MATHEMATICS OF COMPUTATION
Volume 88, Number 316, March 2019, Pages 751–782
https://doi.org/10.1090/mcom/3383
Article electronically published on October 9, 2018

RED–GREEN REFINEMENT OF SIMPLICIAL MESHES IN d DIMENSIONS

JÖRG GRANDE

Abstract. The local red–green mesh refinement of consistent, simplicial meshes in d dimensions is considered. We give a constructive solution to the green closure problem in arbitrary dimension d. Suppose that $\mathcal{T}$ is a simplicial mesh and that R is an arbitrary subset of its faces, which is refined with the Coxeter–Freudenthal–Kuhn (red) refinement rule. Green refinements of simplices $S \in \mathcal{T}$ are generated to restore the consistency of the mesh using a particular placing triangulation. No new vertices are created in this process. The green refinements are consistent with the red refinement on R, the unrefined mesh regions, and all other neighboring green refinements.

1. Introduction

Meshes and nested hierarchies of meshes are fundamental tools for finite element methods, finite volume methods, multigrid methods, and other methods in applied mathematics [10, 20]. They also have applications in topology [34]. A mesh in $\mathbb{R}^d$, $d \in \mathbb{N}$, is simplicial if all of its elements are simplices (triangles, tetrahedra, pentatopes). It is consistent if the intersection of mesh elements is always a common face of them. A consistent simplicial mesh is called a triangulation. In applications, $d \leq 3$ prevails, but space-time methods [3, 14, 21] require $d = 4$. Below, any $d \in \mathbb{N}$ is permitted.

A local mesh refinement $\mathcal{S}$ is produced from a given mesh $\mathcal{T}$ and a user-specified subset $R \subseteq \mathcal{T}$ of elements which should be replaced by "smaller" elements.[1] Mesh refinement is a special case of mesh generation [4] which operates under the assumption that only "few" elements of $\mathcal{T}$ require refinement such that the generation of a completely new mesh would be wasteful. Two well-known classes of mesh refinement algorithms are the bisection methods and the red–green methods.

The basic idea of bisection methods is to cut a simplex (repeatedly) into two pieces using a (hyper-) plane. Methods for triangles are treated in [26, 27, 31, 32] for tetrahedra, e.g., in [9]. They differ in the criteria for the selection of the cut-planes. Bisection methods are generalized to d-dimensional triangulations in [24, 33, 35].

A red–green refinement algorithm has three components, the red refinement rule, the green refinement rule, and a global component, which coordinates the use of the two refinement rules [5]. The red refinement rule may be applied repeatedly to (the descendants of) a simplex. It is mainly responsible for the stability properties of the full algorithm. The green refinement rule is used to restore the consistency

Received by the editor November 10, 2016, and, in revised form, January 2, 2018.
2010 *Mathematics Subject Classification.* Primary 65M50, 65N50, 65D18.
Key words and phrases. Local mesh refinement, red–green refinement, green closure, simplicial meshes, Kuhn triangulation.

[1]The set R is not denoted by a calligraphic letter because this notation is reserved for meshes.

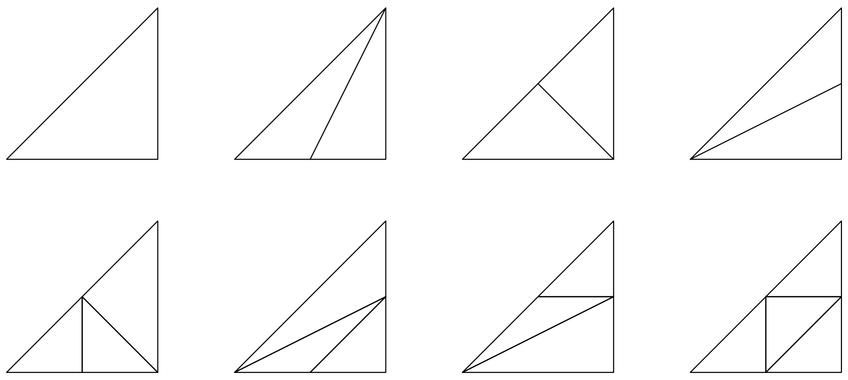

FIGURE 1. The solutions of the Green Closure Problem 1.1 given in Definition 3.11 for $d = 2$.

after the red rule has been applied. For triangles, red–green refinement appears in [1]. Tetrahedral meshes are considered, e.g., in [5] and [2, 18, 23].

The red refinement rule uses the Coxeter–Freudenthal–Kuhn method which is available in d dimensions. It is well known in topology [16, 22], group theory [11], computer graphics [28, 29], and applied mathematics [6, 13]. It partitions a d-dimensional simplex S into 2^d smaller simplices. (This induces a partition of each k-dimensional face of S with the k-dimensional red refinement rule.) The partition is unique provided that an ordering of the vertices of S is fixed.

The green refinement rule is typically generated manually or semi-automatically. This gives rise to the following problem.

Problem 1.1 (Green Closure Problem)**.** Suppose that $\mathcal{T}$ is a triangulation and that $R \subseteq \mathcal{T}$ is the set of all simplices to which the red refinement rule has been applied (including all faces of the user specified simplices). The green closure problem, for any $S \in \mathcal{T} - R$, is to find a green refinement $\mathcal{S}$ of S. This is a triangulation of S with the following properties (where F is an arbitrary face of S):

(1) If $F \in R$, then $\mathcal{S}$ is consistent with the red refinement of F.
(2) If, for all $T \in R$, the intersection $F \cap T$ is empty or contains only vertices of F, then $\mathcal{S}$ does not refine F.
(3) The refinement of F induced by $\mathcal{S}$ depends only on "data" on F. (This condition implies consistency between the refinements of S and $T \in \mathcal{T}$ if F is a common face of S and T.)

The green refinement rule should create as few additional vertices as possible.

For $d = 2$, it is straightforward to solve Problem 1.1 manually and verify the solutions by visual inspection; see Figure 1. Already for $d = 3$, the situation is quite complicated. Depending on which of the six edges of a tetrahedron must be subdivided, there are 2^6 cases to consider. In the software packages UG and DROPS [2, 18], the green refinement rule is available for each refinement pattern. No proof of correctness for these rules seems to exist in the literature. Other authors use an incomplete green refinement rule [1, 5, 7]. If the situation on a simplex is too complicated, the simplex is refined with the red refinement rule. This can lead to an avalanche effect which spreads out the red refinement. This makes the refined

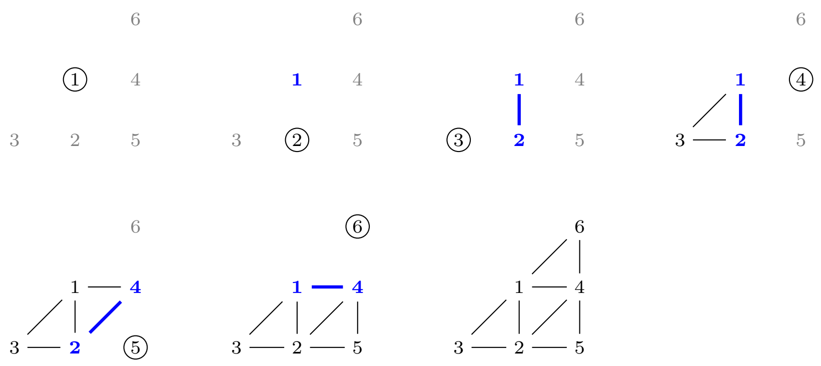

FIGURE 2. Construction of the green refinement with all edge barycenters, $d = 2$. This yields the red refinement. The active node is encircled; visible faces are bold (and blue in the online version).

region of the mesh larger than requested by the user. It also generates additional communication in distributed computer programs.

In this paper, the local red–green mesh refinement of triangulations in $\mathbb{R}^d$ with an arbitrary dimension $d \in \mathbb{N}$ is considered. A constructive solution to the Green Closure Problem 1.1 in dimension d is given. This brings the red–green refinement methods to the same level of generality with respect to the dimension d that was reached for the bisection methods circa twenty years ago [24, 35].

Both bisection methods and red–green methods can produce stable families of refined meshes. In bisection methods, some weak conditions must usually be imposed on the initial mesh to ensure stability. To assess whether a local refinement algorithm is stable, it is applied repeatedly to a simplex and its descendants. The simplices generated by this process are sorted into equivalence classes with respect to the group of scalings and rigid motions. It is desirable to have few equivalence classes. According to Bey [5], the newest vertex bisection from [27] generally produces $d! \cdot 2^{d-2}$ equivalence classes. Bey also shows that the red refinement generates at most $d!/2$ equivalence classes and that this is the minimal number in a certain family of refinement methods.

Results. The main contribution of this paper is a constructive solution of Problem 1.1; cf. Theorem 3.15. As far as the author knows, this yields the first generalization of red–green refinement to triangulations in d dimensions, $d \in \mathbb{N}$. The green refinement rule constructed below introduces *no* additional vertices and handles *every* possible refinement pattern. There is *no* avalanche effect. These optimality properties of the resulting red–green refinement method are stated in Theorem 4.6. A further contribution of this paper is Theorem 5.21, which states that the red refinement of a simplex can be constructed as a placing triangulation. This result is central to proving that the green refinements satisfy property (1) of Problem 1.1. The assertions of Theorems 3.15 and 5.21 are checked in low dimensions with the help of a computer program.

Overview of the construction. The green refinement rule is based on the placing triangulation from convex geometry [12]. A placing triangulation is constructed step by step from a tuple of points w starting with the empty triangulation $\mathcal{T}$. In step i, the "active" point w^i is connected to all faces of $\mathcal{T}$ which are "visible" from w^i; cf. Section 3.2. An example is shown in Figure 2. Software implementations of algorithms to compute the placing triangulation are available in computational geometry packages, e.g., [17, 30]. The placing triangulation has a property that is similar to property (3) in Problem 1.1: The triangulation induced on a boundary face of $\mathcal{T}$ depends only on the sub-tuple of w containing all points in that boundary face.

The ordering of the vertices is important for the result of the placing triangulation. Therefore, one must consider a simplex as a set of points together with an enumeration or ordering of its vertices. This does not impose additional conditions on the initial meshes for a red–green refinement algorithm: To uniquely define the red refinement of a triangulation for $d \geq 3$, an ordering of the vertices is required; cf. Remark 3.4 and Theorem 3.6. Such an enumeration can be constructed on any initial triangulation; cf. Lemma 2.1.

For the green refinement rule, the ordering of the vertices and edge barycenters is constructed such that the placing triangulation of all of these points reproduces the red refinement. This is important for establishing property (1) of Problem 1.1.

Organization of the paper. Section 2 introduces (standard) notation for convex polytopes and triangulations in d dimensions. In Section 3, the red and the green refinement rules are introduced. The focus is on the description of the refinement rules. The main Theorem 3.15 which solves Problem 1.1 is stated, but its proof is deferred to Section 5. In Section 4, some properties of a refinement algorithm with the rules from Section 3 are considered. Theorem 4.6 on the optimality properties is proved. Pseudocode for a red–green refinement algorithm is presented and analyzed in Section 4.2. Section 5 contains the proof of Theorem 3.15. The three properties of Problem 1.1 are treated in separate subsections.

2. Polytopes and complexes

Basic notation of convex geometry is introduced to make the paper broadly accessible. All results on polytopes and complexes stated here are standard in convex geometry; they can be found in the classic monographs [8, 19] or in the more recent books [12, 29].

Let $V \subset \mathbb{R}^d$ be a finite set of points. The linear hull, the affine hull, and the convex hull of V are denoted by $\operatorname{span} V$, $\operatorname{aff} V$, and $\operatorname{conv} V$, respectively. The empty set is convex and its dimension is -1 by convention.

2.1. **Polytopes.**

A bounded convex polytope $P \subset \mathbb{R}^d$ is the convex hull of a finite set of points, $P = \operatorname{conv} V$. As this paper only deals with bounded convex polytopes, they are henceforth simply called polytopes. The set V is a vertex description of P. If the dimension of P is k, one says that P is a k-polytope. For any linear functional $\psi\colon \mathbb{R}^d \to \mathbb{R}$, a face of P is given by the set

$$F = \{x \in P \mid \psi(x) = M(P,\psi)\}, \quad M(P,\psi) = \max\{\psi(x) \mid x \in P\}. \tag{2.1}$$

In particular, P is a face of itself. By convention, $\emptyset$ is a face of P. The (finite) set of all faces is $\mathcal{F}(P)$. Set inclusion defines a partial order on $\mathcal{F}(P)$ which turns it into a lattice, the face lattice. All faces are themselves polytopes. The k-dimensional faces are collected in $\mathcal{F}_k(P)$. The 0- and 1-faces are called vertices and edges; the $(k-1)$-faces of a k-polytope are the facets.

A (closed) half-space is a set $\{x \in \mathbb{R}^d \mid \psi(x) \leq c\}$, where ψ is a linear functional, $c \in \mathbb{R}$. A half-space given by a facet F is obtained by taking ψ from (2.1) and $c = M(P, \psi)$. A fundamental theorem of polytope theory is that P is the intersection of all half-spaces given by its facets. This is the facet representation of P.

An elementary way to extend a polytope by adding a new vertex is the pyramid construction. A d-pyramid is defined as $\operatorname{conv}(\{a\} \cup B)$, where the basis B is a $(d-1)$-polytope and the apex a is a point which does not lie in $\operatorname{aff} B$. The pyramid is written as $\operatorname{pyr}(a, B)$. By definition, there holds $\operatorname{pyr}(a, \emptyset) = \{a\}$.

A d-simplex, $d \geq 0$, is a d-polytope that is the convex hull of exactly $d+1$ points. By convention, the empty set is a simplex. All faces of a simplex are simplices. The face lattice of a d-simplex is isomorphic to the lattice of all subsets of $\{0, \dots, d\}$ ordered by set inclusion. A d-simplex is a pyramid over any of its facets, where the apex is the unique vertex not contained in the facet. For $d \in \{0, \dots, 4\}$, a d-simplex is called a point, a line segment, a triangle, a tetrahedron, and a pentatope.

If A is a mapping $\mathbb{R}^d \to \mathbb{R}^k$, $k \in \mathbb{N}$, the notation $A(F) = \{Ax \mid x \in F\}$ is used, $F \in \mathcal{F}(P)$.

2.2. **Complexes.** A finite set $\mathcal{C}$ of polytopes in $\mathbb{R}^d$ is a complex if the following two assertions hold:

$$\begin{aligned} &\mathcal{F}(S) \subseteq \mathcal{C} && \text{for all } S \in \mathcal{C} && \text{(face-completeness)}, \\ &S \cap T \in \mathcal{F}(S) \cap \mathcal{F}(T) && \text{for all } S, T \in \mathcal{C} && \text{(intersection property)}. \end{aligned} \tag{2.2}$$

In the literature, the more precise term geometric cell complex is used, but this is not necessary below. The members of $\mathcal{C}$ are called faces (or k-faces to specify the dimension). The set of all k-faces is $\mathcal{F}_k(\mathcal{C})$. In the important case that all members of $\mathcal{C}$ are simplices, $\mathcal{C}$ is called simplicial complex or triangulation.

A sub-complex of $\mathcal{C}$ is a subset, which is itself a complex. Let $R \subseteq \mathcal{C}$. The closure of R, written as $\operatorname{cl} R$, is the smallest sub-complex of $\mathcal{C}$, which contains R as a subset. There holds

$$S \in \operatorname{cl} R \quad \Leftrightarrow \quad S \in \mathcal{F}(T) \text{ for some } T \in R. \tag{2.3}$$

The notation $\operatorname{set} \mathcal{C}$ is used for $\cup\{S \mid S \in \mathcal{C}\} \subset \mathbb{R}^d$. The face lattice of a polytope P is a complex. If F is a face of P and $P = \operatorname{set} \mathcal{C}$, the notation $\mathcal{C}|_F$ is used for the restriction of $\mathcal{C}$ to F. This is the sub-complex $\mathcal{C}|_F = \{T \in \mathcal{C} \mid T \subseteq F\}$ of $\mathcal{C}$.

If A is a mapping $\mathbb{R}^d \to \mathbb{R}^k$, $k \in \mathbb{N}$, the notation $A(\mathcal{C})$ is used for $\{A(S) \mid S \in \mathcal{C}\}$. This set is a complex if $\mathcal{C}$ is and if A is affine and invertible on $\operatorname{set} \mathcal{C}$.

2.3. **Consistent enumerations.** An enumeration of a finite set S is a bijective map $e\colon \{0, \dots, |S|-1\} \to S$. It carries the (strict total) ordering of the integers to the elements of S; $e(i)$ is less than $e(j)$ if and only if $i < j$. This is called the ordering given (or induced) by e. Conversely, a strict total ordering of S defines a unique enumeration of S. Let $\mathcal{C}$ be a complex. An enumeration e of $\mathcal{C}$, is a collection of enumerations of the vertices of each member of $\mathcal{C}$,

$$e = (e_S)_{S \in \mathcal{C}}, \quad e_S \text{ is an enumeration of } \mathcal{F}_0(S).$$

Any sub-complex of $\mathcal{D} \subseteq \mathcal{C}$ carries the restricted enumeration $e|_{\mathcal{D}}$ given by all e_S, $S \in \mathcal{D}$. An enumeration of $\mathcal{C}$ is consistent if, for all $S, T \in \mathcal{C}$, the ordering of the common vertices $\mathcal{F}_0(S \cap T)$ given by e_S and given by e_T is the same. More formally, this means that, for all $S, T \in \mathcal{C}$,

$$e_T^{-1} \circ e_S \quad \text{is strictly isotonic on } e_S^{-1}\big(\mathcal{F}_0(S \cap T)\big). \tag{2.4}$$

Here, a map $f\colon A \to B$ between two subsets of the integers is strictly isotonic iff there holds $i < j \Rightarrow f(i) < f(j)$ for any $i, j \in A$. A complex with a consistent enumeration is called consistently numbered and written as a pair $(\mathcal{C}, e)$. A polytope P, particularly a simplex, is called consistently numbered if its face lattice is. A consistently numbered polytope P with enumeration e is written as the pair (P, e).

