BISECTING WITH OPTIMAL SIMILARITY BOUND ON 3D UNSTRUCTURED CONFORMAL MESHES

Guillem Belda-Ferrín[1] Eloi Ruiz-Gironés[1] Xevi Roca[1,2]

[1] *Computer Applications in Science and Engineering, Barcelona Supercomputing Center - BSC, 08034 Barcelona, Spain*
[2] *Corresponding author: xevi.roca@bsc.es*

ABSTRACT

We propose a new method to mark for bisection the edges of an arbitrary three-dimensional unstructured conformal mesh. For these meshes, the approach conformingly marks all the tetrahedra with coplanar edge marks. To this end, the method needs three key ingredients. First, we propose a specific edge ordering. Second, marking with this ordering, we guarantee that the mesh becomes conformingly marked. Third, we also ensure that all the marks are coplanar in each tetrahedron. To demonstrate the marking method, we implement an existent marked bisection approach. Using this implementation, we mark and then locally refine three-dimensional unstructured conformal meshes. We conclude that the resulting marked bisection features an optimal bound of 36 similarity classes per tetrahedron.

Keywords: bisection, marked bisection, newest vertex bisection

1. INTRODUCTION

In adaptive finite element analysis, unstructured tetrahedral meshes have to be locally adapted. To this end, one needs to perform local mesh modifications. One successful modification is to bisect the required tetrahedra. This bisection operation splits a tetrahedron by introducing a new vertex on the selected refinement edge. Then, the vertices not lying on this refinement edge are connected to the new vertex. These connections determine two new tetrahedra. The quality of these tetrahedra depends on the criterion to select refinement edges. This edge selection is commonly based on choosing either the longest edge [1, 2, 3, 4] or the newest vertex [5, 6].

The self-similarity of the newest vertex bisection [5, 6] has been favored over other bisection-based refinements in many three-dimensional applications. Specifically, in adaptive applications [7, 8, 9, 10] where it is possible to start with a three-dimensional reflected mesh [9, 10, 11, 12]. This preference is so since on reflected meshes local refinement with newest vertex bisection has the minimum mesh quality bounded. This bound is a consequence of the bound in the number of similarity classes of mesh tetrahedra. That is, for each initial tetrahedron, successive newest vertex bisection does not generate more than 36 different similarity classes [13]. This number of classes is the smallest bound known for a three-dimensional bisection method.

Unfortunately, this optimal bound has not been met in adaptive applications requiring complex three-dimensional geometries. This lack is so since practical methods to extract a reflection structure are limited to specific meshes. For instance, meshes generated using the Coxeter-Freudenthal-Kuhn algorithm [14, 15, 16] or meshes where all the edges have an even number of incident tetrahedra [9, 10]. Currently, there is no method known to extract a strong reflection structure from an arbitrary three-dimensional unstructured conformal mesh [8, 10, 13, 11, 12].

For three-dimensional unstructured conformal meshes,

there are alternative bisection methods with sub-optimal similarity bound [17, 7, 18, 19, 13]. All these methods lead to analogous locally refined conformal meshes. Nevertheless, Arnold *et al.* [13] establish a key connection between Maubach's newest vertex bisection [8] and marked bisection [13]. Marked bisection leads at most to 72 similarity classes. This bound is two times the number of similarity classes of newest vertex bisection.

To meet this sub-optimal bound, marked bisection [13] features one pre-process stage and two bisection stages. In the pre-processing, for all the faces of the initial mesh, the method conformingly marks the bisection edges. These edge marks determine a finite set of marked tetrahedron types. For each type, there is a specific bisection that leads to two children tetrahedra of the next type. The first stage ensures that different types of tetrahedra are all bisected to the *planar* type. This type, independently for each tetrahedron, is the beginning of the next bisection stage. In this second stage, successive marked bisection cycles every three bisection steps through a subset of the marked types (*unflagged planar*, *flagged planar*, and *adjacent*). This cyclic stage is equivalent to Maubach's newest vertex bisection [8] under specific conditions [13].

The previous overview allows reasoning about the number of similarity classes. On the one hand, the number potentially doubles the bound for the newest vertex bisection due to the initial bisection stage. On the other hand, the cyclic stage guarantees that the rest of the generated similarity classes correspond to those determined by the newest vertex bisection. Accordingly, for some conformingly marked meshes, marked bisection behaves as the newest vertex bisection [13]. Specifically, there are no more than 36 similarity classes if the conformingly marked mesh is composed only of unflagged planar or adjacent tetrahedra.

The question of whether there is a method to conformingly mark as unflagged planar or as adjacent all the tetrahedra of an arbitrary three-dimensional unstructured conformal mesh is still open [13]. A constructive answer is of significant interest. It would lead to the first marked bisection featuring an optimal similarity bound for adaption in complex geometry. The main goal of this work is to answer this question and implement the obtained method.

To meet the goal, our main contribution is to propose a new marking procedure for three-dimensional unstructured conformal meshes. For these meshes, we guarantee that all the tetrahedra become conformingly marked as unflagged planar. To this end, we consider three key ingredients. First, we propose a specific ordering of the global mesh edges. Second, relying on this edge ordering, we deduce that all the mesh tetrahedra become marked as unflagged planar. Third, we guarantee conformingly marked meshes by checking that we fulfill the sufficient conditions for tetrahedral meshes stated in [13]. To illustrate the application, we implement the refine to conformity marked bisection [13] but equipped with our planar marking method. We use the implementation to locally refine three-dimensional unstructured conformal meshes and check the minimum mesh quality.

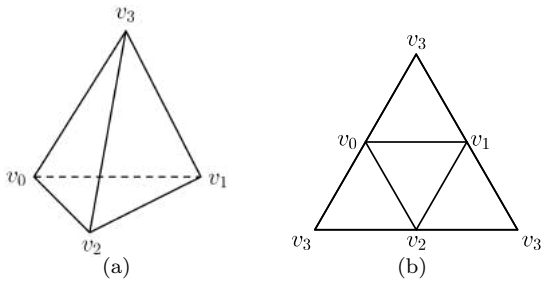


Figure 1: Representations of a tetrahedron composed of the vertices v_1, v_2, v_3, and v_4: (a) volumetric; and (b) planar.

