SIAM J. Sci. Comput.
Vol. 40, No. 6, pp. A3853–A3872

A WEAK COMPATIBILITY CONDITION FOR NEWEST VERTEX BISECTION IN ANY DIMENSION[*]

MARTIN ALKÄMPER[†], FERNANDO GASPOZ[‡], AND ROBERT KLÖFKORN[§]

Abstract. We define a weak compatibility condition for the Newest Vertex Bisection algorithm on simplex grids of any dimension and show that, using this condition, the iterative refinement algorithm terminates successfully. Additionally we provide an $O(n)$ algorithm that renumbers any simplex grid to fulfill this condition. Furthermore we conduct experiments to estimate the distance to the standard compatibility and also study the geometric quality of the produced meshes.

Key words. adaptive method, mesh generation, mesh refinement, Newest Vertex Bisection

AMS subject classifications. 65N30, 65N50, 65N12

DOI. 10.1137/17M1156137

1. Introduction.

Dynamically adaptive conforming unstructured meshes based on simplices usually use either Newest Vertex Bisection (NVB) [6, 17, 9, 10, 22, 2] or Longest Edge Bisection (LEB) [24, 13, 3, 18, 14]. While NVB is a mere topological construct, where refinement simply depends on an ordering of the vertices, LEB uses geometric information to always refine the longest edge. While it is clear for NVB that it only produces a finite number of shape classes, for LEB this is nontrivial and requires some combinatorial effort [3]. This is a strong argument for using NVB, and an overview on NVB can be found in [22].

However, one big drawback of NVB is its nonapplicability to generic unstructured meshes in three (or more) dimensions, as it needs a compatibility condition between neighboring elements on the initial grid. Standard simplicial mesh generators, such as Gmsh [11] or TetGen [26], do not guarantee this condition. This has been an open problem for a long time, and with this work, to the best of our knowledge, for the first time a solution to this problem is presented that is applicable in any dimension and relies only on standard NVB elements (in contrast to [4]). Also, it does not multiply the number of elements by $\frac{1}{2}(d+1)!$, while halving each angle d times, as in [19, 27], and it is surprisingly simple.

The standard (strong) compatibility condition has been developed by Maubach and Traxler [20, 28] and generalized by Stevenson [27]. It ensures that every level

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section November 9, 2017; accepted for publication (in revised form) September 21, 2018; published electronically November 20, 2018.

http://www.siam.org/journals/sisc/40-6/M115613.html

Funding: The first author's work was supported by the German Research Foundation (DFG) within the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart. The second author's work was supported by the German Research Foundation (DFG) (DFG 814/7-1) at the University of Stuttgart. The third author's work was supported by the Research Council of Norway and the industry partners, ConocoPhillips Skandinavia AS, Aker BP ASA, Eni Norge AS, Total E&P Norge AS, Equinor ASA, Neptune Energy Norge AS, Lundin Norway AS, Halliburton AS, Schlumberger Norge AS, Wintershall Norge AS, and DEA Norge AS, of The National IOR Centre of Norway.

[†] Institut für Angewandte Analysis und Numerische Simulation, Universität Stuttgart, Pfaffenwaldring 57, D-70569 Stuttgart, Germany (alkaemper@ians.uni-stuttgart.de).

[‡] Lehrstuhl LSX, Fakultät für Mathematik, Technische Universität Dortmund, Vogelpothsweg 87, D-44227 Dortmund, Germany (fernando.gaspoz@tu-dortmund.de).

[§] NORCE Norwegian Research Centre, Postboks 22, Nygaardstangen, 5838 Bergen, Norway (robert.kloefkorn@norceresearch.no).

of uniform refinement is conforming and that there is a bound on the effort of the conforming closure. There is even a bound on the effort for parallel computations [2].

We introduce a weaker compatibility condition that is applicable to generic unstructured simplicial grids. We gain applicability at the cost of losing some of the convenient properties obtained by the stronger compatibility condition. It generalizes the concept of a mesh being conformingly marked, introduced by Arnold, Mukherjee, and Pouly in [4], and directly relates to what Stevenson calls a compatibly divisible mesh [27]. We will show that NVB terminates successfully using this condition.

Furthermore, we present an algorithm that is capable of relabeling any grid to be weakly compatible and has in principle an effort of $O(n)$, where n is the number of elements in the grid. In our implementation, however, the algorithm shows a complexity of $O(n \log n)$ which is related to a neighbor search that is carried out prior to the relabeling algorithm. In addition, the algorithm is capable of recovering a strongly compatible situation for some meshes.

To estimate the effort of the conforming closure in subsection 3.3, we will investigate metrics that relate to distances to a strongly compatible situation, as a strongly compatible situation minimizes the effort by design. The first metric measures refinement propagation on the macro level, and the second metric exploits the locality of the definition of strong compatibility and counts the number of faces where the corresponding neighbors do not fulfill the strong compatibility condition.

2. Weak compatibility condition.

2.1. Introduction. We briefly introduce NVB, following the notation of Stevenson [27], while the algorithm itself was originally introduced by Maubach and Traxler [20, 28].

Definition 1 (Newest Vertex Bisection). *We identify a d-dimensional simplex T (the convex hull $\operatorname{conv}\{z_i : 0 \le i \le d\}$) with an ordering of the vertices z_i and a type $0 \le t_T \le d-1$, i.e.,*

$$T = [z_0, z_1, \dots, z_d]_{t_T}.$$

Then, the Newest Vertex Bisection (NVB) of the simplex T is defined by its two children

$$T_0 = \left[z_0, \frac{z_0 + z_d}{2}, z_1, \dots, z_{d-1}\right]_{t_T+1 \mod d}$$

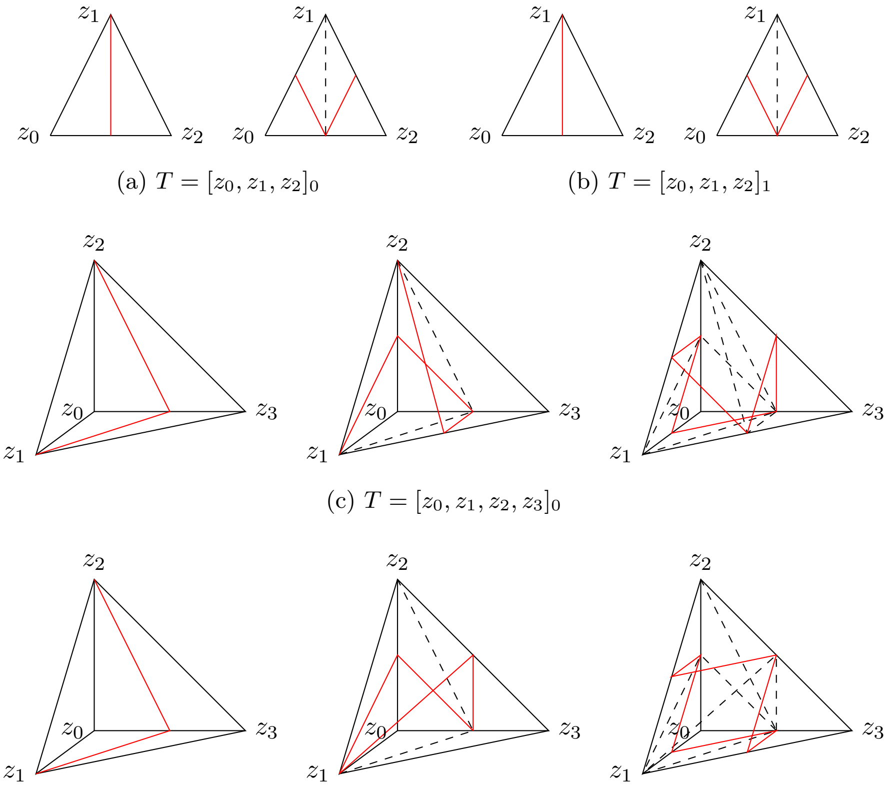
and

$$T_1 = \left[z_d, \frac{z_0 + z_d}{2}, z_1, \dots, z_{t_T}, z_{d-1}, \dots, z_{t_T+1}\right]_{t_T+1 \mod d}.$$

We call the edge $E_T = \overline{z_0 z_d}$ the refinement edge *of the simplex T.*

Remark 2. The type t of an element indicates how many vertices within the element are labeled as "new" and guards them at positions $1, \dots, t$. Definition 1 encodes the identification of the simplex $T = [z_0, \dots, z_d]_t$ with two ordered sets $\mathcal{V}_0 = [z_0, z_{t+1}, \dots, z_d], \mathcal{V}_1 = [z_1, \dots, z_t]$. The children of T then correspond to $\mathcal{V}_0 \setminus \{z_0\}$ or $\mathcal{V}_0 \setminus \{z_d\}$, and $\mathcal{V}_1 = [\frac{z_0+z_d}{2}, z_1, \dots, z_t]$. Once $\#\mathcal{V}_1 = d$ is reached, $\mathcal{V}_1$ is first included into $\mathcal{V}_0$ and then reset to $\mathcal{V}_1 = \emptyset$. Note that $\mathcal{V}_0$ may be flipped, and the resulting children and all further refinement coincide. This has also been observed in [27, section 2], where two simplices are identified (T_R) if they have the same set of children.