Already in the work of Freudenthal [16], [6, Chp. 3.1.6], one finds that a (unique) consistent enumeration of $\mathcal{C}$ can be produced from a single enumeration of all vertices of $\mathcal{C}$:

Lemma 2.1. *If $\mathcal{C}$ is a complex and e is an enumeration of $\mathcal{F}_0(\mathcal{C})$, there is a unique consistent enumeration f of $\mathcal{C}$ such that, for all $S \in \mathcal{C}$, the orderings of $\mathcal{F}_0(S)$ given by e and f_S are equal.*

Proof. Let $S \in \mathcal{C}$. As $\mathcal{F}_0(S) \subseteq \mathcal{F}_0(\mathcal{C})$, the ordering given by e restricts to a unique ordering on $\mathcal{F}_0(S)$. There is precisely one isotonic enumeration of $\mathcal{F}_0(S)$ with respect to this ordering, which uniquely defines f_S. Consistency follows from the fact that f_S and f_T, $S, T \in \mathcal{C}$, both give to $\mathcal{F}_0(S \cap T)$ the same ordering as e. ☐

Let $x, y \in \mathbb{R}^d$. The lex-order $x \preceq y$ is the total order on $\mathbb{R}^d$ defined by

$$x \preceq y \quad \Leftrightarrow \quad \big(x_i \le y_i, \quad i = \max(\{j \in \{1, \dots, d\} \mid x_j \ne y_j\} \cup \{1\})\big). \tag{2.5}$$

This is a variant of the standard lexicographic order. The tuples are compared with the last position (instead of the first) as the most significant. An enumeration of a finite set of points can be produced by enumerating the elements in lex-order. This enumeration is denoted as lex or lex_S if the set under consideration is not clear from the context. The consistent enumeration of a complex $\mathcal{C}$ which is defined by using Lemma 2.1 with $\text{lex}_{\mathcal{F}_0(\mathcal{C})}$ is simply denoted by lex.

2.4. **Isotonic affine maps.**

Let (S, e), (T, f) be consistently numbered k-simplices embedded into $\mathbb{R}^d$ ($k \le d$). Any bijection $\mathcal{F}_0(S) \to \mathcal{F}_0(T)$ can be extended uniquely to an invertible, affine map $\operatorname{aff} S \to \operatorname{aff} T$ which maps S onto T. Let A be the affine map which maps $e_S(i) \mapsto f_T(i)$, $i \in \{0, \dots, k\}$.

Lemma 2.2. *The map A is the unique invertible, affine map $A\colon \operatorname{aff} S \to \operatorname{aff} T$ with $A(S) = T$ such that $f_{A(F)} = A \circ e_F$ holds for all $F \in \mathcal{F}(S)$.*

Proof. By construction, there holds $f_T = f_{A(S)} = A \circ e_S$. By Lemma 2.1, the consistent enumeration of $\mathcal{F}(T)$ is uniquely determined by this enumeration of $\mathcal{F}_0(T)$. Hence, the enumerations agree on all faces of T, or equivalently, on all faces of S. ☐

The map A is called the isotonic (or order preserving) affine map $(S, e) \to (T, f)$. Given (S, e) and an invertible, affine map $A\colon \operatorname{aff} S \to \operatorname{aff} T$ with $A(S) = T$, there is precisely one consistent enumeration f of T which makes A isotonic, namely, $f = Ae = (A \circ e_F)_{F \in \mathcal{F}(S)}$. This is the induced enumeration.

3. Refinement rules for simplices

The red and the green refinement rules for a single, consistently numbered simplex (S, e) are introduced in Sections 3.1 and 3.3, respectively. An important component of the green refinement rule is the placing triangulation, which is introduced in Section 3.2.

A d-refinement rule r maps consistently numbered d-simplices (S, e) to pairs $(\mathcal{C}, f) = r(S, e)$, where $\mathcal{C}$ is a triangulation of S (that is, $\operatorname{set} \mathcal{C} = S$) which is consistently numbered by f. Both $\mathcal{C}$ and f may depend on both S and e. The dimension d is usually clear from the context and therefore omitted. One calls S the parent simplex and the members of $\mathcal{C}$ its child simplices.

Suppose that F is a face of S. The restriction $(\mathcal{C}, f)|_F$ is the pair $(\mathcal{C}|_F, f|_F)$, where $f|_F$ is a shorthand notation for the restricted enumeration $f|_{(\mathcal{C}|_F)}$. Suppose that A is an affine map which is invertible on $\operatorname{aff} S$. The notation $A(\mathcal{C}, f)$ is used for the pair $(A(\mathcal{C}), Af)$ containing the mapped complex and the induced enumeration.

3.1. **The red refinement rule.**

Let

$$\hat{v}^0 = 0, \quad \hat{v}^i = \hat{v}^{i-1} + \delta^i, \quad i \in \{1, \dots, d\}, \tag{3.1}$$

where δ^i is the ith column of the $d \times d$ identity matrix. The enumeration of the points $\hat{v}^i$ with increasing index i coincides with the enumeration in lex-order; cf. (2.5). The scaled reference d-simplex

$$\hat{S} = \operatorname{conv}\{2\hat{v}^0, 2\hat{v}^1, \dots, 2\hat{v}^d\} \tag{3.2}$$

is used. The enumeration of its vertices $i \mapsto 2\hat{v}^i$, $i \in \{0, \dots, d\}$, is also given by the lex-order.

A (k, l)-shuffle π is a permutation in $\operatorname{Sym}(k+l)$, the symmetric group on $\{1, \dots, k + l\}$, with the property that its inverse π^{-1} is isotonic on $\{1, \dots, k\}$ and on $\{k+1, \dots, k+l\}$. For $k = 0$ or $l = 0$, the conditions for the empty set appearing in the definition are considered as vacuously true. The set of (k, l)-shuffles is denoted as $\operatorname{Sh}(k, l) \subseteq \operatorname{Sym}(k+l)$. It is related to the set of binomial coefficients; for example, the number of (k, l)-shuffles is $\binom{k+l}{k}$.

Let $\pi \in \operatorname{Sym}(d)$ operate on $x \in \mathbb{R}^d$ by permuting the components, that is, $(\pi x)_i = x_{\pi i}$, $i \in \{1, \dots, d\}$. Let S_π be the d-simplex

$$S_\pi = \operatorname{conv}\left\{\pi \hat{v}^i \;\middle|\; i \in \{0, \dots, d\}\right\}. \tag{3.3}$$

An enumeration of the vertices of S_π is given by $e^\pi \colon i \mapsto \pi \hat{v}^i$, $i \in \{0, \dots, d\}$. Due to Lemma 2.1, this defines a consistent enumeration of the face lattice of S_π.

Definition 3.1. The pair $\operatorname{redref}(\hat{S}, \mathrm{lex}) = \big(\operatorname{redtri}(\hat{S}, \mathrm{lex}), \operatorname{rednum}(\hat{S}, \mathrm{lex})\big)$ is the red refinement of $(\hat{S}, \mathrm{lex})$. The triangulation $\operatorname{redtri}(\hat{S}, \mathrm{lex})$ is the smallest triangulation containing the following 2^d simplices of dimension d:

$$\hat{v}^k + S_\pi \quad \text{for } \pi \in \operatorname{Sh}(k, d-k) \quad \text{for } k \in \{0, \dots, d\}. \tag{3.4}$$

The enumeration $\operatorname{rednum}(\hat{S}, \mathrm{lex})$ is given (uniquely) by the mappings e^π.

For a consistently numbered d-simplex (S, e), let A be the isotonic, affine map $(\hat{S}, \mathrm{lex}) \to (S, e)$; cf. Lemma 2.2. The red refinement of (S, e) is the pair $\operatorname{redref}(S, e) = A\big(\operatorname{redref}(\hat{S}, \mathrm{lex})\big)$.

Bey [6, La. 3.1.23] proves that $\operatorname{redref}(\hat{S}, \mathrm{lex})$ is a consistently numbered triangulation of $\hat{S}$; see also [13]. The following direct characterization of $\operatorname{rednum}(\hat{S}, \mathrm{lex})$ simplifies the definition and analysis of the green refinement rule below. The author did not find it in the literature.

Lemma 3.2. *The enumeration* $\operatorname{rednum}(\hat{S}, \mathrm{lex})$ *is given by the lex-order on the vertices of* $\operatorname{redtri}(\hat{S}, \mathrm{lex})$*, that is,* $\operatorname{rednum}(\hat{S}, \mathrm{lex}) = \mathrm{lex}$.

Proof. Owing to [6, La. 3.1.23], one must show that the ordering of the vertices of each simplex in (3.4) is given by the lex-order. Fix arbitrary $k \in \{0, \dots, d\}$, $\pi \in \operatorname{Sh}(k, d-k)$. Then, due to (3.3), one must show that $\hat{v}^k + \pi\hat{v}^i \preceq \hat{v}^k + \pi\hat{v}^j$ is equivalent to $i \le j$. As the vertices in (3.1) are pairwise different (also when shifted by $\hat{v}^k$), their enumeration in lex-order is unique. Hence, if one shows that $i \le j$ implies $\hat{v}^k + \pi\hat{v}^i \preceq \hat{v}^k + \pi\hat{v}^j$, the uniqueness of the enumeration in lex-order proves the lemma.

From $i \le j$ and (3.1), one concludes that every component of the vector $\hat{v}^j - \hat{v}^i$ is nonnegative. Therefore, all components of $\pi(\hat{v}^j - \hat{v}^i) = \pi\hat{v}^j - \pi\hat{v}^i$ are nonnegative, which implies $0 \preceq \pi\hat{v}^j - \pi\hat{v}^i$. Adding $\hat{v}^k$ and rearranging this inequality yields the desired result. □

Remark 3.3. It is well known that $\operatorname{redref}(S, e)$ can be defined without reference to the simplex $(\hat{S}, \mathrm{lex})$ (cf. [6, Ch. 3.1.4]), basically by expressing πv^i in barycentric coordinates (more precisely, the action of π). Similar to Lemma 3.2, the enumeration $\operatorname{rednum}(S, e)$ can be described without reference to $(\hat{S}, \mathrm{lex})$ as given by the lex-order on barycentric coordinates; cf. Lemma 5.7 below.

Remark 3.4. A refinement rule is isotonic-affinely invariant if $A\big(r(S, e)\big) = r(T, g)$ holds whenever A is the isotonic affine map $A\colon (S, e) \to (T, g)$. The red refinement rule is isotonic-affinely invariant in all dimensions d.

For $d = 2$, the red refinement is affinely invariant; that is, it commutes with all invertible, affine maps $S \to T$, not only the isotonic one. In this case, the triangulation $\operatorname{redtri}(\hat{S}, \mathrm{lex})$ does not depend on lex. For $d = 3$, this is not true. For any enumeration e, the triangulation $\operatorname{redtri}(\hat{S}, e)$ contains the four child-tetrahedra at the vertices of S. However, the remaining four tetrahedra, which form an octahedron, share one edge (out of a set of three possible edges) which *does* depend on e. As this effect is constrained to the interior of S, it is of minor importance for a red–green refinement algorithm for $d = 3$.

For $d \ge 4$, the choice of e influences how $\operatorname{redtri}(\hat{S}, e)$ triangulates the boundary of S. Hence, a consistent enumeration is required to satisfy the intersection property (2.2).

An important property of the red refinement rule is that it commutes with restriction to faces.

Lemma 3.5 ([6, La. 3.1.24], [13])**.** *Let F be a face of the consistently numbered d-simplex (S, e). Then* $\operatorname{redref}(F, e|_{\mathcal{F}(F)}) = \operatorname{redref}(S, e)|_F$.

A main theorem, valid in any dimension d, that follows from Lemma 3.5 describes the global red refinement of $\mathcal{C}$ [6, Thm. 3.1.25], [13].

Theorem 3.6. *If $(\mathcal{C}, e)$ is a consistently numbered triangulation, then* $\operatorname{redtri}(\mathcal{C}) = \cup\{\operatorname{redtri}(S, e|_S) \mid S \in \mathcal{C}\}$ *is a consistently numbered simplicial complex. In particular, the enumerations* $\operatorname{rednum}(S, e|_S)$*, $S \in \mathcal{C}$, are consistent with each other.*

3.2. **The placing triangulation.** The green refinement rule is defined using a well-known technique from convex geometry to construct triangulations, namely, the placing triangulation; cf. [12, Sec. 4.3], [29]. Let $P \subset \mathbb{R}^d$ be a polytope and let $v \in \mathbb{R}^d$ be a point.

Definition 3.7 (Visibility)**.** A face F of P is visible from v if there is a linear functional ψ on $\mathbb{R}^d$ and $c \in \mathbb{R}$ such that $\psi(v) > c$, $\psi(x) = c$ for all $x \in F$, and $\psi(x) \leq c$ for all $x \in P$.

The definition can also be stated in geometric terms.

Remark 3.8. If $v \notin \operatorname{aff} P$, then every face of P (including P itself) is visible from v. Now suppose that $v \in \operatorname{aff} P$ and that F is a facet of P. There is a unique hyperplane H in $\operatorname{aff} P$ which contains F. The polytope P is contained in exactly one of the closed half-spaces of $\operatorname{aff} P$ bounded by H. The facet F is visible from v if and only if v is contained in the opposite open half-space of $\operatorname{aff} P$. A face $\tilde{F}$ of P is visible from v if and only if there is a facet F of P with $\tilde{F} \subseteq F$ such that F is visible from v.

Definition 3.9 (Placing a vertex)**.** Let $\mathcal{C}$ be a complex with $\operatorname{set} \mathcal{C} = P$ and $\mathcal{V} = \{F \in \mathcal{F}(P) \mid F \text{ is visible from } v.\}$. The complex $\mathcal{D}$ with $\operatorname{set} \mathcal{D} = \operatorname{conv}(P \cup \{v\})$ that results from placing v in $\mathcal{C}$ is $\mathcal{D} = \mathcal{C} \cup \{\operatorname{pyr}(v, F) \mid F \in \mathcal{C}, F \subseteq G \text{ for some } G \in \mathcal{V}\}$. One writes $\mathcal{D} = \operatorname{pla}(\mathcal{C}, v)$.

If $F \subseteq G$ for some $G \in \mathcal{V}$, then $v \notin \operatorname{aff} F$; hence, the pyramids in the definition of $\operatorname{pla}(\mathcal{C}, v)$ are all well-defined. If $v \in P$, then $\mathcal{V} = \emptyset$ and $\mathcal{D} = \mathcal{C}$. If $\mathcal{C}$ is a triangulation, so is $\mathcal{D}$.

Definition 3.10 (Placing triangulation)**.** Given a tuple of n points, $V = (v^i)_{i=1}^n$, the placing triangulation $\operatorname{pla} V$ of $\operatorname{conv}\big\{v^i \bigm| i \in \{1, \dots, n\}\big\}$ is obtained by beginning with the empty set and placing the vertices of the tuple successively in the order of their subscripts $i = 1, \dots, n$.

An example is shown in Figure 2.

3.3. **The green refinement rule.** Compared to the red refinement rule, the green refinement rule on a consistently numbered simplex (S, e) requires a refinement pattern as additional input. This is a set $R_S \subseteq \mathcal{F}(S)$ of faces of S, on which the green refinement must be consistent with the red refinement. First, the reference simplex $(\hat{S}, \text{lex})$ and a refinement pattern $\hat{R} \subseteq \mathcal{F}(\hat{S})$ are considered.

The tuple $\hat{w}$ containing all vertices and edge barycenters of $\hat{S}$ is defined with the auxiliary tuples

$$\begin{gathered} \hat{w}_i^{\text{aux}} = (\hat{v}^i + \hat{v}^{d+1-j})_{j=1}^{d+1-i}, \quad i \in \{0, 1, \dots, d\} \quad \text{as} \\ \hat{w} = \hat{w}_0^{\text{aux}} \sqcup \hat{w}_1^{\text{aux}} \sqcup \dots \sqcup \hat{w}_d^{\text{aux}}. \end{gathered} \tag{3.5}$$

Here, $\sqcup$ denotes the concatenation of tuples. The assertion that all vertices and edge barycenters of $\hat{S}$ occur precisely once in $\hat{w}$ follows from Lemma 5.15 below. As an example, the tuple $\hat{w}$ for $d = 3$ is shown in Table 1. A sub-tuple of a given tuple $(v^i)_{i=1}^n$ is a tuple $(v^{i_j})_{j=1}^m$, where i_j is a strictly isotonic map $\{1, \dots, m\} \to \{1, \dots, n\}$.

Definition 3.11. Let w be the sub-tuple of $\hat{w}$, see (3.5), that contains all vertices of $\hat{S}$ and the barycenters of all edges in $\operatorname{cl} \hat{R}$; cf. (2.3). The green refinement of

TABLE 1. The tuple $\hat{w}$ in (3.5) for $d = 3$.