The rest of the paper is structured as follows. In Section 2, we state the problem. In Section 3, we propose a marking process and show that it generates conformingly marked meshes. Next, in Section 4, we detail the adaptation of Arnold's bisection algorithm to our proposed marking process. In Section 5, we present several examples to show the features of the proposed method. Finally, in Section 6, we detail the conclusions and the future work.

2. PRELIMINARIES AND PROBLEM

We proceed to introduce the necessary notation and concepts. Specifically, we introduce the preliminaries related to conformal simplicial meshes and marked bisection. Finally, we state the problem of conformingly marking unstructured simplicial meshes for bisection.

2.1 Preliminaries: definitions

A *simplex* is the convex hull of $n+1$ points $p_0, \ldots, p_n \in \mathbb{R}^n$ that do not lie in the same hyper-plane. We denote it as $\sigma = conv(p_0, p_1, \ldots, p_n)$. We identify each point p_i with an unique integer identifier v_i that we refer as *vertex*. Thus, a simplex is composed of $n+1$ vertices and we denote it as $\sigma = (v_0, v_1, \ldots, v_n)$. We have an application Π that maps each identifier v_i to the corresponding point such that $\Pi(v_i) = p_i$. In our application, we are interested in tetrahedra, three-dimensional simplices. Herein, as in [13], we represent volumetric tetrahedra composed of the vertices v_1, v_2, v_3, and v_4, see Figure 1(a), in the plane by cutting

and unfolding the corresponding triangular faces, see Figure 1(b).

A tetrahedron has three types of entities: triangles, edges, and vertices, which are sub-simplices composed of 3, 2, and 1 vertices of τ, respectively. We denote the faces, edges and vertices with the letters κ, e and v, respectively. We define the list of local edges of a tetrahedron $\tau = (v_0, v_1, v_2, v_3)$ as the following sorted list of edges

$$(v_0, v_1), (v_0, v_2), (v_0, v_3), (v_1, v_2), (v_1, v_3), (v_2, v_3).$$

We associate each triangular face of a tetrahedron τ with the opposite face to a vertex of τ. As an example, for the tetrahedron $\tau = (v_0, v_1, v_2, v_3)$, the opposite face to the vertex v_0 is the triangular face $\kappa_0 = (v_1, v_2, v_3)$.

A *mesh*, $\mathcal{T}$, associated to an open set $\Omega \in \mathbb{R}^n$ is a finite collection of mutually disjoint tetrahedra such that

$$\bar{\Omega} = \bigcup_{\tau \in \mathcal{T}} \tau.$$

A tetrahedral mesh is *conformal* if, for any $\tau_1, \tau_2 \in \mathcal{T}$, $\tau_1 \cap \tau_2$ is either empty, or a common edge, or a common triangle. We say that two tetrahedra τ_1 and τ_2 are *neighbors* if they share a common triangular face.

2.2 Preliminaries: marked bisection

Arnold *et al.* [13] presented a marked bisection algorithm for unstructured conformal tetrahedral meshes that ensure locally refined conformal meshes and quality stability. Following, we present the terminology and results required to overview their marked bisection algorithm.

The *refinement edge* e_τ is the edge of τ to be bisected. Since an edge is shared by two triangular faces of the tetrahedron, the triangular faces that contain e_τ are the *refinement faces* of τ. The remaining two triangular faces are defined as *non-refinement faces*. For those faces, one edge, referred as *marked edge*, is assigned. We recall that each triangular face κ_i has a refinement edge e_{κ_i}. Particularly, the refinement edge e_τ is the same as the e_{κ_i} of the refinement faces.

Since the non-refinement edges are adjacent or either opposite to the refinement edge, we can classify the marked tetrahedra into four types, see Figure 2: adjacent A, planar P, mixed M, and opposite O.

- Planar, P: the refinement edge and the marked edges are coplanar. A planar tetrahedron is further classified as type P_u or type P_f, according to a boolean flag, see Figures 2(a), and 2(b), respectively.

Algorithm 1 Refining a subset of a mesh.

input: `Mesh` $\mathcal{T}$, set of element $\mathcal{S} \subset \mathcal{T}$ to refine
output: `ConformalMarkedMesh` $\mathcal{T}_2$

```
1: function refineMesh(T, S)
2:     T1 = markMesh(T)
3:     T2 = localRefine(T1, S)
4:     return T2
5: end function
```

- Adjacent, A: the marked edges are adjacent to the refinement edge but are not coplanar, see Figures 2(c).
- Mixed, M: one marked edge is adjacent to the refinement edge, and the other is opposite, see Figures 2(d).
- Opposite, O: both marked edges are opposite to the refinement edge, see Figures 2(e).

These tetrahedron types are the nodes of the directed graph determining the marked bisection sequence, see Figure 3.

Now, we can introduce the definition of *marked tetrahedron*, which is a modification of the one detailed in [13]. Herein, a marked tetrahedron is the 5-tuple

$$\rho = (\tau, e_\tau, e_{\kappa_1}, e_{\kappa_2}, t),$$

where τ is a tetrahedron, e_τ is the refinement edge, e_{κ_1} and e_{κ_2} are the marked edges of the non-refinement faces, and t is the tetrahedron type.

A mesh is *marked* if all its tetrahedra are marked. A marked conformal mesh is *conformingly marked* if each triangular face has a unique marked edge. That is, a triangular face shared by two tetrahedra has the same marked edge from both sides. Accordingly, shared triangular faces are bisected in the same manner from different tetrahedra.

Remark 2.1 (Conditions to conformingly mark). To guarantee that a conformal mesh is conformingly marked, Arnold *et al.* [13] state that it is sufficient to combine a strict total order of the mesh edges with their marking process for tetrahedra. For instance, the mesh edges can be sorted according to their length using a tie-breaking rule when the lengths are equal.

