A WEAK COMPATIBILITY CONDITION FOR NEWEST VERTEX BISECTION IN ANY DIMENSION

MARTIN ALKÄMPER A, FERNANDO GASPOZ A, AND ROBERT KLÖFKORN B

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 algorithm terminates successfully. Additionally we provide an \(O(n)\) algorithm that renumbers any simplex grid to fulfil this condition. Furthermore we conduct experiments to estimate the distance to the standard compatibility and also the geometric quality of the produced meshes.

1. Introduction

Dynamically adaptive conforming unstructured meshes based on simplices usually use either Newest Vertex Bisection (NVB) \[7, 10, 11, 20, 3\] or Longest Edge Bisection (LEB) \[21, 14, 4, 17, 15\]. 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 non-trivial and requires some combinatorial effort \[4\]. This is a strong argument for using NVB and an overview on NVB can be found in \[20\].

However, one big drawback of NVB is its non-applicability to generic unstructured meshes in 3 (or more) dimensions, as it needs a compatibility condition between neighboring elements on the initial grid. Standard simplicial mesh generators, such as Gmsh \[12\] or TetGen \[23\], 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 \[5\] ). Also it does not multiply the number of elements by \(\frac{1}{2}(d + 1)!\), while halving each angle \(d\) times, as in \[18, 24\], and it is surprisingly simple.

The standard (strong) compatibility condition has been developed by Traxler and Maubach \[19, 25\] and generalized by Stevenson \[24\]. It ensures that every level of uniform refinement is conforming and that there is a bound on the effort of the conforming closure. There even is a bound on the effort for parallel computations \[3\].

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 et al. in \[5\] and directly relates to what Stevenson calls compatibly divisible \[24\]. 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.
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 to recover a strongly compatible situation for some meshes. To estimate the effort of the conforming closure, 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 amount of faces where the corresponding neighbors do not fulfill the strong compatibility condition.

2. Weak Compatibility Condition

2.1. Introduction. We shortly introduce NVB following the notation of Stevenson [24], while the algorithm itself was originally introduced by Maubach and Traxler [19, 25].

Definition 2.1 (Newest Vertex Bisection). We identify a $d$-dimensional simplex $T$ (the convex hull of $\{z_i : 0 \leq i \leq d\}$) with an ordering of the vertices $z_i$ and its type $0 \leq t_T \leq d - 1$, i.e.,

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

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

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

and

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

We call the edge $E_T = \frac{z_0 + z_d}{2}$ the refinement edge of the simplex $T$.

Example 2.2 (NVB in 3d). In 3 dimensions the simplex $T = [z_0, z_1, z_2, z_3]_{t_T}$ is of type $t_T \in \{0, 1, 2\}$. Bisecting the refinement edge $E_T = \frac{z_0 + z_3}{2}$ at its center $z_{03} = \frac{z_0 + z_3}{2}$ leads to the two simplices

$T_0 = [z_0, z_{03}, z_1, z_2]_{t_T + 1}$ and

$T_1 = \begin{cases} [z_3, z_{03}, z_2, z_1] & \text{if } t_T = 0, \\ [z_3, z_{03}, z_1, z_2] & \text{if } t_T = 1, \\ [z_3, z_{03}, z_1, z_2] & \text{if } t_T = 2. \end{cases}$ (2.1)

The refinement edges of the children are $E_{T_0} = \frac{z_0 - z_2}{2}$ and $E_{T_1} = \frac{z_0 - z_3}{2}$, if $t_T \in \{0, 1, 2\}$.

From this we can directly deduce the initial refinement edge of all 4 faces of the simplex, namely

$F_0 = \{z_0, z_1, z_2\} : \quad E_{F_0} = \frac{z_0 - z_2}{2}$,

$F_1 = \{z_0, z_1, z_3\} : \quad E_{F_1} = \frac{z_0 - z_3}{2}$,

$F_2 = \{z_0, z_2, z_3\} : \quad E_{F_2} = \frac{z_0 - z_3}{2}$,

$F_3 = \{z_1, z_2, z_3\} : \quad E_{F_3} = \begin{cases} \frac{z_1 - z_2}{2} & \text{if } t_T = 0, \\ \frac{z_1 - z_2}{2} & \text{else.} \end{cases}$

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 et al. [5] 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 non-standard 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 non-degenerating simplices (see [23, 19]). 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 [24].

