Locally Meshable Frame Fields

HENG LIU, University of Bern, Switzerland
DAVID BOMMES, University of Bern, Switzerland

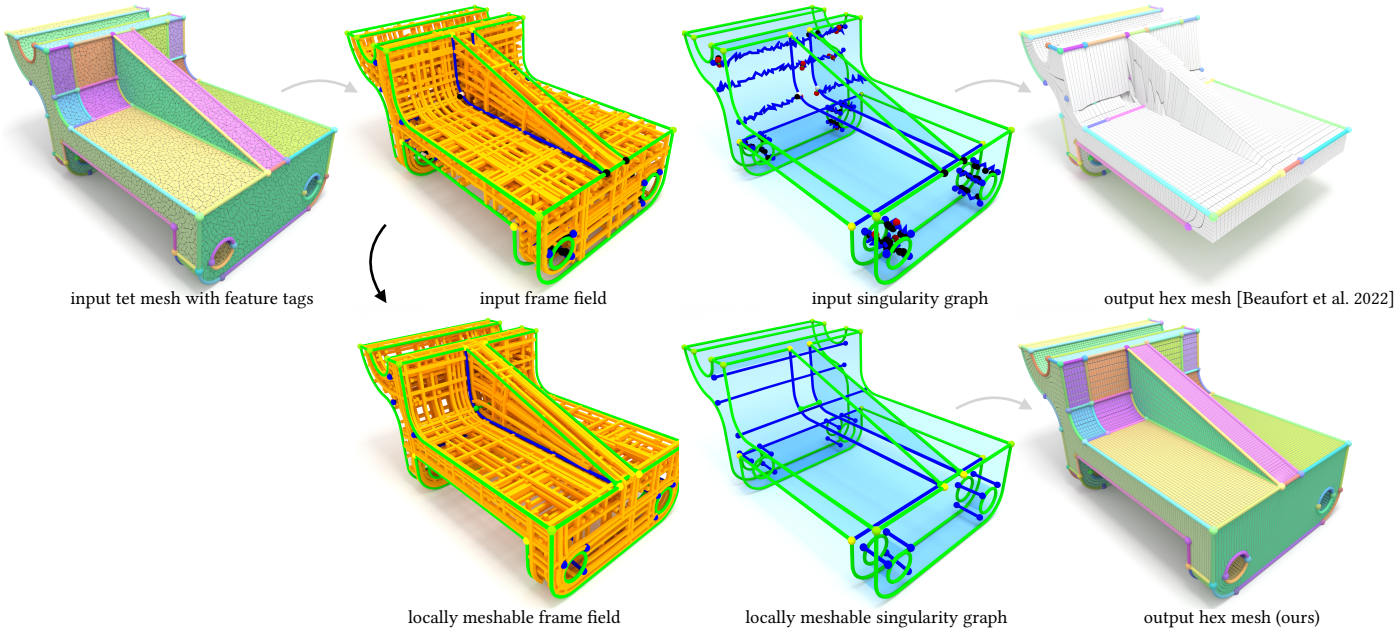


Fig. 1. Overview. Non-meshable topological configurations in frame fields, e.g., invalid singularities or feature structures, induce degenerate integer-grid maps and broken and incomplete hex meshes (top row). Our algorithm (bottom row) automatically turns a given frame field into a locally meshable one, where a valid integer-grid map enables a hex mesh that preserves all input features.

The main robustness issue of state-of-the-art frame field based hexahedral mesh generation algorithms originates from non-meshable topological configurations, which do not admit the construction of an integer-grid map but frequently occur in smooth frame fields. In this article, we investigate the topology of frame fields and derive conditions on their meshability, which are the basis for a novel algorithm to automatically turn a given non-meshable frame field into a similar but locally meshable one. Despite local meshability is only a necessary but not sufficient condition for the stronger requirement of meshability, our algorithm increases the 2% success rate of generating valid integer-grid maps with state-of-the-art methods to 58%, when compared on the challenging HexMe dataset [Beaufort et al. 2022]. The source code of our implementation and the data of our experiments are available at https://lib.algohex.eu.

CCS Concepts: • **Computing methodologies** → **Mesh models**; **Mesh geometry models**; **Volumetric models**.

Authors' addresses: Heng Liu, heng.liu@unibe.ch, University of Bern, Bern, Switzerland; David Bommes, david.bommes@unibe.ch, University of Bern, Bern, Switzerland.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
© 2023 Copyright held by the owner/author(s).
0730-0301/2023/8-ART
https://doi.org/10.1145/3592457

Additional Key Words and Phrases: hexahedral meshing, frame fields, meshability, singularity graph, integer-grid maps

ACM Reference Format:
Heng Liu and David Bommes. 2023. Locally Meshable Frame Fields. *ACM Trans. Graph.* 42, 4 (August 2023), 20 pages. https://doi.org/10.1145/3592457

1 INTRODUCTION

Meshing of volumetric domains is an essential component of various practical applications and research activities. Hexahedral meshes, which decompose a volumetric domain into cuboidal cells, are a popular choice in the context of simulation, e.g. based on finite differences, finite elements or finite volumes. Specifically, when working with basis functions of high polynomial degree, e.g. in Spectral Element Methods [Kopriva 2009], or when requiring higher-order continuity between cells like in Isogeometric Analysis [Cottrell et al. 2006], the tensor-product nature of hexahedra offers advantages as for instance a better performance-to-accuracy tradeoff. Due to global topological constraints, the generation of hexahedral meshes is significantly more challenging than the generation of tetrahedral meshes [Pietroni et al. 2022]. There is an ongoing scientific debate regarding the range of applications, where the additional effort is justifiable [Schneider et al. 2022]. However, to date, there is a high industrial demand for hexahedral meshes, and consequently algorithms to automatically obtain high-quality meshes are very

actively researched [Pietroni et al. 2022]. A promising research direction consists in methods based on *frame fields* and *integer-grid maps* [Bommes et al. 2013a], which have demonstrated enormous practical value for quadrilateral surface mesh generation [Bommes et al. 2013b]. However, so far industrial success of volumetric counterparts for hexahedral meshing is prevented by a lack of robustness. A typical integer-grid map approach consists of five major steps sequentially generating (i) feature aligned frame field, (ii) seamless map, (iii) integer quantization, (iv) integer-grid map, and (v) (potentially higher-order) hexahedral mesh. Provably robust algorithms exist only for the integer quantization (iii) [Brückler et al. 2022a] and the (linear) mesh extraction (v) [Lyon et al. 2016], while steps (i),(ii), and (iv) remain fragile.

Frame Field Meshability. The most problematic robustness issue stems from a topological mismatch between frame fields and integer-grid maps, as described in [Pietroni et al. 2022; Reberol et al. 2019; Viertel et al. 2016]. In frame fields, additional topological structures exist, which do not correspond to any configuration available in a quad/hex mesh, and which make the generation of a locally injective integer-grid map impossible. The 2D setting, where singularities are isolated points in the domain, is already well-understood. The seminal work of Myles et al. [2014] presents an algorithm that is capable of converting an input frame field on a triangular mesh into a (globally) meshable one. Additional theoretical guarantees have been identified in [Shen et al. 2022; Viertel and Osting 2019]. Much less is known about the volumetric setting, where singularities form a graph consisting of singular arcs and singular nodes. The configurations existing in hexahedral meshes have been enumerated [Liu et al. 2018; Nieser et al. 2011], however, non-meshable 3D frame field topologies remain rather unexplored. So far, there are empirical collections of problematic configurations [Reberol et al. 2019; Viertel et al. 2016], and correction algorithms, either requiring non-trivial user interaction [Corman and Crane 2019; Liu et al. 2018], or solely focusing on the meshability of singular arcs [Jiang et al. 2014; Li et al. 2012] but ignoring issues that occur at singular nodes, and certain feature configurations. In Sec. 4 we explain why a certain type of non-meshable singular nodes, called *zipper nodes*, frequently occurs in smoothness optimized frame fields. They are responsible for the low success rate of state-of-the-art algorithms, specifically in case of non-trivial feature constraints.

Goal. Given an input tetrahedral mesh with feature tags, cf. Fig.1, our goal is to generate a locally injective integer-grid map, preserving all feature points, curves and surfaces in its induced hexahedral mesh. We specifically target domains with internal features, which are substantially more challenging in terms of meshability than domains where all features are solely on the boundary.

Overview and Contributions. We mainly target the robustness issues of step (i), namely non-meshable topological configurations that exist in volumetric frame fields and are frequently generated by state-of-the-art algorithms [Reberol et al. 2019]. In Sec. 3.2 we carefully analyze the difference between topological structures that exist in general 2D frame fields but are not possible in quadmesh-induced frame fields. Necessary and sufficient conditions for the meshability of a frame field are identified, and we discuss an algorithm to turn a given frame field into a locally meshable, and with further processing into a (globally) meshable one. Sec. 4 is devoted to the meshability of 3D frame fields. We carefully analyze conditions for local meshability of singular arcs and singular nodes and study their decomposability into fundamental pieces that can be handled algorithmically. With the help of *arc zipping*, we design an algorithm that converts an input frame field into a locally meshable one. In Sec. 5 we describe a discretization enabling a concrete implementation based on piecewise constant frame fields on tetrahedral domains. Despite, only local meshability can be ensured – necessary but not sufficient for meshability – the evaluation of Sec. 7 on basis of the HexMe dataset [Beaufort et al. 2022] shows that our algorithm is able to construct valid integer-grid maps for 58% of the domains. Compared to the 2% success rate of state-of-the-art methods it is significantly more robust. In Sec. 6 we present a novel non-conforming optimization of locally injective seamless and integer-grid maps, again improving robustness w.r.t. available alternatives.

2 RELATED WORK

We review the state-of-the-art hex meshing techniques focusing on the related work of frame field-based methods. For a comprehensive overview, we refer to surveys [Armstrong et al. 2015; Pietroni et al. 2022; Yu et al. 2015]. Solely considering robustness, octree-based methods [Gao et al. 2019; Livesu et al. 2021; Maréchal 2009; Pitzalis et al. 2021] are attractive. However, the lack of a coarse block-structure, motivates mapping-based techniques.

Integer-Grid Maps. One key idea in quadrilateral and hexahedral mesh generation consists in constructing a map that transports cells of a regular Cartesian grid to a boundary-aligned and conforming mesh of the target shape. Some approaches require a block decomposition as input [White et al. 1996], while others restrict the space of considered maps. *Polycube maps* [Huang et al. 2014; Li et al. 2021; Sokolov and Ray 2015; Tarini et al. 2004] require a bijection between a continuous subset of the Cartesian-grid and the target shape, and thus forbid internal singularities. Extensions have been investigated to better handle shapes with handles [Fang et al. 2016; Mandad et al. 2022] or complex features [Guo et al. 2020], or targeting automatic block decompositions [Livesu et al. 2020]. However, the inherent restrictions do not admit high-quality meshes of complex domains with interior feature constraints, e.g. required for computational fluid dynamics involving multiple objects. Integer-grid maps (IGMs) [Pietroni et al. 2022] exist for all quadrilateral/hexahedral meshes and can thus in theory overcome all limitations. The generation of an IGM involves difficult non-linear mixed-integer optimization and is therefore decomposed into several consecutive steps, namely smooth frame field synthesis [Huang et al. 2011], locally injective seamless map construction [Du et al. 2020; Garanzha et al. 2021; Rabinovich et al. 2017], integer-quantization [Brückler et al. 2022a,b], and integer-grid map construction [Nieser et al. 2011]. Additional post-processing is performed to obtain a high-quality hexahedral mesh, including mesh extraction [Lyon et al. 2016], structural mesh improvement [Gao et al. 2015, 2017b], and geometrical mesh improvement [Brewer et al. 2003; Livesu et al. 2015; Marschner et al. 2020; Xu et al. 2018].