The marked bisection method, Algorithm 1, starts by marking the initial unstructured conformal mesh and then applies a local refinement procedure to a subset of tetrahedra of the marked mesh. The marking preprocess is devised to ensure a conformingly marked mesh. Using this marked mesh, the local refinement procedure, Algorithm 2, first refines a set of tetrahedra and then calls a recursive refine-to-conformity strategy. This strategy, Algorithm 3, terminates when

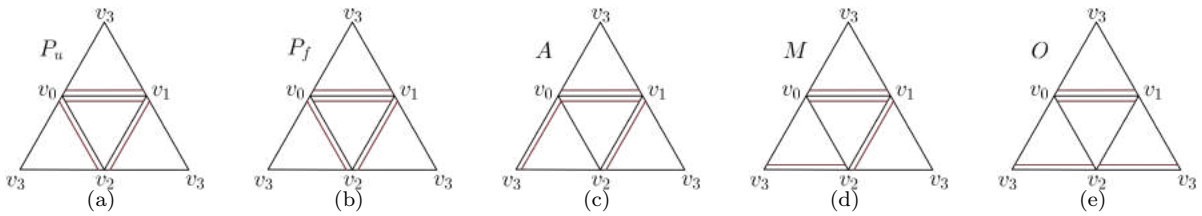


Figure 2: The five different type of marked tetrahedra of Arnold's cycle: (a) unflagged planar tetrahedron, (b) flagged planar tetrahedron, (c) adjacent tetrahedron, (d) mixed tetrahedron, and (e) opposite tetrahedron.

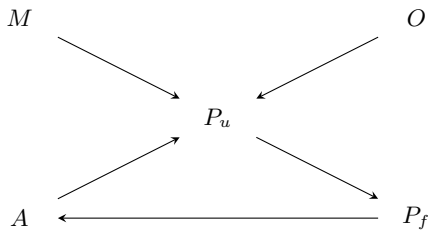


Figure 3: Directed graph of tetrahedron types for marked bisection.

Algorithm 2 Local refinement of a marked mesh.

input: `ConformalMarkedMesh` $\mathcal{T}$ and $\mathcal{S} \subset \mathcal{T}$
output: `ConformalMarkedMesh` $\mathcal{T}_0$
function `localRefine`($\mathcal{T}$, $\mathcal{S}$)
  $\overline{\mathcal{T}} =$ `bisectTetrahedra`$(\mathcal{T}, \mathcal{S})$
  $\mathcal{T}_0 =$ `refineToConformity`$(\overline{\mathcal{T}})$
  return $\mathcal{T}_0$
end function

Algorithm 3 Refine-to-conformity a marked mesh.

input: `MarkedMesh` $\mathcal{T}$
output: `MarkedMesh` $\mathcal{T}'$ without hanging nodes
function `refineToConformity`($\mathcal{T}$)
  $\mathcal{S} =$ `getHangingNodes`$(\mathcal{T})$
  if $\mathcal{S} \neq \emptyset$ **then**
    $\overline{\mathcal{T}} =$ `bisectTetrahedra`$(\mathcal{T}, \mathcal{S})$
    $\mathcal{T}' =$ `refineToConformity`$(\overline{\mathcal{T}})$
  else
    $\mathcal{T}' = \mathcal{T}$
  end if
  return $\mathcal{T}'$
end function

successive bisection leads to a conformal mesh. Both algorithms use marked bisection to refine a set of elements, see Algorithm 4.

Remark 2.2 (Optimal similarity bound). If the conformingly marked mesh is composed only of unflagged planar or adjacent tetrahedra, marked bisection does not generate more than 36 similarity classes [13].

Algorithm 4 Bisect a set of tetrahedra.

input: `MarkedMesh` $\mathcal{T}$, `SetSimplices` $\mathcal{S}$
output: `MarkedMesh` $\mathcal{T}_1$
function `bisectTetrahedra`($\mathcal{T}$, $\mathcal{S}$)
  $\mathcal{T}_1 = \emptyset$
  for $\rho \in \mathcal{T}$ **do**
    if $\rho \in \mathcal{S}$ **then**
      $\rho_1, \rho_2 =$ `bisectTet`(ρ)
      $\mathcal{T}_1 = \mathcal{T}_1 \cup \rho_1$
      $\mathcal{T}_1 = \mathcal{T}_1 \cup \rho_2$
    else
      $\mathcal{T}_1 = \mathcal{T}_1 \cup \rho$
    end if
  end for
  return $\mathcal{T}_1$
end function

2.3 Problem

Our problem is to conformingly mark an unstructured conformal tetrahedral mesh $\mathcal{T}_1$ exclusively with tetrahedra of type P_u. Thus, when applying successive marked bisection, starting on the resulting marked $\mathcal{T}_1$, we can guarantee an optimal number of similarity classes, see Remark 2.2. Specifically, starting on the unflagged planar mesh $\mathcal{T}_1$, if we locally refine a set of elements $\mathcal{S}$, we obtain a new conformal unstructured marked tetrahedral mesh $\mathcal{T}_2$ with the corresponding elements bisected. The marked mesh $\mathcal{T}_2$ is suitable for a posterior local refinement. Furthermore, any successive local refinement process has the minimum element quality bounded.

3. SOLUTION: CONFORMINGLY MARKING AS UNFLAGGED PLANAR

Following, we detail our solution to conformingly mark an unstructured conformal mesh with unflagged planar tetrahedra. To this end, we first introduce the concept of consistent bisection edge. This concept

Algorithm 5 Marking as unflagged planar.

input: `Tetrahedron` τ
output: `MarkedTetrahedron` ρ

1: **function** `markTetrahedron`(τ)
2: e_τ = `consistentBisectionEdge`(τ)
3: $(v_0, v_1) = e_\tau$
4: κ_1 = `oppositeFace`(τ, v_0)
5: κ_2 = `oppositeFace`(τ, v_1)
6: e_{κ_1} = `consistentBisectionEdge`(κ_1)
7: e_{κ_2} = `consistentBisectionEdge`(κ_2)
8: $t = P_u$ ▷ Initialize type of tetrahedron
9: $\rho = (\tau, e_\tau, e_{\kappa_1}, e_{\kappa_2}, t)$
10: **return** ρ
11: **end function**

ensures that we can always select the same bisection edge for a given simplex, independently of its dimension. Based on this selection, we propose an element-based marking process that generates unflagged planar tetrahedra. We also check that our marking process is equivalent to the standard face-based marking process proposed in [13]. Finally, we guarantee that our marking process leads to a conformingly marked mesh. Accordingly, if we use a restricted version of standard marked bisection to refine the resulting marked mesh, we obtain the optimal number of similarity classes.

3.1 Marking edges: strict total order

To mark the mesh edges, we propose a strict total order of the mesh edges. To this end, we use a lexicographic order for the mesh edges that is inherited from the order of the vertices. Specifically, we say that the mesh edge $e_i = (v_{i_1}, v_{i_2})$ has lower global index than the mesh edge $e_j = (v_{j_1}, v_{j_2})$ if $v_{i_1} < v_{j_1}$, or $v_{i_1} = v_{j_1}$ and $v_{i_2} < v_{j_2}$. Note that the proposed lexicographic order is strict and total since it is straight-forward to check that is irreflexive, transitive, asymmetric, and connected. Using this lexicographic order, we identify each mesh edge with a unique integer by sorting all the existing edges of the mesh according to the global index criterion.

