

LOCAL DECOMPOSITION OF HEXAHEDRAL SINGULAR NODES INTO SINGULAR CURVES

Paul Zhang¹ Judy (Hsin-Hui) Chiang² Xinyi (Cynthia) Fan³ Klara Mundilova⁴

¹*Massachusetts Institute of Technology, Cambridge, Massachusetts, U.S.A. pzp1@mit.edu*

²*University of Illinois at Urbana-Champaign, Urbana-Champaign, Illinois, U.S.A. hsinhui2@illinois.edu*

³*Davidson College, Davidson, North Carolina, U.S.A. cyfan@davidson.edu*

⁴*Massachusetts Institute of Technology, Cambridge, Massachusetts, U.S.A. kmundil@mit.edu*

ABSTRACT

Hexahedral (hex) meshing is a long studied topic in geometry processing with many fascinating and challenging associated problems. Hex meshes vary in complexity from structured to unstructured depending on application or domain of interest. Fully structured meshes require that all interior mesh edges are adjacent to exactly four hexes. Edges not satisfying this criteria are considered singular and indicate an unstructured hex mesh. Singular edges join together into singular curves that either form closed cycles, end on the mesh boundary, or end at a singular node, a complex junction of more than two singular curves. While all hex meshes with singularities are unstructured, those with more complex singular nodes tend to have more distorted elements and smaller scaled Jacobian values. In this work, we study the topology of singular nodes. We show that all eight of the most common singular nodes are decomposable into just singular curves. We further show that all singular nodes, regardless of edge valence, are locally decomposable. Finally we demonstrate these decompositions on hex meshes, thereby decreasing their distortion and converting all singular nodes into singular curves. With this decomposition, the enigmatic complexity of 3D singular nodes becomes effectively 2D.

Keywords: hexahedral mesh, singular graph, computational geometry

1. INTRODUCTION

Hexahedral meshes are commonly used to model complex geometries and to solve numerical PDEs. The results they produce with tri-linear basis functions are often superior to those produced with linear basis functions on tetrahedral meshes [1, 2]. They can be preferable to other types of meshes for their natural local coordinate systems and have been shown to perform better with quadratic basis functions in the context of nonlinear elasto-plastic simulation [3]. Due to the persistent demand for hex meshes, a variety of methods have been developed to generate them.

Of particular relevance to our work are frame field based methods, where a smooth boundary aligning coordinate system is computed over the domain, followed

by parameterization and hex extraction [4, 5, 6, 7, 8]. Frame field based hex meshing has especially elucidated the significance of singularities within a hex mesh since frame fields inevitably contain singular structures that are reflected in the resulting mesh. The singularities of a frame field and hex mesh typically consist of a set of singular curves that join up in space at singular nodes. These nodes and curves form the singular graph of a field or mesh as illustrated in [Figure 1](#). Since hex meshable singularities are a subset of frame field singularities, much attention has been devoted to the restriction and correction of singularities in frame field computation [9, 10]. Other works have derived conditions and algorithms to compute frame fields obeying singular constraints [11, 12].

Various works have also targeted the enumeration or

simplification of hex mesh singularities. [13, 14, 15] derive algorithms to simplify the singular structures of hex meshes with collapse operations on a coarsened mesh. [11] provide an enumeration algorithm for all hex mesh singular nodes, as well as an exhaustive list of the most practically relevant singular node types. While this list only contains eight singular nodes, they already form complex junctions that are challenging to parse or manipulate.

In contrast to singular nodes, singular curves are simpler to understand, since their local structure is only a 2D singularity extruded into 3D. In the fully 2D setting, cross field and quadrilateral (quad) mesh singularities are significantly easier to visualize, and in the case of parameterized cross fields, singularities are governed completely by a few simple conditions [16]. In quad meshes, singular vertex pairs are shown to move almost fluidly within a quad mesh [17]. None of these results obviously translates to hex meshes, where 3D singular nodes exist as junctions of multiple 2D singularities.

In this paper, we investigate the structure of hex mesh singular nodes. We uncover that singular nodes can be simplified by pulling their constituent elements apart into singular curves. Our results show theoretically and empirically that singular nodes can be removed from a hex mesh thus reducing the complexity of any local neighborhood in a hex mesh. Our contributions are as follows:

- We show by construction that all eight of the most practically relevant singular nodes are decomposable into just singular curves.
- We show that all singular nodes, regardless of valence, are locally decomposable.
- We apply our decompositions to hex meshes demonstrating that entire singular graphs can be separated into independent singular curves.

2. PRELIMINARIES

Our work is motivated by the following question. What if a singular node is formed when singular curves just barely skim past each other? If that were the case, then we could separate the curves with a sheet and increase its thickness to force the curves away from each other, thereby untangling the singular node. To formalize this idea, we begin with the following definitions.

2.1 Singular Vertices, Curves, and Nodes

We denote a hex mesh as $\{\mathcal{V}, \mathcal{H}\}$ where \mathcal{V} is a list of vertices embedding the mesh and \mathcal{H} is a list of hexes of

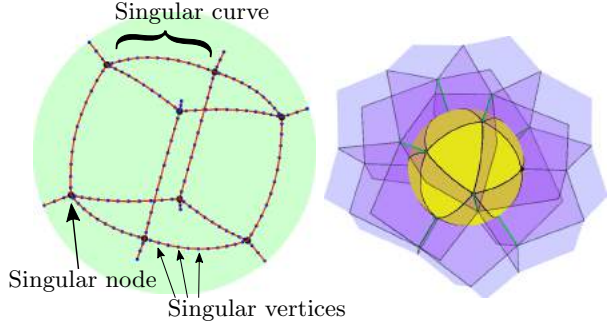


Figure 1: (Left) The singular graph of a hex mesh of a sphere is shown. Singular edges are colored red, singular nodes are large black circles and singular vertices are small blue vertices. (Right) A close-up view of a singular node. Faces adjacent to the singular node are displayed in purple. A yellow sphere is overlaid on top of the singular node. Its intersection with the local hex mesh partitions the sphere into triangular regions.

the mesh. Let \mathcal{F} denote the quadrilateral faces of the mesh, \mathcal{E} denote the edges of the mesh, and $\deg(e \in \mathcal{E})$ denote the number of hexes adjacent to edge e , i.e., its degree or valence. A *singular edge* is an interior edge e satisfying $\deg(e) \neq 4$. We will not treat cases where $\deg(e) \leq 2$ since these are not typically accepted as valid hex meshes. We will also not consider singular boundary edges in this paper but refer the interested reader to [11] for the corresponding definition. For our purposes, one can ignore boundary singularities or push them all to the interior by adding one layer of padding to the hex mesh boundary. A *singular vertex* is a vertex of the mesh that is adjacent to any singular edge. A *singular node* is a vertex of the mesh that is adjacent to more than two singular edges. A *singular curve* is an alternating sequence of singular edges and singular vertices that either forms closed cycles or, ends at a singular node or boundary vertex. Note that we have chosen to deviate from the language of [11] by distinguishing singular nodes from singular vertices. Singular nodes are reserved for the junctions of multiple singular curves and will be the primary focus of this work. [Figure 1](#) depicts a summary of this terminology.