Frame field synthesis. Frame field based hex meshing was introduced by [Nieser et al. 2011], employing a frame field defined by a user-provided meta-mesh. The pioneering work of [Huang et al. 2011] established the spherical harmonics representation of frames, which is invariant to octahedral symmetry and thus suitable for continuous optimization. Improved boundary constraints [Ray et al. 2016], better optimization schemes [Palmer et al. 2020; Ray et al. 2016; Solomon et al. 2017], and odeco frame spaces [Palmer et al. 2020] have been investigated. All these methods optimize a Dirichlet-type energy, enabling smooth fields but not caring about meshability.

Frame field meshability. Not all frame fields are topologically equivalent to a quadrilateral/hexahedral mesh. While the 2D setting is well understood, and [Myles et al. 2014; Shen et al. 2022] offer guarantees regarding meshable topology, the topology of 3D frame fields remains rather unexplored. The seminal work of [Nieser et al. 2011] partially characterizes discrete frame field topology on tetrahedral meshes, with singularities located on edges and vertices. It is shown that singular edges can only be meshable if the rotation along the cycle of incident tetrahedra is aligned with the edge, and that singular vertex types are isomorphic to triangulations of the sphere. A corresponding complete enumeration of singularities of hexahedral meshes has been given in [Liu et al. 2018], which is sufficient to analyze local meshability of frame fields, and can help design singularity graphs for singularity-constrained field generation [Corman and Crane 2019; Liu et al. 2018]. However, globally sufficient conditions for meshability are still not available.

Frame field correction. Automatic correcting of meshability defects in frame fields has already been studied [Jiang et al. 2014; Li et al. 2012]. The key idea of [Li et al. 2012] is to collapse all non-meshable singular edges. The alternative approach [Jiang et al. 2014] splits non-meshable singular arcs into meshable ones. Unlike our approach, both prior methods do not investigate meshability of singular nodes and consequently can only repair a limited range of defects. Viertel at al. [2016] discusses two reasons for non-meshability, namely singular edges not being aligned to the field and limit cycles. They report that it is ineffective to force alignment of the field to static singular edges, motivating our reverse approach, deforming singular edges instead, discussed in Sec. 5.3. Furthermore, [Reberol et al. 2019] investigates non-meshable 3-5 arcs, often occurring in CAD models. Four strategies for correction are proposed, while only a heuristic that snaps singular arcs to the boundary is considered to be practically feasible. However, 3-5 arcs correspond to a special case of a non-meshable singular node, and in Sec. 4.5 we describe a systematic approach to handle all potential configurations.

3 MESHABILITY IN 2D

3.1 2D Frame Field Topology

The topology of a 2D frame field is fully characterized by the behavior of streamlines of its corresponding vector field on a 4-sheeted branched cover of the 2D domain $\Omega \subset \mathbb{R}^2$, as introduced in [Kälberer et al. 2007]. Consequently, we will briefly revisit vector field topology before proceeding with frame field topology. For a detailed and complete discussion of vector field topology we refer the reader to [Asimov 1993; Günther and Baeza Rojo 2021].

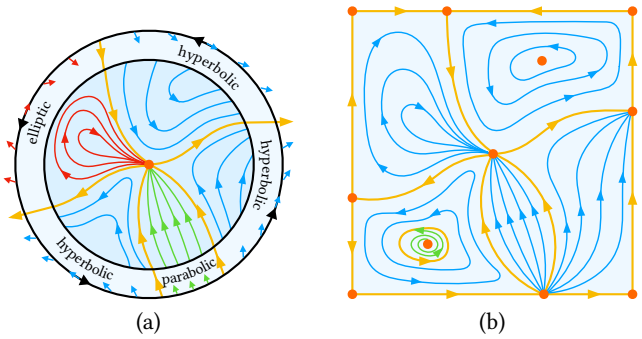


Fig. 2. (a) Local vector field topology of an isolated singularity (orange) with five sectors, formed by streamlines of parabolic(green), hyperbolic (blue), and elliptic(red) flow behavior. Separatrices (yellow) divide space into sectors of different flow behavior. (b) Global vector field topology: The topological skeleton consists of all separatrices (yellow), dividing space into regions of identical flow behavior, potentially including closed orbits. A limit cycle (green) is a streamline that converges to a closed orbit.

Local Vector Field Topology. An important topological feature of a vector field $v(x) : \Omega \to \mathbb{R}^2$ are its singularities, i.e. points, curves or regions where $||v(x)||_2 = 0$. In our setting only isolated point singularities are of importance, where the vector field does not vanish in an ϵ-disk neighborhood D_ϵ of the singularity with circular boundary $\partial D_\epsilon = C_\epsilon$. In such local neighborhood of a singularity, there are only four different types of streamlines that can be distinguished based on their limit behavior in forward and reverse flow directions, as illustrated in Fig. 2. A streamline is called a *closed orbit*, if it cycles on a closed path around the singularity, which in this case is called a *center*. Otherwise, the flow in forward/reverse direction can either converge to the singularity or diverge away from it. A *parabolic* streamline converges in one and diverges in the other direction. It is called *inflow* if the forward direction is converging, otherwise it is called *outflow*. A *hyperbolic* streamline diverges in both directions, and an *elliptic* streamline converges in both directions. Partitioning all streamlines passing through the local neighborhood circle C_ϵ according to their type, results in a cyclic sequence of parabolic, hyperbolic, and elliptic sectors. This cyclic sequence fully specifies the topological type of a point singularity. A *separatrix* is a parabolic or elliptic streamline separating space into sectors of different flow behavior. An example of the local vector field topology at an isolated singularity is shown in Fig. 2a. The *index* of an (interior) singularity quantifies how many full rotations the vector field undergoes when traversing C_ϵ in counter-clockwise (CCW) sense. Considering that parabolic sectors are bounded by separatrices of identical type, i.e. either both inflow or both outflow, while hyperbolic and elliptic sectors are bounded by separatrices of different flow direction, the integer-valued index of a singularity with n_p parabolic, n_h hyperbolic, and n_e elliptic sectors can be computed by $I = 1 - \frac{1}{2}n_h + \frac{1}{2}n_e$. The signs reflect that the field rotates CW in hyperbolic and CCW in elliptic sectors. Due to alternating in↔out-flow and continuity of the vector field $n_h + n_e$ is always even, ensuring that the index is an integer as expected.

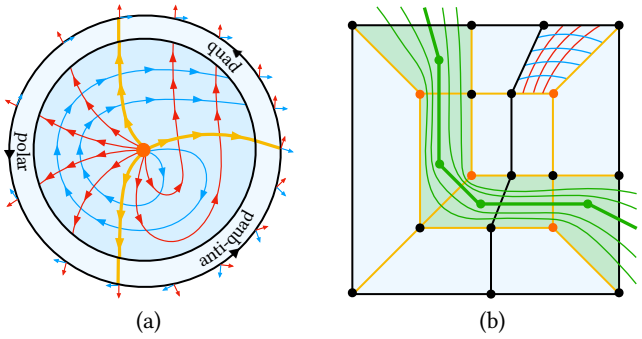
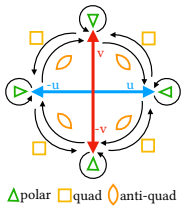


Fig. 3. (a) Local frame field topology at an isolated singularity (orange) with three different sectors of quad, polar, or anti-quad type. Only quad sectors are meshable since otherwise triangular and digon patterns appear. (b) Mesh-induced frame field.

Global Vector Field Topology. The *topological skeleton* encodes the global topology of a vector field by partitioning the domain into regions of identical asymptotic flow behavior. It consists of the union of all singularities and separatrices in the entire domain, including those induced by singularities, and moreover closed orbits at the interface between different flow behavior. On the global scale, there is one additional type of asymptotic streamline behavior that cannot be observed near isolated singularities. A streamline is called a *limit cycle*, if it asymptotically converges to a closed orbit. Fig. 2b depicts a topological skeleton including a limit cycle (green).

Local Frame Field Topology. A *frame* consists of four $\mathbb{R}^2$ vectors $\{u, v, -u, -v\}$, which are pairwise anti-parallel and often specified by a matrix $F = [u, v] \in \mathbb{R}^{2\times2}$. A *frame field* corresponds to a vector field on a 4-sheeted branched cover, where pointwise each branch corresponds to one of $\{u, v, -u, -v\}$, and sheets connecting non-trivially at singularities. For non-singular points with $||F||_2 \neq 0$, we require $\det F > 0$ such that u-streamlines can never be parallel to v-streamlines when projected onto the domain, and a u-vector on one branch cannot vanish independently of the v-vector on the other branch. It would be possible to solely discuss the topology of a frame field in terms of the topology of the vector field on the 4-sheeted branched cover. However, topological sectors will overlap when projected to the original domain, and several topological configurations, existing for entirely independent vector fields, are excluded by the $\det F > 0$ condition. This motivates the definition of new types of frame field sectors, which are meaningful on the domain itself and fully characterize the frame field behaviour. Similarly, to the vector field case, topological sectors at a singular point are defined by subsequent separatrices and the rotational behaviour of the field in-between them. However, in case of a frame field, CCW-subsequent separatrices might belong to different branches, e.g. an outflow-u-separatrix might be followed by an outflow-v-separatrix. Due to continuity of the frame field, when expressed in a common coordinate chart, two CCW-subsequent separatrices can only be on (i) the same branch, (ii) the next branch, or (iii) the previous branch, e.g. an outflow-u separatrix can only be followed by an outflow-u, an outflow-v, or an inflow-v separatrix.

We call the corresponding three types of sectors *polar*, *quad*, and *anti-quad*, all illustrated in Fig. 3a. The complete CCW-transition diagram is shown on the right. The $\frac{1}{4}\mathbb{Z}$-valued index of a frame field singularity with n_q quad, n_p polar, and n_a anti-quad sectors can be computed by



$$I = 1 - \frac{1}{4}n_q + \frac{1}{4}n_a \tag{1}$$

Global Frame Field Topology. In analogy to the vector field case the topological skeleton consists of the union of all singularities and separatrices, which partition the domain into flow-aligned regions. Additional nodes of the topological skeleton are induced by the intersection of separatrices on different branches.

3.2 2D Frame Field Meshability

We call a 2D frame field *meshable* if a quadrilateral mesh of identical topology exists. Comparing the topology of a frame field with the topology of a mesh is possible through the notion of the *mesh-induced frame field topology* introduced next.