Example 3 (NVB in 3D). In three dimensions the simplex $T = [z_0, z_1, z_2, z_3]_{t_T}$ is of type $t_T \in \{0, 1, 2\}$.



(a) $T = [z_0, z_1, z_2]_0$

(b) $T = [z_0, z_1, z_2]_1$

(c) $T = [z_0, z_1, z_2, z_3]_0$

(d) $T = [z_0, z_1, z_2, z_3]_1$

FIG. 2.1. *Newest Vertex Bisection in 2D and 3D.*

Bisecting the refinement edge $E_T = \overline{z_0 z_3}$ at its center $z_{03} = \frac{z_0+z_3}{2}$ leads to the two simplices

$$T_0 = [z_0, z_{03}, z_1, z_2]_{t_T+1 \mod 3} \quad \text{and} \quad \begin{cases} T_1 = [z_3, z_{03}, z_2, z_1]_1 & \text{if } t_T = 0, \\ T_1 = [z_3, z_{03}, z_1, z_2]_2 & \text{if } t_T = 1, \\ T_1 = [z_3, z_{03}, z_1, z_2]_0 & \text{if } t_T = 2. \end{cases} \tag{2.1}$$

The refinement edges of the children are $E_{T_0} = \overline{z_0 z_2}$ and $E_{T_1} = \overline{z_3 z_1}$ if $t_T = 0$ and $E_{T_1} = \overline{z_3 z_2}$ if $t_T \in \{1, 2\}$ (see Figure 2.1).

From this we can directly deduce the initial refinement edge of all four faces of the simplex, namely

$$\begin{aligned} F_0 &= \{z_0, z_1, z_2\}: & E_{F_0} &= \overline{z_0 z_2}, \\ F_1 &= \{z_0, z_1, z_3\}: & E_{F_1} &= \overline{z_0 z_3}, \\ F_2 &= \{z_0, z_2, z_3\}: & E_{F_2} &= \overline{z_0 z_3}, \\ F_3 &= \{z_1, z_2, z_3\}: & E_{F_3} &= \begin{cases} \overline{z_1 z_3} & \text{if } t_T = 0, \\ \overline{z_2 z_3} & \text{else.} \end{cases} \end{aligned}$$

Notice that two 3D neighboring simplices will match if the initial refinement edge on their shared face coincides; otherwise the refinement algorithm will fail. Here follows the idea of Arnold, Mukherjee, and Pouly [4] to set for each face the initial refinement edge as the longest edge. From there they create the simplices containing said faces, but then have to introduce nonstandard initial elements and their respective refinement in order to combat the problem that arises from the fact that the simplices are not necessarily NVB simplices.

2.2. Some NVB properties. By creating a finite number of similarity classes NVB creates a sequence of nondegenerating simplices (see [28, 20]). In order to use this to our advantage, we define the concepts of Refinement Trees and NVB-equivalence; this equivalence defines the same equivalence classes for simplices as in [27].

Definition 4 (Refinement Tree and NVB-equivalence). *Let $T = [z_0, \ldots, z_d]_{t_T}$. We execute d uniform NVB-refinements, which we denote in a graph by setting the refinement edges to be the nodes of the graph and the two children to be the refinement edges of the children. The root is the initial refinement edge $\overline{z_0 z_d}$. We call this graph the Refinement Tree $\mathcal{RT}(T)$ and call a d-simplex T NVB-equivalent to a simplex T' if $\mathcal{RT}(T) = \mathcal{RT}(T')$.*

Remark 5. Let $T = [z_0, \ldots, z_d]_t$ with $0 \leq t < d - 3$. Using $z_{ij} = \frac{z_i + z_j}{2}$ and the identification from Remark 2, the second child of the first child corresponds to $\mathcal{V}_0 = [z_{t+1}, \ldots, z_{d-1}], \mathcal{V}_1 = [z_{0(d-1)}, z_{0d}, z_1, \ldots, z_t]$, and the second child of the second child corresponds to $\mathcal{V}_0 = [z_{t+1}, \ldots, z_{d-1}], \mathcal{V}_1 = [z_{(t+1)d}, z_{0d}, z_1, \ldots, z_t]$. So until $\mathcal{V}_1$ is reset the refinement behavior of these two simplices coincides. This allows us to represent the Refinement Tree for types $0, 1$ in a condensed fashion by identifying subtrees of actually distinct elements.

Example 6 (Condensed Refinement Tree for type 0). Figure 2.2 shows the condensed Refinement Tree of the simplex $T = [z_0, \ldots, z_d]_0$. It is condensed, which means that the number of elements at a position is the number of paths leading to this position, i.e., $\binom{l}{k}$ with the level l and k the position within the row. So each row consists of 2^l descendants. One can easily see that this is also the Refinement Tree of $T' = [z_d, z_{d-1}, \ldots, z_1, z_0]_0$, so T is NVB-equivalent to T'.

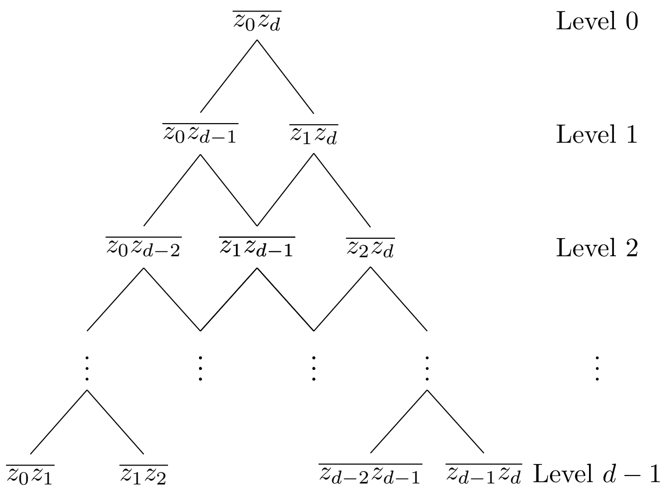


Fig. 2.2. *Condensed Refinement Tree of a d-simplex of type 0 under d uniform refinements.*

Note that in principle one can define Refinement Trees for arbitrary bisection-based refinement. So to turn a Refinement Tree into a simplex ordering, one needs to prove that it is one of the $d+1$ (number of types) Refinement Trees that relate to an actual NVB refinement.

Remark 7. For $d > 2$ the identification in Remark 2 coincides with NVB-equivalence. Note that in principle NVB-equivalence may be defined for arbitrary bisection based refinement.

As we will often deal with d uniform bisection, we use $\mathcal{T}_j$ with a subindex j for j uniform refinements of $\mathcal{T}$. We will use $\mathcal{T}^i$ with a superindex i for lower dimensional subentities as follows.

DEFINITION 8 (d-skeleton of a triangulation). *Let $\mathcal{T}$ be a triangulation of a domain $\Omega \subset \mathbb{R}^d$. For $0 \leq i \leq d$ we denote by $\mathcal{T}^i$ the set of all simplices of dimension i that are contained in $\mathcal{T} = \mathcal{T}^d$ and call $\mathcal{T}^i$ the i-skeleton of $\mathcal{T}$. We also define $\mathcal{T}^i(T)$ to be the i-skeleton of the triangulation that consists of the single element T.*

In particular, $\mathcal{T}^0 = \mathcal{V}$ is the set of all vertices, $\mathcal{T}^1 = \mathcal{E}$ the set of all edges, and $\mathcal{T}^{d-1} = \mathcal{F}$ the set of all faces. Note that $\mathcal{T}_j^i$ is the i-skeleton of the j-times refined grid and not the j-times refined i-skeleton.

LEMMA 9 (type 0). *Let $T = [z_0, \ldots, z_d]_0$ be a d-simplex of type $t_T = 0$ that is bisected d times uniformly.*

- *Then every edge $E \in \mathcal{T}^1(T)$ gets bisected exactly once and no newly created edge gets bisected.*
- *Any subsimplex $F \in \mathcal{T}^{d-1}(T)$ is bisected by a valid NVB uniformly $d-1$ times. F is of type $t_F = 0$ and its ordering coincides with the ordering of T up to NVB-equivalence.*

Proof. The refinement edges of T are depicted in Figure 2.2. The distance of the indices decreases by 1 at each level, so any edge $\overline{z_i z_j}$, $i < j$, is bisected exactly once at level $d-j+i$. Also, none of the vertices in this tree are midpoints of former bisections, so no newly created edges are bisected. This proves the first claim.

For the second claim we notice that for the d uniform NVB-refinements of the simplex T the refinement edges of the induced refinement of the face

$$F_i = \operatorname{conv}\{z_0, \ldots, z_{i-1}, z_{i+1}, \ldots, z_d\} \in \mathcal{T}^{d-1}(T)$$

are refinement edges of the Refinement Tree of T. Moreover, the children of the edge $\overline{z_n z_m}$ in the induced Refinement Tree of F_i are $\overline{z_n z_{m-1}}$ (or $\overline{z_n z_{m-2}}$ if $m-1 = i$) and $\overline{z_{n+1} z_m}$ (or $\overline{z_{n+2} z_m}$ if $n+1 = i$). From this we can clearly see that F_i is NVB-equivalent to $[z_0, \ldots, z_{i-1}, z_{i+1}, \ldots, z_d]_0$. ∎

Remark 10. This implies by induction that for $0 \leq i \leq d-1$ all elements of $\mathcal{T}^i(T)$ are valid NVB simplices of type 0.