For a singular node $v \in \mathcal{V}$, we denote $\mathcal{T}(v)$ as the triangle mesh in bijection with that node according to [6]. This bijection is formed by intersecting the singular node of the hex mesh with an infinitesimally small sphere. Since the intersection of a corner of a hex with a sphere forms a triangle, the hexes adjacent to the singular node partition the sphere into triangular regions thus forming a sphere triangulation. This is depicted in [Figure 1](#). The sphere triangulation encodes the *singular node type*. If two sphere triangulations are isomorphic, then their singular nodes are of the same type. The *signature* of a singular node is a

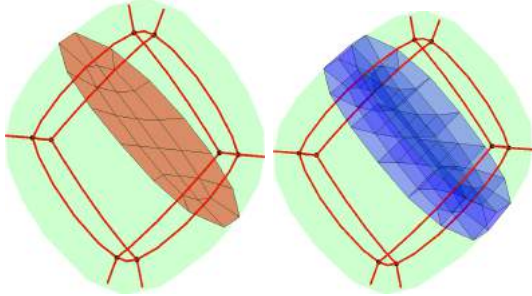


Figure 2: (Left) Red quads indicate a sheet inside of a hex mesh of an ellipsoid. Red curves depict its singular graph. The sheet is manifold with boundary on the boundary of the hex mesh. (Right) Blue quads indicate all faces of the inflated sheet.

list of numbers indicating how many adjacent singular edges are of each degree. Since singular edges have degree 3 or higher, the signature of a node starts with the number of edges with degree 3. For singular nodes whose adjacent singular edges have only valence 3, 4, or 5, the signature also uniquely encodes the singular node type [11]. For this reason we will frequently identify singular nodes by their signature e.g. $(4,0,0)$ is the signature identifying the singular node type generated by subdividing a single tetrahedron into four hexes as illustrated in Figure 4

2.2 Sheet Inflation

Let a *sheet* $Q \subset \mathcal{F}$ be a manifold quad mesh whose boundary is a subset of the boundary of \mathcal{H} . A *sheet inflation* based on sheet Q is a mesh modifying operation by which each $q \in Q$ is thickened from a quad face to a hex cell [18]. A sheet inflation operation is depicted in Figure 2

In the case that the inflated sheet passes through a singular node, the singular type of that node can change. One can interpret this as the sheet passing an infinitesimally small gap between singular curves and forcing them apart, or as cutting a singular node into separate pieces. As singular nodes are often more easily visualized as sphere triangulations we describe here how sheet inflation through a singular node corresponds to a *splitting* of the sphere triangulation. This splitting is illustrated by Figure 3

Given a singular node v , and a sheet $Q \subset \mathcal{F}$ that passes through v , the faces of Q map to edges of $\mathcal{T}(v)$. These edges trace out a cycle in the graph of $\mathcal{T}(v)$ effectively partitioning the sphere into two disk triangulations \mathcal{D}_1 and \mathcal{D}_2 . When the sheet is thickened into a layer of hexes, these two disk triangulations are cut apart. Then the disks are patched into sphere triangulations again by adding one new vertex to each disk and attaching triangles from the boundary of each disk

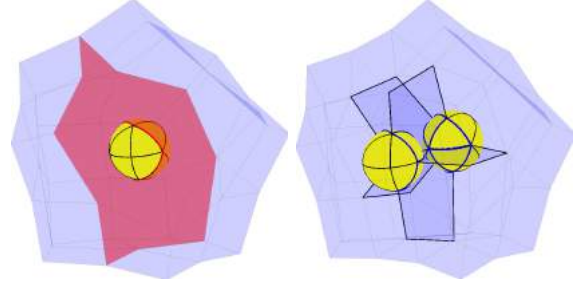


Figure 3: (Left) The yellow sphere triangulation indicates the structure of a singular node. Red quads indicate a sheet intersecting the singular node. The sheet intersects the sphere triangulation on a cycle of red curves that divide the sphere triangulation into two disks. (Right) Two singular nodes are visualized from inflation of the red sheet on the left. Sphere triangulations of the resulting two nodes are shown. Blue quad faces indicate newly created faces from the inflation. Blue edges indicate newly created edges in each sphere triangulation.

to their respective new vertex. This splitting operation is illustrated in Figure 3. The end result is two sphere triangulations one built from \mathcal{D}_1 and one built from \mathcal{D}_2 .

3. SINGULAR NODE DECOMPOSITION (VALENCE 3,4,5)

We are now equipped with the mechanism by which all practically relevant singular nodes can be decomposed into singular curves. These nodes are enumerated in [11] Figure 6] and will also be shown at the beginning of each respective decomposition. Various singular decompositions will be depicted throughout the remainder of this paper. Valence 3 singular curves will be red, valence 5 singular curves will be green and higher valence singular curves will be blue. We omit drawing interior regular edges to minimize clutter. Since these operations can be challenging to understand from static images, we also include supplemental videos for many of the decompositions.

In the first column of Figure 4 we start with the $(4,0,0)$ singular node, which consists of four valence 3 singular curves joined at a junction. With a single sheet inflation, this singular node is revealed to actually be two valence 3 singular curves that pass each other orthogonally. A similar theme follows for $(2,2,2)$ in the second column which is revealed to be a valence 3 and a valence 5 singular curve passing each other orthogonally. Finally in the third column, the $(0,4,4)$ node decomposes into two valence 5 singular curves passing each other orthogonally.