Mesh-induced Frame Field Topology. The edges of a quadrilateral mesh in 2D can be interpreted as a discrete representation of the streamlines of an underlying frame field, as illustrated in Fig. 3b. A discrete streamline following a mesh edge continues through a regular vertex and either forms a closed cycle, or terminates at a boundary point or a singular vertex. The topology of all remaining streamlines through the domain is uniquely determined by the flow along edges of the dual mesh in the following way: All streamlines in-between two opposite edges of a quadrilateral are topologically identical to the corresponding dual streamline, i.e. the channel formed by a strip of subsequent quads, as illustrated in Fig. 3b in green. Please note that we have fully defined the topology of the mesh-induced frame field without explicitly constructing or defining the geometry of the field.

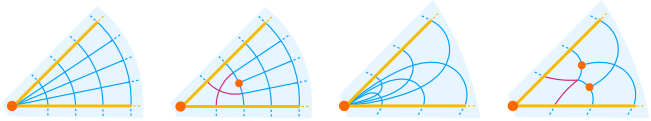
Meshability Conditions. Comparing the topological structures that exist in general frame fields with those induced by a mesh, we deduce the following necessary and sufficient conditions for meshability of a frame field in 2D:

(C1) *Feature Alignment* – The frame field aligns to all feature curves on the domain, to ensure their correct re-production in the quad mesh.
(C2) *Isolated Singularities* – All singularities with $||F||_2 = 0$ are isolated points and there are no curve or patch singularities, which might cause mesh degeneracies.
(C3) *Quad sectors only* – The frame field does not contain any polar or anti-quad sectors, which would induce non-quad cells in the mesh.
(C4) *No limit cycles* – A meshable frame field cannot contain limit cycles since all streamlines of the mesh-induced frame field topology are either closed orbits, or end at singularities or the boundary.

Sufficiency of the conditions can be verified in a constructive way. If (C1)-(C4) are satisfied, all cells of the topological skeleton are either equivalent to a quadrilateral, or an annulus. Enriching

the topological skeleton with all streamlines that are tangential to feature curves, and one streamline for each annulus to cut it into a quadrilateral, is sufficient to obtain a valid quad mesh. Necessity of the conditions is also straightforward since, violation of (C1) is in conflict with meshing of feature curves, violation of (C2) implies singularities that do not exist in mesh-induced frame field topologies, violation of (C3) generates triangles in the topological skeleton, and violation of (C4) induces a topological skeleton, which is not even a cell-complex anymore. Please note that sufficiency of (C1)-(C4) is only true for 2D domains, while for surfaces embedded in $\mathbb{R}^3$, additional degeneracies exist, e.g. Fig. 5 of [Myles et al. 2014].

Local Meshability. We call a frame field *locally meshable*, if at each point of the domain there is an arbitrarily small ϵ-neighborhood that is meshable. Local meshability requires that conditions (C1)-(C3) are fulfilled but omits (C4), which is only relevant on a global scale.

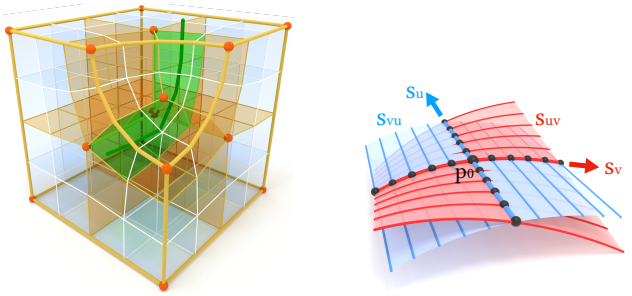


(a) (b) (c) (d)

Fig. 4. The polar sector in (a) is modified into the quad sector in (b) by adding an index $\frac{1}{4}$ singularity. Only a local sector neighborhood is altered, depicted with magenta streamlines. Similarly, the anti-quad sector in (c) is modified into a quad sector by adding two index $\frac{1}{4}$ singularities.

Ensuring Local Meshability. Given a frame field that satisfies (C1) and (C2), it is possible to turn it into a locally meshable field satisfying (C3), solely by modifying an ϵ-neighborhood of its singularities. As illustrated in Fig. 4ab, a polar sector at a singularity can be converted into a quad sector by compensating the added rotation with an $I = \frac{1}{4}$ singularity. The modification is restricted to an ϵ-neighborhood of the polar sector and does not alter the field outside. In an analog way, an anti-quad sector can be turned into a quad sector at the cost of locally adding two $I = \frac{1}{4}$ singularities, as depicted in Fig. 4cd. While a conceptual understanding of local meshability and modifications to ensure it is sufficient in our context, a concrete algorithm can be found in the supplemental material of [Myles et al. 2014], where sector constraints are added to the angle-based formulation of [Bommes et al. 2009].

Sector Modifications. A generalization of the polar/anti-quad sector removal results in a class of potential *sector modifications*. A modification of a flow-aligned local neighborhood $\mathcal{N}$ can be performed if and only if it respects the index theorem, i.e. $\Delta n_a = \Delta n_q$, with Δn_a, Δn_q quantifying the differences of total number of anti-quad and quad sectors of all singularities in $\mathcal{N}$. In the example of Fig. 4ab, $\Delta n_a = \Delta n_q = 0$ since we add one quad sector at the original singularity but remove one at a regular point, which becomes a singularity of index $\frac{1}{4}$, while the index of the original singularity reduces by $\frac{1}{4}$. For the example of Fig. 4cd, $\Delta n_a = \Delta n_q = -1$ since we add one quad sector and remove one anti-quad at the original singularity and remove one quad sector at two previously regular points. Please note that addition or removal of polar sectors does not require modification of other singularities in the neighborhood, and the same is true for addition or removal of pairs of quad and anti-quad sectors.



(a) (b)

Fig. 5. (a) a hexahedral mesh induces a frame field with streamlines aligned to all edges, and faces, and flowing through (green) channels created by dual chords (dark green) everywhere else with separatrices and separating surfaces in orange. (b) A pair of vector fields u and v is not necessarily integrable. Consequently, the red streamsurface S_{uv} formed by streamlines of v and seeded by S_u passing through p_0 is different from the blue streamsurface S_{vu} generated by exchanging the order of vector fields.

Ensuring Meshability. Given a frame field that is locally meshable, it can be turned into a (globally) meshable one with the motorcycle graph based algorithm proposed by Myles et al. [2014]. The zero loop elimination strategy effectively removes limit cycles, typically at the cost of adding pairs of singularities. We leave the discussion of directly converting the topological skeleton into a meshable one for future work, and instead continue with the main topic of this article, frame field meshability in 3D.

4 MESHABILITY IN 3D

Similarly to the 2D case, a hexahedral mesh in 3D can be seen as a discrete representation of streamlines of a 3D frame field, as illustrated in Fig. 5a. The singularities form a graph consisting of *nodes* and *flow-aligned arcs*, referred to as the *singularity graph* [Liu et al. 2018]. In contrast to 2D, there are two major differences that complicate the meshabillity of 3D frame fields, i.e. (i) non-uniqueness of streamsurfaces, and (ii) singular arcs. Both will be investigated in more detail next.

4.1 Streamsurfaces

While under mild continuity conditions, the streamlines of an arbitrary vector field $u(x) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ are uniquely defined by integration, this is no longer true for the streamsurfaces formed by $u(x)$ in combination with a second vector field $v(x) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, as illustrated in Fig. 5b. Assume that s_u and s_v are two streamlines of u respectively v that intersect at a point p_0. Then s_u serves as a seed curve for a well-defined stream surface s_{uv}, which is formed by all streamlines of v that pass through s_u. However, since away from s_u the surface resulting from streamlines along v is not necessarily tangential to u, the shape of the surface depends on the order of vector fields in this process. The stream surface s_{vu} seeded by s_v and formed by streamlines of u in general has a different shape,

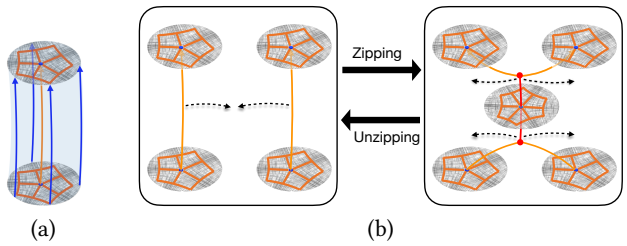


Fig. 6. (a) A footprint of a 2D singularity is extruded along the blue line field to the orange flow-aligned singular arc. (b) Two orange flow-aligned singular arcs of index $-1/4$ are partially zipped to the red singular arc of index $-1/2$, creating two red zipper nodes.

except for the seeds s_u and s_v which by construction are part of both surfaces. Only if the pair of vector fields is *integrable*, meaning that the necessary and sufficient conditions of the Frobenius theorem [Frankel 2011] are satisfied, the streamsurfaces are well defined in the sense that $s_{uv} = s_{vu}$ for any seed point of the domain. Consequently, unless we can guarantee integrability – e.g. for a hexmesh-induced frame field – there are no well-defined streamsurfaces that partition space into hexahedral cells. This prevents a direct extension of the 2D meshability results, resulting from the well-defined quadrilateral cells of the topological skeleton.

4.2 Singular Arcs

A local neighborhood of a 2D frame field, called *footprint*, can be extruded along an additional transversal line field to a tubular swept volume equipped with a 3D frame field, as depicted in Fig. 6a. If the footprint contains one isolated singularity, a sweep results in a *flow-aligned singular arc*. Such singular arcs are special in the sense that only two vectors of a frame $F = [u, v, w] \in \mathbb{R}^{3\times3}$ vanish, inducing a $rank(F) = 1$ property. The *index* of a flow-aligned singular arc is identical to the index of its footprint. If the footprint contains only quad sectors – a necessary condition for meshability – we call the sweep a (locally) *meshable singular arc*. Two flow-aligned arcs are *parallel* w.r.t. a (local) coordinate chart, if both belong to the same sheet of the branched cover. While hexmesh-induced frame fields contain only meshable singular arcs, general frame fields allow additional non-meshable singular arc types, exhibiting one or several of the following defects:

(D1) *Compound Monodromy* – In contrast to flow-aligned arcs, where the frame field rotates around one axis of the frame, other non-meshable monodromies from the Octahedral group are possible, as explained in more detail below.
(D2) *Flow Misalignment* – The singular arc is not tangential to the frame field, cf. Fig.7, which prevents meshability and inevitably violates the $rank(F) = 1$ property.
(D3) *Non-constant Footprint* – While the index of a flow-aligned arc is constant, according to Eqn. 1, the sectors can still continuously change along the arc in an index-invariant fashion, e.g. an isolated separatrix can turn into a polar sector, or into a pair of hyperbolic and elliptic sectors.
(D4) *Non-meshable Footprint* – Meshability requires a footprint equipped only with quad sectors, according to (C3).

