Conforming restricted Delaunay mesh generation for piecewise smooth complexes

Darren Engwirda

Abstract

A Frontal-Delaunay refinement algorithm for mesh generation in piecewise smooth complexes is described. Built using a restricted Delaunay framework, this new algorithm combines a number of novel features, including: (i) a consistent, conforming restricted Delaunay representation for domains specified as a (non-manifold) collection of piecewise smooth surface patches and curve constraints, (ii) a ‘protection’ strategy for domains containing 1-dimensional features that meet at sharply acute angles, and (iii) a new class of ‘off-centre’ refinement rules designed to achieve high-quality point-placement along embedded 1-dimensional constraints. Experimental comparisons show that the new method outperforms a classical (statically weighted) restricted Delaunay-refinement technique for a number of three-dimensional benchmark problems.

Keywords: Three-dimensional mesh generation, restricted Delaunay, Delaunay-refinement, Advancing-front, Frontal-Delaunay, Off-centres, Sharp-features

1. Introduction

Three-dimensional mesh generation is a key component in a variety of mathematical modelling and simulation tasks, including problems in computational engineering, numerical modelling, and computer graphics and animation. Given a general volumetric domain, described by a network of curves $\Gamma \subset \mathbb{R}^3$, a collection of surfaces $\Sigma \subset \mathbb{R}^3$ and an enclosed volume $\Omega$, the three-dimensional meshing problem consists of tessellating $\Gamma$, $\Sigma$ and $\Omega$ into a mesh of non-overlapping simplexes (edges, triangles and tetrahedrons), such that all geometrical, topological and user-defined constraints are satisfied. While some input domains can be described in terms of smooth entities, it is typical to deal with objects that are only piecewise-smooth, consisting of locally manifold surface patches that meet at sharp 0- or 1-dimensional constraints. Additionally, input domains can also incorporate so-called ‘free’ curve and/or vertex constraints, consisting of vertex and curve segments unconnected to the bounding surface $\Sigma$. In this study, a new Delaunay-refinement type algorithm is presented to construct ‘provably-good’ meshes for such domains – forming a Delaunay tetrahedralisation $\text{Del}(X)$ that includes a subset of ‘restricted’ edges, triangles and tetrahedrons that provide ‘good’ topological and geometrical approximations to the input curve network $\Gamma$, surface structure $\Sigma$ and enclosed volume $\Omega$. A method to ‘protect’ sharply acute features present in the input geometry is also described. Additionally, the proposed algorithm is cast in a ‘Frontal-Delaunay’ framework, through the development of a class of ‘off-centre’ refinement rules for curve segments. These new techniques seek to extend the restricted Frontal-Delaunay surface- and volume-meshing algorithms presented by the author in [1, 2, 3] – facilitating the generation of very high-quality Delaunay surface- and volume-meshes for piecewise smooth domains.

*Corresponding author. Tel.: +1-212-678-5521
Email address: engwirda@mit.edu (Darren Engwirda)
1.1. Nomenclature

The following work is based on the restricted Delaunay framework. The reader is referred to [4] for formal definitions, discussions and proofs.

- $X$ A set of points in $\mathbb{R}^3$, associated with the tessellation;
- Del$(X)$ The Delaunay triangulation of the points $X$;
- Vor$(X)$ The Voronoi complex associated with the points $X$;
- $\Omega$ The input domain – a bounded volume in $\mathbb{R}^3$ (See Figure 1.2(i));
- $\Sigma$ The bounding surface $\Sigma = \partial \Omega$ (See Figure 1.2(i));
- $\Gamma$ The input curve segments, either embedded in the volume $\Gamma \subseteq \Omega$ or inscribed on the surface $\Sigma \subseteq \Gamma$;
- Del$|_\Gamma$(X) A sub-complex of the Delaunay tessellation Del$(X)$, restricted to the curve network $\Gamma$. Del$|_\Gamma$(X) contains the set of 1-simplexes $e \in$ Del$(X)$ whose dual Voronoi faces $v_f \subseteq$ Vor$(X)$ intersect the curves $\Gamma$ (See Figure 1.2(ii)–1.2(iii));
- Del$|_\Sigma$(X) A sub-complex of the Delaunay tessellation Del$(X)$, restricted to the surface $\Sigma$. Del$|_\Sigma$(X) contains the set of 2-simplexes $f \in$ Del$(X)$ whose dual Voronoi edges $v_e \subseteq$ Vor$(X)$ intersect the surface $\Sigma$ (See Figure 1.2(ii)–1.2(iii));
- Del$|_\Omega$(X) A sub-complex of the Delaunay tessellation Del$(X)$, restricted to the volume $\Omega$. Del$|_\Omega$(X) contains the set of 3-simplexes $\tau \in$ Del$(X)$ whose dual Voronoi vertices (circumcentres) $v_p \in$ Vor$(X)$ lie within the volume $\Omega$ (See Figure 1.2(ii)–1.2(iii));
- $\rho_d(\tau)$ The radius-edge ratio associated with a $d$-simplex $\tau$. Defined as the ratio of the radius of the circumball of $\tau$ to the length of its shortest edge;
- $\epsilon_1(e)$ The surface discretisation error associated with a 1-simplex $e \in$ Del$|_\Gamma$(X). Defined as the length from the centre of SDB$_1(e)$ to the centre of the diametric ball of $e$;
- $\epsilon_2(f)$ The surface discretisation error associated with a 2-simplex $f \in$ Del$|_\Sigma$(X). Defined as the length from the centre of SDB$_2(f)$ to the centre of the diametric ball of $f$;
- SDB$_1(e)$ The surface Delaunay ball $B(c, r)$ associated with a 1-simplex $e \in$ Del$|_\Gamma$(X). Surface balls are centred at intersections between the associated bipolar Voronoi face $v_f \in$ Vor$(X)$ and the curve network $\Gamma$, such that $c = v_f \cap \Gamma$;
- SDB$_2(f)$ The surface Delaunay ball $B(c, r)$ associated with a 2-simplex $f \in$ Del$|_\Sigma$(X). Surface balls are centred at intersections between the associated bipolar Voronoi edge $v_e \in$ Vor$(X)$ and the surface $\Sigma$, such that $c = v_e \cap \Sigma$;
- $h(x)$ The mesh-size function. A function $f(x) : \mathbb{R}^3 \rightarrow \mathbb{R}^+$ defining the target edge length at points $x \in \Omega$;
- $a(f)$ The area-length ratio associated with a given triangle $f$. Defined as $a(f) = A/\|e\|_{\text{rms}}$, where $A$ is the signed area of $f$ and $\|e\|_{\text{rms}}$ is the root-mean-square edge length. The area-length ratio is a consistent scalar measure of triangular element quality;
- $v(\tau)$ The volume-length ratio associated with a given tetrahedron $\tau$. Defined as $v(\tau) = V/\|e\|_{\text{rms}}^3$, where $V$ is the signed volume of $\tau$ and $\|e\|_{\text{rms}}$ is the root-mean-square edge length. The volume-length ratio is a robust measure of tetrahedral element quality;

1.2. Related Work

Three-dimensional mesh generation is a broad and evolving area of research. Many successful algorithms employ Delaunay-based strategies [5, 6, 7, 8, 9, 10, 11, 12, 13, 14], based on the progressive refinement of a coarse initial Delaunay triangulation. Delaunay-refinement schemes are top-down algorithms – based on the incremental refinement of a bounding Delaunay tessellation. At each step of the algorithm, elements
that violate a set of constraints are identified and the worst offending elements are eliminated. Elimination is achieved through the insertion of additional Steiner-vertices located at the so-called refinement-points associated with the elements in question. Delaunay-refinement algorithms have been developed for planar \cite{5, 6, 7}, surface \cite{12, 13} and volumetric domains \cite{9, 15, 16}. The reader is referred to, for instance, \cite{4} for additional information and summary.