From these three examples, it is tempting to think all singular nodes may consist of singular curves glued to-

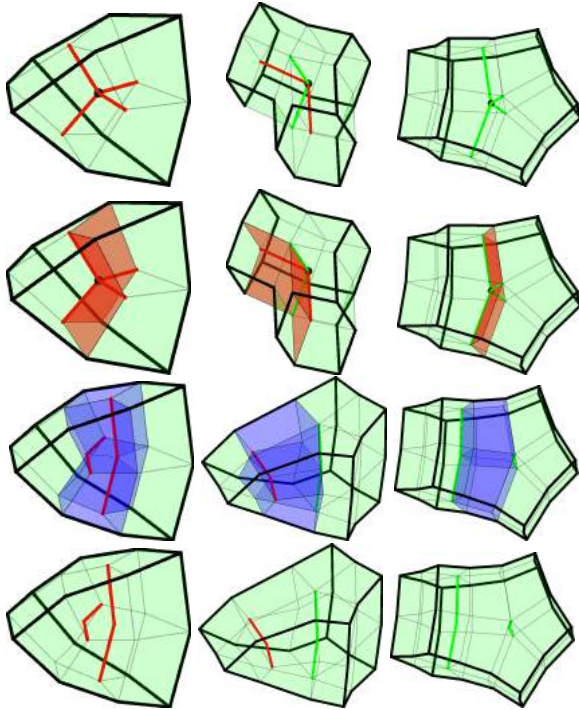


Figure 4: From left to right the $(4,0,0)$, $(2,2,2)$, and $(0,4,4)$ singular nodes are depicted. Top to bottom indicates steps to decompose each singular node. Red quads indicate the sheet to be inflated. Blue quads indicate faces of the newly inflated hexes. Red(Green) edges are valence 3(5) singularities. One sheet inflation is sufficient to decompose each of these nodes.

gether orthogonally. One might conclude as well that the number of singular curves of a particular valence meeting at a singular node from this construction must be even. The $(1,3,3)$ singular node, shown in [Figure 5](#) presents a curious counterexample. Since it consists of one valence 3 singular curve and three valence 5 singular curves, it is impossible to decompose these into two singular curves passing each other orthogonally in a valid hex mesh.

This conundrum is resolved by realizing that one of the regular edges adjacent to this singular node is actually a pair of valence 3 and valence 5 singular curves, glued together in parallel. Another way to understand this node is that one valence 5 singular curve has split an otherwise parallel pair of valence 3 and 5 curves. The decomposition of this node into one valence 3 and two valence 5 singular curves is illustrated in [Figure 5](#)

In [Figure 6](#) we decompose the $(0,3,6)$ singular node into three valence 5 singular curves. It is also valuable to think of the $(0,3,6)$ node as a combination of two $(0,4,4)$ singular nodes. In this way, one only needs to decompose a singular node into constituent nodes that are already known to be decomposable. The fourth image of [Figure 6](#) shows exactly this decomposition which we will notate as:

$$(0, 3, 6) = (0, 4, 4) +_5 (0, 4, 4)$$

The subscript 5 on the plus symbol denotes that two singular nodes are joined along a valence 5 edge, followed by an inverse sheet inflation (sheet collapse). This notation serves only as a shorthand and does not uniquely encode how to glue two singular nodes together. It serves more as a recipe than an equation with any algebraic properties. For completion, we show the rest of the decomposition of the constituent $(0,4,4)$ nodes.

The $(0,2,8)$ singular node is decomposed in [Figure 7](#) into four valence 5 curves. In the fourth image, we see that

$$(0, 2, 8) = (0, 3, 6) +_5 (0, 4, 4)$$

and both constituent singular nodes have already been shown to be decomposable. For completion, the rest of the decomposition steps are also shown.

The $(2,0,6)$ singular node is decomposed in [Figure 8](#) into four valence 5 and two valence 3 curves. By the fourth image we see

$$(2, 0, 6) = (1, 3, 3) +_4 (1, 3, 3).$$

The rest of this decomposition is still interesting however as it introduces a valence 6 singularity in the fifth image and a singular node with signature $(1,3,3,1)$. This valence 6 singular curve is removed in image 9 by decomposing it into two valence 5 curves.

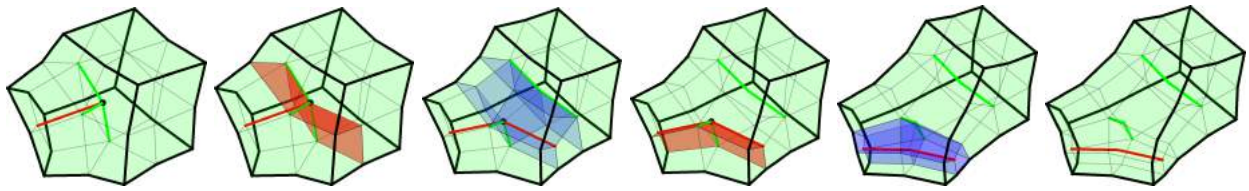


Figure 5: The $(1,3,3)$ singular node is decomposed into two valence 5 and one valence 3 curve via two sheet inflations.

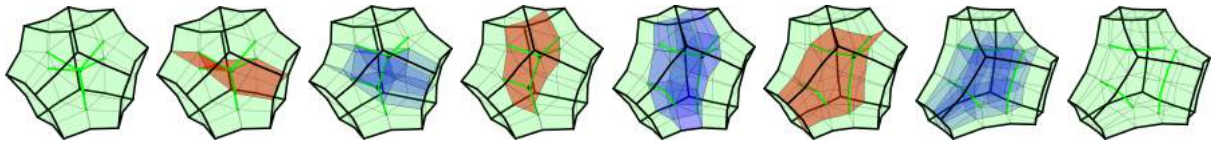


Figure 6: The $(0,3,6)$ singular node is decomposed into three valence 5 curves via three sheet inflations.

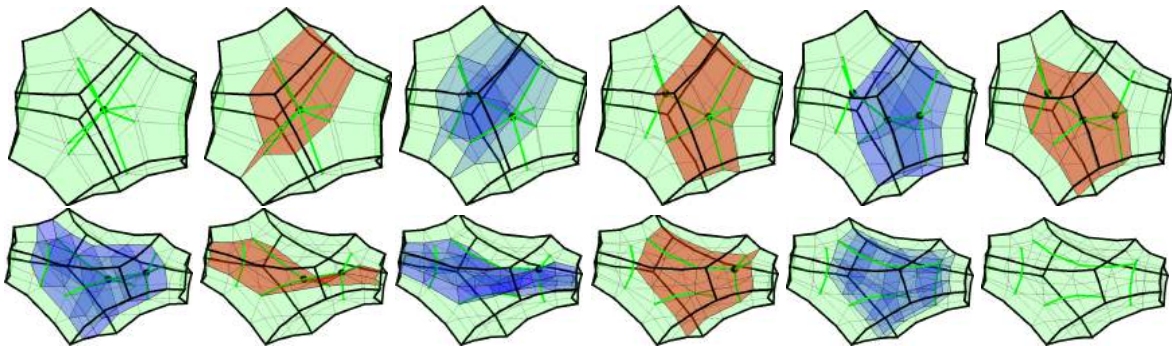


Figure 7: The $(0,2,8)$ singular node is decomposed into four valence 5 curves via five sheet inflations.