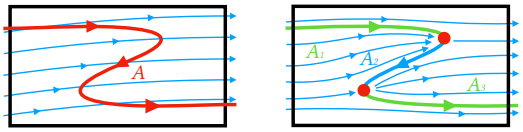


Fig. 7. (Left) the red singular arc A of constant monodromy μ_A is not aligned to the blue streamlines of u. (Right) After aligning $\pm u$ to A in a rotation-minimizing fashion, the u-aligned singular arcs A_1 and A_3 of index I (green) connect at singular nodes(red) to the $-u$-aligned arc A_2 of index $-I$ (blue).

The *monodromy* μ of a singular arc specifies the rotation a frame undergoes when traveling along a cycle enclosing the singular arc but no other singularities. When expressed in the coordinate system of the frame itself the monodromy is an element of the *octahedral group* O, as explained in [Corman and Crane 2019]. Only 10 out of 24 octahedral elements correspond to rotations around one axis of the frame and admit meshable singular arcs. According to [Jiang et al. 2014] we denote the remaining 14 elements as being of *compound type*, implying non-meshability of a singular arc, as well as violation of the $rank(F) = 1$ condition, both proven in [Nieser et al. 2011]. Next, we will describe an approach to convert a general frame field into one containing only meshable singular arcs, which is a necessary condition for meshability of a frame field.

Arc Zipping. A fundamental operation for modifying singular arcs and singular nodes are *zipping* and *unzipping*, both depicted in Fig. 6b. Given two parallel flow-aligned singular arcs A_1 and A_2 with index I_1 and I_2 respectively, the process of letting the arcs approach each other until they partially merge at a novel flow-aligned arc A_z is called *zipping*. The index of A_z is $I_z = I_1 + I_2$. The inverse process is called *unzipping*, and the resulting indices can be any I_1 and I_2 satisfying $I_1 + I_2 = I_z$. Like a zipper that can open and close continuously, both operations can be restricted to a sub-segment of the arcs, in which case two singular nodes, called *zipper nodes* – explained in detail in Sec. 4.3 – are created at the end points of the (un)zipping operation. Arc zipping operations are the 3D analog of the sector modifications of Sec. 3.2, acting inside the sweep volume of a flow-aligned singular arc. A regular streamline, i.e. a flow aligned arc of zero index, can be unzipped into arcs of opposite sign $I_1 = -I_2$. Such a pair of index $\pm\frac{1}{4}$ serve as generators for all other meshable arcs under the zipping operation since they correspond to addition respectively removal of one quad sector in the footprint.

Ensuring Meshable Singular Arcs. A general frame field can be converted into one with solely meshable singular arcs by successively removing defects of types (D1)-(D4). Each modification assumes that defects of all former types have already been resolved and guarantees that none will be re-introduced. Defects of type (D1) can be resolved following the observation of [Jiang et al. 2014] that all non-meshable monodromies are the product of meshable ones, or in other words two non-parallel arcs that have been zipped together. Consequently, a singular arc of compound type can always be decomposed into two arcs of non-compound type by non-parallel unzipping. Modification of the frame field is only required in an arbitrarily small local neighborhood of the compound arc. Defects

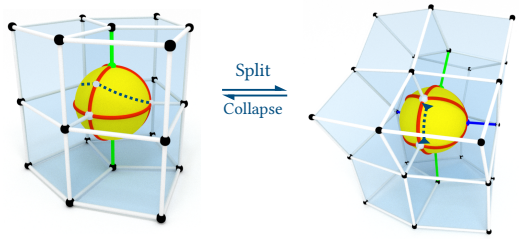


Fig. 8. (Left) unzipping an arc of index $\frac{1}{4}$ adds a quad sector and corresponds to an edge split (dashed green) in the triangulation. (Right) unzipping an arc of index $-\frac{1}{4}$ removes a quad sector and corresponds to an edge collapse (dashed green) in the triangulation.

of type (D2) can be resolved by a continuous transformation of the frame field in an epsilon neighborhood such that the field remains smooth but becomes tangential to the arc, as depicted in Fig.7. Note that the monodromy μ_A of an oriented arc A offers two alignment options $\pm a_\mu$ with $a_\mu \in \{u, v, w\}$, corresponding to the two frame axes remaining invariant under the monodromy transformation. The choice of sign matters for the resulting index since a rotation by α around a_μ is identical to a rotation by $-\alpha$ around $-a_\mu$. Consequently, the sign of a_μ needs to be determined geometrically, e.g. by minimizing the required rotation. If it flips along the arc, singular nodes are implied, splitting the arc into several segments of index with alternating sign, as depicted in Fig.7. Defects of type (D3) are resolved by splitting an arc into sub-arcs of constant footprint by introducing singular nodes at the transition points. All defects of type (D4) can be repaired by unzipping. For instance, the repair of a polar sector is done by unzipping an index $\frac{1}{4}$ singular arc transversally into the polar sector, modifying the footprint as in Fig. 4ab.

After ensuring that all singular arcs are of meshable type, all remaining defects preventing local meshability can only be located at singular nodes, which will be discussed next.

4.3 Singular Nodes

In a hexmesh-induced frame field all *singular nodes* – branching points where several singular arcs meet – are isolated point singularities with $||F||_2 = 0$, a condition identical to (C2) in 2D. All singular node types existing in hexmesh-induced frame fields are called *meshable*, see [Liu et al. 2018] for a complete enumeration. First, we will clarify that meshable singular nodes always result from the interaction of singular arcs, and that meshable arcs of index $\pm\frac{1}{4}$ not only serve as generators for all meshable arcs but also as generators for all meshable nodes.

Theorem 4.1. *All types of meshable singular nodes can be constructed by iteratively unzipping meshable singular arcs of index* $\pm\frac{1}{4}$.

Proof. Each meshable singular node type is homeomorphic to a triangulation of the sphere, as observed in [Nieser et al. 2011]. Each edge of the triangulation corresponds to one quad sector, being incident to the two flow-aligned arcs A_1 and A_2 represented by vertices opposite to the edge. From this perspective, unzipping a meshable singular arc of index $-\frac{1}{4}$ corresponds to an edge collapse

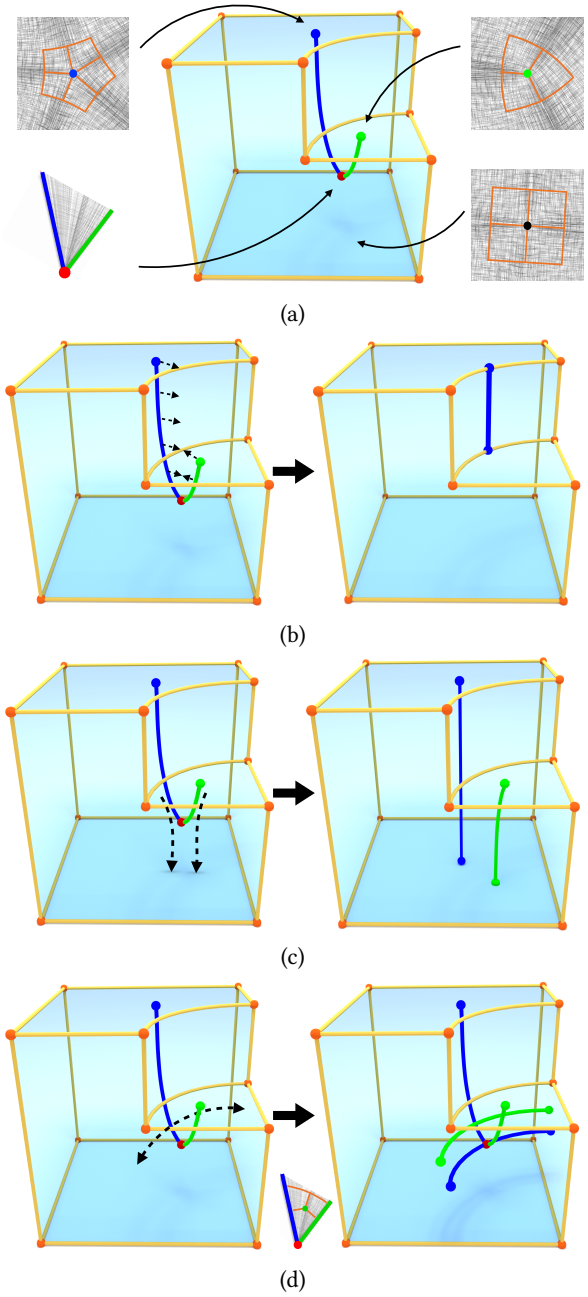


Fig. 9. (a) the smoothest frame field on the notch model contains a zipper node since the features on top require singularities, while the base favours a constant frame field without singularities. (b) resolving a zipper node by zipping, (c) resolving a zipper node by unzipping, (d) resolving a zipper node by unzipping a regular streamline and attaching the blue $-\frac{1}{4}$ arc to the parabolic sector at the zipper node, turning it into a quad sector.

(removing a quad sector from A_1 and A_2), while unzipping one of index $\frac{1}{4}$ is equivalent to an edge split (adding a quad sector to A_1 and A_2), both depicted in Fig. 8. All triangle meshes of genus zero

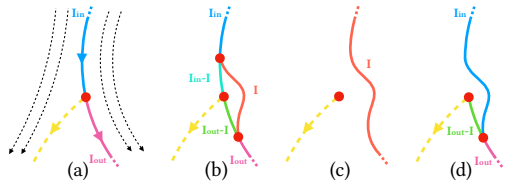


Fig. 10. Arc Unzipping at singular node (red): (a) Initial configuration with incoming arc (blue) and outgoing arc (magenta) forming a hyperbolic sector. (b) Unzipping an arc of index I generates new arcs of indices $I_{in}-I$ and $I_{out}-I$ and two zipper nodes. (c) Singular arc detachment with (T1) corresponds to $I = I_{in} = I_{out}$. (d) Arc unzipping with (T3) corresponds to $I = I_{in} \neq I_{out}$.

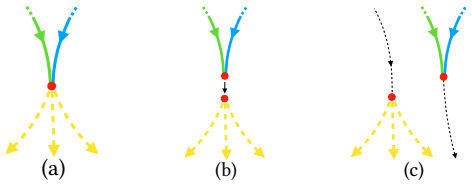


Fig. 11. Zipper Node Detachment (T2): (a) A parabolic sector at a singular node (red) is formed by singular arcs of index $\pm I$. (b) Zipping both singular arcs results in an isolated zipper node on a regular arc. (c) Perturbing the regular streamlines (black) detaches the zipper node from the singular node.

can be generated by a sequence of edge collapses and edge splits when starting from an octahedron (regular node). Consequently, all meshable singular nodes can be generated by iteratively unzipping meshable singular arcs of index $\pm\frac{1}{4}$. □

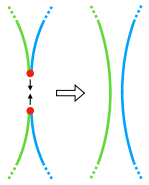
Zipper Nodes. Not all singular nodes resulting from the interaction of meshable singular arcs are meshable. While meshability of arcs ensures that their footprint is intact, it is still possible that two of them converge in a non-meshable fashion to a singular node, creating a node configuration of polar or anti-quad type. Since anti-quad configurations are rare in smooth fields, we leave their treatment for future work and focus the discussion to polar configurations corresponding to (non-meshable) parabolic sectors. The fundamental case is the *zipper node* of Sec. 4.2, resulting from an incomplete zipping operation where two parallel arcs A_0 and A_1 with indices I_0 and I_1 converge at the zipper node in another parallel arc A_z of index $I_z = I_0+I_1$. The special case of $I_z = 0$ is called an *isolated zipper node*, which in contrast to general zipper nodes is never forced to unzip through other singular nodes since the regular streamline A_z allows perturbation as illustrated in Fig. 11bc. Unfortunately, zipper nodes are not rare and can frequently be observed in frame fields that are optimized for smoothness, e.g. the notch model of Fig. 9a. The feature constraints on top require two singular arcs of index $\pm\frac{1}{4}$, however, the quadrilateral base favors a constant frame field. Consequently, the smoothest frame field exhibits an isolated zipper node, where the two singular arcs of opposite index meet in a parabolic sector and then continue as a regular arc.