This study is focused on use of the so-called restricted Delaunay methodology \cite{17, 11, 14} to provide a framework for the approximation of locally 1-, 2- and 3-dimensional topological features via Delaunay sub-complexes. Such techniques have been the focus of previous work, including, for example \cite{12, 13, 11, 4} and previous studies by the author in \cite{1, 2, 3}, where it has been shown that various geometrical and topological guarantees of fidelity are achieved through a careful sampling of the bounding geometrical inputs. Compared to other approaches, the restricted Delaunay framework incorporates a number of desirable characteristics, chiefly: (i) the ability to sample the curve-, surface- and volumetric-features of the domain in a unified manner, and (ii) the development of ‘geometry-agnostic’ meshing algorithms. These characteristics are useful from both a theoretical and software development standpoint: the use of a unified meshing framework obviates non-trivial difficulties associated with the construction of ‘constrained’ Delaunay complexes that conform to lower dimensional constraints \cite{18, 19}, while the use of a geometry-agnostic formulation facilitates the development of meshing software that caters to a broad class of input geometry types and definitions.

Consistent with previous work by the author \cite{1, 2, 3}, the present study combines the restricted Delaunay framework with a so-called ‘Frontal-Delaunay’ methodology – seeking to achieve very high-quality Delaunay-based mesh generation. Specifically, through the use of an appropriate set of ‘off-centre’ refinement rules, a hybrid meshing algorithm, combining the best features of classical Delaunay-refinement and advancing-front techniques, is targeted. It is expected that this class of algorithm may be of interest to users who place a high premium on mesh quality, including those operating in the areas of computational engineering and numerical simulation.

The present study is organised as follows: An overview of the restricted Delaunay framework is presented in Section 2 including a detailed discussion of the methodology used to recover restricted Delaunay edges, triangles and tetrahedrons. The hierarchical restricted Delaunay-refinement algorithm is presented in Section 3 with a new class of curve-conforming off-centre refinement rules described in Section 4. In Section 5 a technique for the ‘protection’ of sharply acute features is presented, extending the restricted Delaunay-refinement algorithm to domains containing curve segments that subtend arbitrarily small angles. An experimental comparison between a conventional (statically weighted) restricted Delaunay-refinement algorithm and the proposed Frontal-Delaunay schemes is presented in Section 7 contrasting output quality and computational performance.
2. Restricted Delaunay Edges, Triangles & Tetrahedrons

The meshing algorithms presented in this study are based on the so-called restricted Delaunay paradigm – a framework utilising a hierarchy of Delaunay sub-complexes to provide consistent and conforming approximations to embedded geometrical features. In the context of 3-dimensional meshing problems, the bounding Delaunay tessellation Del(X) is a tetrahedral complex – of sufficient size to fully enclose the input domain. Embedded within Del(X) are a set of ‘restricted’ Delaunay sub-complexes: Del |Γ(X), Del |Σ(X) and Del |Ω(X), providing discrete approximations to the curve network Γ, surface structure Σ and enclosed volume Ω, respectively. Specifically, the restricted curve triangulation Del |Γ(X) ⊆ Del(X) contains the subset of 1-simplexes e ∈ Del(X) that provide a ‘good’ piecewise linear approximation to the curve network Γ. Similarly, the restricted surface and volume complexes Del |Σ(X) ⊆ Del(X) and Del |Ω(X) ⊆ Del(X) contain the subsets of 2- and 3-simplexes f ∈ Del(X) and τ ∈ Del(X) that provide ‘good’ piecewise linear approximations to the input surface Σ and volume Ω. A comprehensive overview of these concepts is provided in, for example [4, 12, 13, 14, 11] as well as previous work by the author in [1, 2, 3].

Restricted Delaunay techniques exploit the geometrical and topological duality between the Delaunay tessellation and Voronoi complex – using such considerations to compute membership for the restricted Delaunay sub-complexes Del |Γ(X), Del |Σ(X) and Del |Ω(X). Specifically, Del |Σ(X) contains any 3-simplex τ, associated with an internal Voronoi vertex (vν) ∈ Ω, while Del |Σ(X) contains any 2-simplex f, associated with a Voronoi segment vab ∈ Vor(X) that intersects the surface structure Σ, such that vab ∩ Σ ≠ ∅. These are well-known results. Less widely utilised is a mechanism for identifying the restricted 1-simplexes embedded in a tetrahedral complex. In [20], Rineau and Yvinec present a methodology based on dual Voronoi faces. Specifically, Del |Γ(X) contains any 1-simplex ek associated with a Voronoi face vτ ∈ Vor(X) that intersects the curve network Γ, such that vτ ∩ Γ ≠ ∅. Note that, by definition, the faces of the Voronoi complex vτ are convex polygons, oriented normally to their associated Delaunay edges ek. See Figure 2 for additional information.

Each element in a restricted Delaunay sub-complex is also associated with a circumscribing ball. For tetrahedrons in Del |Ω(X), such balls are unique – equivalent to the collection of circumscribing spheres that pass through the four vertices associated with each tetrahedron. For edges in Del |Γ(X) and triangles in Del |Σ(X), each element is instead associated with a so-called Surface-Delaunay-Ball SDB(f). These balls are the set of circumscribing spheres centred upon intersections of the associated Voronoi dual with the input geometry. In the case of multiple intersections, the corresponding Surface Delaunay Ball of maximum radius is selected. See Figure 2 for additional description. Surface Delaunay Balls also support a discrete measure of geometrical fidelity. Specifically, the Euclidean distance ε from the centre of the diametric ball of a given edge or surface element ek ∈ Del |Γ(X) or fjk ∈ Del |Σ(X) to the centre of its associated Surface Delaunay Ball is a one-sided Hausdorff metric – a measure of the geometrical approximation ‘error’ of the piecewise linear Delaunay mesh.

3. A Restricted Delaunay-refinement Algorithm

An algorithm for the meshing of piecewise smooth complexes embedded in \( \mathbb{R}^3 \) is presented here, as an extension of previous work by the author in [1, 2, 3] and by various other authors, including: Rineau and Yvinec [20], Cheng, Dey and Shewchuk [4], Cheng, Dey and Levine [21], and Oudot, Rineau and Yvinec [22]. This method is related to the CGALMESH algorithm, a ‘classical’ restricted Delaunay-refinement approach available as part of the CGAL package, and summarised by Jamin, Alliez, Yvinec and Boissonnat in [14]. The method presented here differs in its methodology for recovering 1-dimensional constraints. Specifically, in the current work, curve constraints are represented as an (unweighted) conforming restricted Delaunay sub-complex, as outlined in Section 2 and consistent with the techniques described by Rineau and Yvinec in [20]. The CGALMESH algorithm is instead based on a so-called ‘protecting-balls’ strategy [4, 21], in which a static decomposition of the curve network Γ is conducted as an initialisation step, with restricted edge segments protected by a weighted Delaunay tessellation in the subsequent surface and volume refinement iterations. Further comparisons between these two feature recovery strategies are made in Section 7.
As per Jamin et al. [14], the development of restricted Delaunay-refinement algorithms is geometry-agnostic, being independent of the specific definition of the underlying geometry inputs. It is required only that the framework support a set of so-called oracle predicates, used to compute: (i) the intersection of convex polygons (Voronoi faces) with the curve network $\Gamma$ (ii) the intersection of line segments (Voronoi edges) with the surface patches $\Sigma$, and (iii) the intersection of points (Voronoi vertices) with the enclosed volume $\Omega$. The Frontal-Delaunay algorithm presented in subsequent sections, additionally requires the computation of intersections between spheres and oriented disks and the curves $\Gamma$ and surfaces $\Sigma$. While a broad class of so-called re-meshing procedures are supported at the theoretical level, in this study attention is restricted to the development of so-called re-meshing operations, in which the input complexes are specified in terms of discrete polylines and triangulated surfaces $\mathcal{P}$. This restriction is made for convenience only – facilitating the construction of simple oracle predicates. Future work is intended to focus on the development of oracles for more general descriptions, including domains defined by implicit, parametric and analytic functions.