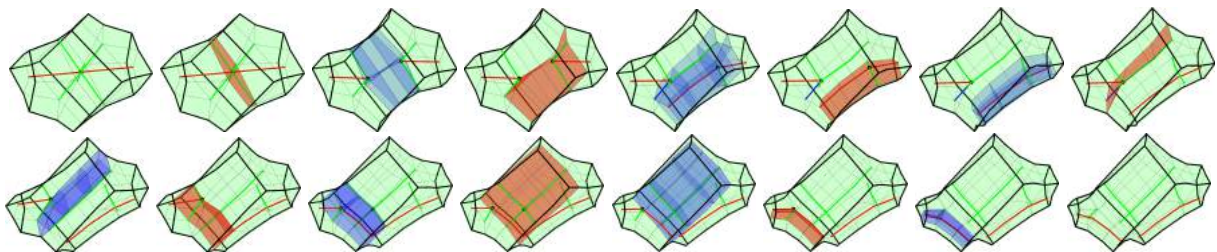


Figure 8: The $(2,0,6)$ singular node is decomposed into four valence 5 and two valence 3 curves via seven sheet inflations.

Finally the $(0,0,12)$ singular node decomposition is shown in [Figure 9](#) to become six valence 5 curves. This singular node is especially interesting as we were unable to show a decomposition of the form

$$(0, 0, 12) = (0, 4, 4) +_n (0, 2, 8).$$

Even though the number of singular curves present is sufficient, we were not able to perform an inverse sheet inflation between $(0,4,4)$ and $(0,2,8)$ to obtain $(0,0,12)$. This shows that the order in which singular curves are combined matters. As this figure is especially complex to comprehend, we offer the following roadmap of how the decomposition is performed.

$$(0, 0, 12) = \underbrace{((0, 3, 6) +_5 (0, 3, 6))}_{(0,2,8,1)} +_6 \underbrace{((0, 4, 4) +_5 (0, 4, 4))}_{(0,4,4,1)}$$

4. DECOMPOSING GENERAL SINGULAR NODES

Given that the eight singular nodes of valence 3, 4, or 5 are decomposable into singular curves, a natural next question is whether decomposition extends to higher valence singular nodes. In fact, the decomposition of the $(0,0,12)$ and $(2,0,6)$ both already required decomposing singular nodes with valence 6: $(0,4,4,1)$, $(0,2,8,1)$ and $(1,3,3,1)$. We will refer to previously known decomposable singular nodes and their associated sphere triangulations as *base cases*. To generalize decomposability of singular nodes we offer the following result.

Proposition 1. Given a sphere triangulation \mathcal{T} with some vertex u of degree larger than 5, there exists a splitting such that either the number of vertices in both resulting triangulations decreases or the resulting triangulations are base cases.

Proof. The local neighborhood of u is an umbrella \mathcal{U} of at least 6 triangles. The boundary of this umbrella is a cycle of at least 6 vertices denoted by \mathcal{C} . To construct a splitting of \mathcal{T} into triangulations of fewer vertices, we need a pair of vertices a and b adjacent to u that are at least 3 edges apart from each other in \mathcal{C} such that there is path p from a to b through the interior of $\mathcal{T} - \mathcal{U}$. This construction is illustrated in [Figure 10](#). The sequence of edges $[(ua), p, (bu)]$ partitions \mathcal{T} into \mathcal{D}_1 and \mathcal{D}_2 where each disk triangulation has at least 2 interior vertices. Since splitting a sphere triangulation replaces all vertices on the interior of either side with just one new vertex each, both resulting triangulations will have fewer vertices than \mathcal{T} . For readability, we leave more detailed construction of the splitting to supplementary materials. \square

Algorithm 1 Decomposes all singular nodes of a hex mesh into singular curves.

```

1: procedure DECOMPOSE-SINGULAR-GRAPH( $\mathcal{H}$ )
2:   do
3:      $N \leftarrow \text{GETRANDOMSINGULARNODE}(\mathcal{H})$ 
4:     if ONLYHASVALENCE345( $\mathcal{H}, N$ ) then
5:        $C \leftarrow \text{GETHARDCODEDCUT}(\mathcal{H}, N)$ 
6:     else
7:        $C \leftarrow \text{GETGENERALCUT}(\mathcal{H}, N)$ 
8:     end if
9:      $S \leftarrow \text{PROPAGATECUT}(\mathcal{H}, C)$ 
10:     $H \leftarrow \text{SHEETINFLATION}(\mathcal{H}, S)$ 
11:    while  $N \neq \emptyset$ 
12:    return  $\mathcal{H}$ 
13: end procedure

```

Applying the splitting in [Prop. 1](#) could result directly in base cases, where the rest of the decomposition is already known. If the splitting does not result in base cases, then it produces triangulations with fewer vertices. This can be repeated until there are not enough vertices to have a degree 6 vertex. Since sheet inflation at a singular node corresponds to splitting of a sphere triangulation, [Prop. 1](#) allows us to find a sequence of sheets whose inflation results in singular nodes that have lower than valence 6 singular edges. We have already enumerated singular decompositions for all singular nodes with valence lower than 6 and can therefore decompose any singular node into singular curves.

A limitation of [Prop. 1](#) is that it restricts attention to individual singular nodes while ignoring the full singular graph of the mesh. It can be challenging to extend a sheet known locally around a singular node to the rest of the hex mesh while guaranteeing no self-intersection occurs. We present our simplistic solution to extending sheets in [subsection 5.1](#) and leave more careful consideration of how to avoid self-intersection to future work.

5. RESULTS

5.1 Singular Graph Decomposition

We develop a procedure to perform singular mesh decompositions on general hex meshes. Pseudocode for this procedure is given in [Alg. 1](#) and [Alg. 2](#). First, we randomly select a singular node. For any singular node with valence restricted to 3, 4, or 5, we hard code a subset of faces adjacent to the node to be inflated. If the node has valence 6 or higher, we use [Prop. 1](#) (denoted `GetGeneralCut` in [Alg. 1](#)) to select these faces. These faces form a partial sheet that decomposes the initially selected node, but need to be extended through the rest of the mesh in order to be inflatable.