The *consistent bisection edge* of a simplex (tetrahedron or triangle) is the edge with the lowest integer assigned in the edge ordering process. Note that the consistent bisection edge of a simplex is unique because we use a strict total order to characterize it.

3.2 Marking tetrahedra: unflagged planar

Using the consistent bisection edge, we propose a marking process of a single tetrahedron that leads to a marked tetrahedron of type P_u, see Algorithm 5. The input of the function is a tetrahedron $\tau = (v_0, v_1, v_2, v_3)$ and the output is the corresponding

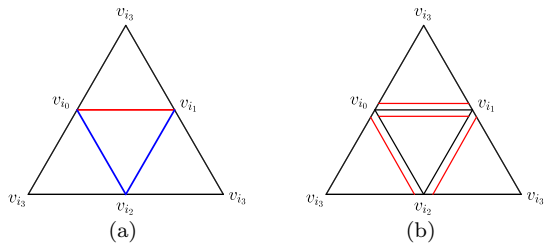


Figure 4: Marked tetrahedra with: (a) our element-based marking method; and (b) the standard face-based marking method.

marked tetrahedron ρ. First, we obtain the consistent bisection edge, e_τ, of the tetrahedron, see Line 2. Then, we obtain the opposite triangular faces of the vertices of the bisection edge e_τ, see Lines 4–5. After that, we obtain the corresponding consistent bisection edges e_{κ_1} and e_{κ_2} of κ_1 and κ_2, see Lines 6–7. Finally, we initialize the tetrahedron type, Line 8, as $t = P_u$.

The proposed marking process always generates an unflagged planar tetrahedron. To check it, we need to ensure that the consistent bisection edges selected in Algorithm 5 define a triangle of the tetrahedron. Let $\tau = (v_0, v_1, v_2, v_3)$ be a tetrahedron and let us reorder the vertices to have $v_{i_0} < v_{i_1} < v_{i_2} < v_{i_3}$. The consistent bisection edge is $e_\tau = (v_{i_0}, v_{i_1})$ since this is the edge with the lowest indices. The opposite faces to e_τ are $\kappa_1 = (v_{i_1}, v_{i_2}, v_{i_3})$ and $\kappa_2 = (v_{i_0}, v_{i_2}, v_{i_3})$, respectively. For those faces, the consistent bisection edges are $e_{\kappa_1} = (v_{i_0}, v_{i_2})$ and $e_{\kappa_2} = (v_{i_1}, v_{i_2})$, respectively. Since e_τ, e_{κ_1} and e_{κ_2} are connected generating the triangle $(v_{i_0}, v_{i_1}, v_{i_2})$, they define a planar configuration. Figure 4(a) shows the obtained marked tetrahedron, where the red edge is the refinement edge and the blue edges are the marked edges corresponding to the non-refinement faces.

We can see that our element-based marking method and the standard face-based one are equivalent, see Figure 4. Note that they might not be equivalent since our marking procedure does not exactly proceed as the standard procedure. Specifically, we do not explicitly mark all the triangular faces of a tetrahedron, see Figure 4(a). In the standard approach, each face of a tetrahedron has a marked edge that indicates which edge has to be bisected, see red edges in Figure 4(b). The refinement edge of the tetrahedron is the only edge that has been marked on both adjacent faces. Thus, after marking all the faces, we obtain that the marked edges of the faces

$$\kappa_1 = (v_{i_0}, v_{i_1}, v_{i_2}), \quad \kappa_2 = (v_{i_0}, v_{i_1}, v_{i_3}),$$
$$\kappa_3 = (v_{i_0}, v_{i_2}, v_{i_3}), \quad \kappa_4 = (v_{i_1}, v_{i_2}, v_{i_3}),$$

are $e_{\kappa_1} = (v_{i_0}, v_{i_1})$, $e_{\kappa_2} = (v_{i_0}, v_{i_1})$, $e_{\kappa_3} = (v_{i_0}, v_{i_2})$

Algorithm 6 Conformingly marking a mesh.

input: `ConformalMesh` $\mathcal{T}$
output: `ConformalMarkedMesh` $\mathcal{T}'$

```
1: function markMesh(T)
2:     T' = ∅
3:     for τ ∈ T do
4:         ρ = markTetrahedron(τ)
5:         T' = T' ∪ ρ
6:     end for
7:     return T'
8: end function
```

and $e_{\kappa_4} = (v_{i_1}, v_{i_2})$, respectively. Therefore, the refinement edge of τ is $e_\tau = e_{\kappa_1} = e_{\kappa_2}$ and the refinement faces are κ_1 and κ_2. The faces κ_3 and κ_4 are the non-refinement faces and their marked edges are e_{κ_3} and e_{κ_4}, respectively. Thus, all the triangular faces are also marked as an unflagged planar tetrahedron. The edge that is marked from two triangular faces corresponds to the refinement edge of the tetrahedron, see Figure 4(b). Thus, the refinement edge and the marked edges obtained with our marking process are equivalent to those obtained with the standard marking process but equipped with our edge ordering. That is, both marking methods generate an equivalent unflagged planar tetrahedron.

3.3 Conformingly marking a mesh

To ensure that we obtain a conformingly marked mesh, we need that our marking procedure fulfills the sufficient conditions required in Remark 2.1. The first condition is fulfilled since our ordering for mesh edges is strict and total. Furthermore, we know that our element-based marking process is equivalent to the standard face-based marking process. Since both sufficient conditions are fulfilled, we can guarantee that the marking process in Algorithm 5 leads to conformingly marked meshes.

Now, we can detail the method to conformingly mark an unstructured conformal tetrahedral mesh, see Algorithm 6. The input is a conformal tetrahedral mesh, $\mathcal{T}$, and the output is a conformingly marked tetrahedral mesh, $\mathcal{T}'$. We initialize an empty marked mesh and generate a marked tetrahedron ρ for each tetrahedra τ of the mesh $\mathcal{T}$. Then, we insert the marked tetrahedra into the marked mesh $\mathcal{T}'$. Finally, we return the conformingly marked mesh $\mathcal{T}'$ after marking all the tetrahedra.

