaries. *International Journal for Numerical Methods in Engineering*, 55(10):1185–1213, 2002.
- [CCY04] D. Cohen-Steiner, E. Colin de Verdière, and M. Yvinec. Conforming delaunay triangulations in 3d. *Computational Geometry: Theory and Applications*, pages 217–233, 2004.
- [CD02] S.-W. Cheng and T. K. Dey. Quality meshing with weighted delaunay refinement. In *SODA '02: Proceedings of the thirteenth annual ACM-SIAM symposium on discrete algorithms*, pages 137–146, 2002.
- [CDE⁺00] S.-W. Cheng, T. K. Dey, H. Edelsbrunner, M. A. Facello, and S.-H. Teng. Silver exudation. *J. ACM*, 47(5):883–904, 2000.
- [CDRR04] S.-W. Cheng, T. K. Dey, E. A. Ramos, and T. Ray. Quality meshing for polyhedra with small angles. In *SCG '04: Proceedings of the twentieth annual symposium on Computational geometry*, pages 290–299. ACM Press, 2004.
- [CGAL] The CGAL Library. Release 3.1 (<http://www.cgal.org>).
- [CL04] F. Chazal and A. Lieutier. Stability and homotopy of a subset of the medial axis. In *SM '04: Proceedings of the 9th ACM symposium on Solid modeling and applications*, 2004.
- [CP03] S.-W. Cheng and S.-H. Poon. Graded conforming delaunay tetrahedralization with bounded radius-edge ratio. In *SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 295–304. Society for Industrial and Applied Mathematics, 2003.
- [DOS99] S. Dey, R. O’Bara, and M. S. Shepard. Curvilinear mesh generation in 3d. In *Proc. 8th Internat. Meshing Roundtable*, pages 407–417, 1999.

- [EG02] H. Edelsbrunner and D. Guoy. An experimental study of sliver exudation. *Engineering With Computers, Special Issue on 'Mesh Generation' (10th IMR 2001)*, 18(3):229–240, 2002.
- [FBG96] P. J. Frey, H. Borouchaki, and P.-L. George. Delaunay tetrahedrization using an advancing-front approach. In *Proc. 5th International Meshing Roundtable*, pages 31–43, 1996.
- [Fed70] H. Federer. *Geometric Measure Theory*. Springer-Verlag, 1970.
- [GHS90] P.-L. George, F. Hecht, and E. Saltel. Fully automatic mesh generator for 3d domains of any shape. *Impact of Computing in Science and Engineering*, 2:187–218, 1990.
- [GHS91] 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.
- [LBG96] P. Laug, H. Bourouchaki, and P.-L. George. Maillage de courbes gouverné par une carte de métriques. Technical Report RR-2818, INRIA Rocquencourt, 1996.
- [LT01] X.-Y. Li and S.-H. Teng. Generating well-shaped delaunay meshed in 3d. In *SODA '01: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 28–37. Society for Industrial and Applied Mathematics, 2001.
- [Mav00] F. Mavriplis. Cfd in aerospace in the new millenium. *Canadian Aeronautics and Space Journal*, 46(4):167–176, 2000.
- [Med] Medit, a scientific visualization tool (<http://www-rocq.inria.fr/gamma/medit/medit.html>).
- [MTT99] G. L. Miller, D. Talmor, and S.-H. Teng. Data generation for geometric algorithms on non uniform distributions. *International Journal of Computational Geometry and Applications*, 9(6):577–599, 1999.
- [Rup95] J. Ruppert. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms*, 18:548–585, 1995.
- [She98] J. R. Shewchuk. Tetrahedral mesh generation by Delaunay refinement. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 86–95, 1998.
- [She02] J. R. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications*, 22:21–74, 2002.

- [SU01] A. Scheffer and A. Ungor. Efficient adaptive meshing of parametric models. In *Proc. 6th ACM Sympos. Solid Modeling and Applications*, pages 59–70, 2001.
- [TW00] S.-H. Teng and C. W. Wong. Unstructured mesh generation: Theory, practice, and perspectives. *Int. J. Comput. Geometry Appl.*, 10(3):227–266, 2000.