We now proceed to present a variant of Lemma 9 for elements of type $t_T = 1$.

LEMMA 11 (type 1). *Let $T = [z_0, z^*, z_1, \ldots, z_{d-1}]_1$ be a d-simplex of type $t_T = 1$ that is bisected d times uniformly.*

- *Then every edge $E \in \mathcal{T}^1(T)$ gets bisected exactly once and no newly created edge gets bisected.*
- *Any subsimplex $F \in \mathcal{T}^{d-1}(T)$ is of type $t_F = 1$ if it contains z^* and of type 0 else. The ordering of F coincides with the ordering of T up to NVB equivalence.*

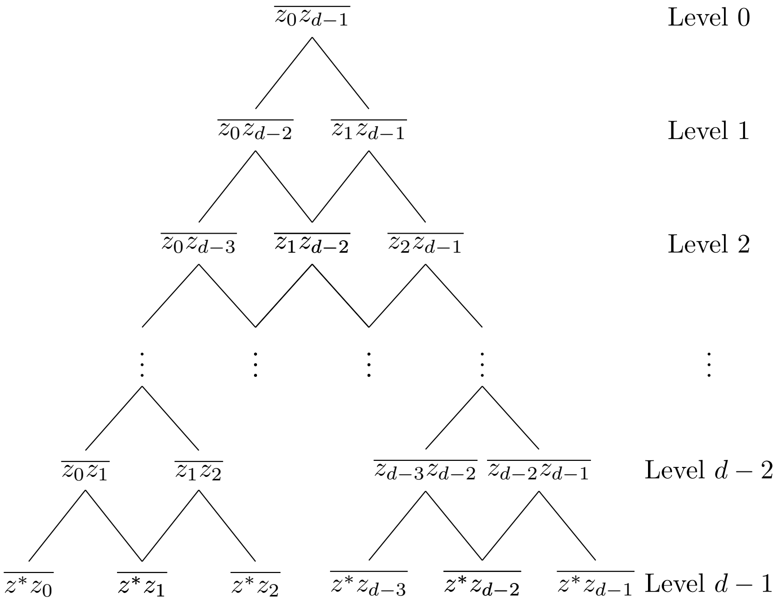


FIG. 2.3. *Condensed Refinement Tree of a d-simplex of type* 1 *under d uniform refinements.*

Proof. The proof is very similar to the proof of Lemma 9. Figure 2.3 shows the refinement edges of T. The distance between the indices decreases by 1 at each level until level $d-2$ and the final level consists of all edges containing z^*. So any edge $\overline{z_iz_j}$, $i<j$, is bisected exactly once at level $d-1-j+i$ and any edge $\overline{z_iz^*}$ is bisected exactly once at level $d-1$. No edge containing a newly created point is bisected, so this proves the first claim.

For the second claim we notice that for the d uniform NVB-refinements of the simplex T the refinement edges of the induced refinement of the face

$$F_i = \operatorname{conv}\{z_0, z^*, \dots, z_{i-1}, z_{i+1}, \dots, z_{d-1}\} \in \mathcal{T}^{d-1}(T)$$

are refinement edges of the Refinement Tree of T. Moreover, the children of the edge $\overline{z_nz_m}$ in the induced Refinement Tree of F_i are $\overline{z_nz_{m-1}}$ (or $\overline{z_nz_{m-2}}$ if $m-1=i$) and $\overline{z_{n+1}z_m}$ (or $\overline{z_{n+2}z_m}$ if $n+1=i$) for $m \neq n+1$. For the case $m=n+1$ the children are $\overline{z^*z_n}$ and $\overline{z^*z_m}$. Finally, for the case $n=i-1, m=i+1$ the children are $\overline{z^*z_n}$ and $\overline{z^*z_m}$. From this we can clearly see that F_i is NVB-equivalent to $[z_0, z^*, \dots, z_{i-1}, z_{i+1}, \dots, z_d]_1$.

The subsimplex $F' = \operatorname{conv}\{z_0, z_1, \dots, z_{d-1}\}$ is NVB-equivalent to $[z_0, z_1, \dots, z_{d-1}]_0$ as every edge $E = \overline{z_iz_j}$ gets bisected at level $d-1-j+i$. This proves the second claim. ◻

Remark 12. We can iterate Lemma 11 in combination with Lemma 9 to see that for a d simplex T with $t_T = 1$ any lower dimensional subsimplex $F \in \mathcal{T}^i(T)$, $2 \leq i \leq d-1$, it holds that $t_F = 1$ if $z^* \in F$ and $t_F = 0$ else.

For types $t_T = 0, 1$ we have the convenient result that the number of refinements to bisect every edge is d. For types greater than 1 this is unfortunately not true. Also it is hard to depict the Refinement Tree, as we cannot use a condensed Refinement Tree anymore.

Lemma 13 (type ≥ 2). *Let $T = [z_0, z_1^*, \ldots, z_t^*, z_1, \ldots, z_{d-t}]_t$ be a d-simplex of type $t_T = t \geq 2$ that is bisected $2d - t$ times uniformly, and let $\mathcal{V}_1 = \{z_1^*, \ldots, z_t^*\}$.*

- *Then every edge of T gets bisected at least once.*
- *Any face $F \in \mathcal{T}^{d-1}(T)$ is of type $t_F = \#(T' \cap \mathcal{V}_1)$. The ordering of F coincides with the ordering of T up to* NVB*-equivalence.*

Proof. Bisecting the element $T = [z_0, z_1^*, \ldots, z_t^*, z_1, \ldots, z_{d-t}]_t$ uniformly $d - t$ times leads to a set of descendants of type 0 that look as follows:

$$T' = [\bar{z}_0, \bar{z}_1, \ldots, \bar{z}_{d-t}, z_1^*, \ldots, z_t^*]_0,$$

where $\bar{z}_k$ are either vertices of T or centers of refinement edges of T. This means that none of the refinement edges $E_{ij} = \overline{z_i^* z_j^*}$, $1 < i < j < t$, has been refined yet. From the proof of Lemma 9 we know that E_{ij} will be refined at level $d - j + i$ of T' and hence at level $d - t + d - j + i = 2d - t - j + i$ of T. In particular for E_{12} this means $2d - t - 1$, so after $2d - t$ uniform bisections every edge in T has been refined at least once.

For the second claim we construct an artificial ancestor $\tilde{T} = [z_0, \ldots, z_d]_0$ by assuming that the simplex T has been created by t bisections always being the first child. From Lemma 9 we know that the ordering of all subsimplices of $\tilde{T}$ coincides with the ordering of $\tilde{T}$, and this translates to the descendant. Furthermore, $\tilde{T}$ has been refined t times introducing the vertices $\mathcal{V}_1$, which means that any subsimplex $\tilde{T}'$ has been refined as many times as it contains vertices from $\mathcal{V}_1$, which proves the second claim.

In particular for $T = [z_0, z_1^*, \ldots, z_t^*, z_1, \ldots, z_{d-t}]_t$ any subsimplex $T' \in \mathcal{T}^{d-1}(T)$ is of type $t_{T'} = \#\{z_i^* | z_i^* \in T'\}$ as the subsimplex of the artificial ancestor has been refined as many times. □

Remark 14. Lemma 13 generalizes Remark 3.4 of [9].

Lemmas 9, 11, and 13 lead to the following corollary by simple induction.

Corollary 15 (trace meshes). *For $0 < i < d$ the meshes $\mathcal{T}_d^i$ can be seen as standard* NVB *meshes on their own.*

This means for any simplex T with type $0 \leq t_T \leq d - 1$ for $0 \leq i \leq d - 1$ all elements of $\mathcal{T}^i(T)$ are valid NVB simplices.

2.3. Compatibility condition. As refinement is defined elementwise with NVB, we have to define a compatibility condition that couples neighboring elements. The weakest possible condition is the following.

Definition 16 (weakest compatibility condition). *A triangulation $\mathcal{T}$ fulfills the weakest compatibility condition if and only if for any pair of neighbors $T_i, T_j \in \mathcal{T}^d$ and their shared face $F \in \mathcal{T}^{d-1}$ with $F = T_i \cap T_j$ the ordering and type of F within both elements coincide up to* NVB*-equivalence.*

This basically means that the induced refinement of a face is the same from both sides. This is called compatibly divisible in [27] or conformingly marked in [4] (for $d = 3$). This condition is necessary but may not be sufficient for the iterative NVB algorithm [27] to terminate. So we define a slightly stronger condition.

Definition 17 (weak compatibility condition). *A triangulation $\mathcal{T}$ is called weakly compatible if and only if it fulfills Definition 16 and there exists a finer triangulation that is conforming and only contains elements of type $t \in \{0, 1\}$.*

Remark 18. The recursive algorithm [21, Algorithm 2.1] does in general not terminate under these conditions, as neither Definition 16 nor Definition 17 imply that the grid is loop-free. On the other hand, the iterative algorithm [19, Algorithm 1] will terminate under Definition 17 (see Corollary 21).

Remark 19. Any two-dimensional triangulation is weakly compatible, as it only contains elements of types 0 and 1.