4. RESTRICTED MARKED BISECTION

To bisect our unflagged planar meshes, we consider a restricted version of the standard marked bisection, see Algorithm 7. The restricted method bisects a tetrahedron according to its type. Moreover, it only needs to consider the bisection cycle of length three for the tetrahedron types P_u, P_f, and A, see Figure 5. In the first case, Line 4, we bisect an unflagged planar tetrahedron. In the second case, Line 6, we bisect a flagged planar tetrahedron. Finally, in the third case, Line 8, we bisect an adjacent tetrahedron.

Algorithm 7 Restriced marked bisection.

input: `MarkedTetrahedron` ρ
output: `MarkedTetrahedron` ρ_1, `MarkedTetrahedron` ρ_2

```
1: function bisectTet(ρ)
2:     t = type(ρ)
3:     if t is P_u then
4:         ρ1, ρ2 = bisectUnflaggedPlanar(ρ)
5:     else if t is P_f then
6:         ρ1, ρ2 = bisectFlaggedPlanar(ρ)
7:     else if t is A then
8:         ρ1, ρ2 = bisectAdjacent(ρ)
9:     end if
10:    return ρ1, ρ2
11: end function
```

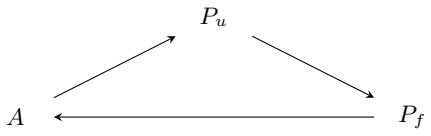


Figure 5: Restricted bisection cycle starting on unflagged planar type.

Figure 6 shows how to assign the refinement edge and the marked edges of the children after bisecting a marked tetrahedron of the proposed refinement cycle, according to standard marked bisection. Without loss of generality, we suppose that in all the cases the refinement edge is $e_\tau = (v_0, v_1)$. The vertex ν is the new vertex after the bisection of the edge e_τ. We colored the refinement edge and the marked edges with red and blue, respectively. The first column corresponds to a marked tetrahedron, and the second and third columns correspond to the left and right children, respectively. In rows, we have three different cases. The first row corresponds to the bisection of an unflagged planar tetrahedron to two flagged planar tetrahedra. The second row corresponds to the bisection of a flagged planar tetrahedron to two adjacent tetrahedra. Finally, the third row corresponds to the bisection of an adjacent tetrahedron to two unflagged planar tetrahedra.

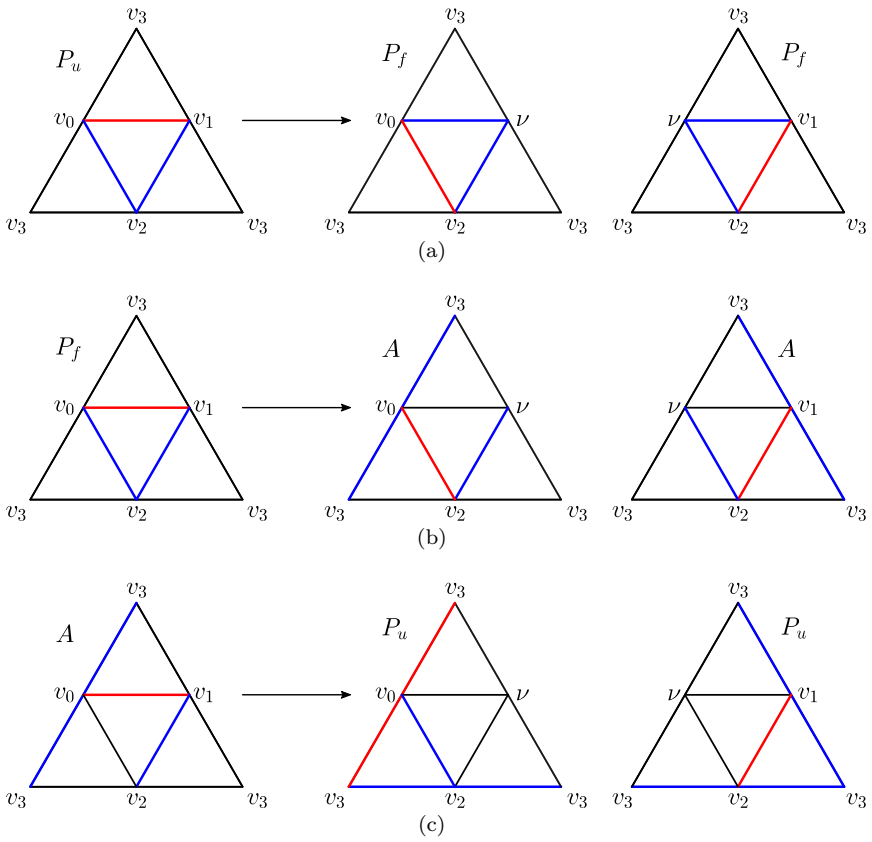


Figure 6: Cases for restricted marked bisection: (a) from a P_u to two P_f; (b) from a P_f to two A; and (c) from a A to two P_u.

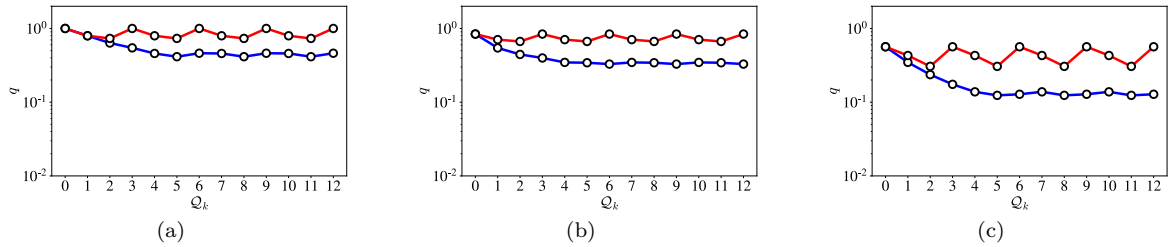
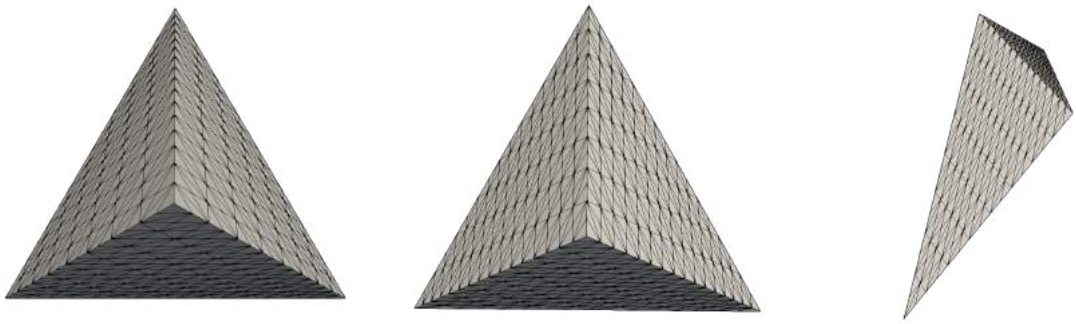


Figure 7: Evolution of the maximum (red line) and minimum (blue line) mesh quality through the mesh refinement iteration: (a) equilateral tetrahedron; (b) cartesian tetrahedron; and (c) random tetrahedron.