Following Jamin et al. [14], the Delaunay-refinement algorithm takes as input a volumetric domain $\Omega$, described by an enclosing (possibly non-manifold) surface $\Sigma \subseteq \mathbb{R}^3$, a network of curve segments $\Gamma \subseteq \mathbb{R}^3$, an upper bound on the allowable element radius-edge ratio $\bar{\rho}$, a mesh size function $h(x)$ defined at all points spanned by the domain and an upper bound on the allowable surface discretisation error $\bar{\epsilon}(x)$. The algorithm returns a discretisation $\mathcal{T}_{\Gamma}$ of the curve network $\Gamma$, a triangulation $\mathcal{T}_{\Sigma}$ of the surface patches $\Sigma$, and a triangulation $\mathcal{T}_{\Omega}$ of the enclosed volume $\Omega$. Here $\mathcal{T}_{\Gamma}$, $\mathcal{T}_{\Sigma}$ and $\mathcal{T}_{\Omega}$ are restricted Delaunay sub-complexes, such that $\mathcal{T}_{\Gamma} = \text{Del}_{\Gamma}(X)$, $\mathcal{T}_{\Sigma} = \text{Del}_{\Sigma}(X)$ and $\mathcal{T}_{\Omega} = \text{Del}_{\Omega}(X)$. Note that $\text{Del}_{\Gamma}(X)$ is an edge complex, $\text{Del}_{\Sigma}(X)$ is a triangular complex, and $\text{Del}_{\Omega}(X)$ and $\text{Del}(X)$ are tetrahedral complexes. The Delaunay-refinement algorithm is summarised in Algorithm 3.1.

The Delaunay-refinement algorithm is designed to provide a number of geometrical and topological guarantees on the output mesh, specifically: (i) that all elements in the volumetric tessellation $\mathcal{T}_{\Omega}$ satisfy constraints on both the element shape and size, such that $\rho(\tau) \leq \bar{\rho}$, and $h(\tau) \leq h(x_{c\tau})$, (ii) that all elements in the embedded surface triangulation $\mathcal{F} \in \mathcal{T}_{\Sigma}$ are guaranteed to satisfy similar element shape and size constraints, in addition to an upper bound on the allowable surface discretisation error, such that $\epsilon(f) \leq \bar{\epsilon}(x_f)$, (iii) that the surface triangulation $\mathcal{T}_{\Sigma}$ is topologically-consistent, ensuring that $\text{Del}_{\Sigma}(X)$ is uniformly 2-manifold in its interior, and consistent with the input geometry at non-manifold features, (iv) that all elements in the embedded curve triangulation $e \in \mathcal{T}_{\Gamma}$ are guaranteed to satisfy similar element shape, size and surface error constraints, and (v) that the curve discretisation $\mathcal{T}_{\Gamma}$ is also topologically-consistent, ensuring that $\text{Del}_{\Gamma}(X)$ is uniformly 1-manifold in its interior, and is consistent with the input geometry at non-manifold features. Making use of properties of the restricted Delaunay tessellation [17], it is known that the triangulations $\mathcal{T}_{\Gamma}$, $\mathcal{T}_{\Sigma}$ and $\mathcal{T}_{\Omega}$ are good piecewise linear approximations to the input curves $\Gamma$, surfaces $\Sigma$ and volumes $\Omega$, provided that the magnitude of the mesh size function $h(x)$ is sufficiently small. Under such conditions it is known that the triangulations $\mathcal{T}_{\Gamma}$, $\mathcal{T}_{\Sigma}$ and $\mathcal{T}_{\Omega}$ are homeomorphic to the underlying

---

**Figure 2:** A restricted Delaunay 1-simplex $e \in \text{Del}_{\Gamma}(X)$ associated with a curve segment $\Gamma$, showing (i) the Surface Delaunay Ball (SDB) associated with the restricted edge, where $r_1(e)$ denotes the SDB radius and $e_1(e)$ the surface discretisation error, (ii) the intersection of the associated dual Voronoi face $v_f \subseteq \text{Vor}(X)$ and the curve network $\Gamma$, and (iii) the off-centre type refinement rule, showing the placement of a locally size-optimal point $c^{(2)}$ about a frontal vertex $x_1$. 

As per Jamin et al. [14], the development of restricted Delaunay-refinement algorithms is geometry-agnostic, being independent of the specific definition of the underlying geometry inputs. It is required only that the framework support a set of so-called oracle predicates, used to compute: (i) the intersection of convex polygons (Voronoi faces) with the curve network $\Gamma$ (ii) the intersection of line segments (Voronoi edges) with the surface patches $\Sigma$, and (iii) the intersection of points (Voronoi vertices) with the enclosed volume $\Omega$. The Frontal-Delaunay algorithm presented in subsequent sections, additionally requires the computation of intersections between spheres and oriented disks and the curves $\Gamma$ and surfaces $\Sigma$. While a broad class of so-called re-meshing procedures are supported at the theoretical level, in this study attention is restricted to the development of so-called re-meshing operations, in which the input complexes are specified in terms of discrete polylines and triangulated surfaces $\mathcal{P}$. This restriction is made for convenience only – facilitating the construction of simple oracle predicates. Future work is intended to focus on the development of oracles for more general descriptions, including domains defined by implicit, parametric and analytic functions.

Following Jamin et al. [14], the Delaunay-refinement algorithm takes as input a volumetric domain $\Omega$, described by an enclosing (possibly non-manifold) surface $\Sigma \subseteq \mathbb{R}^3$, a network of curve segments $\Gamma \subseteq \mathbb{R}^3$, an upper bound on the allowable element radius-edge ratio $\bar{\rho}$, a mesh size function $h(x)$ defined at all points spanned by the domain and an upper bound on the allowable surface discretisation error $\bar{\epsilon}(x)$. The algorithm returns a discretisation $\mathcal{T}_{\Gamma}$ of the curve network $\Gamma$, a triangulation $\mathcal{T}_{\Sigma}$ of the surface patches $\Sigma$, and a triangulation $\mathcal{T}_{\Omega}$ of the enclosed volume $\Omega$. Here $\mathcal{T}_{\Gamma}$, $\mathcal{T}_{\Sigma}$ and $\mathcal{T}_{\Omega}$ are restricted Delaunay sub-complexes, such that $\mathcal{T}_{\Gamma} = \text{Del}_{\Gamma}(X)$, $\mathcal{T}_{\Sigma} = \text{Del}_{\Sigma}(X)$ and $\mathcal{T}_{\Omega} = \text{Del}_{\Omega}(X)$. Note that $\text{Del}_{\Gamma}(X)$ is an edge complex, $\text{Del}_{\Sigma}(X)$ is a triangular complex, and $\text{Del}_{\Omega}(X)$ and $\text{Del}(X)$ are tetrahedral complexes. The Delaunay-refinement algorithm is summarised in Algorithm 3.1.

The Delaunay-refinement algorithm is designed to provide a number of geometrical and topological guarantees on the output mesh, specifically: (i) that all elements in the volumetric tessellation $\mathcal{T}_{\Omega}$ satisfy constraints on both the element shape and size, such that $\rho(\tau) \leq \bar{\rho}$, and $h(\tau) \leq h(x_{c\tau})$, (ii) that all elements in the embedded surface triangulation $\mathcal{F} \in \mathcal{T}_{\Sigma}$ are guaranteed to satisfy similar element shape and size constraints, in addition to an upper bound on the allowable surface discretisation error, such that $\epsilon(f) \leq \bar{\epsilon}(x_f)$, (iii) that the surface triangulation $\mathcal{T}_{\Sigma}$ is topologically-consistent, ensuring that $\text{Del}_{\Sigma}(X)$ is uniformly 2-manifold in its interior, and consistent with the input geometry at non-manifold features, (iv) that all elements in the embedded curve triangulation $e \in \mathcal{T}_{\Gamma}$ are guaranteed to satisfy similar element shape, size and surface error constraints, and (v) that the curve discretisation $\mathcal{T}_{\Gamma}$ is also topologically-consistent, ensuring that $\text{Del}_{\Gamma}(X)$ is uniformly 1-manifold in its interior, and is consistent with the input geometry at non-manifold features. Making use of properties of the restricted Delaunay tessellation [17], it is known that the triangulations $\mathcal{T}_{\Gamma}$, $\mathcal{T}_{\Sigma}$ and $\mathcal{T}_{\Omega}$ are good piecewise linear approximations to the input curves $\Gamma$, surfaces $\Sigma$ and volumes $\Omega$, provided that the magnitude of the mesh size function $h(x)$ is sufficiently small. Under such conditions it is known that the triangulations $\mathcal{T}_{\Gamma}$, $\mathcal{T}_{\Sigma}$ and $\mathcal{T}_{\Omega}$ are homeomorphic to the underlying
Algorithm 3.1 Three-dimensional Restricted Delaunay-refinement