Lemma 20 (every dth uniform refinement is conforming). *Let $\mathcal{T}$ be conforming and only contain elements of type $t \in \{0, 1\}$. Then every dth uniform NVB refinement is conforming.*

Proof. We split the triangulation $\mathcal{T}$ into its elements and investigate each $\mathcal{T}(T)$. Every triangulation $\mathcal{T}(T)$ is refined d times uniformly. As all elements are type 0 or 1, we know from Lemmas 9 and 11 that all $d-1$-dimensional subsimplices $F \in \mathcal{T}^{d-1}(T)$ have been uniformly refined $d-1$ times. Now we reassemble the triangulation $\mathcal{T}$ from the element-triangulations $\mathcal{T}(T)$. Definition 16 ensures that subsimplices that are shared by two elements carry the same refinement structure. So the resulting triangulation is conforming. □

From Lemma 20 follows directly that under Definition 17 there is always a conforming finer grid that only contains elements of type $t \in \{0, 1\}$, which leads to the following corollary.

Corollary 21. *The iterative NVB refinement algorithm terminates on weakly compatible grids for any set $\mathcal{M}$ of elements marked for refinement.*

As we will refer to the standard compatibility condition as the strong compatibility condition in the remainder of this article, we also introduce the notion of reflected neighbors and the strong compatibility condition. These definitions were introduced by Stevenson [27, section 4].

Definition 22 (reflected neighbors). *Two elements $T = \operatorname{conv}\{z_0, \ldots, z_{d-1}, u\}$ and $T' = \operatorname{conv}\{z_0, \ldots, z_{d-1}, v\}$ are called reflected neighbors if and only if the types $t_T = t_{T'}$ and the ordering coincides with u and v being at the same position up to NVB-equivalence.*

Definition 23 (local strong compatibility). *A face $F \in \mathcal{T}^{d-1}$ is called strongly compatible if and only if the two adjacent elements are reflected neighbors, or their direct children that are adjacent to F are reflected neighbors.*

Definition 24 (strong compatibility condition). *A triangulation $\mathcal{T}$ is strongly compatible if all faces $F \in \mathcal{T}^{d-1}$ are strongly compatible.*

Remark 25. Definition 24 implies that all elements in the grid are of the same type.

We extend the local strong compatibility to include neighbors that differ by one in type.

Definition 26 (local quasi-strong compatibility). *Let $F \in \mathcal{T}^{d-1}$ have the two adjacent elements T, T' that differ by one in type with $t_{T'} = (t_T + 1) \mod d$. Then F is called quasi-strongly compatible if and only if the child of T that contains F is a reflected neighbor of $t_{T'}$.*

This definition mimics the behavior within adaptively refined strongly compatible grids. All initial elements in these grids are of the same type, and neighboring refined elements can only differ by one in generation; neighboring elements can also only differ by one in type (modulo d). Such refined elements originating from a strongly compatible grid fulfill Definition 26.

3. Algorithm.

The algorithm we design to actually reach a weakly compatible state is straightforward in any dimension.

Algorithm 3.1: A renumbering algorithm to satisfy Definition 17.

```
Let $\mathcal{V} = \mathcal{T}^0$ be the vertices of the triangulation $\mathcal{T}$. Then we divide $\mathcal{V}$ into
    disjoint subsets $\mathcal{V}_i \subset \mathcal{V}, i \in \{0,1\}, \mathcal{V}_0 \cup \mathcal{V}_1 = \mathcal{V}$ and provide an ordering $>_i$
    for each of them.
For all $T = [z_0, \ldots, z_n]_{t_T} \in \mathcal{T}$ we set $[z_1, \ldots, z_{t_T}] = T \cap \mathcal{V}_1$ in order of $>_1$,
    $[z_0, z_{t_T+1}, \ldots, z_n] = T \cap \mathcal{V}_0$ in order of $>_0$, and $t_T = \#(T \cap \mathcal{V}_1) \mod d$.
If $T \cap \mathcal{V}_0 = \emptyset$, we set $t_T = 0$ and all vertices are sorted ascendingly from $z_0$ to
    $z_n$ with respect to $>_1$.
```

THEOREM 27. *The resulting triangulation of Algorithm* 3.1 *is weakly compatible.*

Proof. The weakest compatibility follows directly from Lemmas 9, 11, and 13.

For the second property of Definition 17 we refine every simplex individually until it is type 0 without conforming closure. This coincides with refining all edges with both vertices contained in $\mathcal{V}_0$ and no other edge for every simplex. Hence the resulting triangulation is conforming and only contains elements of type 0. ∎

Remark 28. Note that the descendants of all faces F that are quasi-strongly compatible in $\mathcal{T}$ will be strongly compatible in the constructed descendant.

3.1. Variants of choice of the sets.

Now we show some variants of the choice of sets to see how this actually relates to usual grid construction.

OT0 Only Type 0: All elements are type 0 (i.e., $\mathcal{V}_0 = \mathcal{V}, \mathcal{V}_1 = \emptyset$).

ILE Initial Longest Edge: Let $C_{ILE} \in \mathbb{N}$ be a constant threshold. Then we define

$$\mathcal{V}_0 = \{v \in \mathcal{V} : v \text{ is in at least } C_{ILE} \text{ Longest Edges }\}$$

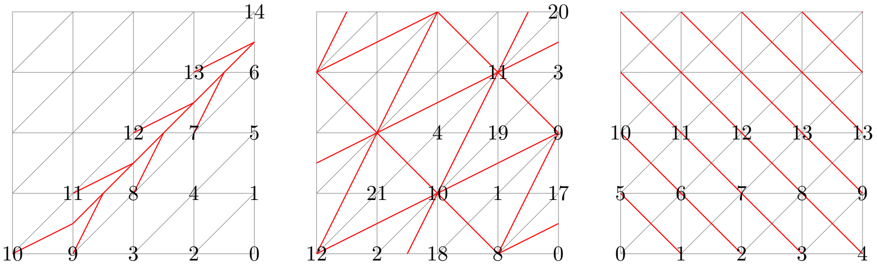
and $\mathcal{V}_1 = \mathcal{V} \setminus \mathcal{V}_0$.

LAE Least Adjacent Elements: The idea is to refine opposite of vertices that have least adjacent elements. Let $C_{LAE} \in \mathbb{N}$ be a constant threshold. Then we define

$$\mathcal{V}_1 = \{v \in \mathcal{V} : v \text{ is in at most } C_{LAE} \text{ Elements}\}$$

and $\mathcal{V}_0 = \mathcal{V} \setminus \mathcal{V}_1$.

Choosing OT0 is the only way to attain strong compatibility by definition. It is also the simplest choice possible. The idea behind ILE is to try mimicking the initial behavior of Longest Edge Bisection (LEB), which is known to improve elements that are not quasi-equilateral [25, 23]. LAE is motivated by the idea of refining opposite of large angles first, which corresponds to refinement edges on faces that are relatively large. These three set choices are in no way a thorough classification of the possible set choices but a possible selection of straightforward implementations. Another possibility would be, for example, to collect all vertices belonging to longest edges that are longer than average into $\mathcal{V}_0$ or collect the vertices with the maximum volume angle into $\mathcal{V}_1$. It is also possible to combine strategies.



(a) Diagonally alternating ordering leading to an expensive conforming closure

(b) SRN ordering leading to strong compatibility

(c) Rowwise and SRN2 ordering leading to strong compatibility

FIG. 3.1. *The conforming closure to different orderings for a 2D Kuhn-tesselation.*

3.2. Variants of the choice of ordering. Up to now, we only constructed the sets $\mathcal{V}_{(0,1)}$. We are still free to choose the ordering. We propose two strategies to construct an ordering, the second one being an improved version of the first one.

SRN Successive Reflected Neighbors: Build a global ordering that includes all vertices in $\mathcal{V}$ successively. Start with an element $T = [z_0, \dots, z_n]_{t_T}$ and add its ordering to the global ordering. Then loop over all neighbors and, if the new vertex has not been added to the ordering, insert the new vertex between the replaced vertex and its follower. Then loop over the neighbors of the neighbors, and so on.

SRN2 Successive Reflected Neighbors with announced refinement edge: We assume that every simplex announces an edge, which it wants to bisect. In our implementation this is the longest edge, but in principle any edge may be announced. We slightly alter the Successive Reflected Neighbors strategy. If the shared face with the neighbor does not contain the refinement edge, we have two options: insert the new vertex either before the first element or after the last. If one of the insertions yields the announced refinement edge, we choose this one.

SRN and SRN2 aim at making neighbors of different type (quasi-)strongly compatible, as types can at most differ by 1 by construction and they create (quasi-)strongly compatible faces opposite every newly inserted vertex. Figure 3.1 depicts different results of Algorithm 3.1 in 2D. The expensive ordering in Figure 3.1a for a structured grid leading to an expensive conforming closure is not easy to construct. In Figure 3.1b the ordering is created by Successive Reflected Neighbors (SRN), and in Figure 3.1c the orderings are created by row-wise ordering, which results in the same behavior as Successive Reflected Neighbors with announced refinement edge (SRN2) where the announced edge is the diagonal. Both grids are strongly compatible.

A possible implementation of OT0 and SRN2 is shown in Algorithm 3.2. Using a structured cube grid filled with Kuhn-cubes, where every simplex announces the diagonal initially, Algorithm 3.1 leads to the standard strongly compatible grid if the initial element is sorted correctly.

3.3. Quality of produced mesh. We are interested in two significant mesh quality goals, namely, the *mesh topological quality* goal that tries to minimize the effort of the conforming closure, and the *geometric quality* goal that tries to maximize the shape regularity of the elements.