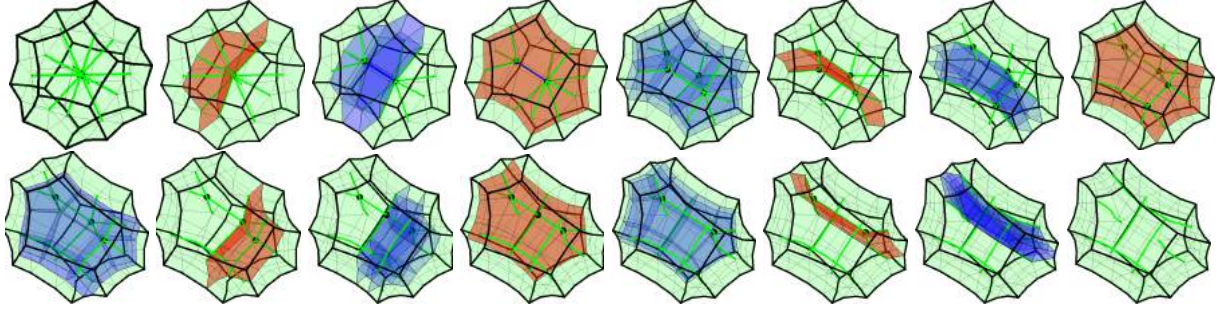


Figure 9: The $(0,0,12)$ singular node is decomposed into six valence 5 curves via seven sheet inflations.

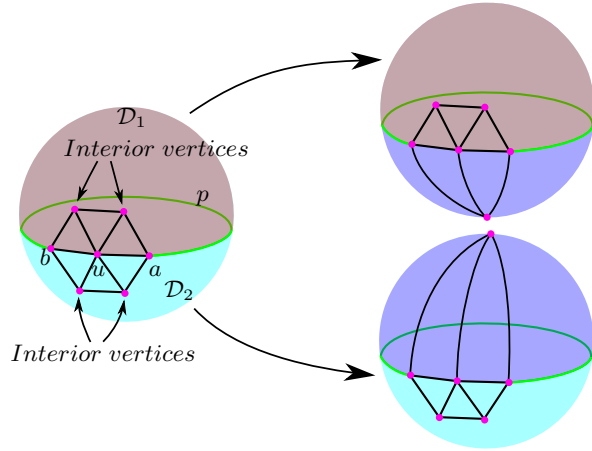


Figure 10: Illustration of how to find a cycle such that splitting along that cycle results in two sphere triangulations, each with fewer vertices. The only requirement is that there is a vertex u of degree ≥ 6 . The required cycle is then $[(ua), p, (bu)]$.

Next we propagate the partial sheet throughout the hex mesh following [Alg. 2](#). Let a face be *parallel* to the partial sheet if they share a regular edge but share no adjacent hexes. We greedily add parallel faces to the partial sheet until no more parallel faces can be found. Next we look for any interior singular vertices on the boundary of the partial sheet. If such a vertex is found, then we compute the smallest number of new faces that need to be added to the partial sheet so that its boundary excludes this singular vertex. This is denoted by `Put-v-In-S` in [Alg. 2](#) and is equivalent to a graph shortest path computation on the triangulation representing this singular vertex.

These two steps are repeated until no more parallel faces can be found, and the boundary of the partial sheet is entirely on the boundary of the hex mesh. If at any stage of the algorithm, the partial sheet became non-manifold then the sheet propagation algorithm has failed. If the sheet is manifold then we in-

Algorithm 2 Propagates a partial sheet into a full sheet recursively.

```

1: procedure PROPAGATECUT( $\mathcal{H}, S$ )
2:    $Q \leftarrow \text{GETFACES}(\mathcal{H})$ 
3:   while  $\exists f \in Q : \text{PARALLEL}(\mathcal{H}, S, f)$  do
4:      $S \leftarrow S \cup f$ 
5:   end while
6:    $\mathcal{V} \leftarrow \text{GETVERTICES}(\mathcal{H})$ 
7:    $\mathcal{V}_S \leftarrow \text{GETINTERIORSINGULARVERTICES}(\mathcal{H})$ 
8:   if  $\exists v \in (\partial S \cap \mathcal{V}_S)$  then
9:      $S \leftarrow \text{PUT-V-IN-S}(\mathcal{H}, S, v)$ 
10:  else
11:    if NONMANIFOLD( $\mathcal{H}, S$ ) then
12:      return ERROR
13:    else
14:      return  $S$ 
15:    end if
16:  end if
17:  return PROPAGATECUT( $\mathcal{H}, S$ )
18: end procedure

```

flate it resulting in the decomposition of at least one singular node. All results shown were generated by [Alg. 1](#)

Applying our decomposition to a hex mesh of a sphere reveals that it has the same singular graph structure as that of a padded tetrahedron. [Figure 11](#) shows this correspondence where inflating one sheet that passes through seven singular nodes, simultaneously decomposes three of them. The end result is a singular graph composed of four $(4,0,0)$ singular nodes. One of these nodes has singular curves that all connect directly to the boundary. The other four of these nodes connect to each other and the boundary via valence 3 singular curves in a tetrahedral arrangement. This singular graph is exactly what one obtains by padding a hex mesh of a regular tetrahedron i.e. padding a $(4,0,0)$ node.

Changing how the sheet cuts through the singular graph produces different intermediate and final singu-

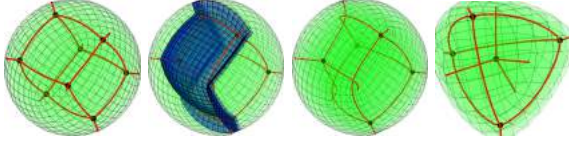


Figure 11: (Left) Hex mesh of sphere with singular graph. (Mid-left) Blue hexes are newly inflated hexes. (Mid-right) Hex mesh post-inflation. (Right) Singular graph of a padded hex mesh of a tetrahedron. The last two images have topologically equivalent singular graphs.

lar graphs. In [Figure 12](#) we decompose a padded cube in two different sequences and show their intermediate singular graphs. To improve clarity, we provide schematics of a subset of the singular graphs. The ending singular graphs from both sequences are also topologically distinct i.e. no purely geometric deformation maps one singular graph into the other. They do however appear to invariably contain a single singular cycle.

The first sheet inflation of the second sequence results in the same singular graph as a padded hex mesh of a triangular prism: a padded $(2,3,0)$. Since the hex mesh of a sphere has the same singular graph as the padded cube, these results indicate that singular graphs for a padded cube, padded tet, and padded triangular prism are identical up to a series of sheet inflation and collapses.