$\hat{w}_0^{\text{aux}}$				$\hat{w}_1^{\text{aux}}$			$\hat{w}_2^{\text{aux}}$		$\hat{w}_3^{\text{aux}}$
$\hat{v}^0 + \hat{v}^3$	$\hat{v}^0 + \hat{v}^2$	$\hat{v}^0 + \hat{v}^1$	$2\hat{v}^0$	$\hat{v}^1 + \hat{v}^3$	$\hat{v}^1 + \hat{v}^2$	$2\hat{v}^1$	$\hat{v}^2 + \hat{v}^3$	$2\hat{v}^2$	$2\hat{v}^3$

$(\hat{S}, \text{lex})$ with the refinement pattern $\hat{R}$ is the consistently numbered triangulation $\text{greenref}(\hat{S}, \text{lex}, \hat{R}) = \big(\text{greentri}(\hat{S}, \text{lex}, \hat{R}), \text{greennum}(\hat{S}, \text{lex}, \hat{R})\big)$. The triangulation and its consistent numbering are given by

$$\text{greentri}(\hat{S}, \text{lex}, \hat{R}) = \text{pla}(w), \quad \text{greennum}(\hat{S}, \text{lex}, \hat{R}) = \text{lex}\,. \tag{3.6}$$

For a consistently numbered simplex (S, e) and the refinement pattern $R_S \subseteq \mathcal{F}(S)$, let A be the isotonic, affine map $(\hat{S}, \text{lex}) \to (S, e)$. Let $\hat{R} \subseteq \mathcal{F}(\hat{S})$ be the refinement pattern with $\{A(\hat{F}) \mid \hat{F} \in \hat{R}\} = R_S$. The green refinement of (S, e, R_S) is the pair $\text{greenref}(S, e, R_S) = A\big(\text{greenref}(\hat{S}, \text{lex}, \hat{R})\big)$.

Remark 3.12. Note that two different orderings of the points w^i appear in Definition 3.11. The ordering induced by the tuple $\hat{w}$ is used in the construction of the triangulation $\text{greentri}(\hat{S}, \text{lex}, \hat{R})$. However, the enumeration $\text{greennum}(\hat{S}, \text{lex}, \hat{R})$ is induced by the lex-order on $\mathbb{R}^d$.

Remark 3.13. The refinement pattern R_S may be an arbitrary subset of $\mathcal{F}(S)$. However, Definition 3.11 is such that the refinement patterns R_S and $\operatorname{cl} R_S$ produce the same green refinement (because the tuple w is constructed from $\operatorname{cl} \hat{R}$). This does not cause an avalanche effect in the red–green refinement method: If the red refinement rule is applied to all $F \in R_S$, then, due to Lemma 3.5, this induces the red refinement of all $F \in \operatorname{cl} R_S$. Hence, all $F \in \operatorname{cl} R_S$ must be considered when restoring the consistency of the triangulation with the green refinement rule.

Remark 3.14. The practical construction of a green refinement of $\hat{S}$ is straightforward. First, one constructs the placing triangulation of the appropriate sub-tuple of $\hat{w}$. Methods for this are discussed, for example, in [12]. Second, the resulting triangulation is enumerated in lex-order (see also Remark 5.13). Two simple examples, where the refinement pattern is in a proper face of $\hat{S}$, are shown in the first two rows of Figure 3. A more complex example is shown in the last row.

The implementation of a red–green refinement algorithm with the refinement rules from Definitions 3.1 and 3.11 is discussed in Section 4.2.

Complementary to the refinement pattern R_S is the set

$$U_S = \{F \in \mathcal{F}(S) \mid F \cap T \subseteq \mathcal{F}_0(S) \text{ for all } T \in R_S\}\,. \tag{3.7}$$

It contains the faces of S which do not require a triangulation to be consistent with the red refinement prescribed by R_S. It is easy to see that U_S is always a sub-complex of the face-lattice (contrary to R_S). The *main theorem* of this paper is the following.

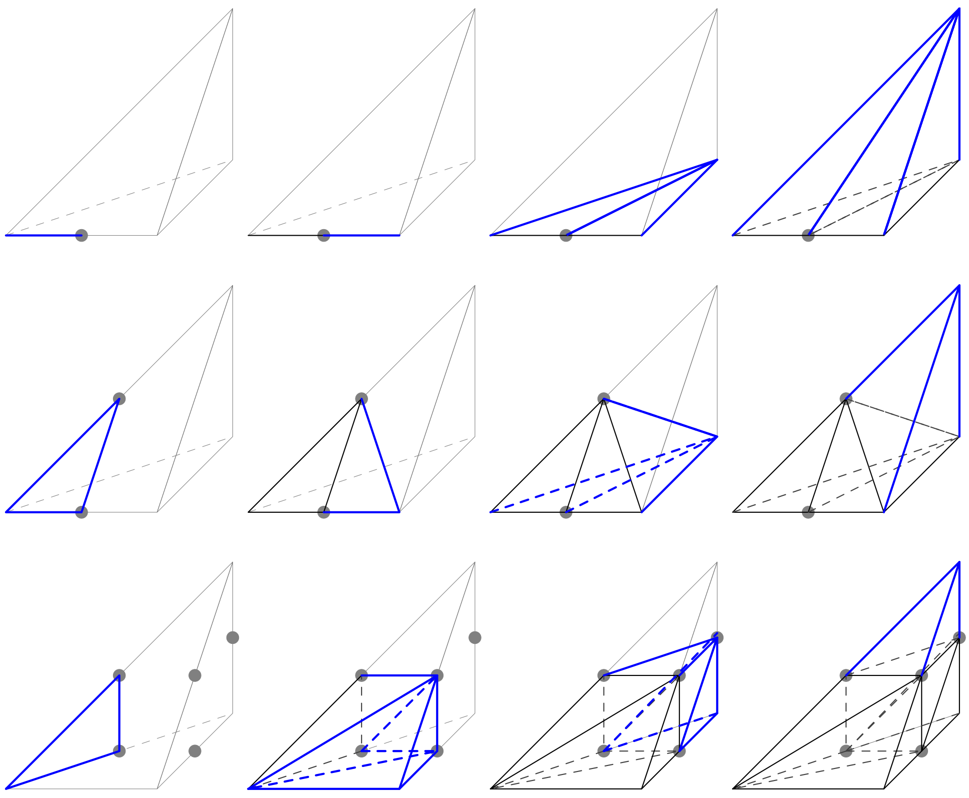

FIGURE 3. Each row shows the construction of a green closure from Definition 3.11 for $d = 3$. The ith image in a row shows the tentative triangulation after placing all points of w up to $\hat{w}_i^{\mathrm{aux}}$, $i \in \{0, \dots, 3\}$. The new edges are bold (and blue in the online version). Top: The edge $2\hat{v}^0 2\hat{v}^1$ is refined. Middle: The edges $2\hat{v}^0 2\hat{v}^1$ and $2\hat{v}^0 2\hat{v}^3$ are refined. Bottom: Only the edge $2\hat{v}^0 2\hat{v}^1$ is not refined.

Theorem 3.15. *For all consistently numbered simplices* (S, e)*, all faces* $F \subseteq S$*, and all refinement patterns* $R_S \subseteq \mathcal{F}(S)$*, there hold*

$$\operatorname{greenref}(S, e, R_S)|_F = \operatorname{greenref}\big(F, e|_{\mathcal{F}(F)}, \operatorname{cl} R_S \cap \mathcal{F}(F)\big), \tag{3.8}$$

$$F \in \operatorname{cl} R_S \implies \operatorname{greenref}(S, e, R_S)|_F = \operatorname{redref}(F, e|_{\mathcal{F}(F)}), \tag{3.9}$$

$$F \in U_S \implies \operatorname{greenref}(S, e, R_S)|_F = (\mathcal{F}(F), e|_{\mathcal{F}(F)}). \tag{3.10}$$

Theorem 3.15 solves the Green Closure Problem 1.1. Assertion (3.8) states that the green refinement of any face is completely determined by the data on this face (the restriction to F of the enumeration and of the closure of the refinement pattern). This makes the green refinements consistent with each other wherever two green refinements meet. Assertion (3.9) makes the green refinement consistent with the red refinement wherever this is required by the refinement pattern. Assertion (3.10) makes the green refinement consistent with any unrefined region of the mesh.

The choice $R_S = \{S\}$ and $F = S$ is admissible in (3.9). Hence, the red refinement of a simplex can be constructed as a placing triangulation, which seems to be a new result in computational geometry. The result is established independently of Theorem 3.15; see Theorem 5.21 below. The latter is then used in the proof of Theorem 3.15.

Remark 3.16 (Numerical experiment). For several dimensions d, the validity of the properties (3.8) and (3.9) is confirmed with a computer program. The green refinement of $\hat{S}$ for a given refinement pattern is computed as in Definition 3.11. First, the tuple of points w is generated. Then, the placing triangulation $\operatorname{pla}(w)$ in (3.6) is generated with the software toolbox `polymake` [17] for convex geometry.

Using this tool, the left-hand side and the right-hand side of (3.8) are computed independently for $S = \hat{S}$, $e = \text{lex}$, all facets F of $\hat{S}$, all refinement patterns $R_{\hat{S}} \subseteq \mathcal{F}(\hat{S})$, and all dimensions $d \in \{1, 2, 3, 4, 5\}$. Property (3.8) is satisfied in every case.

To check property (3.9) (and thus also Theorem 5.21 below), the red refinement of $\hat{S}$ is generated directly from Definition 3.1. For all $d \in \{1, 2, \dots, 20\}$, this is compared to $\operatorname{pla}(\hat{w})$, which is the green refinement of $\hat{S}$ with the refinement pattern $\mathcal{F}(\hat{S})$; cf. Definition 3.11. The triangulations are equal in every case.

Before the proof of Theorem 3.15 is given in Section 5, it is shown in Section 4 how Theorem 3.15 can be used to derive a global red–green refinement algorithm in any dimension d.

4. Red–green refinement with arbitrary refinement pattern

Let $(\mathcal{C}, e)$ be a consistently numbered triangulation. Suppose that all simplices in the refinement pattern $R \subseteq \mathcal{C}$ must be refined with the red refinement rule. In Section 4.1, the basic case is considered that all simplices in the input triangulation may be refined with a green refinement rule. In Section 4.2, an algorithm is presented which takes repeated refinement into account.

4.1. Red–green refinement with arbitrary refinement pattern.

The local refinement $(\mathcal{D}, f)$ of $(\mathcal{C}, e)$ is defined from three parts as $\mathcal{D} = \mathcal{R} \cup \mathcal{U} \cup \mathcal{G}$. Here, $\mathcal{R}$ and $\mathcal{G}$ are constructed with the refinement rules of Section 3; $\mathcal{U}$ is the part of $\mathcal{C}$ which remains unchanged. The first part is generated with the red refinement rule as

$$\begin{aligned} \mathcal{R} &= \cup \left\{ \operatorname{redtri}(S, e|_{\mathcal{F}(S)}) \;\middle|\; S \in R \right\}, \\ f_F &= \operatorname{rednum}(S, e|_{\mathcal{F}(S)})_F \quad \text{for all } S \in R, F \in \mathcal{F}(S). \end{aligned} \tag{4.1}$$

Lemma 4.1. *The pair $(\mathcal{R}, f|_{\mathcal{R}})$ is a consistently numbered complex. The set $\operatorname{cl} R$ is the set of all simplices of $\mathcal{C}$ which are refined by the refinement pattern R.*

Proof. $\mathcal{R}$ is the union of sub-complexes of $\operatorname{redtri}(\mathcal{C}, e)$ and therefore a simplicial complex by Theorem 3.6. The enumeration in (4.1) is the restriction of $\operatorname{rednum}(\mathcal{C}, e)$ to $\mathcal{R}$ and therefore a consistent enumeration.

Due to Lemma 3.5, any simplex in $\operatorname{cl} R$ is refined by the red refinement rule. Conversely, let $\operatorname{redtri}(S, e|_{\mathcal{F}(S)}) \subseteq \mathcal{R}$ for some $S \in \mathcal{C}$. Owing to (4.1), there is a parent simplex $T \in R$ with $\operatorname{redtri}(S, e|_{\mathcal{F}(S)}) \subseteq \operatorname{redtri}(T, e|_{\mathcal{F}(T)})$. This gives $\operatorname{set} S \subseteq \operatorname{set} T$. As $\mathcal{C}$ is a complex and because of (2.3), one obtains $S \in \mathcal{F}(T) \subseteq \operatorname{cl} R$. $\square$

The second part is the subset of $\mathcal{C}$ that requires no refinement,

$$\mathcal{U} = \{S \in \mathcal{C} \mid S \cap T \subseteq \mathcal{F}_0(\mathcal{C}) \text{ for all } T \in R\}\,, \quad f_F = e_F \quad \text{for all } f \in \mathcal{U}. \tag{4.2}$$

The intersection $\mathcal{U} \cap \operatorname{cl} R$ may contain (many) vertices, but no higher dimensional simplices.

Lemma 4.2. *The pair $(\mathcal{U}, f|_{\mathcal{U}})$ is a consistently numbered complex. Moreover, there holds $\mathcal{U} = \{S \in \mathcal{C} \mid S \cap T \subseteq \mathcal{F}_0(\mathcal{C})$ for all $T \in \operatorname{cl} R\}$.*

Proof. Let $S \in \mathcal{U}$, $F \in \mathcal{F}(S)$. As $F \cap T \subseteq S \cap T$ for all T in R, the definition in (4.2) implies that $\mathcal{U}$ is face-complete. As a subset of $\mathcal{C}$, it satisfies the intersection property. The enumeration f is the restriction of e to the sub-complex $\mathcal{U}$. Therefore, it is consistent.

Let $\tilde{\mathcal{U}} = \{S \in \mathcal{C} \mid S \cap T \subseteq \mathcal{F}_0(\mathcal{C})$ for all $T \in \operatorname{cl} R\}$. As $R \subset \operatorname{cl} R$, one concludes $\tilde{\mathcal{U}} \subseteq \mathcal{U}$. Conversely, let $S \in \mathcal{U}$ and $T \in \operatorname{cl} R$ be arbitrary. Due to (2.3), there exists a $\tilde{T} \in R$ with $T \in \mathcal{F}(\tilde{T})$. One computes $S \cap T \subseteq S \cap \tilde{T} \subseteq \mathcal{F}_0(\mathcal{C})$, which implies $S \in \tilde{\mathcal{U}}$. □

The third part is defined with the help of the green refinement rule,

$$\begin{aligned} \mathcal{G} &= \cup \big\{\operatorname{greentri}\big(S, e|_{\mathcal{F}(S)}, \operatorname{cl}(R) \cap \mathcal{F}(S)\big) \bigm| S \in \mathcal{C} - (R \cup \mathcal{U})\big\}\,, \\ f_F &= \operatorname{greennum}\big(S, e|_{\mathcal{F}(S)}, \operatorname{cl}(R) \cap \mathcal{F}(S)\big)_F \quad \text{for all } F \in \mathcal{G}. \end{aligned} \tag{4.3}$$

Remark 4.3. In addition to the red and green refinement rule, a red–green refinement algorithm has a global component that enforces that a simplex which has been generated by a green refinement rule is not refined further. If this is required by the refinement pattern, the green refinement of the parent simplex is rolled back and replaced with a red refinement. Well-known strategies for this are Bank's worker list based approach [1] and Bastian's level oriented approach [5]. Both global components can be generalized to the d-dimensional situation. Specifically, the algorithm in [5] and its correctness analysis work in d dimensions if one replaces "tetrahedron" with "d-simplex" and if one removes all references to the case that a green refinement with a specified refinement pattern is not found. An adjusted version of the algorithm from [5] is presented in Section 4.2.

The remainder of the section is devoted to proving that $(\mathcal{D}, f)$ is a consistently numbered simplicial complex with $\operatorname{set}\mathcal{D} = \operatorname{set}\mathcal{C}$. A simple partial result is the following.

Lemma 4.4. *The pair $(\mathcal{R}\cup\mathcal{U}, f|_{\mathcal{R}\cup\mathcal{U}})$ is a consistently numbered simplicial complex.*

Proof. Due to Lemmas 4.1 and 4.2, $(\mathcal{R}, f|_{\mathcal{R}})$ and $(\mathcal{U}, f|_{\mathcal{U}})$ are consistently numbered triangulations. Hence, their union is face-complete. The intersection property must be verified only for pairs of simplices $S \in \mathcal{R}$, $T \in \mathcal{U}$. Owing to (4.2), $\operatorname{set}\mathcal{U} \cap \operatorname{set} R$ is a subset of $\mathcal{F}_0(\mathcal{C})$. From the definition of a refinement rule, one obtains $\operatorname{set}\mathcal{R} = \operatorname{set} R$. Thus, there holds $\operatorname{set}\mathcal{R} \cap \operatorname{set}\mathcal{U} = \operatorname{set} R \cap \operatorname{set}\mathcal{U} \subseteq \mathcal{F}_0(\mathcal{C})$. For S, T as above, this means that $S \cap T \subseteq \mathcal{F}_0(\mathcal{C})$ is empty or contains a single common vertex v. Hence, the intersection property is satisfied. Finally, the consistency of the enumerations on the intersection must be checked. But as v only has the trivial enumeration, the definitions of f_v in (4.1) and(4.2) agree. □

Theorem 4.5. *The pair $(\mathcal{D}, f)$ is a consistently numbered triangulation of* $\operatorname{set}\mathcal{C}$.

Proof. The set $\mathcal{D}$ is defined as the union of simplicial complexes; cf. Lemmas 4.1, 4.2, and the definition of $\mathcal{G}$ in (4.3). Thus, $\mathcal{D}$ is face-complete. To prove the intersection property, all parent simplices $S, T \in \mathcal{C}$ are considered. Only the consistency on $S \cap T$ must be checked, as the refinements of S, T are consistently numbered simplicial complexes by the definition of a refinement rule.