1: function DELAUNAYMESH(Γ, Σ, Ω, \(\bar{\epsilon}(x)\), \(\hat{h}(x)\), \(T|_\Gamma\), \(T|_\Sigma\), \(T|_\Omega\))

2: Form an initial pointwise sampling \(X\) such that \(X\) is well-distributed on \(\Gamma\) and \(\Sigma\). Compute the Delaunay tessellation \(\text{Del}(X)\) and the restricted curve, surface and volume tessellations \(\text{Del}|_\Gamma(X)\), \(\text{Del}|_\Sigma(X)\) and \(\text{Del}|_\Omega(X)\).

3: If some 1-simplex \(e \in \text{Del}|_\Gamma(X)\) violates BADSIMPLEX1(e), form the Steiner point \(c_e\) associated with \(e\), insert \(c_e\) into \(X\), update \(\text{Del}(X)\) and the restricted tessellations \(\text{Del}|_\Gamma(X)\), \(\text{Del}|_\Sigma(X)\) and \(\text{Del}|_\Omega(X)\) and go to step 3.

4: For all vertices \(p \in \text{Del}|_\Gamma(X)\) compute \(c_p \leftarrow \text{TOPODISK1}(p)\). If \(c_p\) is non-null, insert \(c_p\) into \(X\), update \(\text{Del}(X)\) and the restricted tessellations \(\text{Del}|_\Gamma(X)\), \(\text{Del}|_\Sigma(X)\) and \(\text{Del}|_\Omega(X)\) and go to step 3.

5: If some 2-simplex \(f \in \text{Del}|_\Sigma(X)\) violates BADSIMPLEX2(f), form the Steiner point \(c_f\) associated with \(f\):
   (a) If the point \(c_f\) lies within a surface ball \(B(c_e, r)\) associated with some 1-face \(e \in \text{Del}|_\Gamma(X)\), insert \(c_e\) into \(X\) instead, update \(\text{Del}(X)\) and the restricted tessellations \(\text{Del}|_\Gamma(X)\), \(\text{Del}|_\Sigma(X)\) and \(\text{Del}|_\Omega(X)\) and go to step 3.
   (b) Insert \(c_f\) into \(X\). If \(c_f\) changes the topology of \(\text{Del}|_\Gamma(X)\), find the largest adjacent surface ball \(B(c_e, r)\), delete \(c_f\) from \(X\) and insert \(c_e\) into \(X\), update \(\text{Del}(X)\) and the restricted tessellations \(\text{Del}|_\Gamma(X)\), \(\text{Del}|_\Sigma(X)\) and \(\text{Del}|_\Omega(X)\) and go to step 3.
   (c) Go to step 5.

6: For all vertices \(p \in \text{Del}|_\Sigma(X)\) compute \(c_p \leftarrow \text{TOPODISK2}(p)\). If \(c_p\) is non-null, insert \(c_p\) into \(X\), update \(\text{Del}(X)\) and the restricted tessellations \(\text{Del}|_\Gamma(X)\), \(\text{Del}|_\Sigma(X)\) and \(\text{Del}|_\Omega(X)\) and go to step 3.

7: If some 3-simplex \(\tau \in \text{Del}|_\Omega(X)\) violates BADSIMPLEX3(\(f\tau\)), form the Steiner point \(c_\tau\) associated with \(\tau\):
   (a) If the point \(c_\tau\) lies within a surface ball \(B(c_e, r)\) associated with some 1-face \(e \in \text{Del}|_\Gamma(X)\), insert \(c_e\) into \(X\) instead, update \(\text{Del}(X)\) and the restricted tessellations \(\text{Del}|_\Gamma(X)\), \(\text{Del}|_\Sigma(X)\) and \(\text{Del}|_\Omega(X)\) and go to step 3.
   (b) If the point \(c_\tau\) lies within a surface ball \(B(c_f, r)\) associated with some 2-face \(f \in \text{Del}|_\Sigma(X)\), insert \(c_f\) into \(X\) instead, update \(\text{Del}(X)\) and the restricted tessellations \(\text{Del}|_\Gamma(X)\), \(\text{Del}|_\Sigma(X)\) and \(\text{Del}|_\Omega(X)\) and go to step 3.
   (c) Insert \(c_\tau\) into \(X\). If \(c_\tau\) changes the topology of \(\text{Del}|_\Gamma(X)\), find the largest adjacent surface ball \(B(c_e, r)\), delete \(c_\tau\) from \(X\) and insert \(c_e\) into \(X\), update \(\text{Del}(X)\) and the restricted tessellations \(\text{Del}|_\Gamma(X)\), \(\text{Del}|_\Sigma(X)\) and \(\text{Del}|_\Omega(X)\) and go to step 3.
   (d) Insert \(c_\tau\) into \(X\). If \(c_\tau\) changes the topology of \(\text{Del}|_\Sigma(X)\), find the largest adjacent surface ball \(B(c_f, r)\), delete \(c_\tau\) from \(X\) and insert \(c_f\) into \(X\), update \(\text{Del}(X)\) and the restricted tessellations \(\text{Del}|_\Gamma(X)\), \(\text{Del}|_\Sigma(X)\) and \(\text{Del}|_\Omega(X)\) and go to step 3.
   (e) Go to step 7.

8: Return the final restricted Delaunay curve, surface and volume tessellations \(\text{Del}|_\Gamma(X)\), \(\text{Del}|_\Sigma(X)\) and \(\text{Del}|_\Omega(X)\).

9: end function

curve, surface and volume definitions \(\Gamma\), \(\Sigma\) and \(\Omega\), and that the geometrical properties of \(\mathcal{T}|_\Gamma\), \(\mathcal{T}|_\Sigma\) and \(\mathcal{T}|_\Omega\) converge to the true normals, curvatures, lengths, areas and volumes associated with the input geometry as \(\hat{h}(x) \to 0\).

The Delaunay-refinement algorithm begins by pre-processing the input geometry – seeking to identify any sharp ‘features’ inscribed on the input curve and surface collections. These 0- and 1-dimensional features can be induced by both geometrical and topological constraints, including: (i) features that form sharp ‘creases’ or ‘corners’ in \(\Gamma\) and/or \(\Sigma\), and (ii) features at the apex of non-manifold topological connections.
After pre-processing, an initial point-wise sampling of the input curve and surface segments $\Gamma$ and $\Sigma$ is created. Exploiting the discrete representations available, the initial sampling is obtained in this study as a well-distributed subset of the existing vertices $Y \in \mathcal{P}$, where $\mathcal{P}$ is the polyhedral representation of the curve and surface segments $\Gamma$ and $\Sigma$. In the next step, the initial triangulation objects are formed. In this work, the full-dimensional Delaunay tessellation, $\text{Del}(X)$, is built using an incremental Delaunay triangulation algorithm, based on the Bowyer-Watson technique \cite{23}. The restricted curve, surface and volumetric triangulations, $\text{Del}_{|\Gamma}(X)$, $\text{Del}_{|\Sigma}(X)$ and $\text{Del}_{|\Omega}(X)$, are derived from the topology of $\text{Del}(X)$ by explicitly testing for intersections between the faces of the associated Voronoi complex $\text{Vor}(X)$ and the input curve and surface segments $\Gamma$ and $\Sigma$. These queries are computed efficiently by storing the polyhedral geometry $\mathcal{P}$ in an AABB-tree \cite{23}. The main loop of the algorithm proceeds to incrementally refine any restricted 1-, 2- or 3-simplexes found to be in violation of one or more geometrical or topological constraints. Specifically, in step 3, any 1-simplex $e \in \text{Del}_{|\tau}(X)$ found to violate a set of local topological, mesh-size or surface-error constraints is refined – through the introduction of a new Steiner point $c_e$, located at the centre of its surface ball $B(c_e, r)$. In step 4, the topological consistency of the restricted curve tessellation is enforced, ensuring that the set of 1-simplexes $E_p \in \text{Del}_{|\tau}(X)$ adjacent to each vertex $p \in \text{Del}_{|\tau}(X)$ forms a locally 1-manifold feature, known as a topological 1-disk. Vertices adjacent to non-manifold connections trigger additional refinement operations, with the centres $c_e$ of the largest adjacent surface balls $B(c_e, r)$ with the segments $e \in E_p$ inserted as new Steiner vertices until local topological consistency is recovered. Specifically, a cascade of new Steiner vertices are inserted until the topology of $\text{Del}_{|\tau}(X)$, sampled at all points $p \in \text{Del}_{|\tau}(X)$, is equivalent to that of the input curve complex $\Gamma$.