Algorithm 3.2: Possible implementation of methods OT0 and SRN2.

```
Data: List of active faces $\mathbb{F}$, List of vertices $\mathcal{V}$, Mesh $\mathcal{T}$
$\mathbb{F} = \mathcal{V} = \emptyset$
Choose initial $T \in \mathcal{T}$
Order according to announced refinement edge $E(T)$
$\mathcal{V} = \mathcal{V}(T)$
add all nonboundary $F \in \mathcal{T}^{d-1}(T)$ with $E(T) \in F$ at the beginning of $\mathbb{F}$
add the other nonboundary $F \in \mathcal{T}^{d-1}(T)$ at the end of $\mathbb{F}$ with the flag
  noRefEdge
mark $T$ as treated
while $\exists$ untreated $T \in \mathcal{T}$ ($\Leftrightarrow \mathbb{F} \neq \emptyset$) do
    Get untreated neighbor $T'$ and treated element $T$ of first $F \in \mathbb{F}$
    $v = \mathcal{T}^0(T) \setminus \mathcal{T}^0(F), \quad v' = \mathcal{T}^0(T') \setminus \mathcal{T}^0(F)$
    if $v' \in \mathcal{V}$ then
        /* do nothing */
    else
        if noRefEdge set then
            /* In this case $v$ is the first or last vertex of $T$ */
            insert $v'$ after last or before first vertex of $T$, depending on the
              announced refinement edge
        else
            Insert $v'$ directly after $v$ into $\mathcal{V}$
        end
    end
    sort $T'$ according to $\mathcal{V}$
    mark $T'$ treated
    for nonboundary $F' \in \mathcal{T}^{d-1}(T')$ do
        if $F' \in \mathbb{F}$ then
            /* possibly check for strong compatibility */
            remove $F'$ from $\mathbb{F}$
        else
            if $E(T') \subset F'$ then
                add $F'$ at the beginning of $\mathbb{F}$
            else
                add $F'$ at the end of $\mathbb{F}$
            end
        end
    end
end
```

Strongly compatible grids fulfill the *mesh topological quality* goal by design. So a distance to a strongly compatible situation is a way of measuring this goal. To this aim we define the following two distances.

DEFINITION 29 (distance 1). *Let $\mathcal{T}$ be a weakly compatible grid sorted by one of the above methods, and let $\mathcal{F}_{SC} \subset \mathcal{T}^{d-1}$ be the set of faces that are strongly or* *quasi-strongly compatible. Then we define the distance*

$$d^1_{\mathcal{T}} = \#(\mathcal{T}^{d-1}) - \#(\mathcal{F}_{SC}).$$

This is a simple, easily computable, and straightforward metric that directly tells us whether the grid fulfills the strong compatibility condition. Unfortunately, it does not directly yield information on the effort of the conforming closure, so to estimate this we define a second distance.

Definition 30 (distance 2). *Let $\mathcal{T}$ be a weakly compatible grid sorted by one of the above variants, and let $\mathcal{C}_{\mathcal{T}}(T)$ be the conforming closure of refining T on the initial grid $\mathcal{T}$. Then we define the metric*

$$d^2_{\mathcal{T}} = \max_{T\in\mathcal{T}} \#(\mathcal{C}_{\mathcal{T}}(T)).$$

As the refinement becomes more local (cf. [2]) once the whole grid has been refined d times uniformly, we also investigate $d^2_{\mathcal{T}_d}$ on the d times uniformly refined grid $\mathcal{T}_d$.

Remark 31. We believe that the number of created simplices is bounded depending on the number of elements marked for refinement as in [27, Theorem 6.1] with an additional additive constant depending on the initial grid, which asymptotically should be negligible.

The geometric quality of meshes using bisection (both LEB and NVB) is always an issue. One of the angles may be divided 2^{d-1} times under d uniform bisections. LEB performs a bit better by design, but there is a significant drop in quality for any bisection-based refinement if the initial grid contains elements that are close to equilateral, which we expect from mesh generators such as TetGen. In principle it is advisable to start with elements that are similar to Kuhn-simplices. For NVB we have that d uniform bisections cover all possible similarity classes and hence will only measure our quality indicators on the initial grid and on the d times uniform refined grid.

The set of indicators we use are the d-dimensional sine, aspect ratios of volumes, faces, and edges (min, max, and average), and also the maximum number of adjacent elements of vertices and edges.

4. Numerical experiments.

In this section we study the behavior of the presented options of the reordering algorithm for a variety of different tetrahedral grids. The implementation first checks whether the provided grids are compatible or not. If the grids are already compatible, then no reordering is performed. In general the reordering has to be performed only once per grid. The implementation of the presented algorithm is contained in the open-source Dune module Dune-ALUGrid [1].

4.1. Threshold study.

The first set of experiments we conducted is a threshold study, where we measure the behavior of LAE and ILE with threshold values $C_{ILE}, C_{LAE} \in \{0, \dots, 35\}$. If the threshold value is 0, we obtain in both cases $\mathcal{V}_0 = \mathcal{V}$, so the case OT0 is implicitly included. We use both suggested orderings SRN and SRN2.

First, we test on a sequence of triangulations of the unit cube with decreasing average volume (see Figure 4.1) that have been used in [8]. We will call these grids the "Unitcube Sequence."

Additionally, we will show the same measures for a set of grids representing more complex geometrical structures. We will call these grids "Realistic Grids" and a

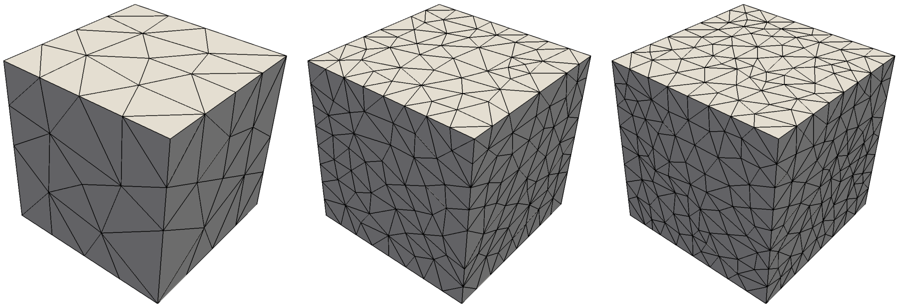


FIG. 4.1. *Series of tetrahedral grids discretizing the unit cube used in* [8].

depiction is presented in Figure 4.2. The tube grid has been previously used for numerical simulation of atherosclerotic plaque formation [12]. The grid representing the head of a human has been used in [16].

First we consider the distribution of $\mathcal{V}_0, \mathcal{V}_1$:

Figure 4.3 shows the proportion of vertices that have been sorted into $\mathcal{V}_0$ for different grids. As expected, LAE and ILE have different "active" regions, i.e., regions where a change of threshold changes the distribution a lot. For ILE this is $C_{ILE} \in \{0, \ldots, 20\}$ and for LAE this is $C_{LAE} \in \{10, \ldots, 30\}$. Keep these active regions in mind when examining the following figures. The distribution of $\mathcal{V}$ into $\mathcal{V}_0$ and $\mathcal{V}_1$ does not depend on the choice of ordering. So the active regions do not change from SRN to SRN2.

We now investigate metric $d^1_{\mathcal{T}}$, i.e., the number of not strongly compatible faces. To be comparable between grids of different size, we display the percentage with respect to all inner faces. Boundary faces are excluded, as they do not need to be compatible.