By Lemma 4.1, the refinement pattern $\operatorname{cl} R$ contains all simplices of $\mathcal{C}$ which are refined with the red refinement rule. It induces the refinement pattern $R_S = \operatorname{cl} R \cap \mathcal{F}(S)$ on S. Due to Lemma 4.4, one only has to consider the case that S is refined with $\operatorname{greenref}(S, e|_{\mathcal{F}(S)}, R_S)$. The intersection $F = S \cap T \in \mathcal{C}$ is a (possibly empty) face of S. The refinement of S induces the refinement $\operatorname{greenref}(S, e, R_S)|_F$ of F. This is now compared with the refinement of F induced by the refinement of T.

Consider the three possible cases for T. If $T \in \operatorname{cl} R$, then T is refined with the red refinement rule because of Lemma 4.1. Due to Lemma 3.5, F is refined by $\operatorname{redref}(F, e|_{\mathcal{F}(F)})$. From $T \in \operatorname{cl} R$, one gets $F \in \operatorname{cl} R$, and therefore $F \in \operatorname{cl} R_S$. Due to property (3.9), $\operatorname{greenref}(S, e|_{\mathcal{F}(S)}, R_S)|_F = \operatorname{redref}(F, e|_{\mathcal{F}(F)})$.

If $T \in \mathcal{U}$, then Lemma 4.2 yields $\mathcal{F}(F) \subseteq \mathcal{U}$ and $f|_{\mathcal{F}(F)} = e|_{\mathcal{F}(F)}$. From (3.7) and the characterization of $\mathcal{U}$ in Lemma 4.2, one concludes $F \in U_S$. Due to property (3.10), one obtains $\operatorname{greenref}(S, e|_{\mathcal{F}(S)}, R_S)|_F = (\mathcal{F}(F), e|_{\mathcal{F}(F)})$.

If $T \in \mathcal{C} - (\mathcal{U} \cup R)$, then T is refined by $\operatorname{greenref}(T, e|_{\mathcal{F}(T)}, R_T)$, $R_T = \operatorname{cl} R \cap \mathcal{F}(T)$. Observing (3.8), this induces the refinement $\operatorname{greenref}(F, e|_{\mathcal{F}(F)}, \tilde{R}_F)$, $\tilde{R}_F = \operatorname{cl} R_T \cap \mathcal{F}(F)$, of F. Owing to (3.8), $\operatorname{greenref}(S, e, R_S)|_F = \operatorname{greenref}(F, e|_{\mathcal{F}(F)}, R_F)$, $R_F = \operatorname{cl} R_S \cap \mathcal{F}(F)$ holds. Using $\operatorname{cl} R_S = R_S$ and $\operatorname{cl} R_T = R_T$, one computes $\tilde{R}_F = \operatorname{cl} R \cap \mathcal{F}(F) = R_F$. Hence, $\operatorname{greenref}(T, e|_{\mathcal{F}(T)}, R_T)|_F = \operatorname{greenref}(S, e|_{\mathcal{F}(S)}, R_S)|_F$ as required. In summary, $\mathcal{D}$ is a consistently numbered simplicial complex.

The assertion $\operatorname{set} \mathcal{D} = \operatorname{set} \mathcal{C}$ holds because every $S \in \mathcal{C}$ is in one of the sets $\operatorname{cl} R$, $\mathcal{U}$, and $\mathcal{C} - (R \cup \mathcal{U})$. ☐

Some optimality properties of the red–green refinement $\mathcal{D}$ are collected in the following theorem.

Theorem 4.6. *The triangulation $\mathcal{R}$ is the minimal complex containing the red refinement of R. Complementary, $\mathcal{U}$ is the maximal subcomplex of $\mathcal{C}$ which is compatible with $\mathcal{R}$. That is, if $\tilde{\mathcal{U}}$ is a complex with $\mathcal{U} \subseteq \tilde{\mathcal{U}} \subseteq \mathcal{C}$ such that $\tilde{\mathcal{U}} \cup \mathcal{R}$ has the intersection property, then there holds $\tilde{\mathcal{U}} = \mathcal{U}$. The new vertices in $\mathcal{D}$ with respect to $\mathcal{C}$ are precisely the barycenters of the edges in* $\operatorname{cl} R$.

Proof. Assume $\tilde{\mathcal{R}}$ contains $\operatorname{redtri}(S, e|_{\mathcal{F}(S)})$ for all $S \in R$. Then, (4.1) implies $\mathcal{R} \subseteq \tilde{\mathcal{R}}$. Hence, $\mathcal{R}$ is minimal. This proves the first assertion of the theorem.

By definition, $\mathcal{U} \subseteq \tilde{\mathcal{U}}$. Assume that there is a simplex $S \in \tilde{\mathcal{U}} - \mathcal{U}$. By (4.2), there is $T \in R$ such that $F = S \cap T \not\subseteq \mathcal{F}_0(\mathcal{C})$. The dimension of F is at least 1. Due to Lemma 3.5, $\mathcal{R}$ contains the barycenters of the edges of F. Let b be such a barycenter. There holds $b \in S$, but $b \notin \mathcal{F}(S)$. This violates the intersection property, which contradicts the assumption that $\tilde{\mathcal{U}}$ is a complex. Hence, one concludes $\tilde{\mathcal{U}} - \mathcal{U} = \emptyset$.

The third assertion of the theorem is proved by considering the new vertices introduced in $\mathcal{R}$, $\mathcal{U}$, and $\mathcal{G}$. There holds $\mathcal{F}_0(\mathcal{R}) = \mathcal{F}_0\big(\operatorname{cl}(R)\big) \cup B$, where B is the set of all barycenters of $\mathcal{F}_1\big(\operatorname{cl}(R)\big)$. This follows from (3.1) and (3.4), cf. Lemma 5.15 below. The vertices of $\mathcal{U}$ satisfy $\mathcal{F}_0(\mathcal{U}) \subseteq \mathcal{F}_0(\mathcal{C})$ due to the definition in (4.2).

Let $S \in \mathcal{C} - (\mathcal{R} \cup \mathcal{U})$ and $\mathcal{S} = \text{greentri}(S, e|_{\mathcal{F}(S)}, \text{cl}(R) \cap \mathcal{F}(S))$. From (3.5) and Definition 3.10, it follows that $\mathcal{F}_0(\mathcal{S}) = \mathcal{F}_0(S) \cup B_S$, where B_S contains the barycenters of the edges in $\text{cl}(R) \cap \mathcal{F}(S)$. From (4.3), one gets $\mathcal{F}_0(\mathcal{G}) \subseteq \mathcal{F}_0(\mathcal{C}) \cup B$. Altogether, using $\mathcal{D} = \mathcal{R} \cup \mathcal{U} \cup \mathcal{G}$, this implies $\mathcal{F}_0(\mathcal{D}) \subseteq \mathcal{F}_0(\mathcal{C}) \cup B$. As $\mathcal{F}_0(\mathcal{C}) \subseteq \mathcal{F}_0(\mathcal{R}) \cup \mathcal{F}_0(\mathcal{U})$, it follows that $\mathcal{F}_0(\mathcal{D}) = \mathcal{F}_0(\mathcal{C}) \cup B$. □

Algorithm 1 Red–green refinement in d dimensions.

```
▷ The auxiliary functions are defined in Algorithm 2. The data structures and functioning
are explained in Sections 4.2.1–4.2.6.
1: function RED_GREEN($\mathcal{C}$, $R$)
2:     $\mathcal{D}, m, r \leftarrow \emptyset$                    ▷ Start with empty auxilliary variables.
3:     for all $S \in \mathcal{C}_{\text{red}}$ with $R(S) = 1$ do      ▷ Propagate refinement marks to the edges.
4:         MARK_ALL_EDGES($S$, $m$)
5:     for $l = l_{\max} - 1, \dots, 0$ do      ▷ Ensure that green children are not refined further.
6:         for all $P \in \mathcal{C}_{\text{p},l}$ do
7:             for all children $S$ of $P$ do
8:                 if $R(S) = 1$ or $m(E) = 1$ for some edge $E \in \mathcal{F}^1(S) - \mathcal{F}^1(P)$ then
9:                     MARK_ALL_EDGES($P$, $m$)
10:                    $r(P) \leftarrow 1$

11:    for all $S \in \mathcal{C}_{\text{red}}$ do                    ▷ Refine the red simplices.
12:        if $R(S) = 1$ then
13:            $\mathcal{D}_{\text{red},l(S)+1} \leftarrow \mathcal{D}_{\text{red},l(S)+1} \cup \text{redtri}(S)$
14:        else
15:            INSERT_GREEN_REFINEMENT($S$, $m$, $\mathcal{D}$)
16:    for all $P \in \mathcal{C}_{\text{p}}$ do          ▷ Refine the parents of the green simplices.
17:        if $r(P) = 1$ then
18:            for all $S \in \text{redtri}(P)$ do
19:                INSERT_GREEN_REFINEMENT($S$, $m$, $\mathcal{D}$)
20:        else
21:            INSERT_GREEN_REFINEMENT($P$, $m$, $\mathcal{D}$)
22:    $\mathcal{C} \leftarrow \mathcal{D}$
23:    return $\mathcal{C}$
```

4.2. **Repeated red–green refinement.** A concrete example of a red–green refinement algorithm, which produces only a finite number of similarity classes of simplices if it is applied repeatedly, is given in this section. The maximal number of similarity classes depends only on the dimension d and the initial triangulation. Algorithm 1 is the generalization of Bastian's level oriented red–green algorithm [5] to d dimensions and a full set of green refinement rules. It is stated for the model case in which

- there is no mesh coarsening,
- only a single mesh is present (i.e., no multilevel data structure),
- only the d-dimensional simplices are tracked in the data structure (the lower-dimensional faces are represented implicitly).

These restrictions are made to conserve space and notation.

In the refinement algorithm, each d-simplex is either “red” or “green”. It is red if it is produced by a series of (zero or more) applications of the red refinement rule (and none of the green refinement rule). It is green if the last refinement is computed with the green refinement rule. The (refinement-) level $l(S)$ of a d-simplex S is the number of times that a refinement rule was applied to create S. For example, all simplices in the initial triangulation are red and satisfy $l(S) = 0$. The children of a simplex S have level $l(S) + 1$.

To ensure that only a finite number of similarity classes of simplices is generated (see property (E) in Section 4.2.7), a standard rollback rule [5] is introduced for green simplices S with refinement pattern R_S and parent P:

(4.4) If $S \in R$ or if $\operatorname{cl} R_S$ contains an edge of S that is not already an edge of P, then the green refinement of P is removed from the triangulation, and the red refinement of P is inserted.

The condition in (4.4) has two parts. The first part, $S \in R$, covers the case that the user requests a refinement of the green simplex S. The second part may occur due to red refinement of neighbor-simplices of S. After a rollback, the children of P are either directly consistent with (the descendants of) the red neighbors of P or a green closure is required. The latter can then be computed because the involved simplices are red. However, if P has a green neighbor P', the latter may require a rollback. The number of rollbacks caused by S is finite because only simplices P' with $l(P') = l(S) - 1$ are affected. So eventually all rollbacks reach level 0, where all simplices are red. Altogether, a naive red–green algorithm allowing repeated refinement is as follows:

(1) Perform the red refinement of all red simplices.
(2) Repeatedly apply the rollback (4.4) to all $S \in \mathcal{C}$ until $\mathcal{C}$ does not change any more.
(3) Compute the green closure of $\mathcal{C}$ (as in Section 4.1).

Algorithm 1 implements this scheme with linear time complexity in the size of the input triangulation; cf. Lemma 4.7. It is explained in the Sections 4.2.1–4.2.6.

4.2.1. *Data structures.* A vertex with coordinates in $\mathbb{R}^d$ is modeled as a d-tuple of floating point numbers. Floating point computations with the coordinates are assumed to be deterministic. This is a very moderate requirement [15]. A consistently numbered d-simplex (S, e) is modeled as a tuple $(v_0, \dots, v_d)$ of vertices, where the enumeration e is given implicitly by the ordering of the vertices in the tuple. The red and the green refinement rules in Algorithm 1 use this implicit numbering, so their usual argument e is not explicitly shown in the notation.

A container data structure that can hold the d-simplices grouped by their level is required. If $\mathcal{C}$ is such a container, the notation $\mathcal{C}_l$ is used for the subset corresponding to level l. Accessing all members in the order of decreasing level must be possible in $\mathcal{O}(1)$ time per member. The order of the members in each level may be arbitrary. Inserting a new member into each level must be possible in $\mathcal{O}(1^+)$ time (constant per inserted member, amortized over many insertions). Such a container data structure exists. For instance, an array that, for each level, contains a doubly linked list or another array (with a suitable memory allocation strategy on growth) satisfies the requirements [25].

To model mappings $f \colon M \to \{0, 1\}$, where M is a set of edges or a set of d-simplices, hash tables are used as a dictionary data structure. This allows for membership tests of the set $\{S \in M \mid f(S) = 1\}$ and insertions into the dictionary in expected time $\mathcal{O}(1)$ [25].

4.2.2. *Input.* The input to the main function RED_GREEN in Algorithm 1 is a consistently numbered triangulation $\mathcal{C}$ and a refinement pattern R. The triangulation is implemented using three containers $\mathcal{C} = (\mathcal{C}_{\text{red}}, \mathcal{C}_{\text{p}}, \mathcal{C}_{\text{green}})$. Initially, $\mathcal{C}_{\text{red}}$ contains all d-simplices of the triangulation, whereas $\mathcal{C}_{\text{p}}$ and $\mathcal{C}_{\text{green}}$ are empty. The refinement pattern R is given as a mapping $\mathcal{C}_{\text{red}} \cup \mathcal{C}_{\text{green}} \to \{0, 1\}$. For this, a dictionary data structure is used (also named R).

4.2.3. *Action.* Red simplices $S \in \mathcal{C}_{\text{red}}$ with $R(S) = 1$ are refined with the red refinement rule. For green simplices $S \in \mathcal{C}_{\text{green}}$ with $R(S) = 1$ or which would themselves require a green refinement, the parent $P \in \mathcal{C}_{\text{p}}$ is refined with the red refinement rule. The green closure is computed.

4.2.4. *Output.* The output of Algorithm 1 is a consistently numbered triangulation (constructed in the auxiliary variable $\mathcal{D} = (\mathcal{D}_{\text{red}}, \mathcal{D}_{\text{p}}, \mathcal{D}_{\text{green}})$, but returned as $\mathcal{C}$), to which the function RED_GREEN can be applied again with a new refinement pattern.

4.2.5. *Structure of Algorithm* 1. There are two phases in the algorithm (as in [5]). In phase I (lines 3–10), the refinement pattern R is evaluated. All edges E that must be refined are marked with $m(E) = 1$. Here, m is a dictionary that represents a mapping on the edges, $m\colon \mathcal{F}^1(\mathcal{C}) \to \{0, 1\}$. Additionally, all parent simplices P whose green refinement must be rolled back are marked with $r(P) = 1$, where r is a dictionary representing a mapping on parent simplices, $r\colon \mathcal{C}_{\text{p}} \to \{0, 1\}$. The new triangulation is constructed in phase II (lines 11–21) based on R, m, and r.

The green closure is completely determined in phase I (as it only depends on R, m, and r). This is an important difference to [5], where an iteration of the green closure process is needed in phase II. This is so because the red refinement rule may be used in [5] for "too complicated" refinement patterns R_S, which may cause additional rollbacks in phase II.

Algorithm 2 Auxiliary functions for Algorithm 1.

```
 1: function MARK_ALL_EDGES($S$, $m$)
 2:     for all edges $e$ of $S$ do
 3:         $m(e) \leftarrow 1$
 4: function INSERT_GREEN_REFINEMENT($S$, $m$, $\mathcal{D}$)
 5:     $R_S \leftarrow \{e \in \mathcal{F}^1(S) \mid m(e) = 1\}$.          ▷ the refinement pattern of $S$
 6:     if $R_S = \emptyset$ then                                   ▷ Copy unchanged simplices.
 7:         $\mathcal{D}_{\text{red},l(S)} \leftarrow \mathcal{D}_{\text{red},l(S)} \cup \{S\}$
 8:     else                                                    ▷ Insert the green closure.
 9:         $\mathcal{D}_{\text{p},l(S)} \leftarrow \mathcal{D}_{\text{p},l(S)} \cup \{S\}$
10:         $\mathcal{D}_{\text{green},l(S)+1} \leftarrow \mathcal{D}_{\text{green},l(S)+1} \cup \text{greentri}(S, R_S)$
```

4.2.6. *Auxiliary functions.* Algorithm 1 uses two auxiliary functions which are shown in Algorithm 2. The first is MARK_ALL_EDGES, which is used only in phase I to mark all edges to which the red refinement rule must be applied. The edge marks in m allow for the determination of the local green refinement patterns R_S in the second auxiliary function INSERT_GREEN_REFINEMENT (line 5), which is called only in phase II. This function inserts the green refinement of its parameter S into $\mathcal{D}$. It also deals with trivial green refinement $R_S = \emptyset$, where S is simply copied to $\mathcal{D}_{\mathrm{red}}$.

4.2.7. *Correctness of Algorithm* 1. The function RED_GREEN maintains three invariants:

- **T:** The simplices in $\mathcal{C}_{\mathrm{red}} \cup \mathcal{C}_{\mathrm{green}}$ are the d-simplices of a consistently numbered triangulation of the whole domain.
- **E:** There are finitely many equivalence classes (with respect to geometric similarity) of simplices in $\mathcal{C}$ independent of the number of prior applications of RED_GREEN.
- **C:** The simplices in $\mathcal{C}_{\mathrm{red}}$ and $\mathcal{C}_{\mathrm{p}}$ are red, whereas the simplices in $\mathcal{C}_{\mathrm{green}}$ are green. (The latter are precisely the children of the simplices in $\mathcal{C}_{\mathrm{p}}$.)