Following the initial refinement of the curve tessellation, the elements of the surface triangulation $\text{Del}_{|\Sigma}(X)$ are refined. In step 5, any 2-simplex $f \in \text{Del}_{|\Sigma}(X)$ found to violate a set of radius-edge, mesh-size or surface-error constraints is conditionally refined – through the introduction of a new Steiner vertex $c_f$, located at the centre of its surface ball $B(c_f, r)$. The insertion of $c_f$ is dependent on several additional constraints, designed to preserve the consistency of the lower dimensional tessellation $\text{Del}_{|\tau}(X)$. In step 5a, if $c_f$ is found to lie within the surface ball $B(c_e, r)$ of an existing curve facet, that facet is instead refined, through the insertion of a Steiner vertex located at the centre of its surface ball $c_e$. This process can be seen simply as an extension of the standard edge-encroachment scheme used in Ruppert’s two-dimensional refinement algorithm. In step 5b, if the insertion of $c_f$ is found to modify the restricted triangulation $\text{Del}_{|\tau}(X)$, the insertion is deferred onto an adjacent curve facet. Specifically, the point $c_f$ is deleted from $\text{Del}(X)$ and a new Steiner vertex $c_e$, corresponding to the centre of the largest adjacent surface ball, is inserted instead. This process ensures that elements in the curve tessellation $\text{Del}_{|\tau}(X)$ are preserved by subsequent surface refinement operations. In step 6, the topological consistency of the restricted surface triangulation is enforced, ensuring that the set of 2-simplexes $F_p \in \text{Del}_{|\Sigma}(X)$ adjacent to each vertex $p \in \text{Del}_{|\Sigma}(X)$ forms a locally 2-manifold feature, known as a topological 2-disk. Vertices adjacent to non-manifold connections trigger additional refinement operations, with the centres $c_f$ of the largest adjacent surface balls $B(c_f, r)$ associated with the segments $f \in F_p$ inserted as new Steiner vertices until local topological consistency is recovered. Specifically, a cascade of new Steiner vertices are inserted until the topology of $\text{Del}_{|\Sigma}(X)$, sampled at all points $p \in \text{Del}_{|\Sigma}(X)$, is equivalent to that of the input surface $\Sigma$.

Following refinement of the curve and surface tessellations, the elements of the volume triangulation $\text{Del}_{|\Omega}(X)$ are refined. In step 7, any 3-simplex $\tau \in \text{Del}_{|\Omega}(X)$ found to violate a set of radius-edge or mesh-size constraints is conditionally refined – through the introduction of a new Steiner vertex $c_\tau$ located at its circumcentre. The insertion of $c_\tau$ is dependent on several additional constraints, designed to preserve the consistency of the lower dimensional tessellations $\text{Del}_{|\tau}(X)$ and $\text{Del}_{|\Sigma}(X)$. In steps 7a and 7b, if $c_\tau$ is found to lie within the surface ball $B(c_e, r)$ or $B(c_f, r)$ of an existing curve or surface facet, that facet is instead refined, through the insertion of a Steiner vertex located at the centre of its surface ball $c_e$ or $c_f$. In steps 7c and 7d, if the insertion of $c_\tau$ is found to modify the restricted triangulations $\text{Del}_{|\tau}(X)$ or $\text{Del}_{|\Sigma}(X)$, the insertion is deferred onto an adjacent curve or surface facet. Specifically, the point $c_\tau$ is deleted from $\text{Del}(X)$ and a new Steiner vertex $c_e$ or $c_f$, corresponding to the centre of the largest adjacent surface ball, is inserted instead. This process ensures that elements in the lower dimensional curve and surface tessellations $\text{Del}_{|\tau}(X)$ and $\text{Del}_{|\Sigma}(X)$ are preserved by subsequent volume refinement operations. The incremental refinement process continues until all radius-edge, mesh-size, surface-error and topolog-
Algorithm 3.2 Topological Disks & Termination Criteria

1: function TopoDisk1\((p)\)
2: \begin{align*}
&\text{Find the set of 1-simplexes } E_p \in \text{Del}_1(X) \\
&\text{adjacent to the vertex } p.
\end{align*}
3: If \(E_p\) is either empty or a valid topological 1-disk, return NULL. Otherwise, find the 1-simplex \(e \in E_p\) that maximises the size of the associated surface Delaunay ball \(B(c_e, r)\) and return \(c_e\).
4: \textbf{end function}

1: function TopoDisk2\((p)\)
2: \begin{align*}
&\text{Find the set of 2-simplexes } F_p \in \text{Del}_2(X) \\
&\text{adjacent to the vertex } p.
\end{align*}
3: If \(F_p\) is either empty or a valid topological 2-disk, return NULL. Otherwise, find the 2-simplex \(f \in F_p\) that maximises the size of the associated surface Delaunay ball \(B(c_f, r)\) and return \(c_f\).
4: \textbf{end function}

4. Feature Conforming Off-centre Steiner Points

Frontal-Delaunay algorithms are a hybridisation of advancing-front and Delaunay-refinement techniques, in which Delaunay triangulations are used to define the topology of a mesh while Steiner vertices are inserted consistent with advancing-front methodologies. In practice, such techniques have been observed to produce very high-quality meshes, inheriting the smooth, semi-structured vertex placement of pure advancing-front methods and the optimal mesh topology and robustness of Delaunay-based approaches. Such techniques have been employed in a number of studies, including, for example \[25\ 26\ 27\ 28\ 29\] and previous work by the author \[1\ 2\ 3\].

4.1. Point-placement Strategy (Curve Refinement)

The ‘off-centre’ point-placement strategy used to refine curve facets is an evolution of the methods previously developed by the author \[1\ 2\ 3\] for surface and volume mesh generation. Two candidate Steiner vertices are considered. Type I vertices, \(c^{(1)}\), are equivalent to conventional element circumcentres (positioned at the centre of associated Surface Delaunay Balls), and are used to preserve convergence in limiting cases. Type II vertices, \(c^{(2)}\), so-called size-optimal points, are designed to satisfy element sizing...
constraints in a locally optimal fashion. Given a refinable 1-simplex \( c \in \text{Del}_\Gamma(X) \), the Type II vertex \( c^{(2)} \) is positioned at an intersection of the curve network \( \Gamma \), and a sphere \( S_\sigma \) of radius \( d_\sigma \), centred on a vertex \( x_1 \in e \). The vertex \( c^{(2)} \) is positioned such that it forms an edge candidate \( \sigma \) about the ‘frontal’ vertex \( x_1 \), such that its size \( h(\sigma) \) satisfies local constraints. Specifically, the length of \( \sigma \) is computed from local mesh-size information, such that

\[
d_\sigma = \frac{1}{2} \left( h(x_1) + h(c^{(2)}) \right)
\]

(1)

For non-uniform \( h(x) \), expressions for the position of the point \( c^{(2)} \) are weakly non-linear, and an iterative procedure is used to obtain an approximate solution. In the case of multiple intersections between the curve network \( \Gamma \) and the sphere \( S_\sigma \), the point \( c^{(2)}_f \) that minimises the angle to the ‘frontal’ vector \( v \) is chosen, where \( v \) is the line joining the frontal vertex \( x_1 \) and the centre of the surface Delaunay ball \( B(c_v, r) \) associated with the edge \( e \). The positioning of size-optimal Type II Steiner vertices for curve facets is illustrated in Figure 2iii.