Repairing Isolated Zipper Nodes. There are three different strategies to get rid of *isolated zipper nodes*, namely *Zipping*, *Unzipping*, and *Transversal Unzipping*, as illustrated in Fig. 9bcd. Identical repair strategies have been suggested in [Reberol et al. 2019], where zipper nodes are called non-meshable 3-5 arcs.

(S1) *Zipping* – Strategy I consists in zipping A_0 and A_1 together, i.e. pushing the zipper node forward such that the parabolic sector closes, as depicted in Fig. 9b.

(S2) *Unzipping* – Strategy II consists in unzipping A_0 and A_1 along A_z, i.e. pulling the zipper node along A_z such that the parabolic sector opens, as depicted in Fig. 9c.

(S3) *Transversal Unzipping* – Strategy III consists in leaving all three arcs meeting at the zipper node untouched but instead unzipping a regular streamline passing transversally through the parabolic sector formed by A_0 and A_1, generating a pair of $\pm\frac{1}{4}$ singular arcs. The $-\frac{1}{4}$ arc is unzipped to the tip of the parabolic sector turning it into a quad sector, as depicted in Fig. 9d. In general, there are two directions that are transversal to the parabolic sector, offering two topologically different modifications. Favorable due to lower geometric error is the one less aligned to the parabolic sector.

Termination of Isolated Zipper Node Repair. 
The unzipping operations of (S2) and (S3) require that a configuration is reached, where the zipper node can be locally resolved. Most commonly, the zipper node will reach a transversal boundary, where it can be disentangled into two separated parallel arcs, e.g. Fig. 9c. The second disentangling option consists in canceling a pair of anti-parallel zipper nodes of identical type, as illustrated in the inset. Both options are special cases of the general unzipping operation at singular nodes, which is described below and illustrated in Fig. 10. The infeasible case corresponds to an unzipping operation continuously moving the zipper node forward without ever terminating, e.g. getting caught in a limit cycle. Fortunately, arbitrary topology-preserving perturbations of the frame field are allowed before unzipping along the regular streamline, offering rich options to find a transversal boundary or an anti-parallel zipper node. Moreover, at any point of the path the unzipping direction can be changed to an orthogonal one by transversal unzipping with (S3). In all our experiments, valid unzipping paths could already be found by field perturbation. However, a complete analysis of termination guarantees is left for future work. The zipping strategy (S1) requires the existence of a surface bounding both singular arcs but not containing other singularities or features where closing the zipper would get stuck. Since the conditions of (S1) are only rarely satisfied in practice, unzipping via (S2) will be the main operation of our algorithm. Strategy (S3) is always feasible but generates an additional zipper node and will thus only be applied when (S2) is infeasible.

Repairing Non-Meshable Nodes. Zipper nodes are only one particular type of non-meshable node and we still need a general strategy to repair other types. Here, we assume that a non-meshable node is only incident to meshable arcs, e.g. ensured by the algorithm of Sec. 4.2. The key observation is that each such non-meshable node can be seen as a superposition of meshable singular arcs and zipper nodes. Consequently, it is not surprising that a generalization of Theorem 4.1 to non-meshable nodes exists, guaranteeing a full

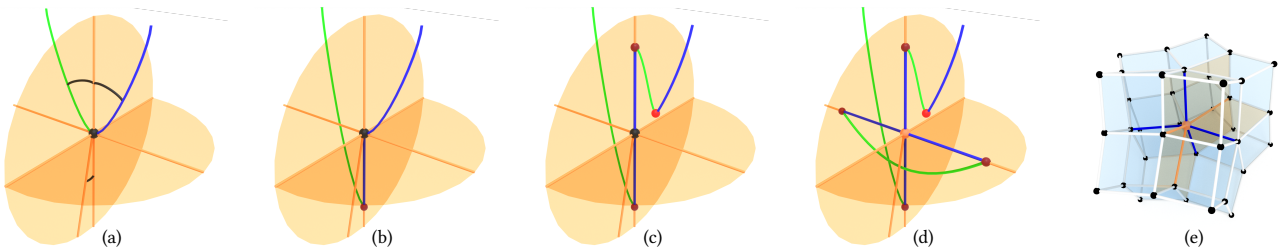


Fig. 12. (a) A non-meshable singular node (black) is incident to singular arcs (green and blue) and feature curves and surfaces (orange). (b) Arc unzipping with (T3) resolves a parabolic sector by pushing the green arc through the node, creating a novel zipper node (dark red). The blue arc is of negative index, such that another (T3) operation would result in a transversal polar sector on the orange feature surface. Instead, the parabolic sector is resolved with the parabolic sector reversal of (T4) shown in (c). The fully constrained parabolic sector between two orange feature curves can only be resolved by the transversal arc unzipping of (T5) shown in (d). The resulting orange singular node is locally meshable and corresponds to the hex mesh configuration of (e).

decomposition of a non-meshable node into a finite set of isolated meshable arcs and isolated zipper nodes. Singular arcs can be detached by unzipping through the singular node as shown in Fig. 10c, while zipper nodes can be detached by zipping, depicted in Fig. 11. Unzipping each zipper node finalizes the decomposition and clarifies that the reverse process constructs an arbitrary (non-meshable) singular node from a set of isolated singular arcs.

Theorem 4.2. *Every singular node of a frame field that is only incident to meshable singular arcs of index* $\pm\frac{1}{4}$ *can be decomposed into a finite set of isolated meshable singular arcs, and isolated zipper nodes by only modifying an arbitrarily small local neighborhood.*

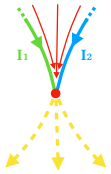
Proof. It suffices to observe that each quad sector incident to a singular arc A_1 continues along at least one other arc A_2, both forming a hyperbolic sector at the singular node. If the corresponding indices are equal, i.e. $I_1 = I_2$, we can detach a singular arc by unzipping as depicted in Fig. 10c. If two parallel arcs of opposite index form a parabolic sector, we can detach them by zipping to an isolated zipper node, as shown in Fig. 11. Iterating both operations until all singular arcs are detached concludes the decomposition. □

Theorem 4.2 is a generalization of the hex mesh singularity decomposition of the concurrent work [Zhang et al. 2023] to the setting of frame fields with meshable singular arcs but potentially non-meshable singular nodes. Note that the decomposition is not unique, depending on the order of unzipping/zipping operations.

4.4 Handling Feature Constraints

So far, we assumed that field alignment constraints are only caused by singular arcs, which themselves can freely deform within the domain. However, in our targeted hexahedral meshing setting, the domain is additionally equipped with an arbitrary set of immutable feature curves and feature surfaces, where the frame field needs to align as well. Please note that junctions of piecewise smooth feature curves and surfaces always induce singularities in the field, potentially enforcing non-meshable parabolic sectors in their vicinity.

Repairing Feature-Constrained Singular nodes. In case of feature constraints, the full decomposition of singular nodes based on Theorem 4.2 is no longer always possible. The required local modification of streamlines to detach arcs by unzipping and detach zipper nodes by zipping might be infeasible without creating new non-meshable sectors enforced by feature alignment constraints. Nevertheless, by adding three additional operations, i.e. *singular arc unzipping*, *parabolic sector reversal*, and *transversal unzipping*, a transformation into a locally meshable node and sets of meshable arcs and zipper nodes can still be obtained in a systematic way. Our local singular node repair is driven by four design principles, (i) all non-meshable (parabolic) sectors incident at the singular node are successively resolved, (ii) newly introduced parabolic sectors are not incident to the singular node and are of zipper node type with unzipping direction away from the singular node, (iii) all singular arcs remain meshable, and (iv) modifications enabling a lower number of singular arcs are preferred over other alternatives. The transformation considers five different operations to resolve parabolic sectors. Each of them targets a parabolic sector formed by a pair of arcs A_1 and A_2 with indices I_1 and I_2. We call the parabolic sector *fully constrained* if both arcs are tangential to different features, *partially constrained* if only A_1 is tangential to a feature and otherwise *unconstrained*. An operation is not allowed if new parabolic sectors incident to the singular node are created. Specifically, it is not allowed to degenerate constrained quad sectors into polar sectors, or to generate new arcs that form parabolic sectors with feature curves or surfaces. If multiple operations are possible, they are prioritized in the given order.



(T1) *Singular Arc Detachment* – A parabolic sector is resolved by detaching the singular arc A_2 with the unzipping operation illustrated in Fig.10c. Existence of an outgoing arc of index I_2 is required. Fully constrained parabolic sectors cannot be resolved by (T1) and the operation might be inadmissible for $I_2 < 0$ since the number of transversal quad sectors incident to the singular node is reduced. However, unzipping with $I_2 > 0$ is always allowed since quad sectors are added.

(T2) *Zipper Node Detachment* – Two parallel singular arcs of opposite index $I_1 = -I_2$ are detached from the singular node by zipping, resulting in an isolated zipper node, illustrated in Fig. 11. The operation is only allowed if both singular arcs can move freely, i.e. for unconstrained parabolic sectors, and no feature surface separates both arcs.

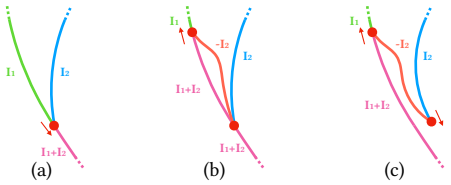


Fig. 13. Parabolic Sector Reversal (T4): (a) The initial zipper node (red) is unzipping the blue arc of index I_2 from the magenta arc. Unzipping direction and unzipping index can be reversed by first unzipping the red arc as shown in (b), creating a reversed zipper node unzipping with $-I_2$, and subsequently detaching the magenta arc of index $I_1 + I_2$ creating an additional isolated zipper node shown in (c).

(T3) *Arc Unzipping* – A parabolic sector is pushed through the singular node by unzipping A_2, illustrated in Fig. 10d. The operation is similar to (T1), however, allowing an outgoing arc of different index, resulting in a zipper node unzipping along the outgoing arc. Similarly to (T1) it is inadmissible for fully constrained parabolic sectors and might be inadmissible if $I_2 < 0$, as shown in the example of Fig. 12bc.