To prove (E), note that there is only a finite number of local green refinement patterns R_S for a d-simplex S. (Each edge of S either is in $\operatorname{cl} R_S$ or not; cf. Definition 3.11.) Each corresponding green refinement contains finitely many child simplices. Let the finite number of all of these children be g. The (repeated) red refinement of a d-simplex S produces at most $d!/2$ equivalence classes under geometric similarity. As green simplices are never refined further, the number of equivalence classes of all red and green descendants of any simplex in the initial triangulation is at most $gd!/2$.

The invariants (T) and (C) have an inductive proof over the number of applications of the function RED_GREEN. For the initial triangulation, (T) and (C) are satisfied thanks to the assumptions in Section 4.2.2. In the induction step, one has to show that (T) and (C) hold for $\mathcal{D}$ if they hold for $\mathcal{C}$ (immediately before line 22).

To establish (C), one checks that the simplices inserted into $\mathcal{D}$ have the right color. In line 13 of RED_GREEN, a red simplex is inserted into $\mathcal{D}_{\mathrm{red}}$. All other insertions happen in INSERT_GREEN_REFINEMENT (lines 15, 19, and 21). Using the induction hypothesis, one concludes that the first argument S (respectively, P) is always red. The function itself inserts its parameter S into $\mathcal{D}_{\mathrm{red}}$ (line 7) or $\mathcal{D}_{\mathrm{p}}$ (line 9). The only insertion into $\mathcal{D}_{\mathrm{green}}$ happens in line 10, where a nontrivial green refinement of S is inserted. As required, these simplices are green.

To show (T), the key is to check the correctness of the lines 5–10 of RED_GREEN, which implement the rollback. This is the main difference from computing a green closure as in Section 4.1. For a parent simplex $P \in \mathcal{C}_{\mathrm{p},l}$, the rollback criterion for its children S in line 8 is a restatement of the condition in (4.4) with the data structures R and m. In step (2) of the naive red–green refinement algorithm on page 766, the rollback (4.4) is applied until $\mathcal{C}$ does not change any more. It is now shown that the loops in the lines 5–10 have the same effect. This is the case if (i) all writes to $m(E)$ precede the read of $m(E)$ in line 8, and (ii) all reads of $m(E)$ follow the write in MARK_ALL_EDGES in line 9.

To check (i), assume that $m(E)$ is read in line 8 for some edge of S.[2] Then, E is an edge of $\operatorname{redtri}(P)$ or $E \in \mathcal{F}^1(S) - \mathcal{F}^1(P) - \mathcal{F}^1(\operatorname{redtri}(P))$. In the latter case, there is no write to $m(E)$ because such edges of a green simplex can only have green neighbors, but $m(E)$ is only set for edges of red simplices. So one may assume $E \in \mathcal{F}^1(\operatorname{redtri}(P))$. There are two cases in which $m(E) = 1$ is written. First, there

[2]If E is an edge of P, then $m(E)$ is not considered in line 8 because E is simply added to R_P resulting in a different green refinement of P.

might be a red neighbor of S with $R(S) = 1$. Its edge marks are written to m in the loop in line 3, thus preceding the read in line 8. Second, there might be a neighbor P' of S with a green refinement that is rolled back. As P' is a neighbor across the edge $E \in \mathcal{F}^1(\text{redtri}(P))$, there holds $l(P') = l(P) + 1$. Therefore, the edge marks on P' have already been written in the previous iteration of the loop in line 5.

To check (ii), assume that $m(E)$ is set in MARK_ALL_EDGES in line 9 for some parent simplex P. This value is read in line 8 only if $E \in \mathcal{F}^1(S) - \mathcal{F}^1(P')$ for a green simplex S with parent P'. As E is an edge of the red simplex P, one concludes that $E \in \mathcal{F}^1(\text{redtri}(P'))$ which implies $l(P) = l(P') + 1$. Consequently, $l(P') < l(P)$ such that the read of $m(E)$ is in a later iteration of the loop in line 5 than the write.

4.2.8. *Time complexity.* The dimension d is treated as a constant in the $\mathcal{O}$-notation. For example, the function MARK_ALL_EDGES does a constant amount of work for each edge of S because insertions into the dictionary m take $\mathcal{O}(1)$ time; cf. Section 4.2.1. Thus, the time complexity is $\mathcal{O}(d^2)$, and this is treated as $\mathcal{O}(1)$.

Lemma 4.7. *Let N be the number of simplices in $\mathcal{C}_{red} \cup \mathcal{C}_{green}$. The time complexity of* RED_GREEN *is $\mathcal{O}(N)$.*

Proof. MARK_ALL_EDGES and INSERT_GREEN_REFINEMENT have the time complexity $\mathcal{O}(1)$. Therefore, the loops in lines 3 and 11 run in $\mathcal{O}(N)$ time.

Together, the loops in lines 5 and 6 enumerate all members of $\mathcal{C}_\text{p}$ and so does the loop in line 16. They perform $\mathcal{O}(N)$ iterations because $|\mathcal{C}_\text{p}| \leq N$. The number of (green or red) children visited in lines 7 and 18 is bounded by $\mathcal{O}(1)$ independent of the particular refinement. Thus, both loops do at most a constant amount of work per iteration. □

5. The proof of Theorem 3.15

Each of the properties (3.8)–(3.10) is treated in a separate subsection below. In Section 5.1, key properties of the placing triangulation are stated.

The proofs of property (3.8) (Section 5.2) and property (3.9) (Section 5.3) are both rather technical. For property (3.8), one must deal with many objects: a simplex, one of its faces, the corresponding reference simplices, the affine mappings between these simplices, and the tuples defining the respective green refinements; see Figure 4. On the other hand, the placing triangulation commutes with invertible, affine maps and with the restriction to faces (cf. Section 5.1), which simplifies the analysis. Once property (3.8) is established, it is used to simplify the proof of properties (3.9) and (3.10): As made precise in Lemma 5.14, it is sufficient to consider only the full refinement pattern $\mathcal{F}(S)$ and certain trivial refinement patterns on S to prove (3.9) and (3.10).

In the proof of property (3.9), a partition of the reference simplex into many parts is used. The simplex is covered with cubes, and the cubes are partitioned using half-spaces. The visible parts of the tentative triangulations in the computation of the placing triangulation are traced through the partition, which makes the proof technical. Furthermore, the partitioning is shown to be consistent with the red refinement. (That is, the interiors of the d-dimensional simplices are not cut.) This enables an inductive proof, in which the red refinement and the green refinement with the full refinement pattern are compared.

5.1. **Properties of the placing triangulation.** The following uniqueness property of the placing triangulation is required in Section 5.3.

Lemma 5.1 ([12, La. 4.3.2]). *Let $\mathcal{C}$ be a complex with the property that* $\operatorname{set}\mathcal{C}$ *is convex. The complex $\mathcal{D}$ obtained by placing a point v in $\mathcal{C}$ is the unique complex with* $\operatorname{set}\mathcal{D} = \operatorname{conv}(\operatorname{set}\mathcal{C} \cup \{v\})$ *that contains $\mathcal{C}$ as the sub-complex.*

Let $V = (v^i)_{i=1}^n$ be a tuple of points and $P = \operatorname{conv}\{v^i \mid i \in \{1, \dots, n\}\}$.

Lemma 5.2. *The placing triangulation commutes with invertible affine mappings, that is,* $\operatorname{pla}\big((Av^i)_{i=1}^n\big) = A\big(\operatorname{pla} V\big)$*, where A is an affine mapping which is invertible on* $\operatorname{aff} P$.

Proof. The proof is by induction. If the tuple contains only one point, there holds $\operatorname{pla}\big((Av^i)_{i=1}^1\big) = A\big(\operatorname{pla}(v^i)_{i=1}^1\big)$. Let $\mathcal{C} = \operatorname{pla}(v^i)_{i=1}^{m-1}$ for some $m \le n$, let $A(\mathcal{C})$ be the mapped complex, and $S = \operatorname{set}\mathcal{C}$. In view of Definition 3.9, one must show that for all $F \in \mathcal{C}$

$$F \text{ is visible from } v^m \quad \Leftrightarrow \quad A(F) \text{ is visible from } Av^m. \tag{5.1}$$

As A is invertible on $\operatorname{aff} P$, it is sufficient to prove only the implication from left to right. If F is visible from v^m, one gets from Definition 3.7 the functional ψ on $\mathbb{R}^d$ and $c \in \mathbb{R}$ with $\psi(v^m) > c$, $\psi(x) = c$ for all $x \in F$, and $\psi(x) \le c$ for all $x \in S$. Clearly, $\phi = \psi \circ A^{-1}$ is a linear functional. One gets $\phi(Av^m) > c$, $\phi(y) = c$ for all $y \in A(F)$, and $\phi(y) \le c$ for all $y \in A(S)$. Thus, $A(F)$ is visible from Av^m. Using (5.1), one obtains $\operatorname{pla}\big((Av^i)_{i=1}^m\big) = A\big(\operatorname{pla}(v^i)_{i=1}^m\big)$ from the induction hypothesis. ☐

Lemma 5.3. *The placing triangulation commutes with the restriction to faces. That is,* $\operatorname{pla}(V_F) = \big(\operatorname{pla}(V)\big)|_F$.

To prove Lemma 5.3, some properties of the (well-known) construction of a regular complex are used [12, Sec. 4.3], [29]: Given a tuple of weights $\alpha = (\alpha_i)_{i=1}^n \subset \mathbb{R}$, the regular complex of P is defined as follows.

Definition 5.4 (Regular complex). Let $Q \subset \mathbb{R}^{d+1}$ be the convex hull of the set $\{(v^1, \alpha_1), \dots, (v^n, \alpha_n)\} \subset \mathbb{R}^{d+1}$. A lower facet of Q is a facet as in (2.1) with $\psi(x) = a^T x$, $a \in \mathbb{R}^{d+1}$, where $a_{d+1} < 0$. The regular complex $\operatorname{rc}(V, \alpha)$ of P is the smallest complex which contains the projections of all lower facets of Q onto $\mathbb{R}^d \times \{0\}$.

Let F be a face of P and let $V_F = (v^{i_j})_{j=1}^m$ be the sub-tuple of V containing all $v^i \in F$; let $\alpha_F = (\alpha_{i_j})_{j=1}^m$.

Lemma 5.5 ([12, La. 2.3.15]). *The regular complex commutes with the restriction to faces. That is,* $\operatorname{rc}(V_F, \alpha_F) = \big(\operatorname{rc}(V, \alpha)\big)|_F$.

Lemma 5.6 ([12, La. 4.3.4]). *All placing triangulations are regular complexes. Moreover, there is a constant $c \ge 1$ (depending on V) such that* $\operatorname{pla}(V) = \operatorname{rc}(V, \alpha)$ *if the weights satisfy*

$$\alpha_1 > 0, \quad \alpha_{i+1} > c\alpha_i, \quad i \in \{1, \dots, n-1\}. \tag{5.2}$$

Proof of Lemma 5.3. Due to Lemma 5.6, there is a constant c_S and weights $(\alpha_i^S)_{i=1}^n$ satisfying (5.2) such that $\operatorname{pla}(V) = \operatorname{rc}(V, \alpha^S)$. Using Lemma 5.6 again, there is a constant c_F and weights $(\alpha_i^F)_{i=1}^m$ satisfying (5.2) such that $\operatorname{pla}(V_F) = \operatorname{rc}(V_F, \alpha^F)$.

Let $c = \max\{c_S, c_F\}$ and $\alpha_1 > 0$, $\alpha_{i+1} > c\alpha_i$, $1 \le i \le n-1$. The tuple α satisfies (5.2) with the constant c_S, and the tuple $(\alpha_{i_j})_{j=1}^m$ satisfies (5.2) with the constant c_F. Using this and Lemma 5.5, one finds

$$\operatorname{pla}(V)|_F = \operatorname{rc}(V,\alpha)|_F = \operatorname{rc}\big(V_F, (\alpha_{i_j})_j\big) = \operatorname{pla}(V_F).$$

□

5.2. **Consistency within the green refinement.** Property (3.8) is proved. One must show that both $\operatorname{greennum}(S, e, R_S)|_F$ and $\operatorname{greentri}(S, e, R_S)|_F$ only depend on the data on F, which are the enumeration $e|_{\mathcal{F}(F)}$ and the restriction of (the closure of) the refinement pattern to F. There are two key lemmas: Lemma 5.10 relates the lex-order on the reference k-simplex and the lex-order on a k-face of the reference d-simplex. This is important for enumeration-part of the green refinement. Lemma 5.11 relates the tuple $\hat{w}$ from (3.5) on the k-dimensional and the d-dimensional reference simplex. Together with the invariance properties of the placing triangulation, this makes the green refinement of F independent of its embedding into S.

Suppose that (S, e) is a consistently numbered d-simplex and that $x \in \operatorname{aff}(S)$. The barycentric coordinates of x with respect to (S, e) are the unique vector $\lambda = (\lambda_0, \dots, \lambda_d)^T = \operatorname{bary}(x, S, e) \in \mathbb{R}^{d+1}$ with

$$x = \sum_{i=0}^{d} \lambda_i e_S(i), \quad \sum_{i=0}^{d} \lambda_i = 1. \tag{5.3}$$

Lemma 5.7. *Let $x, y \in \mathbb{R}^d$. Let $\lambda, \mu \in \mathbb{R}^{d+1}$ be their barycentric coordinates with respect to $(\hat{S}, \mathrm{lex})$. Then, $x \preceq y$ is equivalent to $\lambda \preceq \mu$.*

The components of barycentric coordinates are counted from 0 to d, whereas the lex-order in (2.5) requires counting from 1 to $d+1$. In the following proof, 0-based counting is retained for λ, μ, and (2.5) is interpreted accordingly.

Proof of Lemma 5.7. The definition of barycentric coordinates and (3.1), (3.2) imply

$$x_i = 2\sum_{j=i}^{d} \lambda_j, \quad i \in \{1, \dots, d\}.$$

Elementary manipulations reveal

$$\lambda_0 = \frac{1}{2}(2 - x_1), \quad \lambda_i = \frac{1}{2}(x_i - x_{i+1}) \quad \text{for } i \in \{1, \dots, d-1\}, \quad \lambda_d = \frac{1}{2}x_d.$$

These relations also hold with x, λ replaced by y, μ, respectively.

Assume $x \preceq y$. If $x = y$; then $\lambda = \mu$. Now assume $x \ne y$, which implies that $x_i \le y_i$ in (2.5) is a strict inequality. If there holds $i = d$ in (2.5), then one gets $\lambda_d < \mu_d$, which implies $\lambda \preceq \mu$. Otherwise, $i \in \{1, \dots, d-1\}$, and $x_k = y_k$ for all $k \in \{i+1, \dots, d\}$. Thus, $\lambda_k = \mu_k$ for all $k \in \{i+1, \dots, d\}$. From $x_i < y_i$, one gets

$$\lambda_i = \frac{1}{2}(x_i - x_{i+1}) < \frac{1}{2}(y_i - y_{i+1}) = \mu_i.$$

Therefore, $\lambda \preceq \mu$.

Assume $\lambda \preceq \mu$. If $\lambda = \mu$; then $x = y$. Now assume $\lambda \ne \mu$, which implies that $\lambda_i \le \mu_i$ in (2.5) is a strict inequality. If $i = d$ in (2.5), then one gets $x_d < y_d$, which implies $x \preceq y$. Otherwise, $i \in \{1, \dots, d-1\}$. The case $i = 0$ does not occur because of the assumption $\lambda \neq \mu$ and because the components of λ (and μ) add up to 1. One finds $x_k = y_k$ for all $k \in \{i+1, \dots, d\}$. Hence,

$$x_i = \lambda_i + \sum_{j=i+1}^{d} \lambda_j < \mu_i + \sum_{j=i+1}^{d} \mu_j = y_i.$$

Therefore, $x \preceq y$. □

Barycentric coordinates are invariant under isotonic invertible affine maps.

Lemma 5.8. *Suppose that $A\colon (S, e) \to (T, f)$ is an isotonic invertible affine map. There holds* $\operatorname{bary}(x, S, e) = \operatorname{bary}(Ax, T, f)$ *for all $x \in \operatorname{aff} S$.*

Proof. This follows immediately from (5.3) and $A \circ e_S = f_{A(S)}$; cf. Lemma 2.2. □

Lemma 5.9. *Let (S, e) be a consistently numbered d-simplex, let $F \subseteq S$ be a k-face of S, and let $x \in F$ be arbitrary. Suppose that $\bar\lambda = \operatorname{bary}(x, S, e)$ and $\lambda = \operatorname{bary}(x, F, e|_F)$ are barycentric coordinates of x. Then, λ is a sub-tuple of $\bar\lambda$. There holds*

$$\begin{gathered}\bar\lambda_{f(i)} = \lambda_i, i \in \{0, \dots, k\}, \quad \bar\lambda_i = 0, i \notin \operatorname{image} f,\\ \text{where } f = e_S^{-1} \circ e_F\colon \{0, \dots, k\} \to \{0, \dots, d\}.\end{gathered} \tag{5.4}$$

Proof. From the definition of f in (5.4), one gets $e_S\big(f(i)\big) = e_F(i)$, $i \in \{0, \dots, k\}$, which implies $x = \sum_{i=0}^{k} \lambda_i e_F(i) = \sum_{i=0}^{k} \lambda_i e_S\big(f(i)\big)$. The definition of $\bar\lambda$ in (5.4) is such that this equals $\sum_{i=0}^{d} \bar\lambda_i e_S(i)$. □

Let $\hat F$ be a k-face of $\hat S$, let $\hat S^k$ be the k-dimensional reference simplex, and let $\hat A$ be the isotonic, affine map

$$\hat A\colon (\hat S^k, \text{lex}) \to (\hat F, \text{lex}), \tag{5.5}$$

where the first lex-order is on $\mathbb{R}^k$ and the second is on $\mathbb{R}^d$.