5. EXAMPLES

We present several examples to illustrate that our proposed algorithm refines unstructured tetrahedral meshes, generates locally adapted conformal meshes, a finite number of similarity classes, and has a lower-bounded quality. For all the examples, we have computed the shape quality [20] of the mesh elements. Then, we plot the minimum and maximum shape quality of the mesh in each refinement step to check that the minimum quality is lower bounded and cycles. Moreover, in the examples where we locally refine the mesh, our code asserts that the mesh is conformal by faces and that Euler's characteristic of the mesh remains constant.

The results have been obtained on a MacBook Pro with one dual-core Intel Core i5 CPU, at a clock frequency of 2.7GHz, and with a total memory of 16GBytes. As a proof of concept, a mesh refiner has



(a) (b) (c)

Figure 8: Final mesh after twelve iterations of uniform refinement for: (a) equilateral tetrahedron; (b) cartesian tetrahedron; and (c) random tetrahedron.

been fully developed in Julia 1.4. The Julia prototype code is sequential (one execution thread), corresponding to the implementation of the method presented in this work.

5.1 Minimum quality is lower bounded and cycles with uniform refinement

In this example, we show that the minimum quality is lower bounded and cycles. To this end, we uniformly refine a single tetrahedron several times. We denote as $\mathcal{Q}_k$the obtained mesh after k uniform refinements,

$$\mathcal{Q}_k = \texttt{bisectTetrahedra}(\mathcal{Q}_{k-1}, \mathcal{Q}_{k-1}),$$

where $\mathcal{Q}_0 = \tau$. Thus, the mesh $\mathcal{Q}_k$ is composed of 2^k tetrahedra and the accumulated number of generated tetrahedra is $2^{k+1} - 1$.

The method needs at least five iterations of uniform refinement to generate 36, different similarity classes. This number of iterations is so since the process only accumulates 31 generated tetrahedra after four successive uniform refinements. Assuming all of these 31 tetrahedra are of a different similarity class, the pigeonhole principle ensures that they cannot correspond to 36 different similarity classes. On the contrary, at the end of iteration five, the accumulated number of generated tetrahedra is 63, greater than 36, and thus, all the similarity classes might be generated. After that iteration, further refinements do not generate new similarity classes, and therefore, the minimum quality remains lower bounded. Moreover, if we perform three additional uniform refinements, we obtain an entire cycle of the quality of length three. To illustrate those quality cycles, we perform a series of additional refinements.

Figure 7 plots the evolution of the minimum (blue line) and maximum (red line) qualities during the uniform refinement process. Figures 7(a), 7(b), and 7(c) illustrate the quality of an equilateral, cartesian and a perturbed tetrahedra, respectively. At most, we have to perform five uniform refinements to generate all the similarity classes. Thus, we perform 12 uniform refinements to see how the quality cycles. We can see in Figure 7(a), for the most symmetric tetrahedron, how the minimum quality achieves its minimum at iteration five, and then it remains cycling. For the cartesian and perturbed tetrahedra, we can see in Figures 7(b) and 7(c) that we also have to perform five uniform refinements to generate all the similarity classes, achieving the minimum quality of the mesh and start to cycle. In Figures 8(a), 8(b), and 8(c) correspond to the meshes $\mathcal{Q}_{12}$ of the equilateral, cartesian and random tetrahedra after 12 uniform refinements.

We have generated all the similarity classes for the three tetrahedra, and thus, the minimum mesh quality is achieved. Thus, this example illustrates that the method is stable and the mesh quality does not degenerate during successive refinement.

5.2 3D unstructured mesh: locally refining a sphere

This example shows that the proposed refinement scheme can be applied to locally refine unstructured tetrahedral meshes. We recreate the first example from Maubach [8] and Arnold *et al.* [13] but for a sphere. Specifically, we generate an unstructured three-dimensional mesh of a sphere of radius 2 and centered at the origin. Let H be a hemisphere of a sphere of radius one centered at $(1/2, 1/2, 1/2)$ defined by the equations

$$\left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2 + \left(z - \frac{1}{2}\right)^2 = 1,\, x \geq \frac{1}{2}.$$

We want to adapt the tetrahedral mesh $\mathcal{T}_0$ to the hemisphere H. At each local refine iteration, we choose

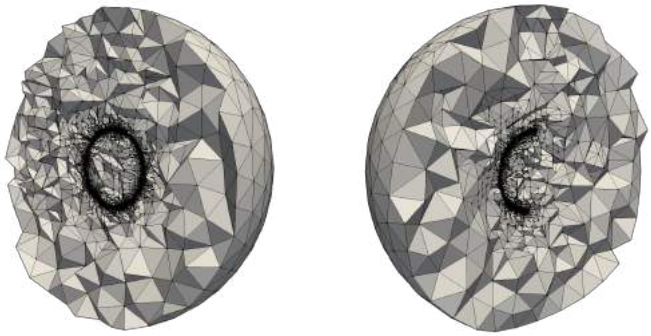


Figure 9: Slice of the mesh $\mathcal{T}_{40}$ with the plane: (a) $x = 1/2$; and (b) $y = 1/2$.