(T4) *Parabolic Sector Reversal* – A not fully-constrained parabolic sector can be resolved at the cost of creating two new zipper nodes as depicted in Fig. 13. One of them is isolated while the other one lies on A_1 and unzips away from the singular node. Since the index of the unzipping arc turns into $-I_2$, the (T4) operation unlocks situations where (T1) and (T3) operations are blocked due to a negative I_2.

(T5) *Transversal Arc Unzipping* – Perform arc unzipping via (T3) but transversally to the parabolic sector and with an index of $\frac{1}{4}$. The parabolic sector is turned into a quad sector at the cost of generating two novel (outgoing) zipper nodes on transversal arcs of the singular node. Since the unzipping is done with a positive index, the (T5) operation is always admissible. It corresponds to a local version of the (S3) strategy for zipper node repair.

Fig. 12 shows an example of a feature-constrained singular node, which is transformed into a locally meshable one by performing operations (T3), (T4), and (T5).

Theorem 4.3. *By only modifying an arbitrarily small local neighborhood, every singular node of a frame field that is incident to arbitrary feature curves and surfaces and only meshable singular arcs, can be modified into a locally meshable singular node connected to a set of outgoing zipper nodes, and sets of additional isolated meshable singular arcs, and isolated zipper nodes.*

Proof. Operation (T5) resolves any parabolic sector at the singular node by adding two outgoing zipper nodes. Hence, iteratively applying (T5) is sufficient to obtain the targeted decomposition by successively adding quad sectors to arcs incident at the singular node until all parabolic sectors are pushed to the outgoing zipper nodes. Including operations (T1)-(T4) is not necessary to obtain the decomposition, however, important to minimize the resulting number of zipper nodes. □

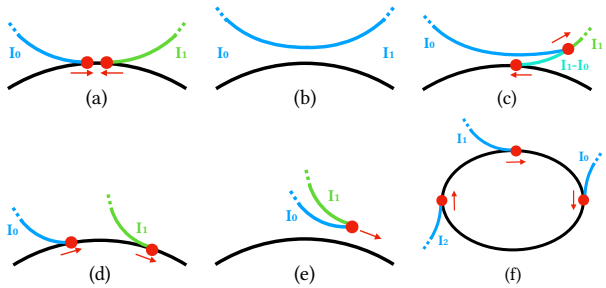


Fig. 14. (a) Collision of two anti-parallel zipper nodes, where the blue and green arc are unzipped from the black feature curve. (b) For identical indices $I_0 = I_1$, both zipper nodes can be disentangled into separate arcs with (T3). (c) Given that both indices are of equal sign, arc unzipping with (T3) turns one zipper node into an isolated one. (d) Collision of two parallel zipper nodes, where the blue and green arc are unzipped from the black feature curve. (e) For opposite indices $I_0 = -I_1$, an isolated zipper node can be detached with (T2). (f) Collision of other parallel zipper nodes can often be avoided by changing the unzipping order, unless cyclic dependencies exist that need to be unlocked through transversal unzipping with (T5).

Resolving Non-Isolated and Feature-Constrained Zipper Nodes. After repairing all singular nodes based on Theorem 4.3, the only remaining defects are zipper nodes. The repair of isolated zipper nodes has already been clarified but non-isolated and feature-constrained zipper nodes require additional considerations. While isolated zipper nodes can move rather freely by perturbing regular streamlines, the trajectory of non-isolated or feature-constrained zipper nodes is significantly more restricted such that collision with singular nodes or other zipper nodes might be unavoidable. Zipper nodes forming partially constrained or unconstrained parabolic sectors allow unzipping of at least one arc. If the unzipping arc is of positive index, operation (T3) ensures that unzipping through a singular node is always possible without creating novel defects. If unzipping with negative index gets blocked at a singular node, the reversal operation (T4) is performed, resulting in a reversed positive unzipping index and one additional isolated zipper node. Collision of two zipper nodes is allowed for two specific cases and treated in the following way: (i) anti-parallel unzipping arcs with index of identical sign trigger a (T3) operation, resolving both zipper nodes if $I_1 = I_2$, and otherwise turning one zipper node into an isolated one, both shown in Fig. 14abc. (ii) in case of parallel unzipping arcs of opposite index ($I_1 = -I_2$), an isolated zipper node is detached via (T2), as shown in Fig. 14e. For anti-parallel cases of opposite sign, the unzipping arc of negative index is reverted via (T4). Other parallel collisions can be avoided by changing the order of unzipping, i.e. first unzipping the zipper node in front. Cyclic dependencies between parallel zipper arcs might exist, e.g. several parallel unzipping arcs of identical index on a cyclic feature curve as shown in Fig. 14f. These can be unlocked by transversal unzipping with (T5). The last case that requires special attention is zipper nodes with fully constrained parabolic sectors caused by feature alignment constraints, where only transversal unzipping with (T5) can avoid novel parabolic sectors.

4.5 Algorithm: Locally Meshable Frame Field

Combining all components of Sections 4.2, 4.3, and 4.4, we obtain an algorithm to establish a locally meshable frame field. Starting from a given domain equipped with arbitrary feature curves, feature surfaces and a smooth frame field tangential to all features, the algorithm performs four major steps, each targeting specific types of defects and not re-introducing previously resolved ones.

(A1) *Establish Meshable Arcs* – All defects (D1)-(D4) related to non-meshable singular arcs and feature arcs are repaired following the procedure of Sec. 4.2.

(A2) *Establish Meshable Nodes* – All non-meshable singular nodes, except zipper nodes, are transformed into locally meshable ones. Parabolic sectors incident at a singular node are iteratively resolved with the prioritized operations (T1)-(T5) of Sec. 4.4, potentially adding zipper nodes.

(A3) *Resolve Non-Isolated and Feature-Constrained Zipper Nodes* – All non-isolated and feature-constrained zipper nodes are eliminated following the rules of Sec. 4.4, potentially adding isolated zipper nodes.

(A4) *Resolve Isolated Zipper Nodes* – All isolated zipper nodes are resolved one after the other by unzipping with (S2), falling back to transversal unzipping with (S3) if necessary.

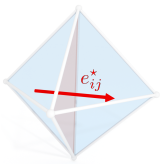
Steps (A1) and (A2) are guaranteed to succeed and solely modify arbitrarily small local neighborhoods of the non-meshable arcs and nodes. In contrast, the resolution of zipper nodes in steps (A3) and (A4) requires non-local frame field modifications and lacks a formal termination guarantee.

5 DISCRETE LOCAL MESHABILITY

The operations necessary for obtaining local meshability with the algorithm of Sec. 4.5 can be implemented in the discrete setting of tetrahedral meshes. We will first clarify the discretization and optimization of frame fields and then explain the discrete repair operations as well as a complementary handling of noisy singularity graphs, frequently occurring in the discrete setting.

Domain and Feature Representation. The domain is represented with a piecewise linear tetrahedral mesh $\mathcal{T} = (V, E, T, C)$ formed by vertices V, edges E, triangles T, and cells C. All feature nodes, curves and surfaces are embedded in the tetrahedral mesh. Accordingly, the subsets $V_F \subset V$, $E_F \subset E$, and $T_F \subset T$ correspond to discretized feature nodes, curves, and surfaces. In our algorithm the tetrahedral mesh is not static, and will be modified or refined whenever necessary, always ensuring that the topology of all feature entities and the domain remains intact.

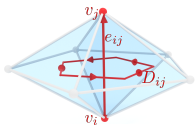
Frame Field Representation. The frame field F is represented by elements of $SO(3)$ in combination with octahedral group $O \subset SO(3)$ valued matching transformations M, which define the connection between neighboring charts. The frame field is discretized in a piecewise constant manner by assigning a frame $F_i = [u_i, v_i, w_i] \in \mathbb{R}^{3\times3}$ to each tetrahedron $c_i \in C$, where the columns define three vectors in the chart of c_i, and $det(F_i) = 1$. Consequently, the matchings are given as $M_{ij} \in \mathbb{R}^{3\times3}$ for oriented dual edges $e^\star_{ij} \in E^\star$ between cells c_i and c_j. A frame F_i can be expressed in a neighboring chart by the transformation $\eta_{i\to j}(F_i) = F_i M_{ij}$, also clarifying that $M_{ij} = M_{ji}^{-1}$. Please note that our discretization is similar to the one of [Liu et al. 2018] and our matchings are inverse to the linear part of their integer-grid map transitions, i.e. $M_{ij} = R_{j\to i}$.



Feature Alignment Constraints. Each feature edge imposes the constraint that in all incident cells one of the frame axes $\{\pm u, \pm v, \pm w\}$ aligns to it. Similarly, each feature triangle requires that one frame axis within neighboring cells aligns to its normal vector, ensuring that the orthogonal ones are tangential. Whenever a cell is incident to multiple features resulting in a locally over-constrained situation, we split edges of the corresponding tetrahedron to isolate alignment constraints, similarly to [Liu et al. 2018]. Only in case of a feature face incident to a feature edge, a cell with two alignment constraints is allowed, consistently specifying the full frame.

Separation of Singularities and Features. Singular nodes and arcs are represented by subsets of vertices $V_S \subset V$ and edges $E_S \subset E$ of the tetrahedral mesh. To avoid special cases, we assume that singular edges, singular vertices and features are maximally isolated and do not interfere unnecessarily. Specifically, in the submesh $\mathcal{T}_v \subset \mathcal{T}$ containing all cells incident to a singular/feature vertex v, all other features/singularities are incident to v and none are on the boundary $\partial\mathcal{T}_v$. The same is required for singular/feature edges and both requirements can be easily fulfilled by iteratively splitting violating mesh edges until satisfied.

Edge Monodromy. Given an (interior) oriented edge $e_{ij} \in E$ of the tetrahedral mesh pointing from vertex v_i to vertex v_j, the monodromy of the consistently ccw-oriented dual cycle D_{ij} enclosing e_{ij} and visiting a sequence of cells $c_{i_0}, c_{i_1}, \ldots, c_{i_k}$ can be determined by concatenating the corresponding matching transformations along the dual edges of the cycle. The resulting monodromy of an oriented edge e_{ij} expressed in the chart of c_{i_0} is



$$\mu_{ij} = M_{i_0 i_1} M_{i_1 i_2} \ldots M_{i_k i_0} \tag{2}$$

It characterizes the integral rotation of the frame when traveling along the dual cycle expressed in the coordinate system of the frame itself, resulting in an orientation-preserving permutation. All edges of non-identity monodromy are singular. However, the monodromy alone is neither sufficient to determine all singularities, nor able to fully characterize the index and sector types of flow-aligned arcs. The problem is that the $SO(3)$ valued monodromy, as defined above, is insensitive to full rotations, e.g. a regular flow-aligned arc of index 0 cannot be distinguished from a singular one with integer index $I \in \mathbb{Z}$. Fortunately, in case of flow-aligned singular edges, the ambiguity can be resolved.