Lemma 5.10. *For all $x, y \in \mathbb{R}^k$, there holds that $x \preceq y$ is equivalent to $\hat A x \preceq \hat A y$.*

Proof. Let $\lambda = \operatorname{bary}(x, \hat S^k, \text{lex})$, $\mu = \operatorname{bary}(y, \hat S^k, \text{lex})$ be the barycentric coordinates of x, y. Using Lemma 5.7 (on $\hat S^k$), $\hat x \preceq \hat y$ is equivalent to $\lambda \preceq \mu$. Due to Lemma 5.8, there holds $\lambda = \operatorname{bary}(\hat A x, \hat F, \text{lex})$, $\mu = \operatorname{bary}(\hat A y, \hat F, \text{lex})$. Let $\bar\lambda = \operatorname{bary}(\hat A x, \hat S, \text{lex})$, $\bar\mu = \operatorname{bary}(\hat A y, \hat S, \text{lex})$. As $\hat F$ is a face of $\hat S$, λ and $\bar\lambda$ satisfy (5.4) (where e is the enumeration induced by the lex-order on $\hat S$). Likewise, μ and $\bar\mu$ satisfy (5.4). The mapping f in (5.4) is strictly isotonic because $(\hat S, \text{lex})$ is consistently numbered; cf. (2.4). From this and (2.5), one gets $\lambda \preceq \mu \iff \bar\lambda \preceq \bar\mu$. Owing to Lemma 5.7, $\bar\lambda \preceq \bar\mu$ is equivalent to $\hat A x \preceq \hat A y$. □

Recall the tuple $\hat w$ defined in (3.5). Let $\tilde w$ be the same tuple on the k-dimensional reference simplex $\hat S^k$ (instead of the d-dimensional reference simplex $\hat S = \hat S^d$).

Lemma 5.11. *The tuple $(\hat A \tilde w^j)_j$ is a sub-tuple of $\hat w$.*

Proof. Let $y \in \mathbb{R}^d$ be an element of the tuple $\hat w$ in (3.5). Slightly abusing notation, this is written as $y \in \hat w$ below. Let $\bar\lambda = \operatorname{bary}(y, \hat S, \text{lex})$. The point y is of the form $\hat v^i + \hat v^{d+1-j}$ with $i \in \{0, \dots, d\}$, $j \in \{1, \dots, d+1-i\}$. If $j = d+1-i$, then y is

$$\begin{tikzcd}
\hat{F} \subseteq \hat{S} \arrow[r, "A_S"] & S \supseteq F \\
\hat{S}^k \arrow[u, "\hat{A}"] \arrow[ur, "A_F"'] &
\end{tikzcd}$$

FIGURE 4. The simplices and the affine isotonic maps for Lemma 5.12.

a vertex of $\hat{S}$. Otherwise, y is the barycenter of the edge with the vertices $2\hat{v}^i$ and $2\hat{v}^{d+1-j}$. In both cases, $\bar{\lambda}$ is given by

$$\bar{\lambda} = \frac{1}{2}(\delta^i + \delta^{d+1-j}) \in \mathbb{R}^{d+1}. \tag{5.6}$$

For the auxiliary tuples $\hat{w}_i^{\text{aux}}$ in (3.5), one has the following characterization:

$$y \in \hat{w}_i^{\text{aux}} \quad \Leftrightarrow \quad \Big(\bar{\lambda}_i \in \big\{\tfrac{1}{2}, 1\big\}, \quad \bar{\lambda}_j = 0 \quad \text{for all } 0 \le j < i, \\ \bar{\lambda}_j \in \{0, \tfrac{1}{2}\} \quad \text{for all } i < j \le d\Big). \tag{5.7}$$

The implication "$\Rightarrow$" follows from (5.6) and $j \le d+1-i$. To prove the implication "$\Leftarrow$", two cases are considered. If $\bar{\lambda}_i = 1$, then all other barycentric coordinates are zero. Thus, y is the vertex in $\hat{w}_i^{\text{aux}}$ (given by $j = d+1-i$). If $\bar{\lambda}_i = \frac{1}{2}$, then there is precisely one index k, $i < k \le d$, with $\bar{\lambda}_k = \frac{1}{2}$. Writing $k = d+1-l$ with $1 \le l < d+1-i$, one gets $y \in \hat{w}_i^{\text{aux}}$.

Let $x \in \tilde{w}$ be a point with the barycentric coordinates $\lambda = \text{bary}(x, \hat{S}^k, \text{lex})$. One obtains analogues of (5.6) and (5.7) for x and λ by replacing d with k and using indices $l \in \{0, \dots, k\}$, $m \in \{1, \dots, k+1-l\}$ instead of i, j. Let $y = \hat{A}x$. Due to Lemmas 5.8 and 5.9, λ is a sub-tuple of $\bar{\lambda}$ as in (5.4). Hence, by (5.7), there holds $y \in \hat{w}_i^{\text{aux}}$ with $i = f(l)$ and f as in (5.4). One concludes that

$$\big(\hat{A}\tilde{w}_l^{\text{aux},m}\big)_{m=1}^{k+1-l} \text{ enumerates a subset of } \Big\{\hat{w}_i^{\text{aux},j} \;\Big|\; j \in \{1, \dots, d+1-i\}\Big\}. \tag{5.8}$$

After these preliminaries, consider the points $x = \tilde{w}^\alpha$ and $\xi = \tilde{w}^\beta$ with indices $\alpha < \beta$. If x, ξ are in different auxiliary tuples $\tilde{w}_l^{\text{aux}}$, $\tilde{w}_m^{\text{aux}}$, respectively, then $l < m$. As the mapping f of the indices is strictly isotonic, one has $\hat{A}x \in \hat{w}_i^{\text{aux}}$ and $\hat{A}\xi \in \hat{w}_j^{\text{aux}}$ for some indices $i < j$. That is, the mapped points occur in the same order as x and ξ. Otherwise, x and ξ are in the same auxiliary tuple $\tilde{w}_l^{\text{aux}}$. From (5.8), one gets $\hat{A}x, \hat{A}\xi \in \hat{w}_i^{\text{aux}}$. Observing (3.5), (5.6), and (5.7), one finds that the tuples $\tilde{w}_l^{\text{aux}}$ and $\hat{w}_i^{\text{aux}}$ both enumerate their points from large to small with respect to the lex-order. Hence, there holds $\xi \preceq x$. Using Lemma 5.10, one obtains $\hat{A}\xi \preceq \hat{A}x$. Therefore, $\hat{A}x$ is enumerated before $\hat{A}\xi$ in $\hat{w}_i^{\text{aux}}$. □

Let (S, e) be a consistently numbered d-simplex and let $F \subseteq S$ be a k-face. Let $R_S \subseteq \mathcal{F}(S)$ be a refinement pattern on S. Suppose that

$$A_F \colon \mathbb{R}^k \to \text{aff}\, F, \quad (\hat{S}^k, \text{lex}) \to (F, e|_{\mathcal{F}(F)}),$$
$$A_S \colon \mathbb{R}^d \to \text{aff}\, S, \qquad (\hat{S}, \text{lex}) \to (S, e).$$

Recall $\hat{A}$, $\hat{F}$ from (5.5), and suppose that A_S maps $\hat{F} \to F$. The setup is shown in Figure 4.

Lemma 5.12. *Property* (3.8) *is satisfied.*

Proof. The assertions in (3.8) on the triangulations and on the enumerations are considered separately.

From $A_S(\hat{F}) = F$ and Lemma 2.2, one gets $A_F = A_S \circ \hat{A}$. Let $\hat{R} \subseteq \mathcal{F}(\hat{S})$ be the refinement pattern on $\hat{S}$ with $\{A_S(\hat{G}) \mid \hat{G} \in \hat{R}\} = R_S$. Then, greentri$(\hat{S}, \mathrm{lex}, \hat{R})$ is the placing triangulation pla(w) with the sub-tuple w of $\hat{w}$ given in Definition 3.11. By Lemma 5.3, there holds greentri$(\hat{S}, \mathrm{lex}, \hat{R})|_{\hat{F}} = \mathrm{pla}\big((w^{i_j})_j\big)$, where $(w^{i_j})_j$ is the sub-tuple of w containing all points in $\hat{F}$. Using Lemma 5.11, this can be written as $(w^{i_j})_j = (\hat{A}\tilde{w}^{i_j})_j$. The tuple $(\tilde{w}^{i_j})_j \subset \mathbb{R}^k$ contains the vertices and edge barycenters (of $\hat{S}^k$) for the refinement pattern $\hat{R}_F \subseteq \mathcal{F}(\hat{S}^k)$ with $\{\hat{A}(\hat{G}_F) \mid \hat{G}_F \in \hat{R}_F\} = \mathrm{cl}\,\hat{R} \cap \mathcal{F}(\hat{F})$. The appearance of $\mathrm{cl}\,\hat{R}$ instead of $\hat{R}$ in the right-hand side is due to the definition of w in Definition 3.11. Owing to Lemma 5.2,

$$\begin{aligned}\mathrm{greentri}(\hat{S}, \mathrm{lex}, \hat{R})|_{\hat{F}} &= \mathrm{pla}\big((w^{i_j})_j\big) \\ &= \hat{A}\big(\mathrm{pla}((\tilde{w}^{i_j})_j)\big) = \mathrm{greentri}(\hat{F}, \mathrm{lex}, \mathrm{cl}\,\hat{R} \cap \mathcal{F}(\hat{F})).\end{aligned}$$

Mapping the preceding chain of equations with A_S, one obtains greentri$(S, e, R_S)|_F$ $=$ greentri$(F, e|_{\mathcal{F}(F)}, \mathrm{cl}\,R_S \cap \mathcal{F}(F))$.

In Definition 3.11, the enumeration greennum(S, e, R_S) is defined as the induced enumeration $A_S\,\mathrm{lex}$, where lex denotes the enumeration of greentri$(\hat{S}, \mathrm{lex}, \hat{R})$ defined by the lex-order on $\mathbb{R}^d$. This means that, given $x = A_S\hat{x} \in F$ and $y = A_S\hat{y} \in F$, x is enumerated before y if and only if $\hat{x} \preceq \hat{y}$ for the lex-order on $\mathbb{R}^d$. Likewise, the enumeration greennum$\big(F, e|_{\mathcal{F}(F)}, \mathrm{cl}\,R_S \cap \mathcal{F}(F)\big)$ is defined as the induced enumeration under A_F, where the lex-order is now on $\mathbb{R}^k$. Thus, for $x = A_F\hat{x}^k$ and $y = A_F\hat{y}^k$, x is enumerated before y if and only if $\hat{x}^k \preceq \hat{y}^k$ for the lex-order on $\mathbb{R}^k$. Using $A_F = A_S \circ \hat{A}$, one finds $\hat{x} = \hat{A}\hat{x}^k$ and $\hat{y} = \hat{A}\hat{y}^k$. Lemma 5.10 asserts that $\hat{x}^k \preceq \hat{y}^k$ if and only if $\hat{x} \preceq \hat{y}$. Hence, the enumerations of $\mathcal{F}(F)$ are identical. □

Remark 5.13. The green refinements are defined on the reference simplex and then mapped to specific simplices. This is convenient in an implementation to precompute results or to take advantage of the fact that only integer arithmetic is required to compute the placing triangulation on $\hat{S}$. However, the proof of Lemma 5.12 shows that the green refinement of a pattern R_S can be computed as the placing triangulation on S. Moreover, the enumeration of the refinement is given by the lex-order of the barycentric coordinates with respect to (S, e).

Property (3.8) simplifies the proof of (3.9) and (3.10) as follows.

Lemma 5.14. *Suppose that, for all consistently numbered simplices* (S, e)*, there hold*

$$\mathrm{greenref}\big(S, e, \mathcal{F}(S)\big) = \mathrm{redref}(S, e), \tag{5.9}$$

$$\mathrm{greenref}(S, e, R_S) = (\mathcal{F}(S), e) \quad \textit{for all } R_S \subseteq \mathcal{F}_0(S). \tag{5.10}$$

Then, (3.9) *and* (3.10) *are satisfied.*

Proof. Let (S, e) and R_S be as in Theorem 3.15. To prove (3.9), let $F \in \mathrm{cl}\,R_S$ be arbitrary. Applying (5.9) to the consistently numbered simplex $(F, \tilde{e})$, $\tilde{e} = e_{\mathcal{F}(F)}$, yields redref$(F, \tilde{e})$ = greenref$(F, \tilde{e}, \mathcal{F}(F))$. As $\mathrm{cl}\,R_S$ is face-complete, cf. (2.3), $F \in \mathrm{cl}\,R_S$ implies $\mathcal{F}(F) = \mathrm{cl}\,R_S \cap \mathcal{F}(F)$. Due to Lemma 5.12, property (3.8) holds.

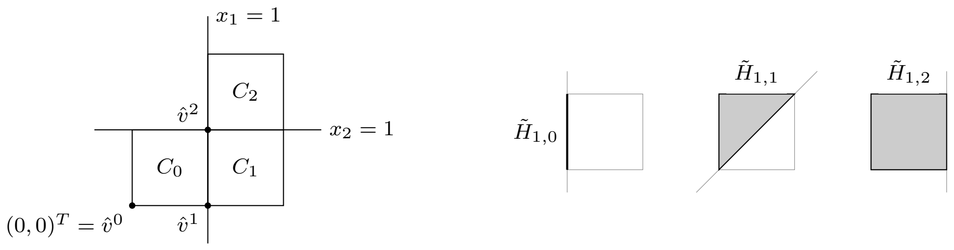

FIGURE 5. Left: The cubes C_i and hyper-planes $\{x_i = 1\}$ for $d = 2$. Right: The polytopes $\tilde{H}_{1,j} \subseteq C_1$, $j \in \{0, 1, 2\}$ for $d = 2$.

One obtains $\operatorname{greenref}\big(F, \tilde{e}, \mathcal{F}(F)\big) = \operatorname{greenref}(S, e, R_S)|_F$. Altogether, one finds

$$\operatorname{redref}(F, \tilde{e}) = \operatorname{greenref}\big(F, \tilde{e}, \operatorname{cl} R_S \cap \mathcal{F}(F)\big) = \operatorname{greenref}(S, e, R_S)|_F.$$

As $F \in \operatorname{cl} R_S$ was arbitrary, this proves (3.9).

To prove property (3.10), let $F \in U_S$ (cf. (3.7)) be arbitrary. From this, one gets $R_F := \operatorname{cl} R_S \cap \mathcal{F}(F) \subseteq \mathcal{F}_0(F)$. Applying (5.10) to the consistently numbered simplex $(F, \tilde{e})$, yields $(\mathcal{F}(F), \tilde{e}) = \operatorname{greenref}(F, \tilde{e}, R_F)$. Using (3.8), one gets

$$(\mathcal{F}(F), \tilde{e}) = \operatorname{greenref}(F, \tilde{e}, R_F) = \operatorname{greenref}(S, e, R_S)|_F,$$

which implies that (3.10) holds. □

5.3. **Consistency between the red and the green refinement.** Thanks to Lemma 5.14, the proof of (3.9) is reduced to proving (5.9). The basic idea of the proof of (5.9) is to carefully trace the visible parts of the tentative polytopes and the current point of $\hat{w}$ in the construction of $\operatorname{pla}(\hat{w})$. For this, $\hat{S}$ is covered with cubes C_i; each cube is partitioned with half-spaces into polytopes $\tilde{H}_{i,j}$. The $\tilde{H}_{i,j}$ are nested into each other; the visible parts from specific points of $\hat{w}$ are characterized (Lemmas 5.16 and 5.17; this is the most technical part of the paper). Furthermore, the $\tilde{H}_{i,j}$ are compatible with the red refinement in the sense that they are triangulated by subtriangulations of $\operatorname{redtri}(\hat{S}, \operatorname{lex})$; cf. Lemmas 5.18 and 5.19. These facts are combined in an inductive proof in which $\operatorname{pla}(\hat{w})$ is compared with $\operatorname{redtri}(\hat{S}, \operatorname{lex})$ (Lemma 5.20 and Theorem 5.21).

The facet representation of $\hat{S}$ and S_π, $\pi \in \operatorname{Sym}(d)$, in (3.2) and (3.3) is given by

$$\hat{S} = \left\{x \in \mathbb{R}^d \;\middle|\; 2 \geq x_1 \geq x_2 \geq \cdots \geq x_d \geq 0\right\}, \tag{5.11}$$

$$S_\pi = \left\{x \in \mathbb{R}^d \;\middle|\; 1 \geq x_{\pi 1} \geq x_{\pi 2} \geq \cdots \geq x_{\pi d} \geq 0\right\}, \tag{5.12}$$

respectively. The shorthand notation $\{x_i = 1\}$ is used for the hyperplane $\{x \in \mathbb{R}^d \mid x_i = 1\}$, $i \in \{1, \ldots, d\}$. The cube C_i and the union U_i are given by

$$C_i = \hat{v}^i + [0,1]^d, \quad i \in \{0, \ldots, d\}, \tag{5.13}$$

$$U_i = \cup \{C_j \mid 0 \leq j < i\}, \quad i \in \{0, \ldots, d+1\}; \tag{5.14}$$

cf. Figure 5. For C_i, the facet representation is

$$C_i = \left\{x \in \mathbb{R}^d \;\middle|\; 2 \geq x_k \geq 1 \text{ for } 1 \leq k \leq i, 1 \geq x_k \geq 0 \text{ for } i < k \leq d\right\}. \tag{5.15}$$

For $i \in \{0, \ldots, d\}$, let

$$\hat{W}_i = \left\{x \in (\hat{S} \cap C_i) - U_i \;\middle|\; x \text{ is an integral point}\right\}. \tag{5.16}$$

The vertices and edge barycenters of the simplex $\hat{S}$ have integer coordinates; cf. (3.1), (3.2). Such points are called integral points of $\hat{S}$. The next lemma classifies them.