Definition 2.3 (Refinement Tree and NVB-equivalence). Let $T = [z_0, \ldots, z_d]_T$. 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 $z_0z_d$. We call this graph the Refinement Tree $RT(T)$ and call a $d$-simplex $T$ NVB-equivalent to a simplex $T'$, if $RT(T) = RT(T')$.

Example 2.4 (Refinement Tree for Type 0). Figure 2.1 shows the refinement tree of the simplex $T = [z_0, \ldots, z_d]_T$. One can easily see that this is also the refinement tree of $T' = [z_d, z_{d-1}, \ldots, z_1, z_0]_T$, so $T$ is NVB-equivalent to $T'$.

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.

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

Definition 2.5 ($d$-Skeleton of a triangulation). Let $T$ be a triangulation of a domain $\Omega \subset \mathbb{R}^d$. For $0 \leq i \leq d$ we denote by $T^i$ the set of all simplices of dimension $i$ that are contained in $T = T^d$ and call $T^i$ the $i$-skeleton of $T$. We also define $T^i(T)$ to be the $i$-skeleton of the triangulation that consists of the single element $T$. 

![Figure 2.1. Refinement edges of a $d$-simplex of type 0 under $d$ uniform refinements.](image-url)
In particular $T^0 = \mathcal{V}$ is the set of all vertices, $T^1 = \mathcal{E}$ the set of all edges and $T^{d-1} = \mathcal{F}$ the set of all faces. Note that $T^j$ is the $i$-skeleton of the $j$-times refined grid and not the $j$-times refined $i$-skeleton.

**Lemma 2.6 (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 $E \in T^1(T)$ gets bisected exactly once and no newly created edge gets bisected.
- Any sub-simplex $F \in T^{d-1}(T)$ is bisected by a valid NVB uniformly $d - 1$ times. It 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 2.1. The distance of the indices decreases by 1 each level, so any edge $\frac{z_i}{z_j}, i < j$ 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

$$F_1 = \text{conv}\{z_0, \ldots, z_{i-1}, z_{i+1}, \ldots, z_d\} \in T^{d-1}(T)$$

are refinement edges of the Refinement Tree of $T$. Moreover the children of the edge $\frac{z_i}{z_m}$ in the induced refinement tree of $F_1$ are $\frac{z_i}{z_{m-1}}$ (or $\frac{z_i}{z_{m-2}}$ if $m - 1 = i$) and $\frac{z_{i+1}}{z_m}$ (or $\frac{z_{i+2}}{z_m}$ if $n + 1 = i$). From this we can clearly see that $F_1$ is NVB-equivalent to $[z_0, \ldots, z_{i-1}, z_{i+1}, \ldots, z_d]_0$. □

**Remark 2.7.** This implies by induction that for $0 \leq i \leq d - 1$ all elements of $T^i(T)$ are valid NVB simplices of type 0.

We now proceed to present a variant of Lemma 2.6 for elements of type $t_T = 1$.

**Lemma 2.8 (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 $E \in T^1(T)$ gets bisected exactly once and no newly created edge gets bisected.
- Any sub-simplex $F \in T^{d-1}(T)$ 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.

**Proof.** The proof is very similar to the proof of Lemma 2.6

Figure 2.2 shows the refinement edges of $T$. The distance between the indices decreases by 1 each level until level $d - 2$ and that the final level consists of all edges containing $z^*$. So any edge $\frac{z_i}{z_j}, i < j$ is bisected exactly once at level $d - 1 - j + i$ and any edge $\frac{z_i}{z_j}$ 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

$$F_1 = \text{conv}\{z_0, z^*, \ldots, z_{i-1}, z_{i+1}, \ldots, z_d\} \in T^{d-1}(T)$$

are refinement edges of the Refinement Tree of $T$. Moreover the children of the edge $\frac{z_i}{z_m}$ in the induced refinement tree of $F_1$ are $\frac{z_i}{z_{m-1}}$ (or $\frac{z_i}{z_{m-2}}$ if $m - 1 = i$) and $\frac{z_{i+1}}{z_m}$ (or $\frac{z_{i+2}}{z_m}$ if $m + 1 = i$) for $m \neq n + 1$. For the case $m = n + 1$ the children are are $\frac{z^*}{z_m}$ and $\frac{z^*}{z_m}$. Finally for the case $n = i - 1, m = i + 1$ the children are $\frac{z^*}{z_m}$ and $\frac{z^*}{z_m}$. From this we can clearly see that $F_1$ is NVB-equivalent to $[z_0, z^*, \ldots, z_{i-1}, z_{i+1}, \ldots, z_d]_1$.