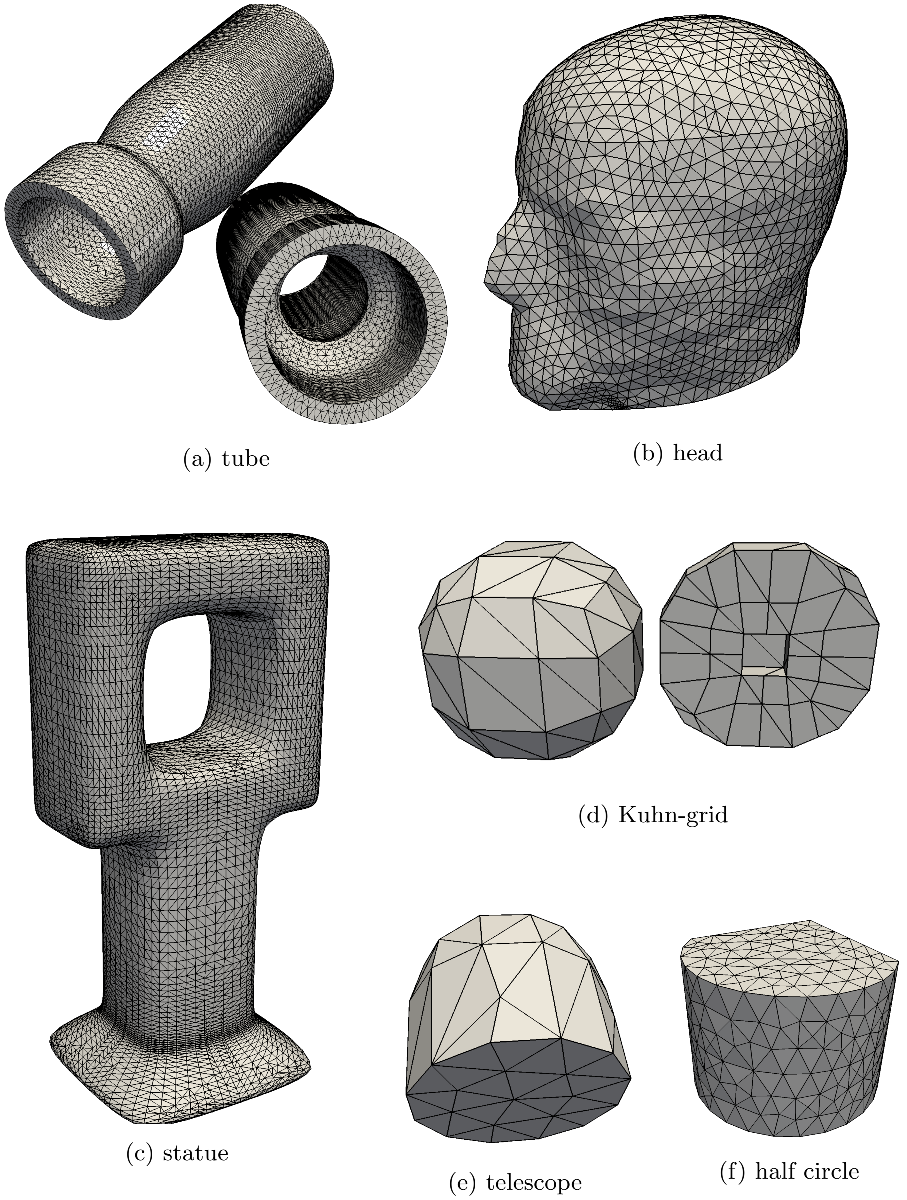
Figure 4.4 displays the number of not strongly compatible faces in the grid. There are a few things to note. In most cases SRN2 outperforms SRN by about 10% of the faces. The Kuhn-grid is able to recover the strong compatibility. Surprisingly, this is reached in the inactive region (i.e., OT0) combined with SRN.

Using the abbreviations Volume (V), Longest Edge (LE), Shortest Edge (SE), Largest Face (LF), Smallest Face (SF), Face Volume (F), we examine the following geometric metrics of the grid: $\mathrm{V/LE}^3$, $\mathrm{V/SE}^3$, $\mathrm{V/LF}^{3/2}$, $\mathrm{V/SF}^{3/2}$, $\mathrm{F/LE}^2$, $\mathrm{F/SE}^2$, and the d-sine [7]. We observed that $(\mathrm{V/LE}^3) \approx (\mathrm{V/LF}^{3/2})\times$ $(\mathrm{F/LE}^2)$ and additionally $(\mathrm{V/LF}) \approx$ d-sine. Qualitatively the metrics behave similarly, so we choose to display only the d-sine here.

The first value in these graphs with threshold value $C_{ILE/LAE} = -1$ denotes the unrefined grid, which is obviously the same for all thresholds. The geometric quality indicators have been measured on the 3-times uniformly refined grid.

Figure 4.5 displays the average d-sine for the investigated grids. For almost all grids we see the expected significant drop of geometric quality from the initial grid to the 3-times uniformly refined mesh and then some improvement in the active region. The only exception is the Kuhn-grid, which actually gets worse in the active region, where it is no longer strongly compatible. Also, there is no initial drop, as the mesh is based on Kuhn-simplices. For other grids LAE seems to perform a bit better than ILE, and choosing between SRN and SRN2 does not seem to have much impact.

A way to investigate the minimum angle is to consider the maximum number



(a) tube

(b) head

(c) statue

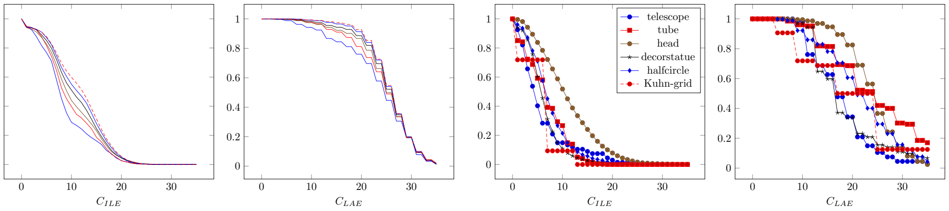
(d) Kuhn-grid

(e) telescope

(f) half circle

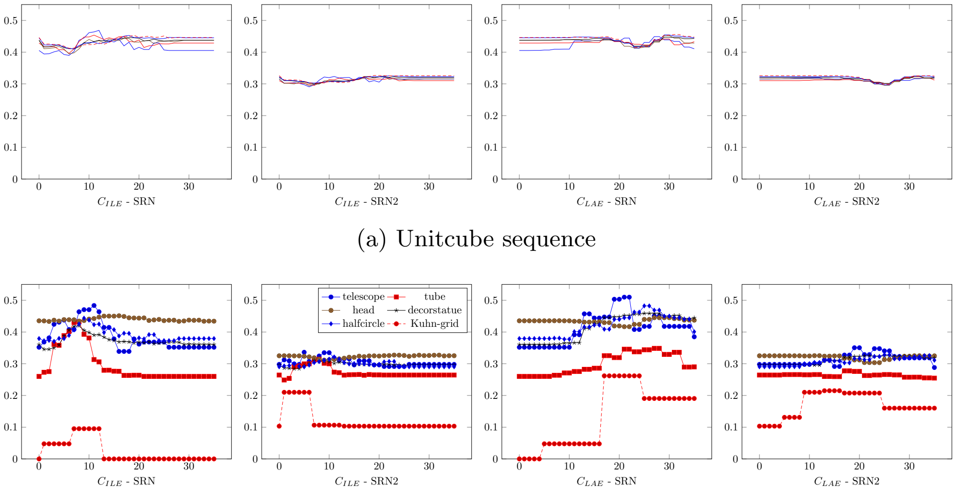
Fig. 4.2. *Realistic tetrahedral meshes used for studying the proposed algorithms. Figure* 4.2a *has been previously used for numerical simulation of atherosclerotic plaque formation* [12], *Figure* 4.2b *has been used in* [16] *for electrical impedance tomography simulations, Figure* 4.2c *has been downloaded from* [15] *and generated with* `TetGen`, *Figure* 4.2e *is part of the test grids in the* Dune-Grid *module* [5], *Figure* 4.2d *stems from a Kuhn-grid and has been modified (vertex renumbering and projection), and Figure* 4.2f *has been generated with* `Gmsh`.

of adjacent elements at a vertex. Figure 4.6 indicates that there is a big difference between LAE and ILE. In the active region of ILE this gets worse, while in the active region of LAE this value improves. To get an impression of the size of this value note the following calculation. In the equilateral case at every vertex there are 20 adjacent elements. Uniform bisection in 3D can quadruple the number of elements at a vertex,



(a) Unitcube sequence (b) Realistic grids

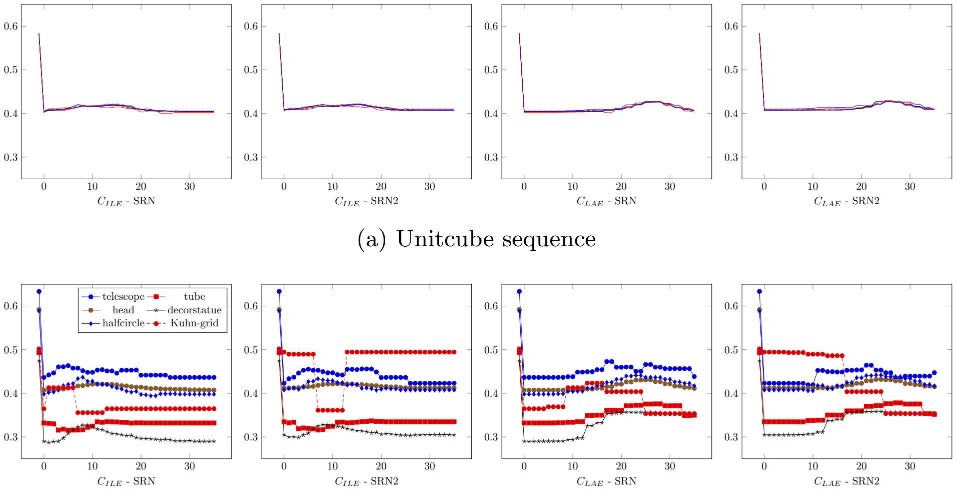
FIG. 4.3. *The proportion of vertices sorted into $\mathcal{V}_0$ over different thresholds.*