In [Figure 13](#) we apply our decomposition to more complex singular graphs. The first two rows depict the decomposition of the **G1** hex mesh. The starting singular graph consists of 12 nodes connected by 36 singular curves. This graph is successfully decomposed into seven singular curves, one of which is a closed cycle. The last two rows depict the decomposition of the **G2** hex mesh. The starting singular graph consists of 16 nodes connected by 40 singular curves. This graph is successfully decomposed into 12 singular curves, two of which are closed cycles. While the starting singular graphs are different, both meshes are fully decomposed with the same number of sheet inflations.

Finally, we apply our decomposition in [Figure 14](#) to the cactus mesh from [\[19\]](#). While the mesh starts with only singularities of valence 3, 4 and 5, the decomposition results in intermediate singular graphs with nodes of signature $(2,3,0,2)$. The final configuration is seen to contain singular curves of valence 6. Since our goal is only to remove singular nodes, we terminate with valence 6 curves.

5.2 Scaled Jacobians

The minimum scaled Jacobian of a hex mesh is a common metric by which to evaluate distortion of the mesh

[\[20\]](#). We maximize the minimum scaled Jacobian before and after singular decomposition of each singular node with free boundaries and present the resulting minimum scaled Jacobians in [Table 1](#). Unsurprisingly, singular nodes have lower scaled Jacobians than singular curves.

By symmetry, the minimum scaled Jacobian of any hex mesh, regardless of resolution, containing a $(4,0,0)$ node is upper bounded by $\frac{4}{3\sqrt{3}} = .7698$. The same bound for a hex mesh containing a $(0,0,12)$ node is $\frac{\sqrt{2(5+\sqrt{5})}}{5} = .761$. These bounds are exactly attained in [Table 1](#) for the $(4,0,0)$ and $(0,0,12)$ nodes. The same upper bound computed for meshes containing valence 3 singular curves is $\sin(\frac{2\pi}{3}) = .866$ and for meshes containing valence 5 singular curves is $\sin(\frac{2\pi}{3}) = .951$.

By performing a sheet inflation to split singular nodes into singular curves, the minimum scaled Jacobian of the $(4,0,0)$ node is increased to .86, almost the theoretic upper bound. For $(0,4,4)$ as well, decomposing the singular node into two valence 5 curves brings the minimum scaled Jacobian to almost the theoretic upper bound. Decomposing $(0,0,12)$ node into six valence 5 curves brings significant improvement to the minimum scaled Jacobian, though it is not as close to the theoretic upper bound due to interactions between singular curves.

Moving towards full singular graphs, we perform the same scaled Jacobian optimization for a sphere mesh. Maximization of its minimum scaled Jacobian results in a value of .768, close to the upper bound for any mesh containing a $(4,0,0)$ node. We apply our decomposition to this mesh and re-optimize its scaled Jacobian resulting in a significant improvement to .849. We run the same optimizations on the padded tetrahedron, **G1**, and **G2** resulting in similar increases in the minimum scaled Jacobian. These results are summarized in [Table 1](#).

6. CONCLUSIONS AND FUTURE WORK

This paper presents singular nodes as the result of gluing singular curves together at a point and shows that the reverse can be done via sheet inflation to untangle singular nodes into simple singular curves. This removes the 3D complexity of singular nodes leaving meshes with lower distortion. We demonstrate this procedure on a variety of meshes showing in all cases that no singular nodes are left behind.

The main limitation of our work is that the local sheets we prescribe for decomposing a singular node are not guaranteed to propagate globally while avoiding self-intersection. This can result in the inability to decompose a singular graph by removing all of its singular

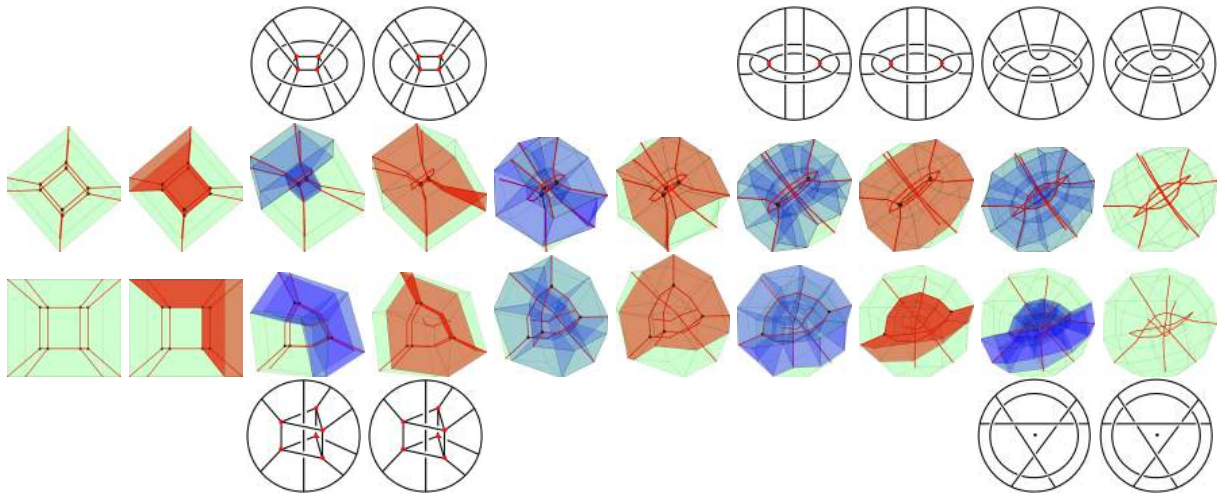


Figure 12: We show two sequences of singular graph decomposition starting from the same hex mesh on the left to a fully decomposed singular graph on the right. The first row indicates select singular graph schematics for the first sequence. The last row indicates select singular graph schematics for the second sequence. Even though both singular graphs start out identical, the ending singular graphs are different due to different sheet inflations.