The sub-simplex $F' = \text{conv}\{z_0, z_1, \ldots, z_{d-1}\}$ is NVB-equivalent to $[z_0, z_1, \ldots, z_{d-1}]_0$ as every edge $E = \frac{z_i}{z_j}$ gets bisected at level $d - 1 - j + i$. This proves the second claim. □
Remark 2.9. We can iterate Lemma 2.8 in combination with Lemma 2.6 to see that for a $d$-simplex $T$ with $t_T = 1$ any lower dimensional sub-simplex $F \in T^i(T)$, $2 \leq i \leq d - 1$ it holds $t_F = 1$ if $z^* \in F$ and $t_F = 0$ else.

For types $T = 0, 1$ 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.

Lemma 2.10 (Type $\geq 2$). Let $T = [z_0, z_1, \ldots, z_t, z_1, \ldots, z_{d-1}]$ be a $d$-simplex of type $t_T = t \geq 2$ that is bisected $2d - t$ times uniformly and let $\mathcal{V}_i = \{z_{i1}, \ldots, z_{it}\}$.

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

Proof. Bisecting the element $T = [z_0, z_1, \ldots, z_t, z_1, \ldots, z_{d-1}]$ uniformly $d-t$ times leads to a set of descendants of type 0 that look as follows

$$T' = [\bar{z}_0, \bar{z}_1, \ldots, \bar{z}_{d-1}, \bar{z}_{t1}, \ldots, \bar{z}_{ti}]_0$$

where $\bar{z}_k$ are either vertices of $T$ or centers of refinement edges of $T$. This means that none of the refinement edges $E_{ij} = \bar{z}_i \bar{z}_j$, $1 < i < j < t$ has been refined yet.

From the proof of Lemma 2.6 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 $\bar{T} = [z_0, \ldots, z_{d}]_0$ by assuming that the simplex $T$ has been created by $t$ bisections always being the first child. From Lemma 2.6 we know that the ordering of all subsimplices of $\bar{T}$ coincides with the ordering of $T$ and this translates to the descendant. Furthermore $\bar{T}$ has been refined $t$ times introducing the vertices $\mathcal{V}_i$, which means that any subsimplex $T'$ has been refined as many times as it contains vertices from $\mathcal{V}_i$, which proves the second claim.
In particular for \( T = [z_0, z_1^*, \ldots, z_t^*, \ldots, z_{d-1}] \), any subsimplex \( T' \in \mathcal{T}^{d-1}(T) \) is of type \( t_{T'} = \# \{ z_i^* \mid z_i^* \in T' \} \) as the subsimplex of the artificial ancestor has been refined as many times.

**Remark 2.11.** Lemma 2.10 generalizes Remark 3.4 of [10].

The Lemmata 2.6, 2.8 and 2.10 lead to the following corollary by simple induction.

**Corollary 2.12 (Trace Meshes).** For \( 0 < i < d \) the meshes \( T_i \) 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 \( T^i(T) \) are valid NVB simplices.

### 2.3. Compatibility Condition.

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

**Definition 2.13 (Weakest Compatibility Condition).** A triangulation \( \mathcal{T} \) fulfills the weakest compatibility condition, 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 [24] or conformingly marked in [5] (for \( d = 3 \)). This condition is necessary, but may not be sufficient for the iterative NVB algorithm ([24]) to terminate. So we define a slightly stronger condition.

**Definition 2.14 (Weak Compatibility Condition).** A triangulation \( \mathcal{T} \) is called weakly compatible, if it fulfills condition 2.13 and there exists a finer Triangulation that is conforming and only contains elements of type \( t \in \{ 0, 1 \} \).

**Remark 2.15.** The recursive algorithm [22] does in general not terminate under these conditions, as neither condition 2.13 nor condition 2.14 imply that the grid is loop-free.