(a) Unitcube sequence

(b) Realistic grids

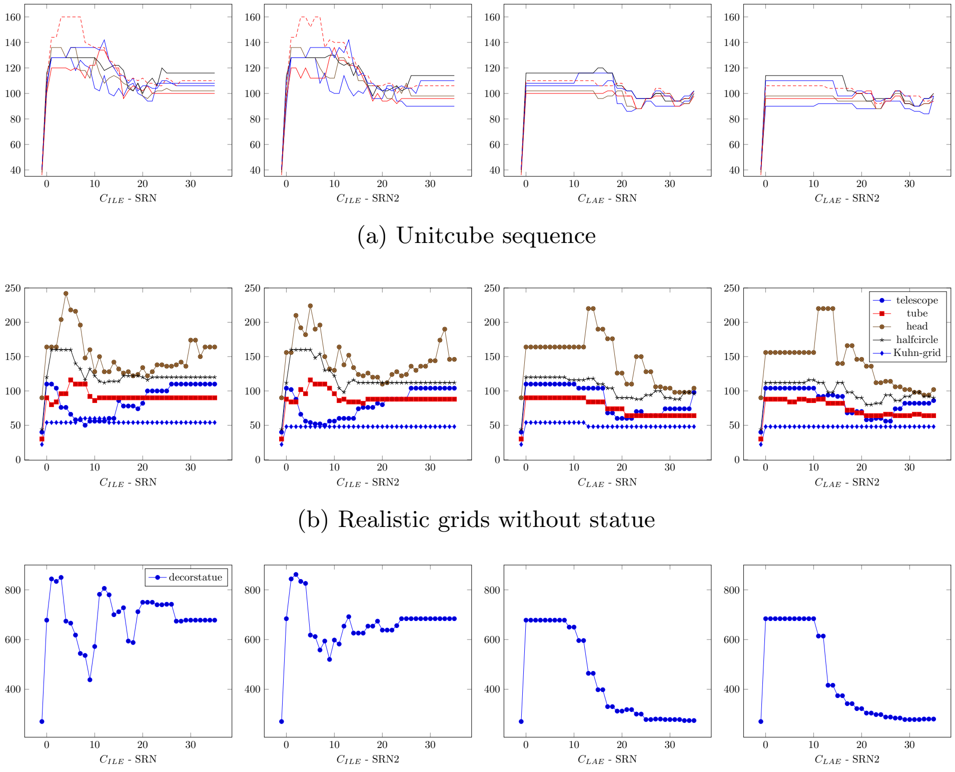
FIG. 4.4. *Percentage of not strongly compatible faces over different threshold values.*



(a) Unitcube sequence

(b) Realistic grids

FIG. 4.5. *Average d-sine.*



(a) Unitcube sequence

(b) Realistic grids without statue

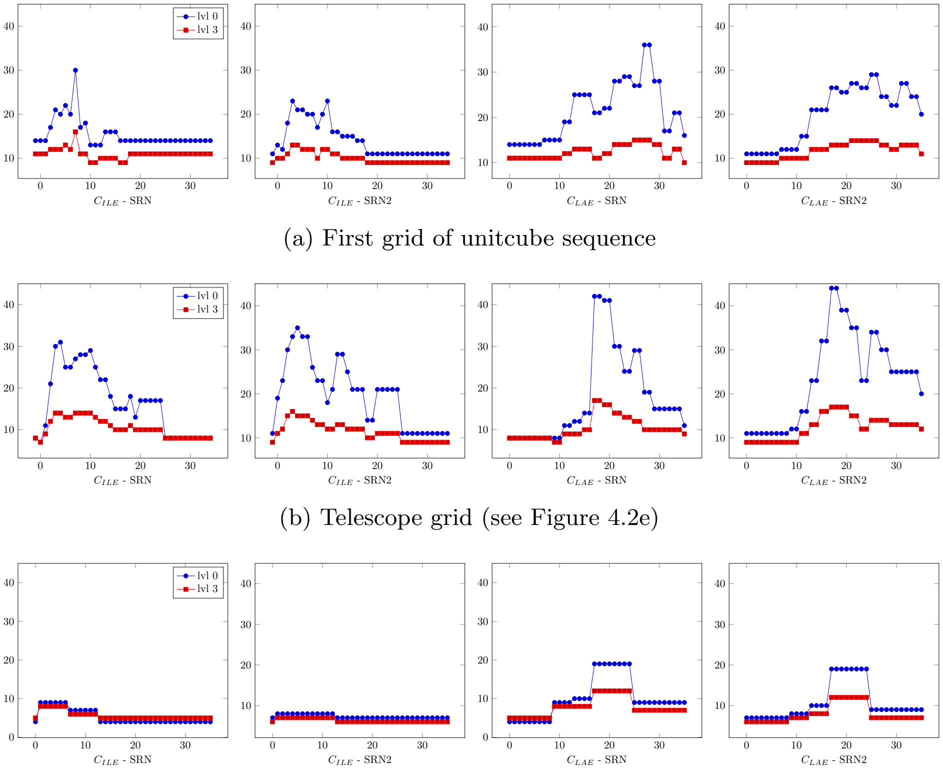
(c) Statue

FIG. 4.6. *Maximal number of elements at a vertex.*

which yields an expected 80 elements as a maximum after 3 uniform refinements for equilateral initial grids. The best values of LAE are not far from that. The special grid originating from Kuhn-cubes (see Figure 4.2d) performs best under bisection with respect to both measures. The second special case among the realistic grids is the statue grid in Figure 4.2c. It has by far the most elements at a vertex in the initial triangulation, and we see that ILE is not improving that, while LAE almost recovers the initial value at a high threshold.

4.2. Conforming closure. We discuss the connection between the distances $d^1_{\mathcal{T}}$, $d^2_{\mathcal{T}}$, and $d^2_{\mathcal{T}_3}$. Unfortunately, the computational cost of $d^2_{\mathcal{T}}$ is at least $O(n^2)$ and that of $d^2_{\mathcal{T}_3}$ is at least $O((8n)^2)$. Hence, we did not compute these measures for all grids and possible parameters but just for a small selection. Also in the implementation we make the following simplification: if refinement of an element T is part of the conforming closure of another element T', then the conforming closure of T is included in the conforming closure of T' and we do not compute the closure of T. This means the maximum conforming closure is computed exactly, while we just provide an upper bound for the average conforming closure.

Figure 4.7 displays an upper bound for the average conforming closure of each element in a small grid of the `TetGen` sequence and the special grid, which become strongly compatible. Figure 4.8 actually displays $d^2_{\mathcal{T}}$ and $d^2_{\mathcal{T}_3}$, i.e., the maximum conforming closure for both grids. Both times we measure both on the initial grid



(a) First grid of unitcube sequence

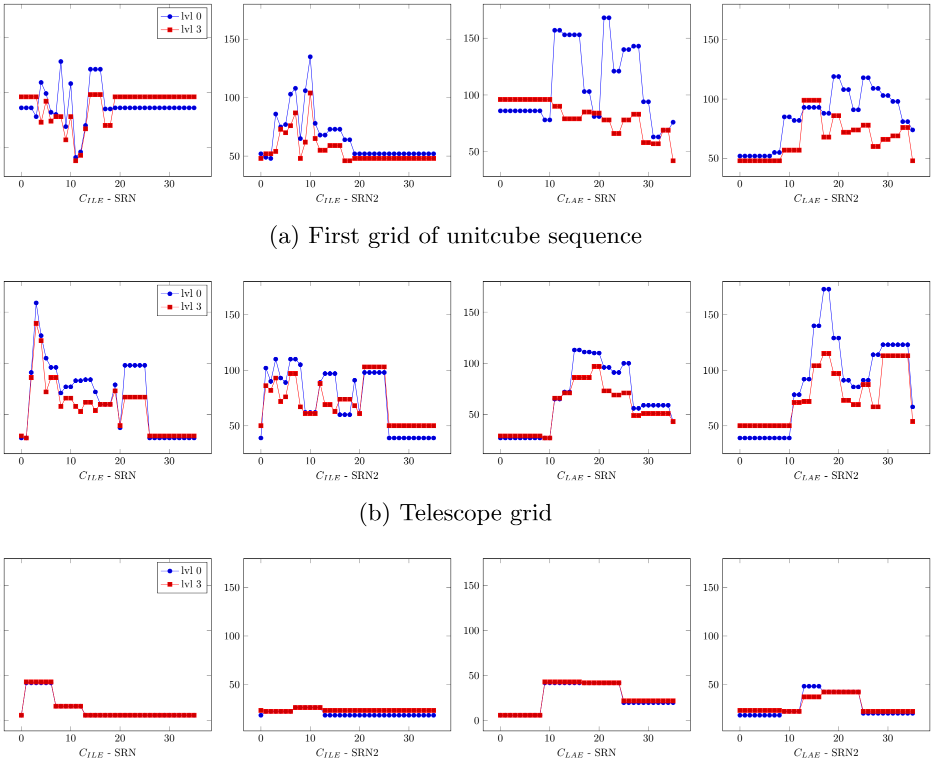
(b) Telescope grid (see Figure 4.2e)

(c) Kuhn-grid

FIG. 4.7. *Average number of elements in the conforming closure.*

(blue) and on the 3-times uniformly refined grid (red) (color available online). The behavior is almost always better in the 3-times uniformly refined case. The conforming closure is more costly in the active region, as we additionally introduce refinement propagation by having neighbors of different type. In the case of SRN we see for the `TetGen` mesh that the maximum conforming closure actually diminishes in the active region, but SRN2 performs better in total. The Kuhn-grid of course performs best in the inactive region, where it actually becomes strongly compatible.

4.3. Algorithm complexity. We also studied the runtime of the algorithm over the sequence of unit cube meshes that we used to study the threshold values. Figure 4.9 shows that the runtime actually is $O(n \log n)$ with n being the number of elements. The measured runtime includes recalculating the grid neighborhood information, checking whether a grid is compatible initially, setting up $\mathcal{V}_0, \mathcal{V}_1$, sorting the mesh, and finally checking how compatible the mesh is. The algorithm can be implemented in $O(n)$ if the recalculation of the neighborhood information is not necessary. In the current implementation the association of neighboring elements is done via comparison of vertex indices of faces and storage of those in a `std::map` with $O(\log n)$ member access, where n is the number of faces. Therefore, the overall algorithm complexity becomes $O(n \log n)$. In our tests we have run the various algorithms on meshes with about one million grid cells in under 10 seconds, which we consider sufficiently fast, especially since the algorithm has to be applied only once.