4.2. Point-placement Strategy (Surface & Volume Refinement)

The off-centre point-placement strategies used to refine surface and volume elements follow a similar methodology to that described previously for curve segments, with two candidate Steiner vertices used to balance local optimality with guaranteed convergence and provably-good behaviour. These methods are presented in detail in [1,2].

4.3. Point-placement Strategy (Off-centre Selection)

Given the Type I and Type II off-centres \( c^{(1)} \) and \( c^{(2)} \) available for curve, surface and volume elements, the positions of the associated refinement points \( c_c, c_f \) and \( c_\tau \) are calculated. These points are selected to satisfy the limiting local constraints, setting

\[
c_c = \begin{cases} 
  c^{(2)}_c, & \text{if } (d^{(2)}_c \leq d^{(1)}_c) \\
  c^{(1)}_c, & \text{otherwise}
\end{cases} \quad \text{and} \quad c_f = \begin{cases} 
  c^{(2)}_f, & \text{if } (d^{(2)}_f \leq d^{(1)}_f) \text{ and } (d^{(2)}_f \geq r_0) \\
  c^{(1)}_f, & \text{otherwise}
\end{cases}
\]

(2)

and

\[
c_\tau = \begin{cases} 
  c^{(2)}_\tau, & \text{if } (d^{(2)}_\tau \leq d^{(1)}_\tau) \text{ and } (d^{(2)}_\tau \geq r_0) \\
  c^{(1)}_\tau, & \text{otherwise}
\end{cases}
\]

(3)

where the \( d^{(i)} = \|c^{(i)} - c_0\| \) are distances from the centre of the frontal facet to the Type I and Type II off-centres, respectively. This cascading selection criteria is designed to ensure that the refinement scheme smoothly degenerates to that of a conventional circumcentre-based Delaunay-refinement strategy in limiting cases, while using locally optimal points where possible. Specifically, these constraints guarantee that the refinement points for curve, surface and volume elements lie within a ‘safe’ region – being positioned on an adjacent sub-face of the Voronoi complex and bound between the circumcentre of the element itself and the diametric ball of the associated frontal facet. See previous work by the author [1,2] for additional details.

4.4. Refinement Order

In addition to the use of off-centre point-placement schemes, the Frontal-Delaunay paradigm also introduces changes to the order in which elements are refined. To better mimic the behaviour of advancing-front type methods, elements are refined only if they are adjacent to existing ‘frontal’ entities. In the case of curve facets \( e \in \text{Del}_\Gamma(X) \), the frontal vertex \( x_i \in e \) must be shared by at least one adjacent facet \( e_j \in \text{Del}_\Gamma(X) \) that is ‘converged’ – satisfying its associated topological, mesh-size and surface-error constraints. In the case of surface triangles \( f \in \text{Del}_\Sigma(X) \), the frontal edge \( e_0 \in f \) must either be a converged edge facet \( e_j \in \text{Del}_\Gamma(X) \) or be shared by an adjacent surface triangle \( f_j \in \text{Del}_\Sigma(X) \) that satisfies its associated
radius-edge and mesh-size, surface-error and topological constraints. In the case of interior tetrahedral elements \( \tau \in \text{Del}\{\Omega}\{X\} \), the frontal face \( f_i \in \tau \) must either be a converged surface facet \( f_j \in \text{Del}\{\Sigma\}\{X\} \) or be shared by an adjacent tetrahedron \( \tau_j \in \text{Del}\{\Omega}\{X\} \) that satisfies its associated radius-edge and mesh-size constraints. The idea of defining ‘frontal’ entities as a dynamic boundary between converged and un-converged elements is a common feature of Frontal-Delaunay algorithms, with similar approaches presented in, for example, \( \cite{26, 27, 28, 29} \).

4.5. Convergence Guarantees & Robustness

The new off-centre point-placement schemes are derived with respect to the fundamental properties associated with the underlying Voronoi diagram. Importantly, by constraining new Steiner vertices to lie along the sub-faces of \( \text{Vor}(X) \), it is guaranteed that the distribution of mesh vertices remains well-separated throughout the refinement process. This behaviour ensures that the algorithm does not create arbitrarily short edges. For the sake of brevity, a full proof of termination or correctness is not included here, but it is important to note that constraints on element radius-edge ratios \( \rho(f) \), element size \( h(x) \), surface discretisation error \( \epsilon(f) \) and topological consistency are satisfied by definition, provided that termination of the algorithm is achieved in practice. The development of a suitable theoretical model for the new Frontal-Delaunay algorithm is the subject of a forthcoming publication.

5. Protecting Sharp Angles Between Curves

The restricted Frontal-Delaunay algorithm presented previously, like all Delaunay-refinement type methods, suffers from issues of non-convergence when the input domain contains sharply acute geometrical features. In such cases, a special-case pre-processing phase is required to identify and to ‘protect’ such features throughout the subsequent refinement process. In this study, a technique to protect sharp angles subtended by curve segments is described. The development of a generalised procedure catering to sharp features between surface patches is deferred for future investigation. Specifically, a process modelled on the ‘protective-collars’ techniques of Rand and Walkington \( \cite{30, 31} \) is adopted, in which a small subset of the domain, adjacent to any sufficiently sharp features, is pre-processed and ‘quarantined’ from subsequent refinement operations.

Given a sharply acute internal angle \( A_{ij} \) formed by any pair of segments \( i, j \) in the curve network \( \Gamma \), the protection process proceeds in a staged fashion: (i) a vertex \( x_A \) is introduced at the apex of the sharp angle \( A_{ij} \), (ii) two new vertices \( x_i, x_j \) are positioned along the incident curve segments \( i, j \in \Gamma \), such that a (well-centred) isosceles triangle is formed. Specifically, the points \( x_i, x_j \) are positioned at the intersection of a ball of radius \( r_A \), centred at \( x_A \), and the curve network \( \Gamma \). Ideally, the radius \( r_A \) should be chosen to reflect the local-feature-size at the apex of the sharp angle \( \text{lfs}(x_A) \). In practice, such a quantity is hard to compute reliably, and the radius \( r_A \) is computed using an iterative procedure in the present work as a result. In this process, the local mesh-size \( h(x_A) \) is used as an initial value for the radius \( r_A \). The radius is then iteratively reduced until: (a) there are exactly two intersections between the ball \( B(x_A, \beta r_A) \) and the curve network \( \Gamma \), and (b) the ball \( B(x_A, \beta r_A) \) is empty of intersections with other balls centred on protected features in \( \Gamma \). Here, the scalar \( \beta \geq 1 \) is a ‘safety-factor’, ensuring that adjacent protecting balls are sufficiently well separated. In this study \( \beta = \frac{3}{2} \) is used.

Noting that such a set of protecting balls is disjoint, and that the candidate vertices describe sets of well-centred isosceles triangles, it is clear that the Delaunay tessellation \( \text{Del}(X) \) contains both the protected edge segments \( x_A, x_i \) and \( x_A, x_j \), in addition to the triangles \( x_A, x_i, x_j \), thus constituting a conforming Delaunay triangulation of the sharp features in \( \Gamma \). Clearly, these protected elements are also automatically included in the restricted sub-complexes \( \text{Del}\{\tau\}\{X\} \) and \( \text{Del}\{\Sigma\}\{X\} \). Subsequent to the initialisation phase, the Delaunay-refinement algorithm must also be modified to ensure that these protected elements are preserved throughout the refinement passes. In this study, a simple topological constraint is enforced: any new Steiner

---

1Simplex elements possessing interior circumcentres.
vertex found to delete a protected edge \( e_A \in \text{Del} \big|_{\Gamma}(X) \) is rejected. In practice, this means that a narrow halo of low quality elements adjacent to sharp features are tolerated. The use of topology-based vertex rejection, as opposed to the standard geometry-based filtering, was found to reduce the size of this halo region.