**Remark 2.16.** Any 2-dimensional triangulation is weakly compatible, as it only contains elements of type 0 and 1.

**Lemma 2.17 (Every \( d \)-th uniform refinement is conforming).** Let \( \mathcal{T} \) be conforming and only contain elements of type \( t \in \{ 0, 1 \} \). Then every \( d \)-th 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 2.6 and 2.8 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 \( \mathcal{T}(T) \). Condition 2.13 ensures that subsimplices that are shared by two elements carry the same refinement structure. So the resulting triangulation is conforming.

From Lemma 2.17 follows directly that under Condition 2.14 there always is a conforming finer grid, that only contains elements of type \( t \in \{ 0, 1 \} \), which leads to the following corollary.

**Corollary 2.18.** 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 strong compatibility condition in the remainder of this article we also introduce the notion of reflected neighbors and the strong compatibility condition.
Definition 2.19 (Reflected Neighbors). Two elements $T = \text{conv}\{z_0, \ldots, z_{d-1}, u\}$ and $T' = \text{conv}\{z_0, \ldots, z_{d-1}, v\}$ 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 2.20 (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 2.21 (Strong Compatibility Condition). A triangulation $\mathcal{T}$ is strongly compatible, if all faces $F \in \mathcal{T}^{d-1}$ are strongly compatible.

Remark 2.22. Condition 2.21 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 2.23 (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 behaviour 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 fulfil Definition 2.23.

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 Condition 2.14

1. 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.
2. For all $T = [z_0, \ldots, z_n] \in \mathcal{T}$ 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 $t_T = \#(T \cap \mathcal{V}_1) \mod d$.
3. 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 3.1. The resulting triangulation of Algorithm 3.1 is weakly compatible.

Proof. The weakest compatibility follows directly from Lemmas 2.6, 2.8, 2.10. For the second property of Definition 2.14 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 3.2. 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. \( V_0 = V, V_1 = \emptyset \)). This may be used to reconstruct a strongly compatible mesh in the standard way.

**ILE** Initial Longest Edge:
Let \( C_{ILE} \in \mathbb{N} \) be a constant threshold. Then we define
\[
V_0 = \{ v \in V : v \text{ is in at least } C_{ILE} \text{ Longest Edges} \}
\]
and \( V_1 = V \setminus V_0 \). This results in refining the initial longest edge first for some simplices.

**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
\[
V_1 = \{ v \in V : v \text{ is in at most } C_{LAE} \text{ Elements} \}
\]
and \( V_0 = V \setminus V_1 \). This results in refining opposite of the biggest angle for some simplices.

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 \( V_0 \) or collect the vertices with the maximum volume angle into \( V_1 \). It is also possible to combine strategies.

3.2. Variants of the Choice of Ordering. Up to now, we only constructed the sets \( V_{(0,1)} \). We are still free to choose the ordering. Figure 3.1 depicts different results of the algorithm 3.1 in 2d. The ordering in Figure 3.1b is lowest in the lower left corner and highest in the upper right and increases row-wise. This naive ordering leads to an expensive conforming closure. On the other hand in Figures 3.1b and 3.1c the orderings are created with alternative methods (SRN and SRN2) and lead to all neighbors being strongly compatible.

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 \( V \) successively. Start with an element \( T = [z_0, \ldots, z_n]_T \), 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: either insert the new vertex 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 compati-
able, as types can at most differ by 1 by construction and they create (quasi-)strongly
compatible faces opposite of every newly inserted vertex.

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 [5,7] 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 qual-
ity 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.