Flow-Aligned Edge Index and Sector Types. Assume an edge e_{ij} with monodromy μ_{ij}, which is not of compound type. In this case, when traveling along the dual cycle at least one frame axis comes back to itself, which is the rotation axis of μ_{ij}. Consequently, it is possible to locally align the frames of all cells incident to e_{ij} such that the local field topology is fully characterized by the rotation of the 2D frame field of frame vectors orthogonal to e_{ij}. Additionally

assuming that the rotation angle between neighboring frames is below π – e.g. by adequate sampling – the index of an (interior) edge e_{ij} can be computed via

$$I_{ij} = \frac{1}{2\pi} \sum_{e^\star_{kl} \in D_{ij}} \alpha_{ij}(F_k M_{kl}, F_l) \tag{3}$$

with $\alpha_{ij}(F, G)$ measuring the signed rotation angle to transform frame F into frame G by a rotation around axis e_{ij}. Please note that the index does not depend on the orientation of edges, i.e. $I_{ij} = I_{ji}$ holds since the sign change of α is compensated by reversal of dual edges. Sector types can be determined in a similar fashion by investigating sub-segments of the dual cycle D_{ij}. A sub-segment $D \subset D_{ij}$ corresponds to a fan of tetrahedra incident at e_{ij}. In case the frame field is not only aligned to e_{ij} but also to the bounding triangles of the tetrahedral fan, the sector type is revealed by measuring the difference between the opening angle of the fan and the rotation of the frame. The corresponding *sector angle* γ_D is defined as

$$\gamma_D = \sum_{c_{ijkl} \in D} \beta_{ij}^{kl} - \sum_{e^\star_{kl} \in D} \alpha_{ij}(F_k M_{kl}, F_l) \tag{4}$$

with β_{ij}^{kl} being the dihedral angle incident to edge e_{ij} of the tetrahedron with vertices (i, j, k, l). For *quad* sectors $\gamma_D = \frac{1}{2}\pi$, for *polar sectors* $\gamma_D = 0$, and for *anti-quad sectors* $\gamma_D = -\frac{1}{2}\pi$. This is consistent with Eqn. (1) since for the full dual cycle $I_{ij} = 1 - \frac{1}{2\pi}\gamma_{D_{ij}}$. In general, there is no reason that the faces of the tetrahedral mesh conform to all sectors at a singular edge. However, for constrained sectors that are bounded by features faces (or feature/singular edges), the alignment holds and γ_D can be computed robustly, revealing local meshability defects caused by polar or anti-quad sectors in the footprint of a singular arc.

Algorithm 1 FeatureAlignedSmoothField

Input: Tet mesh $\mathcal{T}$, initial field $\mathcal{F} = (F, M)$, alignment constr. $\mathcal{A}$
Output: Optimized field $\mathcal{F}_{new} = (F_{new}, M)$

```
for k = 1 ... NumIters do
    for c_i ∈ C do
        for e*_ij ∈ E* do
            g ← f_j m_ji                          ▷ express f_j w.r.t. f_i
            f_i ← f_i + sgn(g · f_i) g            ▷ accum. with correct sign
        end for
        f_i ← normalize(f_i)
        f_i ← project(A, f_i)                     ▷ align to features
    end for
end for
```

5.1 Frame Field Optimization

One important component of our algorithm is the optimization of a smooth frame field subject to fixed topology and feature alignment constraints. We adopt a simple iterative averaging strategy inspired by [Gao et al. 2017a], where the frames and matchings are represented as unit quaternions $f_i, m_i \in \mathbb{R}^4$. However, in contrast to [Gao et al. 2017a] we do not re-compute the matchings on the fly since we want to preserve the field topology. The average of two rotations represented as unit quaternions q_0 and q_1 can be obtained by normalizing their sum $q_s = q_0 + sgn(q_0 \cdot q_1)q_1$, which corresponds to the unit quaternion closest to their average. The sign of the dot product is necessary to deal with the double-covering of $SO(3)$ by unit quaternions, resulting in the ambiguity that one element of $SO(3)$ corresponds to quaternions $\pm q$. A meaningful average is obtained by choosing signs such that both quaternions lie in the same hemisphere. Correctly averaging several elements of $SO(3)$ is significantly more difficult since a suitable choice of signs is not always possible and the computation of a Fréchet mean would be required. However, iteratively accumulating several quaternions with sign correction is a cheap approximation, which is sufficient to optimize smooth frame fields. Feature alignment constraints can be enforced by projecting the frame of the averaged quaternion onto the closest aligned one as described in the supplemental material of [Gao et al. 2017a]. The complete frame field optimization algorithm, iteratively averaging and projecting quaternions, is shown in Algorithm 1. The optimization can either be performed on the entire mesh, or alternatively be restricted to a submesh, e.g. to obtain a smooth field in a local neighborhood where the topology has been changed by a repair operation. It should be noted that the matchings M constrain the frame field topology according to their induced monodromies but only specify sector angles modulo 2π. We did not observe undesired topology changes in our experiments such that development of alternatives with full topology control is left for future work.

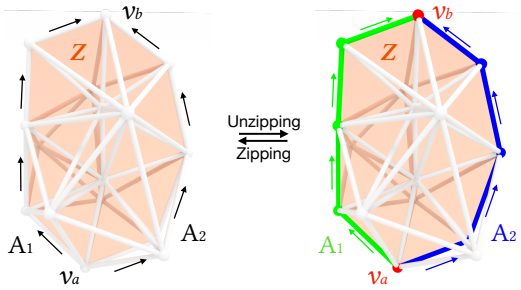


Fig. 15. Discrete Zipping/Unzipping requires a disc-topology patch of triangles Z, where the boundary edges are partitioned into oriented arcs A_1 and A_2. Consistently changing the matchings dual to Z and aligning frames to A_1, A_2 changes the index of A_1 by $+\Delta I$ and of A_2 by $-\Delta I$

5.2 Discrete Local Meshability Repair

The core operation of local meshability repair is the arc zipping operation, such that we will first clarify its discrete counterpart.

Discrete Arc Zipping. Consider a triangle mesh of disc topology $Z \subset T$ that is formed by a subset of triangles of $\mathcal{T}$. For simplicity, assume that all matchings are identity. By changing the matchings of all dual edges that pierce triangles of Z in a consistent orientation to $M_z \in O$, only monodromies at the boundary ∂Z are changed since dual cycles of interior edges always traverse Z in both directions such that M_z and M_z^{-1} cancel out. Choosing two vertices v_a, v_b on ∂Z, partitions the boundary into two arcs A_1 and A_2, both oriented from v_a to v_b. Combining the monodromy change with alignment

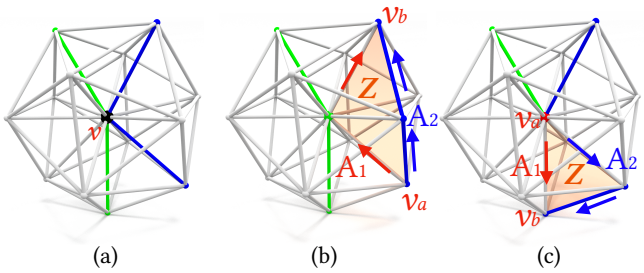


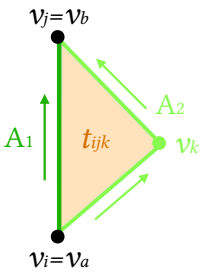
Fig. 16. (a) Two singular arcs of index $\pm I$ touch at vertex v. (b) Discrete singular arc detachment (DT1) is realized by discrete unzipping. (c) Discrete zipper node detachment also corresponds to discrete zipping. Both require a fan of triangles Z bounding a sector of suitable flow behavior.

of the frame field to A_1 and A_2 – choosing a frame axis consistent with M_z – alters the zero index of edges of A_1 to I and those of A_2 to $-I$, while not changing the index of any other edge in $\mathcal{T}$. Hence, we have unzipped a regular streamline to two singular arcs merging at zipper nodes given by v_a and v_b. Reverting the process by an inverse M_z^{-1} change corresponds to zipping, both shown in Fig. 15.

The general *discrete arc zipping* operation is specified by the tuple $[Z, A_1, A_2, \Delta I]$ consisting of an oriented disc topology triangle surface Z, parallel oriented arcs A_1, A_2 partitioning ∂Z, and the desired index change ΔI. From this information, novel matchings for dual edges crossing Z are determined such that the index of edges of A_1 increases by $+\Delta I$, while the index of edges of A_2 decreases by $-\Delta I$. A prerequisite of discrete arc zipping is that all interior edges of Z are non-singular such that their monodromy is the identity. In practice, we do not require that the neighborhood of Z is represented in a common chart as assumed in the introductory example. Considering that the target indices and thus the required monodromies are known, we can uniquely determine the single unknown matching of a boundary edge of ∂Z through Eqn. (2). Proceeding with edges of a single unknown matching until all matchings of Z are determined coincides with the *chart zippering* procedure of [Liu et al. 2018].

Establishing Meshable Edges. Edges of the tetrahedral mesh represent singular arcs and consequently might suffer from all four meshability defects discussed in Sec. 4.2. Defects of a single edge e_{ij} can be resolved locally by the discrete analog of modifications discussed in Sec. 4.2.

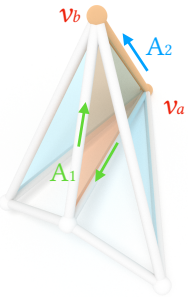
A discrete (D1) defect, i.e. an e_{ij} with compound monodromy μ_{ij}, is fixed by discrete arc unzipping on any incident triangle t_{ijk}, where only e_{ij} is singular or feature constrained. The monodromy of e_{ij} is turned into a meshable one at the cost of generating a new singular arc formed by e_{ik} and e_{kj}, as shown on the right. If all incident triangles have additional singular or feature edges, a valid candidate triangle can always be obtained by a $1:3$ split of any tetrahedron incident to e_{ij}.



A discrete (D2) defect, i.e. flow misalignment is resolved by first splitting all tetrahedra exhibiting multiple conflicting alignment constraints, and then obtaining a feature and singularity aligned smooth field through Algorithm 1 restricted to the one-ring of e_{ij}.

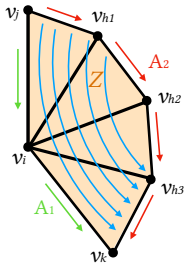
After aligning the field to all singular edges and features, non-constant footprint defects of type (D3) can be identified by computing sector angles following Eqn. 4. Adding a singular node at each vertex where the footprint changes, resolves all defects of type (D3).

Discrete (D4) defects, i.e. constrained non-quad sectors incident at e_{ij}, might be caused by singular edges with non-meshable index $I \geq 1$ or sector angles less than $\frac{1}{2}\pi$ sandwiched between feature triangles. Unzipping arcs with index $\Delta I = \frac{1}{4}$ on a triangle t_{ijk} inside the non-meshable sector increases the sector angle such that repeated execution is sufficient to resolve all non-meshable sectors. Suitable triangles within non-meshable sectors can always be ensured through edge splits. A subsequent (D2) repair is necessary to warrant field alignment of newly created singular arcs.