6. Sliver Suppression

Slivers are a class of low-quality tetrahedral elements that occur in three-dimensional Delaunay tessellations. Consisting of four vertices positioned in a thin ‘kite’-like configuration, sliver elements are typically of very low shape-quality – possessing pathologically small dihedral angles, but relatively small radius-edge ratios. Sliver elements are not guaranteed to be eliminated by standard Delaunay-based refinement schemes, including the Frontal-Delaunay algorithm presented previously. Various strategies designed to remove sliver elements are known to exist, including non-linear optimisation methods based on sliver-exudation \cite{32} and topological-optimisation \cite{33}. In this study, a simple method for the suppression of sliver elements is employed, in which slivers are eliminated through additional refinement. Following \cite{34}, any tetrahedron \( \tau_i \in \text{Del} \big|_{\Omega}(X) \) with a small volume-length ratio \( v(\tau_i) \leq \bar{v} \) is marked for refinement, where \( \bar{v} \) is a user-defined lower-bound on element volume-length ratios. Previous studies \cite{34,3} have shown that this modified refinement algorithm is convergent for \( \bar{v} \leq 1/3 \). Noting that the volume-length ratio is a robust measure of element quality, known to detect all classes of low-quality tetrahedrons, the resulting meshes are of guaranteed quality, with bounded element dihedral angles and aspect ratios. The Frontal-Delaunay algorithm presented previously was modified to impose additional bounds on element volume-length ratios during the tetrahedral refinement phase. Additional details can be found in \cite{2,3}.

7. Experimental Results

The performance of the Frontal-Delaunay algorithm presented in Sections 3, 4 and 5 was investigated experimentally, with the method used to mesh a series of benchmark problems. The algorithm was implemented in C++ and compiled as a 64-bit executable. The Frontal-Delaunay algorithm has been implemented as part of the JIGSAW meshing package, currently available online \cite{35} or by request from the author. The Frontal-Delaunay implementation is referred to as JGSW-FD throughout, with the suffix ‘-FD’ denoting the ‘Frontal-Delaunay’ method. In order to provide additional performance information, the well-known CGALMESH implementation \cite{14} was also used to mesh the set of benchmark problems. The CGALMESH algorithm was sourced from version 4.6 of the CGAL package \cite{36,37} and was compiled as a 64-bit library. The CGALMESH algorithm is referred to as CGAL-DR throughout the following exposition, with the suffix ‘-DR’ denoting ‘Delaunay-refinement’. All tests were completed on a Linux platform using a single core of an Intel i7 processor. Visualisation and post-processing was completed using MATLAB.

7.1. Preliminaries

The JGSW-FD and CGAL-DR algorithms were used to mesh a set of surface- and volume-based benchmark problems (Figure 7), comprising the ISO, FANDISK and BRACKET test-cases. The ISO test-case is an iso-surface problem, consisting of a multiply-connected collection of smooth surface patches with open boundary contours. The FANDISK object is a piecewise smooth surface model of a turbine component, incorporating a network of curve constraints inscribed at connections between adjacent surface patches. The FANDISK problem also includes sharply acute angle constraints, with a minimum angle of 20.1° subtended by segments in the input geometry. The BRACKET object is a discrete CAD representation of a mechanical component, incorporating curve constraints at connections between adjacent surface patches, as per the FANDISK example. The BRACKET problem also incorporates acute angle constraints, with curve segments in the input geometry subtending angles as small as 6.5°.

In all test cases, constant radius-edge ratio thresholds were specified for both surface and volume elements, such that \( \rho_f = 1.25 \) and \( \rho_v = 2 \), corresponding to \( \theta_{\text{min}} \geq 23.5^\circ \) for surface facets. Additionally, uniform mesh-size and surface discretisation constraints were enforced, setting \( h(x) = \alpha \) and \( \epsilon(x) = \beta h(x) \), with \( \alpha \)
Figure 3: Meshes for the ISO, FANDISK and BRACKET test-cases, showing output for the JGSW-FD and CGAL-DR algorithms. Detailed mesh statistics are shown including normalised histograms of elements area-length and volume-length ratios, dihedral-angles and relative edge-lengths. Element counts and total refinement times are also shown.

A scalar length equivalent to approximately 2% of the mean bounding-box dimension associated with each input geometry and $\beta = \frac{1}{4}$.

For all test problems, detailed statistics on element quality are presented, including histograms of element volume-length ratios $v(\tau)$, dihedral-angles $\theta(\tau)$, and relative edge-length $h_r$. The element volume-length ratio is a robust measure of element quality, where high-quality elements attain scores that approach unity. The relative edge-length is defined to be the ratio of edge-length $|e|$ to desired edge-length $h(x_e)$, where $x_e$ is the edge midpoint. Relative edge-lengths close to unity indicate conformance to the mesh-size function. High-quality surface triangles contain angles of $60.0^\circ$, while high-quality tetrahedrons contain angles approaching $70.5^\circ$. Histograms further highlight the minimum, maximum and mean values of the relevant distributions as appropriate.
7.2. A Comparison of JGSW-FD and CGAL-DR

The results in Figure 7 show that, overall, the new JGSW-FD algorithm typically outperforms the CGAL-DR implementation – generating slightly smaller meshes with improved element quality characteristics and mesh-size conformance. Overall computational expense for both algorithms was observed to be similar. In terms of element counts, the new method leads to a reduction of approximately 7%. Focusing on the distributions of element shape-quality, it can be seen that the JGSW-FD algorithm achieves significant improvements in mean area-length and plane-angle distributions in the case of the ISO and FANDISK problems, with smaller improvements in the volume-length and dihedral-angle metrics achieved in the BRACKET test-case. In all cases, it can be seen that a subset of very high-quality elements ($a_f \approx 1$, $v_\tau \approx 1$ and $\theta_f \approx 60^{\circ}$, $\theta_\tau \approx 70^{\circ}$) are generated by the JGSW-FD algorithm. Comparisons of distributions of element relative-length reveal the largest relative differences between algorithms, with the JGSW-FD implementation showing significantly improved conformance to the imposed mesh-size function, with a tight clustering of $h_e$ about 1 observed in all cases. This result is not unexpected – confirming that the new size-optimal off-centre point-placement scheme leads to high-quality vertex distributions that follow the imposed sizing distribution. These results are consistent with those previously obtained for smooth manifold geometries [1, 2] using a simplified version of the three-dimensional Frontal-Delaunay-refinement algorithm presented here. These results demonstrate that, when combined with a suitable off-centre point-placement scheme, the restricted Frontal-Delaunay paradigm can be extended to support piecewise smooth geometric inputs, defined as a collection of curves, surfaces and enclosed volumes.

Acknowledgements. This work was conducted at the Massachusetts Institute of Technology and the University of Sydney with the support of a NASA–MIT cooperative agreement and an Australian Postgraduate Award. The author wishes to thank the anonymous reviewers for their helpful comments and feedback.

References

References

[1] D. Engwirda, D. Ivers, Off-centre Steiner points for Delaunay-refinement on curved surfaces Computer-Aided Design 72 (2016) 157 – 171, 23rd International Meshing Roundtable Special Issue: Advances in Mesh Generation. doi:http://dx.doi.org/10.1016/j.cad.2015.10.007

URL http://www.sciencedirect.com/science/article/pii/S0010448515001608

[2] D. Engwirda, [Voronoi-based Point-placement for Three-dimensional Delaunay-refinement} Procedia Engineering 124 (2015) 330 – 342, 24th International Meshing Roundtable. doi:http://dx.doi.org/10.1016/j.proeng.2015.10.143

URL http://www.sciencedirect.com/science/article/pii/S1877705815032240

[3] D. Engwirda, Locally optimal Delaunay-refinement and optimisation-based mesh generation, Ph.D. thesis (2014).