Strongly compatible grids fulfil 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 3.3** (Distance 1). Let \( T \) be a weakly compatible grid sorted by one of
the above methods and let \( F_{SC} \subset T^{d-1} \) be the set of faces that are strongly or
quasi-strongly compatible. Then we define the distance

\[
d_{1}^{T} = \#(T^{d-1}) - \#(F_{SC}).
\]

This is a simple, easily computable and straight-forward metric that directly tells
us whether the grid fulfils 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 3.4** (Distance 2). Let \( T \) be a weakly compatible grid sorted by one of
the above variants and let \( C_{T}(T) \) the conforming closure of refining \( T \) on the initial
grid \( T \). Then we define the metric

\[
d_{2}^{T} = \max_{T \in T} \#(C_{T}(T)).
\]

As the refinement becomes more local (cf. [8]) once the whole grid has been refined
d times uniformly, we also investigate \( d_{2}^{T_{d}} \) on the \( d \) times uniformly refined grid \( T_{d} \).
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 bisectios.
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.
Algorithm 3.2: Possible implementation of methods OT0 and SRN2

Data: List of active faces $F$, List of vertices $V$, Mesh $T$

1. $F = V = \emptyset$
2. Choose initial $T \in T$
3. Order according to announced refinement edge $E(T)$
4. $V = V(T)$
5. add all non-boundary $F \in T^{d-1}(T)$ with $E(T) \in F$ at the beginning of $F$
6. add the other non-boundary $F \in T^{d-1}(T)$ at the end of $F$ with the flag noRefEdge
7. mark $T$ as treated
8. while $\exists$ untreated $T \in T \leftrightarrow F \neq \emptyset$ do
9.   Get untreated neighbor $T'$ and treated element $T$ of first $F \in F$
10. $v = T^0(T) \setminus T^0(F)$  $v' = T^0(T') \setminus T^0(F)$
11. if $v' \in V$ then /* do nothing */
12.   else if noRefEdge set then /* In this case $v$ is the first or last vertex of $T$ */
13.      insert $v'$ after last or before first vertex of $T$, depending on the announced refinement edge
14.      else
15.         Insert $v'$ directly after $v$ into $V$
16.      end
17.  end
18.  sort $T'$ according to $V$
19.  mark $T'$ treated
20. for non-boundary $F' \in T^{d-1}(T)$ do
21.   if $F' \in F$ then /* possibly check for strong compatibility */
22.      remove $F'$ from $F$
23.   else
24.      if $E(T') \subset F'$ then
25.         add $F'$ at beginning of $F$
26.      else
27.         add $F$ at the end of $F$
28.      end
29.   end
30. end
31. end
32. end

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. In the implementation first a check is performed whether the provided grids is compatible or not. If the grid is 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 [2].
4.1. Threshold Study. The first set of experiments we conducted is a threshold study, where we measure the behaviour of LAE and ILE with threshold values $C_{ILE}, C_{LAE} \in \{0, \ldots, 35\}$. If the threshold value is 0 we obtain in both cases $V_0 = V$, 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, that have been used in the 3D Benchmark on Discretization Schemes for Anisotropic Diffusion Problems on General Grids \cite{9}. We will call these grids the "Unitcube Sequence".

![Series of tetrahedral grids discretizing the unit cube used in the 3D Benchmark on Discretization Schemes for Anisotropic Diffusion Problems on General Grids \cite{9}.

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 depiction is presented in Figure 4.2. The grid representing the head of a human has been used in \cite{16}. First we consider the distribution of $V_0, V_1$:

![Figure 4.3 shows the proportion of vertices that have been sorted into $V_0$ 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 $V$ into $V_0$ and $V_1$ does not depend on the choice of ordering. So the active regions do not change from SRN to SRN2.

We now investigate metric $d_T$, i.e. the amount 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 amount of not strongly compatible faces in the grid. There are a few things to note. In most cases SN2 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: $V/LE^3$, $V/SE^3$, $V/LF^{3/2}$, $V/SF^{3/2}$, $F/LE^2$, $F/SE^2$ and the d-sine \cite{8}. We observed that $(V/LE^3) \approx (V/LF^{2/3}) \times (F/LE^2)$ and additionally $(V/LF) \approx d$-sine. Qualitatively the metrics behave similar, so we choose to display only the d-sine here.
Figure 4.2. Realistic tetrahedral meshes used for studying the proposed algorithms. 4.2a has been previously used for numerical simulation of atherosclerotic plaque formation [13], 4.2b has been used in [16] for electrical impedance tomography simulations, 4.2c has been downloaded from [1] and generated with TetGen, 4.2d is part of the test grids in the DUNE-GRID module [6], 4.2e stems from a Kuhn-grid and has been modified (vertex renumbering and projection), and 4.2f has been generated with Gmsh.

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
Figure 4.3. The proportion of vertices sorted into $V_0$ over different thresholds.