(a) First grid of unitcube sequence

(b) Telescope grid

(c) Kuhn-grid

FIG. 4.8. $d^2_{\mathcal{T}}$ *and* $d^2_{\mathcal{T}_3}$.

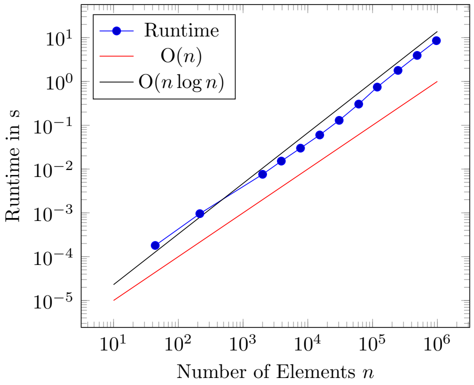


FIG. 4.9. *Runtime of the algorithm for different grid sizes.*

5. Summary and outlook.

We have presented a weak compatibility condition that is sufficient for the iterative NVB algorithm to terminate. Additionally, we provide a simple algorithm to relabel any d-dimensional grid to fulfill said condition with effort $O(n)$. The methods we designed try to achieve a compromise between LEB and NVB. We suggest using (LAE)/(SRN2) with a parameter around 27 for grids created by mesh generators that aim at equilateral tetrahedrons. For meshes created by Kuhn-cubes it is best to just use the strongly compatible enumeration and not to try to improve the behavior.

It is still an open question whether the *weak compatibility condition* is enough to obtain an estimate like the one in [27] for the computational effort of the conforming closure or if this can be achieved with some extra mild conditions in the mesh.

REFERENCES

[1] M. Alkämper, A. Dedner, R. Klöfkorn, and M. Nolte, *The DUNE-ALUGrid module*, Arch. Numer. Software, 4 (2016), pp. 1–28, https://doi.org/10.11588/ans.2016.1.23252.

[2] M. Alkämper and R. Klöfkorn, *Distributed newest vertex bisection*, J. Parallel Distributed Comput., 104 (2017), pp. 1–11, https://doi.org/10.1016/j.jpdc.2016.12.003.

[3] G. Aparicio, L. G. Casado, E. M. T. Hendrix, B. G.-Tóth, and I. Garcia, *On the minimum number of simplex shapes in longest edge bisection refinement of a regular n-simplex*, Informatica (Vilnius), 26 (2015), pp. 17–32, https://doi.org/10.15388/Informatica.2015.36.

[4] D. N. Arnold, A. Mukherjee, and L. Pouly, *Locally adapted tetrahedral meshes using bisection*, SIAM J. Sci. Comput., 22 (2000), pp. 431–448, https://doi.org/10.1137/S1064827597323373.

[5] M. Blatt, A. Burchardt, A. Dedner, C. Engwer, J. Fahlke, B. Flemisch, C. Gersbacher, C. Gräser, F. Gruber, C. Grüninger, D. Kempf, R. Klöfkorn, T. Malkmus, S. Müthing, M. Nolte, M. Piatkowski, and O. Sander, *The distributed and unified numerics environment, version* 2.4, Arch. Numer. Software, 4 (2016), pp. 13–29, https://doi.org/10.11588/ans.2016.100.26526.

[6] J. M. Cascon, C. Kreuzer, R. H. Nochetto, and K. G. Siebert, *Quasi-optimal convergence rate for an adaptive finite element method*, SIAM J. Numer. Anal., 46 (2008), pp. 2524–2550, https://doi.org/10.1137/07069047X.

[7] F. Eriksson, *The law of sines for tetrahedra and n-simplices*, Geom. Dedicata, 7 (1978), pp. 71–80, https://doi.org/10.1007/BF00181352.

[8] R. Eymard, G. Henry, R. Herbin, F. Hubert, R. Klöfkorn, and G. Manzini, 3*D benchmark on discretization schemes for anisotropic diffusion problems on general grids*, in Finite Volumes for Complex Applications VI: Problems & Perspectives, Springer Proc. Math. 4, J. Fort et al., eds., Springer, Berlin, Heidelberg, 2011, pp. 895–930, https://doi.org/10.1007/978-3-642-20671-9_89.

[9] D. Gallistl, M. Schedensack, and R. P. Stevenson, *A remark on newest vertex bisection in any space dimension*, Comput. Methods Appl. Math., 14 (2014), pp. 317–320, https://doi.org/10.1515/cmam-2014-0013.

[10] F. D. Gaspoz, C.-J. Heine, and K. G. Siebert, *Optimal grading of the newest vertex bisection and H^1-stability of the L_2-projection*, IMA J. Numer. Anal., 36 (2016), pp. 1217–1241, https://doi.org/10.1093/imanum/drv044.

[11] C. Geuzaine and J.-F. Remacle, *Gmsh: A* 3*-D finite element mesh generator with built-in pre- and post-processing facilities*, Internat. J. Numer. Methods Engrg., 79 (2009), pp. 1309–1331, https://doi.org/10.1002/nme.2579.

[12] S. Girke, R. Klöfkorn, and M. Ohlberger, *Efficient parallel simulation of atherosclerotic plaque formation using higher order discontinuous Galerkin schemes*, in Finite Volumes for Complex Applications VII, Springer Proc. Math. Statist. 78, J. Fuhrmann et al., eds., Springer, Cham, 2014, pp. 617–625, https://doi.org/10.1007/978-3-319-05591-6_61.

[13] A. Hannukainen, S. Korotov, and M. Křížek, *On global and local mesh refinements by a generalized conforming bisection algorithm*, J. Comput. Appl. Math., 235 (2010), pp. 419–436, https://doi.org/10.1016/j.cam.2010.05.046.

[14] R. Horst, *On generalized bisection of n-simplices*, Math. Comp., 66 (1997), pp. 691–698, https://doi.org/10.1090/S0025-5718-97-00809-0.

[15] INRIA GAMMA Group, 3D *Meshes Research Database*, https://team.inria.fr/gamma3/gamma-software/.

[16] M. Jehl, A. Dedner, T. Betcke, K. Aristovich, R. Klöfkorn, and D. Holder, *A fast parallel solver for the forward problem in electrical impedance tomography*, IEEE Trans. Biomed. Engrg., 62 (2015), pp. 126–137, https://doi.org/10.1109/TBME.2014.2342280.

[17] M. Karkulik, D. Pavlicek, and D. Praetorius, *On* 2*D newest vertex bisection: Optimality of mesh-closure and H^1-stability of L_2-projection*, Constr. Approx., 38 (2013), pp. 213–234, https://doi.org/10.1007/s00365-013-9192-4.

[18] S. Korotov, M. Křížek, and A. Kropáč, *Strong regularity of a family of face-to-face partitions generated by the longest-edge bisection algorithm*, Comput. Math. Math. Phys., 48 (2008), pp. 1687–1698, https://doi.org/10.1134/S0965542508090170.

[19] I. Kossaczký, *A recursive approach to local mesh refinement in two and three dimensions*, J. Comput. Appl. Math., 55 (1994), pp. 275–288, https://doi.org/10.1016/0377-0427(94)90034-5.

[20] J. M. Maubach, *Local bisection refinement for n-simplicial grids generated by reflection*, SIAM J. Sci. Comput., 16 (1994), pp. 210–227, https://doi.org/10.1137/0916014.

[21] W. F. Mitchell, *Unified Multilevel Adaptive Finite Element Methods for Elliptic Problems*, Ph.D. thesis, Department of Computer Science, University of Illinois, Urbana, IL, 1988.

[22] W. F. Mitchell, 30 *years of newest vertex bisection*, AIP Conference Proc., 1738 (2016), 020011, https://doi.org/10.1063/1.4951755.

[23] Á. Plaza, J. P. Suárez, M. A. Padrón, S. Falcón, and D. Amieiro, *Mesh quality improvement and other properties in the four-triangles longest-edge partition*, Comput. Aided Geom. Design, 21 (2004), pp. 353–369, https://doi.org/10.1016/j.cagd.2004.01.001.

[24] M.-C. Rivara, *New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations*, Internat. J. Numer. Methods Engrg., 40 (1997), pp. 3313–3324.

[25] M.-C. Rivara and G. Iribarren, *The* 4*-triangles longest-side partition of triangles and linear refinement algorithms*, Math. Comp., 65 (1996), pp. 1485–1502.

[26] H. Si, *TetGen, a Delaunay-based quality tetrahedral mesh generator*, ACM Trans. Math. Software, 41 (2015), 11, https://doi.org/10.1145/2629697.

[27] R. Stevenson, *The completion of locally refined simplicial partitions created by bisection*, Math. Comp., 77 (2008), pp. 227–241, https://doi.org/10.1090/S0025-5718-07-01959-X.

[28] C. T. Traxler, *An algorithm for adaptive mesh refinement in n dimensions*, Computing, 59 (1997), pp. 115–137, https://doi.org/10.1007/BF02684475.