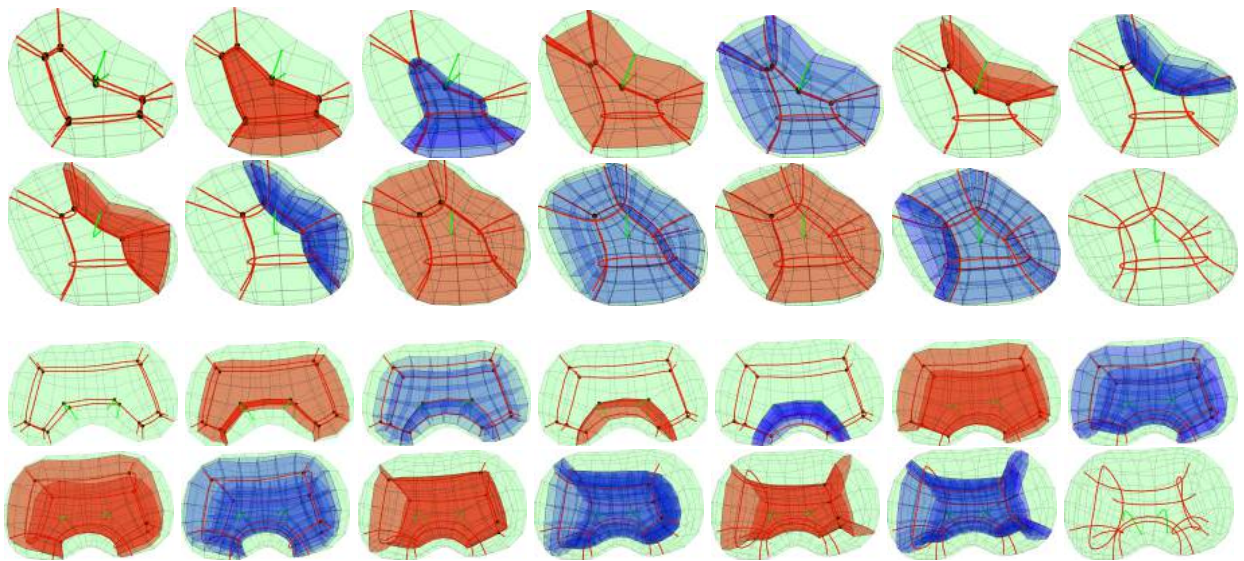


Figure 13: We apply singular decomposition to the **G1** and **G2** hex meshes. The first two rows correspond to the sequence of singular graphs from decomposing **G1**. The last two rows correspond to the sequence of singular graphs from decomposing **G2**. The number of singular nodes decreases each sheet inflation ultimately resulting in a singular graph with no nodes at all.

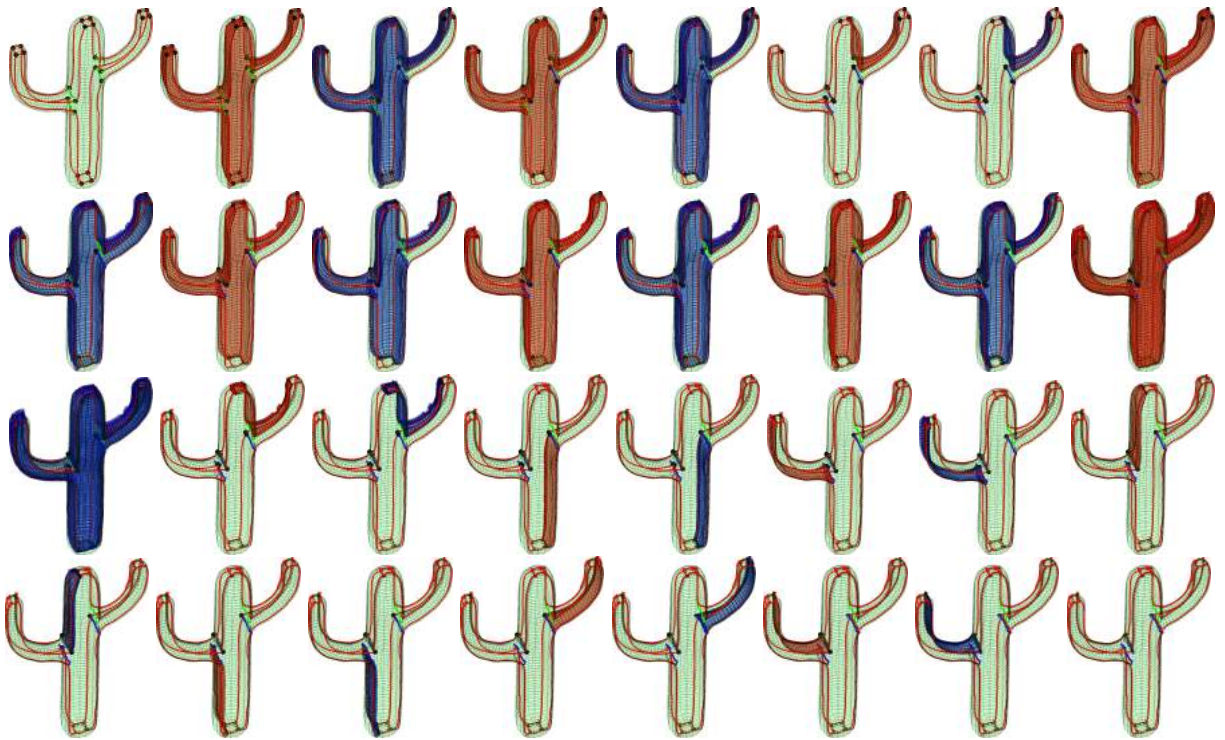


Figure 14: We apply singular decomposition to the cactus mesh from [19](#). The number of singular nodes decreases with each sheet inflation ultimately resulting in a singular graph with no nodes at all. While the original singular graph consisted of only valence 3, 4 and 5 nodes, intermediate singular graphs from this sequence contain singular nodes with signature $(2,3,0,2)$. The fully decomposed singular graph has valence 6 curves.

Mesh	Original	Decomposed	UpperBound
(4,0,0)	0.769	0.86	0.866
(2,2,2)	0.807	0.862	0.866
(0,4,4)	0.896	0.943	0.951
(1,3,3)	0.822	0.865	0.866
(0,3,6)	0.863	0.939	0.951
(0,2,8)	0.82	0.937	0.951
(2,0,6)	0.745	0.856	0.866
(0,0,12)	0.761	0.926	0.951
Sphere	0.768	0.849	0.866
Padded Tet	0.715	0.812	0.866
G1	0.769	0.811	0.866
G2	0.757	0.820	0.866
Ellipsoid	0.767	0.825	0.866

Table 1: For various hex meshes, we indicate the maximized minimum scaled Jacobian before and after singular decomposition. The first column indicates the mesh, the second column indicates before singular decomposition, and the third column indicates after. The fourth column indicates a theoretic upper bound on the minimum scaled Jacobian for the decomposed mesh. It essentially indicates the presence of a valence 3 or 5 singular curve. The maximized minimum scaled Jacobian is invariably higher post singular decomposition.

nodes. We expect that a valid sheet inflation can always be found and leave its efficient computation to future work.

While our method decreases the number of singular nodes in a mesh, its *base complex* [14] may increase in size. This tradeoff should be considered by the user as they may have to choose between a larger scaled Jacobian or maintaining a small number of base complex cells.