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

Figure 4.5. Average $d$-sine region, where it is not strongly compatible anymore. 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 SRN versus SRN2 does not seem to have much impact.

A way to investigate the minimum angle is to consider the maximum number 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, 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 decorstatue grid 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 $d^1_T$, $d^2_T$, and $d^3_T$. Unfortunately the computation cost of $d^2_T$ is at least $O(n^2)$ and $d^3_T$ 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 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. 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 becomes strongly compatible. Figure 4.8 actually displays $d^T_1$ and $d^T_3$, i.e. the maximum conforming closure for both grids. Both times we measure both on the initial grid (blue) and on the 3 times uniformly refined grid (red). 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 applied the various algorithms to 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.
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 to use (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 behaviour.

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 to use (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 behaviour.
It is still an open question whether the weak compatibility condition is enough to obtain an estimate like the one in [24] for the computational effort of the conforming closure or if this can be achieved with some extra mild conditions in the mesh.

Acknowledgements

The author Martin Alkämper would like to thank the German Research Foundation (DFG) for financial support of the project within the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart.

Fernando Gaspoz acknowledges the German Research Foundation (DFG) for financial support of the project DFG 814/7-1 at the University of Stuttgart.

Robert Klöfkorn acknowledges the Research Council of Norway and the industry partners: ConocoPhillips Skandinavia AS, Aker BP ASA, Eni Norge AS, Maersk Oil Norway AS, DONG Energy A/S, Denmark, Statoil Petroleum AS, ENGIE E&P NORGE AS, Lundin Norway AS, Halliburton Norge AS, Schlumberger Norge AS, Wintershall Norge AS, DEA Norge AS of The National IOR Centre of Norway for support.

References

[1] 3D Meshes Research Database by INRIA GAMMA Group. URL: https://www.rocq.inria.fr/gamma/gamma/download/download.php

[2] Martin Alkämper, Andreas Dedner, Robert Klöfkorn, and Martin Nolte. The DUNE-ALUGrid module. Archive of Numerical Software, 4(1):1–28, 2016. doi:10.11588/ans.2016.1.23925

[3] Martin Alkämper and Robert Klöfkorn. Distributed Newest Vertex Bisection. Journal of Parallel and Distributed Computing, 104:1 – 11, 2017. doi:10.1016/j.jpdc.2016.12.003

[4] Guillermo Aparicio, Leocadio G. Casado, Eligius M. T. Hendrix, Boglárka G.-Tóth, and Inmaculada Garcia. On the minimum number of simplex shapes in longest edge bisection refinement of a regular n-simplex. Informatica (Vilnius), 26(1):17–32, 2015. doi:10.15388/Informatica.2015.36

[5] Douglas N. Arnold, Arup Mukherjee, and Luc Poully. Locally Adapted Tetrahedral Meshes Using Bisection. SIAM J. Sci. Comput., 22(2):431–448, 2000. doi:10.1137/S1064827597323373

[6] M. Blatt, A. Burchardt, A. Dedner, Ch. Engwer, J. Fahlke, B. Flemisch, Ch. Gersbacher, C. Grser, F. Gruber, Ch. Grninger, D. Kempf, R. Klöfkorn, T. Malkmus, S. Mthing, M. Nolte, M. Piatkowski, and O. Sander. The Distributed and Unified Numerics Environment, Version 2.4. Archive of Numerical Software, 4(100):13–29, 2016. doi:10.11588/ans.2016.100.26526

[7] J. Manuel Cascon, Christian Kreuzer, Ricardo H. Nochetto, and Kunibert G. Siebert. Quasi-optimal convergence rate for an adaptive finite element method. SIAM J. Numer. Anal., 46(5):2524–2550, 2008. doi:10.1137/07069047X

[8] Folke Eriksson. The law of sines for tetrahedra and n-simplices. Geometriae Dedicata, 7(1):71–80, Mar 1978. doi:10.1007/BF00181352

[9] R. Eymard, G. Henry, R. Herbin, F. Hubert, R. Klöfkorn, and G. Manzini. 3D Benchmark on Discretization Schemes for Anisotropic Diffusion Problems on General Grids. In J. Fort, J. Fürt, J. Halama, R. Herbin, and F. Hubert, editors, Finite Volumes for Complex Applications VI Problems & Perspectives, volume 4 of Springer Proceedings in Mathematics, pages 895–930. Springer Berlin Heidelberg, 2011. doi:10.1007/978-3-642-20671-9_89