Lemma 5.15. *All integral points $x \in \hat{S}$ have the form $x = \hat{v}^i + \hat{v}^j$ for some $i, j \in \{0, \ldots, d\}$. The set $\hat{W}_i$ contains precisely the $d + 1 - i$ points $\hat{v}^i + \hat{v}^{i+j}$, $j \in \{0, \ldots, d-i\}$, $i \in \{0, \ldots, d\}$.*

A consequence of Lemma 5.15 is that the sets $\hat{W}_i$ partition the set of integral points of $\hat{S}$. The auxiliary tuple $\hat{w}_i^{\mathrm{aux}}$ in (3.5) enumerates $\hat{W}_i$. Thus, Lemma 5.15 proves that $\hat{w}$ enumerates the vertices and edge barycenters of $\hat{S}$.

Proof of Lemma 5.15. Due to (5.11), the components of any integral point $x \in \hat{S}$ are a monotonically decreasing sequence which can only take values in $\{2, 1, 0\}$. If x is a constant sequence, then $x \in \{2\hat{v}^d, \hat{v}^d, \hat{v}^0\}$. If x contains precisely two of the values from $\{2, 1, 0\}$, let i be the biggest index such that x_i is the greater of the two values. Then, one has $x \in \{\hat{v}^i, 2\hat{v}^i, \hat{v}^i + \hat{v}^d\}$ depending on whether the third value is 2, 1, or 0. Finally, if all three values are present in x, then $x = \hat{v}^i + \hat{v}^j$, where i is the biggest index with $x_i = 1$ and j is the biggest index with $x_j = 2$. This proves the first assertion of the lemma.

The set U_0 is empty. Thus $\hat{W}_0$ contains precisely the $d+1$ vertices of S_{id}. Suppose $i \in \{1, \ldots, d\}$. The integral points in C_i are of the form $x = \hat{v}^i + (u_k)_{k=1}^d$ with each $u_k \in \{0, 1\}$. Due to (5.13) and (5.14), the integral points in U_i satisfy $x_i \le 1$, and the integral points in C_i satisfy $x_i \ge 1$. Hence, all integral points in the intersection $C_i \cap U_i$ are in $\{x_i = 1\}$. Therefore, all $x \in \hat{W}_i$ have $x_i = 2$; see (5.16). As $x \in \hat{S}$, (5.11) implies $x_k = 2$, $1 \le k \le i$. For $k > i$, there holds $\hat{v}_k^i = 0$, which implies $x_k \le 1$. The monotonicity of x_k as function of k, cf. (5.11), implies that there can be at most one transition from 1 to 0 in $(x_k)_{k=i+1}^d$. All integral points satisfying this condition are given by x^j, $j \in \{0, \ldots, d-i\}$, with components

$$x_k^j = \begin{cases} 2, & 1 \le k \le i, \\ 1, & i+1 \le k \le i+j, \\ 0, & i+j+1 \le k \le d. \end{cases}$$

□

For the remainder of Section 5.3, an arbitrary $i \in \{1, \ldots, d\}$ is fixed. (The easy case $i = 0$ is treated separately in the proof of Theorem 5.21.) To study how $\hat{S} \cap C_i$ can be constructed in a step by step fashion, the family of $d + 2 - i$ half-spaces

$$\begin{aligned} H_{i,0} = \hat{v}^i + \left\{x \in \mathbb{R}^d \;\middle|\; x_i \le 0\right\}, \quad H_{i,d+1-i} = \hat{v}^i + \left\{x \in \mathbb{R}^d \;\middle|\; x_i \le 1\right\}, \\ H_{i,j} = \hat{v}^i + \left\{x \in \mathbb{R}^d \;\middle|\; x_i \le x_{d+1-j}\right\}, \quad 1 \le j \le d-i, \end{aligned} \tag{5.17}$$

is needed. An equivalent representation of these half-spaces is

$$\begin{aligned} H_{i,0} = \left\{x \in \mathbb{R}^d \;\middle|\; x_i \le 1\right\}, \quad H_{i,d+1-i} = \left\{x \in \mathbb{R}^d \;\middle|\; x_i \le 2\right\}, \\ H_{i,j} = \left\{x \in \mathbb{R}^d \;\middle|\; x_i - 1 \le x_{d+1-j}\right\}, \quad 1 \le j \le d-i. \end{aligned} \tag{5.18}$$

For any $H_{i,j}$, let

$$\tilde{H}_{i,j} = H_{i,j} \cap \hat{S} \cap C_i, \quad \mathcal{C}\tilde{H}_{i,j} = \overline{\tilde{H}_{i,d+1-i} - \tilde{H}_{i,j}}; \tag{5.19}$$

see Figure 5. The set $\mathcal{C}\tilde{H}_{i,j}$ is the closure of the complement of $\tilde{H}_{i,j}$ in $\tilde{H}_{i,d+1-i}$.

Lemma 5.16. *The sets $\tilde{H}_{i,j}$ are an ascending chain with respect to set inclusion; that is, $\tilde{H}_{i,j} \subseteq \tilde{H}_{i,k}$ for all $0 \le j \le k \le d+1-i$. Furthermore, there hold*

$$\tilde{H}_{i,0} = \{x_i = 1\} \cap \hat{S} \cap C_i = \{x_i = 1\} \cap \hat{S}, \quad \tilde{H}_{i,d+1-i} = \hat{S} \cap C_i, \quad \textit{and} \tag{5.20}$$

$$\hat{S} \cap U_i = \hat{S} \cap H_{i,0}. \tag{5.21}$$

Proof. Comparing the facet representation (5.15) of C_i with (5.18) yields

$$C_i \subset H_{i,d+1-i}, \quad C_i \cap H_{i,0} \subset \{x_i = 1\}, \tag{5.22}$$

$$C_j \subset H_{i,0} \quad \Leftrightarrow \quad 0 \le j < i. \tag{5.23}$$

First, (5.20) is proved. From (5.19) and the first part of (5.22), one gets $\tilde{H}_{i,d+1-i} = \hat{S} \cap C_i$, which is the rightmost equation in (5.20). The second part of (5.22) implies $\tilde{H}_{i,0} = \{x_i = 1\} \cap \hat{S} \cap C_i$. From this, one concludes $\tilde{H}_{i,0} \subseteq \{x_i = 1\} \cap \hat{S}$. For the reverse inclusion, consider an arbitrary $x \in \{x_i = 1\} \cap \hat{S}$. Then, (5.11) implies $2 \ge x_j \ge 1$, $1 \le j \le i$, and $1 \ge x_j \ge 0$, $i < j \le d$. From (5.15), one obtains $x \in C_i$ and, consequently, $x \in \tilde{H}_{i,0}$.

To prove (5.21), one uses (5.23) and (5.14) to obtain $U_i \subseteq H_{i,0}$. Therefore, one has $\hat{S} \cap U_i \subseteq \hat{S} \cap H_{i,0}$. For the reverse inclusion, let $x \in \hat{S} \cap H_{i,0}$ be arbitrary. If $x_1 \le 1$, then (5.11) and (5.15) yield $x \in C_0 \subseteq U_i$. If $x_1 > 1$, then let $j = \max\{k \in \{1, \ldots, d\} \mid x_k > 1\}$. Due to (5.11), there hold $2 \ge x_k > 1$, $1 \le k \le j$, and $1 \ge x_k \ge 0$, $j < k \le d$. This implies $x \in C_j$. As $x \in H_{i,0}$, there holds $x_i \le 1$, hence, $j < i$. From (5.14), one obtains $x \in \hat{S} \cap U_i$.

The inclusions $\tilde{H}_{i,j} \subseteq \tilde{H}_{i,k}$, $0 \le j \le k \le d+1-i$, are considered. The cases with $j = k$ are trivial. The cases with $0 \le j < k = d+1-i$ hold because (5.19) implies $\tilde{H}_{i,j} \subseteq \hat{S} \cap C_i$, $0 \le j < d+1-i$, and (5.20) yields $\hat{S} \cap C_i = \tilde{H}_{i,d+1-i}$.

Next, the cases with $0 = j < k \le d-i$ are considered. Let $x \in \tilde{H}_{i,0}$ be arbitrary. From (5.20), one gets $x_i - 1 = 0$. Furthermore, using $x \in C_i$, one gets $x_{d+1-k} \ge 0$, $1 \le k \le d-i$. This implies $x_i - 1 \le x_{d+1-k}$, $1 \le k \le d-i$. Using (5.18) and (5.19), one concludes $x \in \tilde{H}_{i,k}$ for all $1 \le k \le d-i$.

It remains to treat the cases with $1 \le j < k \le d-i$. Let $x \in \tilde{H}_{i,j}$ be arbitrary, which implies $x_i - 1 \le x_{d+1-j}$; cf. (5.18). As $x \in \hat{S}$ and $k \ge j$, one gets from (5.11) that there holds $x_{d+1-k} \ge x_{d+1-j}$. Using (5.18) and (5.19), one concludes $x \in \tilde{H}_{i,k}$. □

Lemma 5.17. *The point $\hat{v}^i + \hat{v}^{d-j} \in \hat{W}_i$ is the only integral point in $\tilde{H}_{i,j+1} - \tilde{H}_{i,j}$, $0 \le j \le d-i$.*

Proof. Fix an arbitrary integer j with $0 \le j \le d-i$. Due to Lemma 5.16, there holds $\tilde{H}_{i,j+1} - \tilde{H}_{i,j} \subseteq \hat{S} \cap C_i$. Let $x \in \hat{S} \cap C_i$ be an arbitrary integral point. Owing to Lemma 5.15, x is of the form $x = \hat{v}^m + \hat{v}^k$ for some $m \in \{0, \ldots, d\}$, $k \in \{m, \ldots, d\}$. There holds

$$m \ne i \quad \text{implies } x \notin \tilde{H}_{i,j+1} - \tilde{H}_{i,j}. \tag{5.24}$$

This can be seen as follows. If $i = d$, there is no $d \ge m > i$. Otherwise, $d \ge m > i$ and (3.1) imply that $x_{i+1} = 2$. Due to (5.15), this implies $x \notin C_i$, and therefore $x \notin \tilde{H}_{i,j+1} - \tilde{H}_{i,j}$. In the case $m < i$, one obtains $x_i \le 1$ from (3.1). Using (5.18) and Lemma 5.16, one concludes that $x \in \tilde{H}_{i,0} \subseteq \tilde{H}_{i,j}$. This implies $x \notin \tilde{H}_{i,j+1} - \tilde{H}_{i,j}$.

Thanks to (5.24), one only has to examine which of the points $x = \hat{v}^i + y \in \hat{W}_i$, $y = \hat{v}^k$, $k \in \{i, \dots, d\}$, are in $\tilde{H}_{i,j+1} - \tilde{H}_{i,j}$. Next, it is shown that, for $0 \leq l \leq d+1-i$,

$$x \in \tilde{H}_{i,l} \quad \text{is equivalent to } l \geq d+1-k. \tag{5.25}$$

To prove the lemma using (5.25), one computes

$$x \in \tilde{H}_{i,j+1} - \tilde{H}_{i,j} \quad \Leftrightarrow \quad j+1 \geq d+1-k > j \quad \Leftrightarrow \quad d+1-j > k \geq d-j.$$

It remains to establish (5.25). From $x \in \hat{S} \cap C_i$, (5.19), and (5.17), one infers that $x \in \tilde{H}_{i,l}$ is equivalent to

$$(y_i \leq 0 \wedge l = 0) \;\vee\; (y_i \leq 1 \wedge l = d+1-i) \;\vee\; (y_i \leq y_{d+1-l} \wedge 1 \leq l \leq d-i).$$

From (3.1) and $i \leq k \leq d$, one gets $y_i = 1$. This simplifies the condition to

$$l = d+1-i \;\vee\; (1 \leq y_{d+1-l} \wedge 1 \leq l \leq d-i).$$

Using (3.1) a second time yields that $1 \leq y_{d+1-l}$ can be replaced by $k \geq d+1-l$ in the previous condition. Writing this as $l \geq d+1-k$, one gets

$$l = d+1-i \;\vee\; (l \geq d+1-k \wedge 1 \leq l \leq d-i),$$

which is equivalent to $l \geq d+1-k$. This proves (5.25). □

Lemma 5.18. *For all $1 \leq j \leq d+1-i$ and all simplices $S = \hat{v}^i + S_\pi$, $\pi \in Sh(i, d-i)$, there holds: If $\pi^{-1}i \geq \pi^{-1}(d+1-j)$, then $S \subseteq \tilde{H}_{i,j}$, or else $S \subseteq \mathcal{C}\tilde{H}_{i,j}$.*

Proof. First, consider the cases $i \in \{1, \dots, d-1\}$, $1 \leq j \leq d-i$. These conditions on i, j imply $i < d+1-j \leq d$. As any shuffle $\pi \in \mathrm{Sh}(i, d-i)$ is a bijective map, there holds either $\pi^{-1}i > \pi^{-1}(d+1-j)$ or $\pi^{-1}i < \pi^{-1}(d+1-j)$. Consider the case $\pi^{-1}i > \pi^{-1}(d+1-j)$. Observing that $\pi^{-1}i$ is the position at which i occurs in the tuple $(\pi 1, \pi 2, \dots, \pi d)$ and using (5.12), one obtains $S_\pi \subseteq \{x \in \mathbb{R}^d \mid x_i \leq x_{d+1-j}\}$. By (5.17), there holds $S = \hat{v}^i + S_\pi \subseteq H_{i,j}$. From Definition 3.1, one infers $S \subseteq \hat{S} \cap C_i$.

Now, consider $\pi^{-1}i < \pi^{-1}(d+1-j)$. One gets $S \subseteq \hat{v}^i + \{x \in \mathbb{R}^d \mid x_i \geq x_{d+1-j}\}$. This is the closure of $\mathbb{R}^d - H_{i,j}$. As $S \subseteq \hat{S} \cap C_i$, one obtains $S \subseteq \mathcal{C}\tilde{H}_{i,j}$.

The remaining cases are $i \in \{1, \dots, d\}$, $j = d+1-i$, which correspond to $\pi^{-1}i = \pi^{-1}(d+1-j)$. Due to (5.20), one has $\tilde{H}_{i,j} = \hat{S} \cap C_i$ in these cases. From $S \subseteq C_i$, one obtains $S \subseteq \tilde{H}_{i,j}$ for all $\pi \in \mathrm{Sh}(i, d-i)$. □

The d-simplices S_π, cf. (3.3), satisfy $S_\pi \subseteq [0,1]^d$. Using this and Definition 3.1, one sees that there is a subtriangulation $\mathcal{T}_{i,d+1-i}$ of the red refinement $\mathrm{redtri}(\hat{S}, \mathrm{lex})$ which triangulates $\hat{S} \cap C_i$. The set $\{x_i = 1\} \cap \hat{S} \subset \partial(\hat{S} \cap C_i)$ is a facet of $\hat{S} \cap C_i$. Hence, there is a ($(d-1)$-dimensional) subtriangulation $\mathcal{T}_{i,0}$ of $\mathrm{redtri}(\hat{S}, \mathrm{lex})$ which triangulates $\{x_i = 1\} \cap \hat{S}$. Due to (5.20), there hold $\mathrm{set}\, \mathcal{T}_{i,d+1-i} = \hat{S} \cap C_i = \tilde{H}_{i,d+1-i}$ and $\mathrm{set}\, \mathcal{T}_{i,0} = \{x_i = 1\} \cap \hat{S} = \tilde{H}_{i,0}$. More generally, let

$$\mathcal{T}_{i,j} = \left\{ S \in \mathrm{redtri}(\hat{S}, \mathrm{lex}) \;\middle|\; S \subseteq \tilde{H}_{i,j} \right\}, \quad 0 \leq j \leq d+1-i. \tag{5.26}$$

Lemma 5.19. *For $0 \leq j \leq d+1-i$, $\mathcal{T}_{i,j}$ is a subtriangulation of* $\mathrm{redtri}(\hat{S}, \mathrm{lex})$, *and there holds* $\mathrm{set}\, \mathcal{T}_{i,j} = \tilde{H}_{i,j}$.

Proof. The two cases $j \in \{0, d+1-i\}$ are treated in the paragraph before (5.26).

Let $0 < j < d+1-i$. Due to (5.26), there holds $\mathcal{T}_{i,j} \subseteq \operatorname{redtri}(\hat{S}, \mathrm{lex})$, and the latter has the intersection property. Consequently, $\mathcal{T}_{i,j}$ has the intersection property; cf. (2.2). The set $\tilde{H}_{i,j}$ is (topologically) closed, which implies that $\mathcal{T}_{i,j}$ is face-complete. Hence, $\mathcal{T}_{i,j}$ is a triangulation.