Our results can be extended to design new ways of modifying the singular graph of a mesh. Instead of only decomposing nodes into curves, one can *rewire* singular curves by merging them at a node with sheet collapse, and decomposing them in a different way from how they were combined. For example, consider the (4,0,0) node in Figure 4. Its sphere triangulation is a tetrahedron which contains three distinct cycles of length four. Therefore, it is possible to bring two valence 3 singular curves together to form a (4,0,0) node and split them apart again in three distinct ways. Each one results in a different singular graph, none of which require introducing new singularities.

Many works aim to build minimal degree smooth parameterizations of quad meshes with singularities [21, 22]. These methods do not clearly generalize to the volumetric case where singular nodes may suffer decreased continuity from methods designed for 2D singularities. A promising approach following our work is then to decompose any given singular graph

so that no singular nodes exist. We expect that it is easier to adapt quad mesh singular parameterization methods to singular curves that are just 2D singularities extruded into 3D than it is to adapt parameterization methods for singular nodes. Even if one derived a singular node parameterization method for a specific singular node type, there is no guarantee that it extends to any other node type. This problem is made easier by only needing to consider singular curves after decomposition.

ACKNOWLEDGEMENTS

This project was launched at the Summer Geometry Initiative (SGI) 2021, supported by National Science Foundation grant DMS-2103933, Army Research Office grant W911NF2110095, and generous donations from corporate partners. Paul Zhang acknowledges the support of the Department of Energy Computer Science Graduate Fellowship and the Mathworks Fellowship. The authors thank Justin Solomon and David Bommers for many valuable discussions.

References

- [1] Weingarten V.I. “The controversy over hex or tet meshing.” *Machine design*, vol. 66, no. 8, 74–76, 1994
- [2] Cifuentes A., Kalbag A. “A performance study of tetrahedral and hexahedral elements in 3-D finite element structural analysis.” *Finite Elements in Analysis and Design*, vol. 12, no. 3-4, 313–318, 1992
- [3] Benzley S., Perry E., Merkle K., Clark B., Sjaardema G. “A Comparison of All Hexagonal and All Tetrahedral Finite Element Meshes for Elastic and Elasto-Plastic Analysis.” *International Meshing Roundtable*, vol. 17, 01 1995
- [4] Ray N., Sokolov D., Lévy B. “Practical 3D frame field generation.” *ACM Transactions on Graphics*, vol. 35, no. 6, 1–9, Nov. 2016. URL <http://dl.acm.org/citation.cfm?doi=2980179.2982408>
- [5] Lyon M., Bommers D., Kobbelt L. “HexEx: Robust Hexahedral Mesh Extraction.” *ACM Trans. Graph.*, vol. 35, no. 4, 123:1–123:11, Jul. 2016
- [6] Nieser M., Reitebuch U., Polthier K. “CubeCover–Parameterization of 3D Volumes.” *Computer Graphics Forum*, vol. 30, no. 5, 1397–1406, Aug. 2011. URL <http://doi.wiley.com/10.1111/j.1467-8659.2011.02014.x>
- [7] Huang J., Tong Y., Wei H., Bao H. “Boundary aligned smooth 3D cross-frame field.”

- SIGGRAPH Asia*, p. 1. ACM Press, Hong Kong, China, 2011. URL <http://dl.acm.org/citation.cfm?doid=2024156.2024177>
- [8] Solomon J., Vaxman A., Bommes D. “Boundary Element Octahedral Fields in Volumes.” *ACM Transactions on Graphics*, vol. 36, no. 3, 1–16, May 2017. URL <http://dl.acm.org/citation.cfm?doid=3087678.3065254>
- [9] Li Y., Liu Y., Xu W., Wang W., Guo B. “All-hex meshing using singularity-restricted field.” *ACM Trans. Graph.*, vol. 31, no. 6, 177, 2012
- [10] Jiang T., Huang J., Wang Y., Tong Y., Bao H. “Frame Field Singularity Correction for Automatic Hexahedralization.” *IEEE Trans. on Visualization & Computer Graphics*, vol. 20, no. 8, 1189–1199, Aug. 2014
- [11] Liu H., Zhang P., Chien E., Solomon J., Bommes D. “Singularity-constrained octahedral fields for hexahedral meshing.” *ACM Transactions on Graphics*, vol. 37, no. 4, 1–17, Jul. 2018. URL <http://dl.acm.org/citation.cfm?doid=3197517.3201344>
- [12] Corman E., Crane K. “Symmetric Moving Frames.” *ACM Trans. Graph.*, vol. 38, no. 4, Jul. 2019. URL <https://doi.org/10.1145/3306346.3323029>
- [13] Gao X., Panozzo D., Wang W., Deng Z., Chen G. “Robust structure simplification for hex re-meshing.” *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, 1–13, 2017
- [14] Gao X., Deng Z., Chen G. “Hexahedral mesh re-parameterization from aligned base-complex.” *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, 1–10, 2015
- [15] Xu G., Ling R., Zhang Y.J., Xiao Z., Ji Z., Rabczuk T. “Singularity Structure Simplification of Hexahedral Meshes via Weighted Ranking.” *Computer-Aided Design*, vol. 130, 102946, 2021
- [16] Campen M., Shen H., Zhou J., Zorin D. “Seamless parametrization with arbitrary cones for arbitrary genus.” *ACM Transactions on Graphics (TOG)*, vol. 39, no. 1, 1–19, 2019
- [17] Peng C.H., Zhang E., Kobayashi Y., Wonka P. “Connectivity editing for quadrilateral meshes.” *Proceedings of the 2011 SIGGRAPH Asia conference*, pp. 1–12. 2011
- [18] Ledoux F., Shepherd J. “Topological modifications of hexahedral meshes via sheet operations: a theoretical study.” *Engineering with Computers*, vol. 26, no. 4, 433–447, 2010
- [19] Bracci M., Tarini M., Pietroni N., Livesu M., Cignoni P. “HexaLab.net: An online viewer for hexahedral meshes.” *Computer-Aided Design*, vol. 110, 24–36, 2019. URL <https://www.sciencedirect.com/science/article/pii/S0010448518304238>
- [20] Quadros R. “The CUBIT Geometry and Meshing Toolkit.” <https://cubit.sandia.gov/> 2021
- [21] Karčiauskas K., Peters J. “Minimal bi-6 G2 completion of bicubic spline surfaces.” *Computer Aided Geometric Design*, vol. 41, 10–22, 2016
- [22] Karčiauskas K., Peters J. “Refinable smooth surfaces for locally quad-dominant meshes with T-gons.” *Computers & graphics*, vol. 82, 193–202, 2019