[10] Dietmar Gallistl, Mira Schedensack, and Rob P. Stevenson. A remark on newest vertex bisection in any space dimension. Comput. Methods Appl. Math., 14(3):317–320, 2014. doi:10.1515/cmam-2014-0013

[11] Fernando D. Gaspoz, Claus Justus Heine, and Kunibert G. Siebert. Optimal grading of the newest vertex bisection and $H^1$-stability of the $L_2$-projection. IMA J. Numer. Anal., 36(3):1217–1241, 2016. doi:10.1093/imanum/drv044

[12] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. International Journal for Numerical Methods in Engineering, 79(11):1309–1331, 2009. doi:10.1002/nme.2579

[13] S. Girke, R. Klöfkorn, and M. Ohlberger. Efficient Parallel Simulation of Atherosclerotic Plaque Formation Using Higher Order Discontinuous Galerkin Schemes. In J. Fuhrmann et al., editor, Finite Volumes for Complex Applications VII, volume 78 of Springer Proceedings in Mathematics & Statistics, pages 617–625. Springer, 2014. doi:10.1007/978-3-319-05591-6_61
[14] Antti Hannukainen, Sergey Korotov, and Michal Krížek. On global and local mesh refinements by a generalized conforming bisection algorithm. *J. Comput. Appl. Math.*, 235(2):419–436, 2010. doi:10.1016/j.cam.2010.05.046
[15] Reiner Horst. On generalized bisection of n-simplices. *Math. Comp.*, 66(218):691–698, 1997. doi:10.1090/S0025-5718-97-00809-0
[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 Transactions on Biomedical Engineering*, 62(1):126–137, 2015. doi:10.1109/TBME.2014.2342280
[17] S. Korotov, M. Krížek, 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(9):1687–1698, 2008. doi:10.1134/S0965542508090170
[18] I. Kossaczký. A recursive approach to local mesh refinement in two and three dimensions. *J. Comput. Appl. Math.*, 55:275–288, 1994. doi:10.1016/0377-0427(94)90034-5
[19] Joseph M. Maubach. Local bisection refinement for n-simplicial grids generated by reflection. *SIAM J. Sci. Comput.*, 16(1):210–227, 1995. doi:10.1137/0916014
[20] William F. Mitchell. 30 years of newest vertex bisection. *AIP Conference Proceedings*, 1738(1), 2016. doi:10.1063/1.4951755
[21] María-Cecilia Rivara. New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations. *Internat. J. Numer. Methods Engrg.*, 40(18):3313–3324, 1997. doi:10.1002/(SICI)1097-0207(19970930)40:18<3313::AID-NME214>3.0.CO;2-R
[22] Alfred Schmidt and Kunibert G. Siebert. *Design of adaptive finite element software*, volume 42 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 2005. The finite element toolbox ALBERTA, With 1 CD-ROM (Unix/Linux).
[23] Hang Si. TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator. *ACM Trans. on Mathematical Software*, 41(2), 2015. doi:10.1145/2629697
[24] Rob Stevenson. The completion of locally refined simplicial partitions created by bisection. *Math. Comp.*, 77(261):227–241 (electronic), 2008. doi:10.1090/S0025-5718-07-01959-X
[25] C. T. Traxler. An algorithm for adaptive mesh refinement in n dimensions. *Computing*, 59:115–137, 1997. doi:10.1007/BF02684475

URL: [www.ians.uni-stuttgart.de/mnh/](http://www.ians.uni-stuttgart.de/mnh/)
E-mail address: alkaemper@ians.uni-stuttgart.de
E-mail address: fgaspoz@ians.uni-stuttgart.de
E-mail address: Robert.Kloefkorn@iris.no

A Institut für Angewandte Analysis und Numerische Simulation, Fachbereich Mathematik, Universität Stuttgart, Pfaffenwaldring 57, D-70569 Stuttgart, Germany

B International Research Institute of Stavanger, Thormeulensgt. 55, 5006 Bergen, Norway