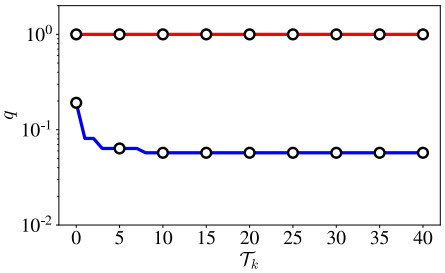


Figure 10: Quality of Example 5.2: Evolution of the maximum (red line) and minimum (blue line) mesh quality through the mesh refinement iterations.

the tetrahedra that intersect the hemisphere H as the refinement set. After forty iterations the mesh $\mathcal{T}_{40}$ is composed by 5806615 tetrahedra and 1045175 vertices. Figures 9(a) and 9(b) show the $\mathcal{T}_{40}$ sliced with the planes $x = 1/2$ and $y = 1/2$. Figure 10 shows how the maximum quality remains constant because it is achieved in each iteration of the local refinement. The minimum quality decreases until its minimum is achieved and then remains constant.

The final mesh is conformal and captures the chosen hemisphere with smaller elements, while it contains larger elements at the exterior boundary.

5.3 3D space-time mesh: locally refining a iso-potential surface

The main goal of this example is to capture a three-dimensional manifold defined by the movement of a two-dimensional object. We show the evolution of the gravitational potential defined by two particles that move along the y-axis. Let

$$V(\mathbf{x}, t) = -G\left(\frac{m_1}{\|\mathbf{x} - \mathbf{p}_1(t)\|} + \frac{m_2}{\|\mathbf{x} - \mathbf{p}_2(t)\|}\right)$$

$$\mathbf{p}_1(t) = \mathbf{p}_1 + (0, vt),\ t \in [0, 1]$$
$$\mathbf{p}_2(t) = \mathbf{p}_2 - (0, vt),\ t \in [0, 1]$$

the equation that defines the gravitational potential. For a given iso-value V_0, $V(\mathbf{x}, t) = V_0$ defines a two-dimensional embedded manifold in three-dimensional space. Let H be the hyper-cylinder with spherical basis defined by the equations

$$\left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2 = 1, \quad 0 \le t \le 1.$$

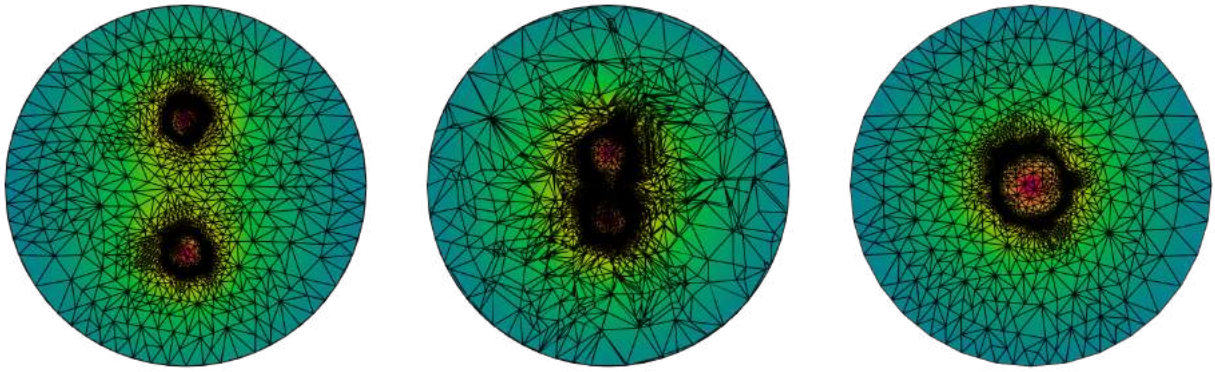
In this example, we choose the iso-value $V_0 = -10$ and the parameters $G = 1$, $m_1 = 1$, $m_2 = 1$, $\mathbf{p}_1 = (1/2, 1/8)$, $\mathbf{p}_2 = (1/2, 7/8)$ and $v = 3/8$.

We generate an adapted tetrahedral mesh by locally refining an initial mesh around the manifold. The initial mesh, $\mathcal{T}_0$, is composed of 3781 tetrahedra and 712 vertices. We generate the set of tetrahedra that intersect H, $F_k = \{\tau \in \mathcal{T}_{k-1} \,|\, \sigma \cap H \neq \emptyset\}$. Then, for each tetrahedron in F_k we compute the curvature of $V(\mathbf{x}, t)$ at each simplex using the formula

$$e_\sigma = \sum_{i=0}^{3} \left| h_i^T \nabla^2 V(\mathbf{x}_i, t_i) h_i \right|,$$

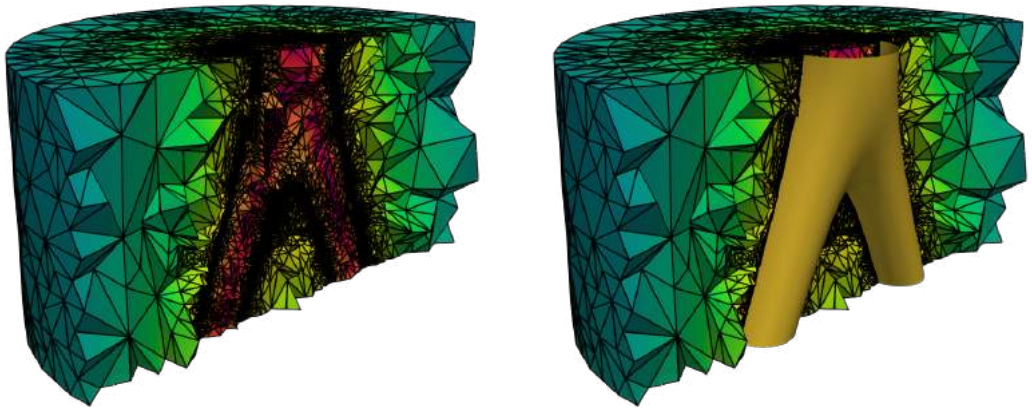
where $\nabla^2 V(\mathbf{x}_i, t_i)$ is the Hessian matrix of the potential $V(\mathbf{x}, t)$ evaluated at the vertices $(\mathbf{x}_i, t_i)$ of τ, and $h_i = (\mathbf{x}_i, t_i) - c_M$, where c_M is the center of mass of τ. After that, we choose as refinement set $\mathcal{S}_k$ the 10% of the tetrahedra of F_k with more curvature. The idea is to adapt the tetrahedral mesh not only to the elements that intersect the iso-surface, but also to the areas of the iso-surface with more curvature.