From (5.26), one gets immediately that $\operatorname{set} \mathcal{T}_{i,j} \subseteq \tilde{H}_{i,j}$. Let $x \in (\tilde{H}_{i,j})^{\circ}$ be an arbitrary point in the interior of $\tilde{H}_{i,j}$. As $\operatorname{set} \mathcal{T}_{i,d+1-i} = \hat{S} \cap C_i$, there is a d-simplex $S \in \mathcal{T}_{i,d+1-i}$ with $x \in S$. As S intersects $(\tilde{H}_{i,j})^{\circ}$, Lemma 5.18 implies $S \subseteq \tilde{H}_{i,j}$. This yields $S \in \mathcal{T}_{i,j}$. Thus, $(\tilde{H}_{i,j})^{\circ} \subseteq \operatorname{set} \mathcal{T}_{i,j}$; taking the closure completes the proof. □

Lemma 5.20. *Placing the vertices of the tuple $\hat{w}_i^{aux}$ from* (3.5) *successively in $\mathcal{T}_{i,0}$ yields $\mathcal{T}_{i,d+1-i}$.*

Proof. Let $\mathcal{S}_0 = \mathcal{T}_{i,0}$, and let $\mathcal{S}_j$ be the triangulation obtained from placing $\hat{w}_i^{\mathrm{aux},j} = \hat{v}^i + \hat{v}^{d+1-j}$ in $\mathcal{S}_{j-1}$, $1 \le j \le d+1-i$. One shows $\mathcal{T}_{i,j} = \mathcal{S}_j$ by induction on $0 \le j \le d+1-i$. The case $j = 0$ holds by definition.

Assume that $\mathcal{T}_{i,j} = \mathcal{S}_j$ for some $0 \le j \le d-i$. The goal is to show $\mathcal{T}_{i,j+1} = \mathcal{S}_{j+1}$. By Lemma 5.19, $\mathcal{T}_{i,k}$ triangulates $\tilde{H}_{i,k}$, $k \in \{j, j+1\}$. In particular, $\mathcal{T}_{i,k}$ is the restriction of the red refinement $\operatorname{redtri}(\hat{S}, \mathrm{lex})$ to $\tilde{H}_{i,k}$, $k \in \{j, j+1\}$. Due to Lemma 5.16, one has $\tilde{H}_{i,j} \subseteq \tilde{H}_{i,j+1}$. This implies

$$\mathcal{T}_{i,j} \subseteq \mathcal{T}_{i,j+1}. \tag{5.27}$$

Let $v = \hat{v}^i + \hat{v}^{d+1-(j+1)}$ be the vertex placed in $\mathcal{S}_j$ to obtain $\mathcal{S}_{j+1}$. The next step is to show that

$$\tilde{H}_{i,j+1} = \operatorname{conv}(\tilde{H}_{i,j} \cup \{v\}). \tag{5.28}$$

Let $S \in \mathcal{T}_{i,j+1}$ be a simplex. If $S \in \mathcal{T}_{i,j}$, then $S \subseteq \tilde{H}_{i,j}$. If $S \notin \mathcal{T}_{i,j}$, then at least one vertex w of S satisfies $w \notin \tilde{H}_{i,j}$. Due to Lemma 5.17, there is exactly one vertex in $\tilde{H}_{i,j+1} - \tilde{H}_{i,j}$, and this is v. It follows that $w = v$ and $S = \operatorname{pyr}(v, F)$ for some face $F \in \mathcal{T}_{i,j}$. This implies $S \subseteq \operatorname{conv}(\tilde{H}_{i,j} \cup \{v\})$, and therefore (5.28) holds.

From Lemma 5.1, one obtains that the unique triangulation satisfying (5.27) and (5.28) is $\operatorname{pla}(\mathcal{T}_{i,j}, v)$. Using the induction hypothesis and the definition of $\mathcal{S}_{j+1}$, one gets

$$\mathcal{T}_{i,j+1} = \operatorname{pla}(\mathcal{T}_{i,j}, v) = \operatorname{pla}(\mathcal{S}_j, v) = \mathcal{S}_{j+1}.$$

□

Theorem 5.21. *There holds* $\operatorname{pla}(\hat{w}) = \operatorname{redtri}(\hat{S}, \mathrm{lex})$.

Proof. Let $\mathcal{S}_0 = \operatorname{pla}(\hat{w}_0^{\mathrm{aux}})$ be the placing triangulation of the $d+1$ points in the tuple $\hat{w}_0^{\mathrm{aux}}$; cf. (3.5). Comparing this to (3.3) yields $\mathcal{S}_0 = \mathcal{F}(S_{\mathrm{id}})$. For $i \in \{1, \dots, d\}$, let $\mathcal{S}_i$ be the triangulation obtained by placing the points in $\hat{w}_i^{\mathrm{aux}}$ in $\mathcal{S}_{i-1}$. From (5.11) and (5.12), one gets $\hat{S} \cap C_0 = S_{\mathrm{id}}$. Hence, using Definition 3.1, the subtriangulation $\mathcal{F}(S_{\mathrm{id}}) =: \mathcal{T}_{0,d+1}$ of $\operatorname{redtri}(\hat{S}, \mathrm{lex})$ triangulates $\hat{S} \cap C_0$. There holds $\mathcal{S}_0 = \mathcal{F}(S_{\mathrm{id}}) = \mathcal{T}_{0,d+1}$. Below, it is shown that

$$\mathcal{S}_i = \bigcup_{k=0}^{i} \mathcal{T}_{k,d+1-k}, \quad i \in \{0, \dots, d\}. \tag{5.29}$$

Due to (5.20) and Lemma 5.19, the right-hand side is the subtriangulation of $\operatorname{redtri}(\hat{S}, \operatorname{lex})$ which triangulates $\hat{S} \cap (\bigcup_{k=0}^{i} C_k)$. The case $i = d$ is the assertion of the theorem because there holds $\mathcal{S}_d = \operatorname{pla}(\hat{w})$ (cf. (3.5) and Definition 3.10) and, furthermore, $\bigcup_{k=0}^{d} \mathcal{T}_{k,d+1-k} = \operatorname{redtri}(\hat{S}, \operatorname{lex})$.

Assertion (5.29) is proved by induction on i. In the base case $i = 0$, there holds $\mathcal{S}_0 = \mathcal{T}_{0,d+1}$. Suppose that $\mathcal{S}_{i-1} = \bigcup_{k=0}^{i-1} \mathcal{T}_{k,d+1-k}$ holds for some $i \in \{1, \ldots, d\}$. Recalling U_i from (5.14), one uses (5.21) to obtain

$$\operatorname{set} \bigcup_{k=0}^{i-1} \mathcal{T}_{k,d+1-k} = \hat{S} \cap U_i = \hat{S} \cap H_{i,0} =: P.$$

The boundary of the polytope P is given by

$$\partial P = (\{x_i = 1\} \cap \hat{S}) \cup (H_{i,0} \cap \partial \hat{S}) = \tilde{H}_{i,0} \cup (H_{i,0} \cap \partial \hat{S}),$$

where (5.20) was used to obtain the rightmost expression. Let x be an arbitrary point from the tuple $\hat{w}_i^{\mathrm{aux}}$. Using $x \in \partial \hat{S}$ and Definition 3.7, one finds that no facet F of P with $F \subseteq H_{i,0} \cap \partial \hat{S}$ is visible from x. As $x_i = 2$ (cf. (3.1) and (3.5)) and $P \subset H_{i,0}$, the facet $\tilde{H}_{i,0}$ of P is visible from x. Owing to the induction hypothesis, $\mathcal{S}_{i-1}$ is a subtriangulation of $\operatorname{redtri}(\hat{S}, \operatorname{lex})$; for the facet $\tilde{H}_{i,0}$, this implies $\mathcal{S}_{i-1}|_{\tilde{H}_{i,0}} = \mathcal{T}_{i,0}$. As only $\tilde{H}_{i,0}$ is visible from the points in $\hat{w}_i^{\mathrm{aux}}$, one can apply Lemma 5.20 to conclude

$$\mathcal{S}_i = \mathcal{S}_{i-1} \cup \mathcal{T}_{i,d+1-i} = \bigcup_{k=0}^{i} \mathcal{T}_{k,d+1-k},$$

where the second equality follows from the induction hypothesis. □

Corollary 5.22. *Property* (3.9) *is satisfied.*

Proof. Taking into account Lemma 5.14, it suffices to prove (5.9) for all consistently numbered simplices (S, e). As the refinement pattern in (5.9) is $R_S = \mathcal{F}(S)$, all vertices and edge barycenters of $\hat{S}$ are used in Definition 3.11. Consequently, one has $w = \hat{w}$ in (3.6). From Theorem 5.21, one obtains $\operatorname{greentri}\big(\hat{S}, \operatorname{lex}, \mathcal{F}(\hat{S})\big) = \operatorname{redtri}(\hat{S}, \operatorname{lex})$. This implies $\operatorname{greentri}(S, e, R_S) = \operatorname{redtri}(S, e)$ because both the green and the red refinement of S are defined by mapping the refinements of $\hat{S}$ to S (with the same isotonic affine map).

The enumeration $\operatorname{greennum}(S, e, R_S)$ is the mapping of the lex-enumeration (on $\hat{S}$) of the points in $\hat{w}$. Due to Lemma 3.2, this agrees with $\operatorname{rednum}(S, e)$. □

5.4. **Consistency between the green refinement and unrefined simplices.**

Lemma 5.23. *Property* (3.10) *is satisfied.*

Proof. Thanks to Lemma 5.14, one only has to prove (5.10) for all consistently numbered simplices (S, e) and all $R_S \subseteq \mathcal{F}_0(S)$. Due to the condition on R_S, no edge barycenters appear in the refinement pattern of S. Hence, the sub-tuple w of $\hat{w}$ in Definition 3.11 consists of the vertices of $\hat{S}$, and the placing triangulation in (3.6) is $\mathcal{F}(\hat{S})$. The triangulation of S is given by $A\big(\mathcal{F}(\hat{S})\big) = \mathcal{F}(S)$, where $A\colon \hat{S} \to S$ is the mapping from Definition 3.11. Furthermore, A maps the enumeration lex on $\hat{S}$ to the enumeration e given on S. □

Acknowledgment

The author thanks Arnold Reusken for reading the first version of this paper. His feedback helped to improve the presentation.

References

[1] R. E. Bank, A. H. Sherman, and A. Weiser, *Refinement algorithms and data structures for regular local mesh refinement*, Scientific computing (Montreal, Que., 1982), IMACS Trans. Sci. Comput., I, IMACS, New Brunswick, NJ, 1983, pp. 3–17. MR751598

[2] P. Bastian, K. Birken, K. Johannsen, S. Lang, N. Neuß, H. Rentz-Reichert, and C. Wieners, *UG – a flexible software toolbox for solving partial differential equations*, Computing and Visualization in Science **1** (1997), no. 1, 27–40, doi:10.1007/s007910050003.

[3] M. Behr, *Simplex space-time meshes in finite element simulations*, Internat. J. Numer. Methods Fluids **57** (2008), no. 9, 1421–1434, DOI 10.1002/fld.1796. MR2435099

[4] M. Bern and D. Eppstein, *Mesh generation and optimal triangulation*, Computing in Euclidean geometry, Lecture Notes Ser. Comput., vol. 1, World Sci. Publ., River Edge, NJ, 1992, pp. 23–90, DOI 10.1142/9789814355858_0002. MR1239190

[5] J. Bey, *Tetrahedral grid refinement* (English, with English and German summaries), Computing **55** (1995), no. 4, 355–378, DOI 10.1007/BF02238487. MR1370107

[6] J. Bey, *Finite-Volumen- und Mehrgitter-Verfahren für elliptische Randwertprobleme* (German, with German summary), Advances in Numerical Mathematics, B. G. Teubner, Stuttgart, 1998. MR1649911

[7] F. Bornemann, B. Erdmann, and R. Kornhuber, *Adaptive multilevel methods in three space dimensions*, Internat. J. Numer. Methods Engrg. **36** (1993), no. 18, 3187–3203, DOI 10.1002/nme.1620361808. MR1236370

[8] A. Brøndsted, *An Introduction to Convex Polytopes*, Graduate Texts in Mathematics, vol. 90, Springer-Verlag, New York-Berlin, 1983. MR683612

[9] E. Bänsch, *Local mesh refinement in* 2 *and* 3 *dimensions*, Impact Comput. Sci. Engrg. **3** (1991), no. 3, 181–191, DOI 10.1016/0899-8248(91)90006-G. MR1141298

[10] P. G. Ciarlet and J.-L. Lions (eds.), *Handbook of Numerical Analysis. Vol. II*, Handbook of Numerical Analysis, II, North-Holland, Amsterdam, 1991. Finite element methods. Part 1. MR1115235

[11] H. S. M. Coxeter, *Discrete groups generated by reflections*, Ann. of Math. (2) **35** (1934), no. 3, 588–621, DOI 10.2307/1968753. MR1503182

[12] J. A. De Loera, J. Rambau, and F. Santos, *Triangulations*, Algorithms and Computation in Mathematics, vol. 25, Springer-Verlag, Berlin, 2010. Structures for algorithms and applications. MR2743368

[13] H. Edelsbrunner and D. R. Grayson, *Edgewise subdivision of a simplex*, Discrete Comput. Geom. **24** (2000), no. 4, 707–719, DOI 10.1145/304893.304897. ACM Symposium on Computational Geometry (Miami, FL, 1999). MR1799608

[14] K. Eriksson and C. Johnson, *Adaptive finite element methods for parabolic problems. I. A linear model problem*, SIAM J. Numer. Anal. **28** (1991), no. 1, 43–77, DOI 10.1137/0728003. MR1083324

[15] IEEE Working Group for Floating-Point Arithmetic, *IEEE standard for floating-point arithmetic*, no. 754-2008, IEEE, New York, 2008, doi:10.1109/IEEESTD.2008.4610935.

[16] H. Freudenthal, *Simplizialzerlegungen von beschränkter Flachheit* (German), Ann. of Math. (2) **43** (1942), 580–582, DOI 10.2307/1968813. MR0007105

[17] E. Gawrilow and M. Joswig, *Polymake: A Framework for Analyzing Convex Polytopes*, Polytopes—combinatorics and computation (Oberwolfach, 1997), DMV Sem., vol. 29, Birkhäuser, Basel, 2000, pp. 43–73. MR1785292

[18] S. Groß, J. Peters, V. Reichelt, and A. Reusken, *The DROPS package for numerical simulations of incompressible flows using parallel adaptive multigrid techniques*, IGPM preprint 211, IGPM, RWTH Aachen University, 2002.

[19] B. Grünbaum, *Convex Polytopes*, With the cooperation of Victor Klee, M. A. Perles and G. C. Shephard. Pure and Applied Mathematics, Vol. 16, Interscience Publishers John Wiley & Sons, Inc., New York, 1967. MR0226496

[20] W. Hackbusch, *Multigrid Methods and Applications*, Springer Series in Computational Mathematics, vol. 4, Springer-Verlag, Berlin, 1985, doi:10.1007/978-3-662-02427-0.

[21] T. J. R. Hughes and G. M. Hulbert, *Space-time finite element methods for elastodynamics: formulations and error estimates*, Comput. Methods Appl. Mech. Engrg. **66** (1988), no. 3, 339–363, DOI 10.1016/0045-7825(88)90006-0. MR928689

[22] H. W. Kuhn, *Some combinatorial lemmas in topology*, IBM J. Res. Develop. **4** (1960), 518–524, DOI 10.1147/rd.45.0518. MR0124038

[23] A. Liu and B. Joe, *Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision*, Math. Comp. **65** (1996), no. 215, 1183–1200, DOI 10.1090/S0025-5718-96-00748-X. MR1348045

[24] J. M. Maubach, *Local bisection refinement for n-simplicial grids generated by reflection*, SIAM J. Sci. Comput. **16** (1995), no. 1, 210–227, DOI 10.1137/0916014. MR1311687

[25] K. Mehlhorn and P. Sanders, *Algorithms and Data Structures*, Springer-Verlag, Berlin, 2008. The basic toolbox. MR2444537

[26] J.-M. Mirebeau and A. Cohen, *Greedy bisection generates optimally adapted triangulations*, Math. Comp. **81** (2012), no. 278, 811–837, DOI 10.1090/S0025-5718-2011-02459-2. MR2869038

[27] W. F. Mitchell, *A comparison of adaptive refinement techniques for elliptic problems*, ACM Trans. Math. Software **15** (1989), no. 4, 326–347 (1990), DOI 10.1145/76909.76912. MR1062496

[28] D. Moore, *Subdividing simplices*, Graphics Gems III (David Kirk, ed.), Academic Press Professional, Inc., San Diego, CA, USA, 1992, pp. 244–249.

[29] J. E. Goodman and J. O'Rourke (eds.), *Handbook of Discrete and Computational Geometry*, CRC Press Series on Discrete Mathematics and its Applications, CRC Press, Boca Raton, FL, 1997. MR1730156

[30] The CGAL Project, *CGAL User and Reference Manual*, 4.9 ed., CGAL Editorial Board, 2016.

[31] M.-C. Rivara, *Design and data structure of fully adaptive, multigrid, finite-element software*, ACM Trans. Math. Software **10** (1984), no. 3, 242–264, DOI 10.1145/1271.1274. MR791990

[32] M.-C. Rivara and G. Iribarren, *The 4-triangles longest-side partition of triangles and linear refinement algorithms*, Math. Comp. **65** (1996), no. 216, 1485–1502, DOI 10.1090/S0025-5718-96-00772-7. MR1361811

[33] R. Stevenson, *The completion of locally refined simplicial partitions created by bisection*, Math. Comp. **77** (2008), no. 261, 227–241, DOI 10.1090/S0025-5718-07-01959-X. MR2353951

[34] M. J. Todd, *The Computation of Fixed Points and Applications*, Springer-Verlag, Berlin-New York, 1976. Lecture Notes in Economics and Mathematical Systems, Vol. 124. MR0410732

[35] C. T. Traxler, *An algorithm for adaptive mesh refinement in n dimensions*, Computing **59** (1997), no. 2, 115–137, DOI 10.1007/BF02684475. MR1475530

Institut für Geometrie und Praktische Mathematik, RWTH Aachen University, Templergraben 55, D-52056 Aachen, Germany

Email address: grande@igpm.rwth-aachen.de