[4] S. W. Cheng, T. K. Dey, J. R. Shewchuk, Delaunay mesh generation, Taylor & Francis, New York, 2013.

[5] L. P. Chew, Guaranteed-quality Triangular Meshes, Tech. rep., Cornell University, Department of Computer Science, Ithaca, New York (1989).

[6] J. Ruppert, A New and Simple Algorithm for Quality 2-dimensional Mesh Generation, in: Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms, SODA ’93, 1993, pp. 83–92.

[7] A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation, Journal of Algorithms 18 (3) (1995) 548 – 585. doi:http://dx.doi.org/10.1006/jagm.1995.1021

[8] J. R. Shewchuk, Delaunay Refinement Mesh Generation, Ph.D. thesis, Pittsburgh, Pennsylvania (1997).

[9] J. R. Shewchuk, Tetrahedral Mesh Generation by Delaunay Refinement, in: Proceedings of the fourteenth annual symposium on Computational geometry, SCG ’98, ACM, New York, NY, USA, 1998, pp. 86–95. doi:10.1145/276884.276894

[10] S. Cheng, T. Dey, Quality Meshing with Weighted Delaunay Refinement, SIAM Journal on Computing 33 (1) (2003) 69–93. doi:10.1137/00097539703418809

[11] S. W. Cheng, T. K. Dey, E. A. Ramos, Delaunay Refinement for Piecewise Smooth Complexes, Discrete & Computational Geometry 43 (1) (2010) 121–166. doi:10.1007/s00454-008-9109-3

[12] J. D. Boissonnat, S. Oudot, Provably Good Surface Sampling and Approximation, in: ACM International Conference Proceeding Series, Vol. 43, 2003, pp. 9–18.

[13] J. D. Boissonnat, S. Oudot, Provably Good Sampling and Meshing of Surfaces, Graphical Models 67 (5) (2005) 405–451.

[14] C. Jamin, P. Alliez, M. Yvinec, J. D. Boissonnat, CGALmesh: A Generic Framework for Delaunay Mesh Generation, Tech. rep., INRIA (2013).
[15] H. Si, Constrained Delaunay Tetrahedral Mesh Generation and Refinement, Finite Elements in Analysis and Design 46 (12) (2010) 33 – 46, mesh Generation - Applications and Adaptation. doi:http://dx.doi.org/10.1016/j.finel.2009.06.017
URL http://www.sciencedirect.com/science/article/pii/S0168874X09000808

[16] H. Si, TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator, ACM Trans. Math. Softw. 41 (2) (2015) 11:1–11:36. doi:10.1145/2629697
URL http://doi.acm.org/10.1145/2629697

[17] H. Edelsbrunner, N. R. Shah, Triangulating Topological Spaces, International Journal of Computational Geometry & Applications 7 (04) (1997) 365–378.

[18] J. R. Shewchuk, General-dimensional Constrained Delaunay and Constrained Regular Triangulations, I: Combinatorial Properties, Discrete & Computational Geometry 39 (1-3) (2008) 580–637.

[19] H. Si, Adaptive Tetrahedral Mesh Generation by Constrained Delaunay Refinement, International Journal for Numerical Methods in Engineering 75 (7) (2008) 856–880. doi:10.1002/nme.2318
URL http://dx.doi.org/10.1002/nme.2318

[20] L. Rineau, M. Yvinec, Meshing 3D Domains Bounded by Piecewise Smooth Surfaces, in: M. L. Brewer, D. Marcum (Eds.), Proceedings of the 16th International Meshing Roundtable, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 443–460. doi:10.1007/978-3-540-75103-8_25

[21] S. W. Cheng, T. K. Dey, J. A. Levine, A Practical Delaunay Meshing Algorithm for a Large Class of Domains, in: Proceedings of the 16th international meshing roundtable, Springer, 2008, pp. 477–494.

[22] S. Oudot, L. Rineau, M. Yvinec, Meshing Volumes Bounded by Smooth Surfaces, in: Proceedings of the 14th International Meshing Roundtable, Springer, 2005, pp. 203–219.

[23] A. Bowyer, Computing Dirichlet Tessellations, The Computer Journal 24 (2) (1981) 162–166.

[24] P. Alliez, S. Tayeb, C. Wormser, AABB Tree, Tech. rep. (2009).

[25] H. Edelsbrunner, A. Lischinski, and M. Sharir. 2001. Guaranteed -approximate Delaunay triangulation. Algorithmica 28, 2 (2000), 183–198. doi:10.1007/s004540010012

[26] S. Rebay, Efficient Unstructured Mesh Generation by Means of Delaunay Triangulation and Bowyer-Watson Algorithm, Journal of Computational Physics 106 (1) (1993) 125 – 138. doi:http://dx.doi.org/10.1006/jcph.1993.1097
URL http://www.sciencedirect.com/science/article/pii/00219991939010971

[27] D. J. Mavriplis, An Advancing Front Delaunay Triangulation Algorithm Designed for Robustness, Journal of Computational Physics 117 (1) (1995) 90 – 101. doi:http://dx.doi.org/10.1006/jcph.1995.1047
URL http://www.sciencedirect.com/science/article/pii/S00219991959010479

[28] P. J. Frey, H. Borouchaki, P. L. George, 3D Delaunay Mesh Generation Coupled with an Advancing-Front Approach, Computer Methods in Applied Mechanics and Engineering 157 (12) (1998) 115 – 131. doi:http://dx.doi.org/10.1016/S00457825(97)00222-3
URL http://www.sciencedirect.com/science/article/pii/S0045782597002223

[29] J. F. Remacle, F. Henrotte, T. Carrier-Baudouin, E. Béchet, E. Marchandise, C. Geuzaine, T. Mouton, A Frontal-Delaunay Quad-Mesh Generator using the $L^\infty$-norm, International Journal for Numerical Methods in Engineering 94 (5) (2013) 494–512. doi:10.1002/nme.4458
URL http://dx.doi.org/10.1002/nme.4458

[30] A. Rand, N. Walkington, 3D Delaunay refinement of sharp domains without a local feature size oracle, in: Proceedings of the 17th International Meshing Roundtable, Springer, 2008, pp. 37–54.

[31] A. Rand, N. Walkington, Collars and intestines: Practical conforming Delaunay refinement, in: Proceedings of the 18th International Meshing Roundtable, Springer, 2009, pp. 481–497.

[32] S. W. Cheng, T. K. Dey, H. Edelsbrunner, M. A. Facello, S. H. Teng, Silver Exudation, J. ACM 47 (5) (2000) 883–904. doi:10.1145/355483.355487

[33] B. M. Klingner, J. R. Shewchuk, Aggressive Tetrahedral Mesh Improvement, in: M. L. Brewer, D. Marcum (Eds.), Proceedings of the 16th International Meshing Roundtable, Springer Berlin Heidelberg, 2008, pp. 3–23. doi:10.1007/978-3-540-75103-8_25

[34] S. Gosselin, C. Ollivier-Gooch, Tetrahedral Mesh Generation using Delaunay Refinement with Non-standard Quality Measures, International Journal for Numerical Methods in Engineering 87 (8) (2011) 795–820. doi:10.1002/nme.3138
URL http://dx.doi.org/10.1002/nme.3138

[35] D. Engwirda, JIGSAW: An unstructured mesh generation package, http://www.github.com/dengwirda/jigsaw-matlab
(2016).

[36] C. Jamin, S. Pion, M. Teillaud, 3D Triangulations, in: CGAL User and Reference Manual, 4.6.1 Edition, CGAL Editorial Board, 2015.

[37] C. Jamin, S. Pion, M. Teillaud, 3D Triangulation Data Structure, in: CGAL User and Reference Manual, 4.6.1 Edition, CGAL Editorial Board, 2015.