After 50 iterations of the local refinement process, the generated mesh $\mathcal{T}_{50}$ has 8356894 tetrahedra and



(a) (b) (c)

Figure 11: Slices of $\mathcal{T}_{50}$ with the plane: (a) $t = 0.0$; (b) $t = 0.5$; and (c) $t = 1.0$.



(a) (b)

Figure 12: Slice of the $\mathcal{T}_{50}$ with the plane $x = 1/2$, (a) with, and (b) without the iso-surface.

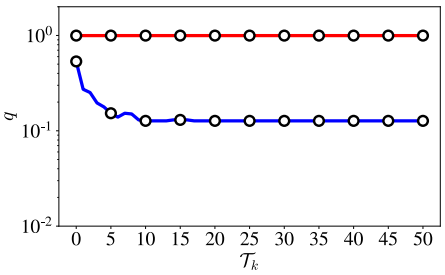


Figure 13: Quality of Example 5.3: Evolution of the maximum (red line) and minimum (blue line) mesh quality through the mesh refinement iterations.

1504344 vertices. Figures 11(a), 11(b) and 11(c) show a slice of the tetrahedral mesh with the planes $t = 0$, $t = 0.5$ and $t = 1$, respectively. The mesh has been locally refined around the iso-surface and therefore, we have smaller elements near the iso-surface and large elements far from the iso-surface. Figure 12 shows a slice of the tetrahedral mesh with the plane $x = 0.5$. We can see how the mesh captures the time evolution of the iso-surface defined by $V(\mathbf{x}, t)$. Figure 12(b) shows the iso-surface that is extracted from the space-time mesh. Figure 13 shows how the maximum quality remains constant because it is achieved in each iteration of the local refinement. The minimum quality decreases until its minimum is achieved and then remains constant.

6. CONCLUDING REMARKS

In conclusion, we have shown the first bisection method meeting the bound of 36 similarity classes on three-dimensional unstructured conformal meshes. For these meshes, we have guaranteed that our approach conformingly marks all the tetrahedra as unflagged planar. In this case, marked bisection behaves as the newest vertex bisection, and thus, it features the optimal bound. We have also checked, with our implementation, that the minimum quality cycles for three-dimensional unstructured conformal meshes.

We have answered an open question. Specifically, we have proved that it is possible to mark as unflagged planar all the tetrahedra of an arbitrary three-dimensional unstructured conformal mesh. To explore alternative answers, we will study whether it is possible to mark all the tetrahedra as adjacent or as a mixture of unflagged planar and adjacent elements.

In perspective, our marked bisection allows refining with optimal similarity bound in adaptive applications on three-dimensional complex geometry. The complexity can be handled by the geometrical flexibility of unstructured conformal meshes. On these meshes, our marked bisection meets all the advantages of the newest vertex bisection.

7. ACKNOWLEDGMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 715546. This work has also received funding from the Generalitat de Catalunya under grant number 2017 SGR 1731. The work of X. Roca has been partially supported by the Spanish Ministerio de Economía y Competitividad under the personal grant agreement RYC-2015-01633.

References

[1] Rivara M.C. "Algorithms for refining triangular grids suitable for adaptive and multigrid techniques." *International journal for numerical methods in Engineering*, vol. 20, no. 4, 745–756, 1984

[2] Rivara M.C. "Local modification of meshes for adaptive and/or multigrid finite-element methods." *Journal of Computational and Applied Mathematics*, vol. 36, no. 1, 79–89, 1991

[3] Plaza A., Carey G.F. "Local refinement of simplicial grids based on the skeleton." *Applied Numerical Mathematics*, vol. 32, no. 2, 195–218, 2000

[4] Plaza A., Rivara M.C. "Mesh Refinement Based on the 8-Tetrahedra Longest-Edge Partition." *IMR*, pp. 67–78. 2003

[5] Mitchell W.F. "Adaptive refinement for arbitrary finite-element spaces with hierarchical bases." *Journal of computational and applied mathematics*, vol. 36, no. 1, 65–78, 1991

[6] Mitchell W.F. "30 years of newest vertex bisection." *AIP Conference Proceedings*, vol. 1738, p. 020011. AIP Publishing, 2016

[7] Kossaczký I. "A recursive approach to local mesh refinement in two and three dimensions." *Journal of Computational and Applied Mathematics*, vol. 55, no. 3, 275–288, 1994

[8] Maubach J.M. "Local bisection refinement for n-simplicial grids generated by reflection." *SIAM Journal on Scientific Computing*, vol. 16, no. 1, 210–227, 1995

[9] Maubach J.M. "The efficient location of neighbors for locally refined n-simplicial grids." *5th Int. Meshing Roundable*, vol. 4, no. 6, 14, 1996

[10] Traxler C.T. "An algorithm for adaptive mesh refinement in n dimensions." *Computing*, vol. 59, no. 2, 115–137, 1997

[11] Stevenson R. "The completion of locally refined simplicial partitions created by bisection." *Mathematics of computation*, vol. 77, no. 261, 227–241, 2008

[12] Alkämper M., Gaspoz F., Klöfkorn R. "A Weak Compatibility Condition for Newest Vertex Bisection in Any Dimension." *SIAM Journal on Scientific Computing*, vol. 40, no. 6, A3853–A3872, 2018

[13] Arnold D.N., Mukherjee A., Pouly L. "Locally adapted tetrahedral meshes using bisection." *SIAM Journal on Scientific Computing*, vol. 22, no. 2, 431–448, 2000

[14] Coxeter H. "Discrete groups generated by reflections." *Ann. Math.*, pp. 588–621, 1934

[15] Freudenthal H. "Simplizialzerlegungen von beschrankter flachheit." *Ann. Math.*, pp. 580–582, 1942

[16] Kuhn H.W. "Some combinatorial lemmas in topology." *IBM Journal of research and development*, vol. 4, no. 5, 518–524, 1960

[17] Bänsch E. "Local mesh refinement in 2 and 3 dimensions." *IMPACT of Computing in Science and Engineering*, vol. 3, no. 3, 181–191, 1991

[18] Liu A., Joe B. "On the shape of tetrahedra from bisection." *Mathematics of computation*, vol. 63, no. 207, 141–154, 1994

[19] Liu A., Joe B. "Quality local refinement of tetrahedral meshes based on bisection." *SIAM Journal on Scientific Computing*, vol. 16, no. 6, 1269–1291, 1995

[20] Knupp P.M. "Algebraic mesh quality metrics." *SIAM journal on scientific computing*, vol. 23, no. 1, 193–218, 2001