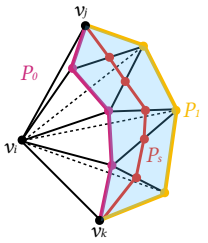


Establishing Meshable Vertices. Vertices of the tetrahedral mesh represent singular nodes and consequently might suffer from non-meshability of parabolic sectors formed by zipper configurations as discussed in Sec. 4.3. The required repair operations (T1)-(T5) discussed in Sec. 4.4 are all specializations of arc zippering, enabling discrete counterparts achieving local meshability of a vertex v_i within its one-ring of incident tetrahedra.

Given a singular edge e_{ij}, the discrete singular arc detachment (DT1) is realized by discrete arc unzipping as illustrated in Fig. 16ab. It requires a fan of triangles formed by v_i and a chain of one-ring neighbors $v_j v_{h_1} \dots v_{h_m} v_k$ such that (i) edges e_{ij} and e_{ik} are of identical index, (ii) all interior edges $e_{ih_1} \dots e_{ih_m}$ as well as edges along the chain $e_{h_l h_{l+1}}$ are regular and non-feature, and (iii) the streamline entering through e_{ji} forms a hyperbolic sector with the streamline exiting along e_{ik} represented by the holonomy of the frame field restricted to the triangle fan. For a given incoming singular edge e_{ji} we perform a breadth-first search to determine both, a suitable outgoing singular edge e_{ik} and the corresponding fan of triangles. We rely on a dual path search such that the field holonomy is well-defined by the matchings.

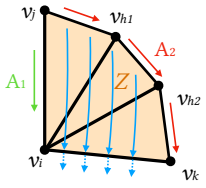


A valid triangle fan for the discrete arc unzipping operation is then obtained either by triangles forming the side walls of the dual region spanning from e_{ji} to e_{ik} corresponding to paths P_1, P_2, or alternatively by splitting edges of all interior triangles of the dual region resulting in path P_s, all depicted on the right. We run the breadth-first search from all tetrahedra incident to e_{ij} and exclude dual edges corresponding to feature surfaces with a flow parallel to e_{ij}. In order to avoid spurious edge pairs not incident to a common hyperbolic sector, possible since 2π rotations are not disambiguated, the search is required to roughly follow streamlines. We forbid dual edges, where the frame axis corresponding to the

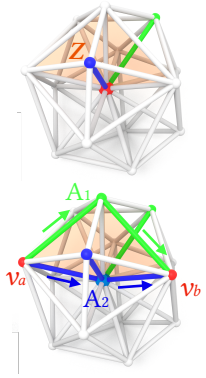


streamline in question deviates by more than $\beta = 100°$ from the outgoing normal vector of its primal face.

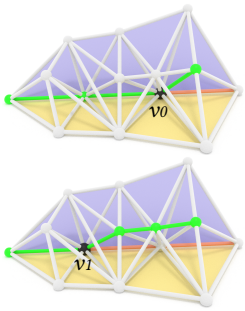
In addition to interior hyperbolic sectors formed by e_{ji} and e_{ik}, the path search also considers the case of a boundary truncated hyperbolic sector, where the streamline exits transversally, as depicted on the right. If multiple valid detachment candidates are found, we prioritize the boundary case and otherwise choose randomly.



The discrete zipper node detachment (DT2), depicted in Fig. 16c, is mostly identical to (DT1), except that edges e_{ij} and e_{ik} of opposite index are searched that form an outgoing parabolic sector. Discrete arc unzipping (DT3) and discrete parabolic sector repair (DT4) are identical to (DT1) and (DT2) respectively, except that different indices for both participating sector-edges are allowed. Discrete transversal arc unzipping (DT5) starts from a pair of edges forming a parabolic sector represented by a triangle fan. However, discrete arc unzipping is performed on a transversal Z as depicted on the right. To simplify the implementation, we do not explicitly search the edges corresponding to transversal separatrices incident at v_i. Consequently, in contrast to the continuous (T5) operation its discrete counterpart might create novel parabolic sectors. Those will be resolved subsequently by (DT4) at the cost of unnecessary isolated zipper nodes, which are effectively eliminated by the continuous singularity optimization described in Sec. 5.3. In agreement with the continuous operations, we forbid arc unzipping (DT1),(DT3) with negative ΔI whenever novel parabolic sectors are generated. It is reverted if an exhaustive enumeration of all parabolic sectors between pairs of (singular/feature) edges reveals novel defects.



Resolving Zipper Vertices. Non-isolated zipper vertices can be iteratively unzipped by the (DT3) operation as depicted on the right. Identically to the continuous setting, unzipping with negative index through singular vertices might be forbidden, triggering a (DT4) reversal. Isolated zipper vertices are different in the sense that the unzipping path is not constrained to a given set of mesh edges. Consequently, we perform a flow-guided dual path search to determine an optimal unzipping arc. The objective is to find either a transversal boundary triangle, or an anti-parallel zipper node of identical type since in both cases the zipper node can be resolved. The path should also be short and roughly aligned to the field. Assuming a coordinate system where the unzipping direction is along u, we explore dual paths w.r.t. the distance function $d^2 = u^2 + s \cdot (v^2 + w^2)$ with $s = 100$ such that traveling close to the streamline is preferred over deviating from it. The u, v, w values are computed by integrating the frame field along the dual path. The resulting path search is identical to the breadth-first search, except that we follow the priority dictated by the distance function. To exclude paths, which would require huge field perturbations, we again forbid dual edges which deviate by more than $\beta = 100°$ from the local u-direction. The dual path is not allowed to cross parallel feature surfaces, causing new parabolic sectors. 'The search is restricted to a feature surface if at least one arc is tangential to it. It terminates when the first transversal boundary or appropriate complementary zipper node is found. Once a valid path is available, unzipping via (DT3) can be performed, where based on the previously computed (u, v, w) coordinates we minimize spiraling effects, not relevant for local meshability but potentially influencing global meshability and the distortion of the integer-grid map.



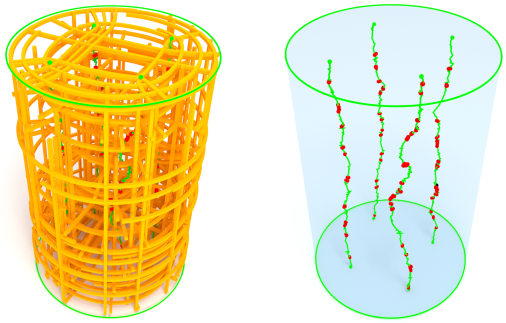


Fig. 17. Despite perfectly smooth streamlines (left), the singular arcs on a tetrahedral mesh can be highly noisy (right). The misalignment of field and singular arcs causes a large number of zipper nodes (red).

5.3 Handling Noisy Singularity Graphs

While the streamlines of frame fields on tetrahedral meshes are sufficiently smooth, this is often not true for discrete singularity graphs, as illustrated in Fig. 17. The set of mesh edges often does not admit a smooth approximation of flow-aligned arcs. Moreover, optimization of frame fields with the popular spherical harmonics formulation [Huang et al. 2011; Palmer et al. 2020] does not enforce alignment of the field to singular arcs. One remedy consists in re-optimizing the frame field with additional alignment constraints once the singularities are known, e.g. with Alg. 1. However, such an approach is highly detrimental to the field quality since the smooth streamlines are forced to align to the noisy singular edges. Consequently, we advocate for the opposite – optimizing the geometry of the singularity graph to align to the (more reliable) streamlines. We formulate an optimization problem, where vertices of the tetrahedral mesh deform in order to align singular arcs to the flow. Moreover, we include terms to continuously resolve chains of zipper nodes, achieve smooth singular arcs, prevent flipped tetrahedra, and prevent singular arcs from getting too close to each other.

Singularity Relocation. Let $x_i, p_i \in \mathbb{R}^3$ be the new and current position of the singular vertex v_i, also including all vertices incident to singular edges. The optimal relocation of $v_i \in V_S$ is posed as the

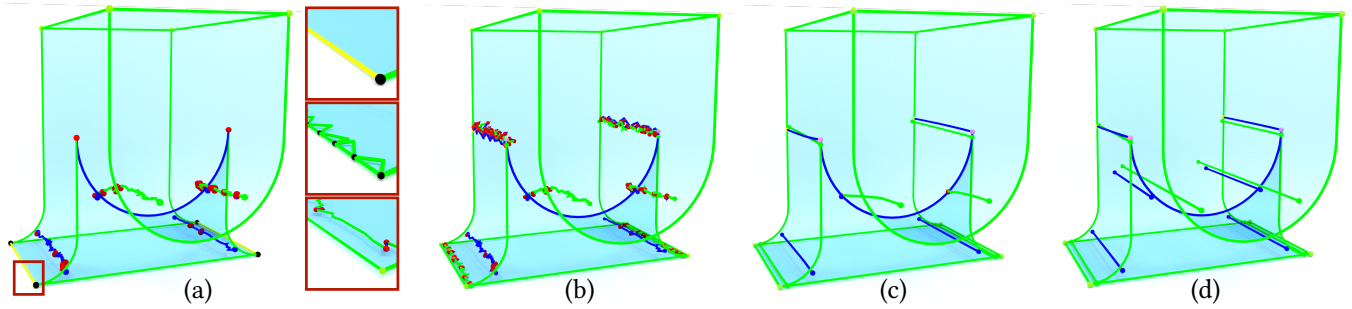


Fig. 18. Algorithm Local Meshability: (a) The input frame field contains non-meshable nodes (black and red), and non-meshable arcs (yellow). The yellow arc is repaired by first adding quad sectors at individual edges and then detaching arcs at black vertices, shown in closeups. (b) Result after one iteration of singularity repair, resolving various defects. (c) After Stage I, two isolated zipper nodes (red) remain. (d) Locally meshable result after Stage II.

following non-linear optimization problem with linear constraints for feature preservation:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & E_a + E_s + E_c + E_d + E_r \\ \text{subject to} \quad & x_i - p_i = 0, \ \forall v_i \in V_F \\ & (x_i - p_i) \times t_i = 0, \ \forall e_{ij} \in E_F \\ & (x_i - p_i) \cdot n_i = 0, \ \forall t_{ijk} \in T_F, \end{aligned}$$

where $t_i \in \mathbb{R}^3$ is a feature tangent incident to v_i and n_i the normal vector of an incident feature surface.

Alignment energy E_a – Aligning singular edges $e_{ij} \in E_S$ with the average frame field direction d_{ij} of incident tetrahedra.

$$E_a = \frac{w_a}{s} \sum_{e_{ij} \in E_M} \frac{|(x_i - x_j) \times d_{ij}|^2}{l_{e_{ij}}}$$

with the edge length $l_{e_{ij}}$, and $E_M \subset E_S$ being the subset of singular edges with meshable monodromy.

Shrink energy E_s – Contracting singular edges of non-meshable monodromy, or incident to zipper nodes both combined in $E_Z \subset E_S$.

$$E_s = \frac{w_s}{s} \sum_{e_{ij} \in E_C \cup E_Z} \frac{|x_i - x_j|^2}{l_{e_{ij}}}$$

Curvature energy E_c – Acting as smoothness regularizer by minimizing curvature of singular arcs with meshable monodromy

$$E_c = \frac{w_c}{s} \sum_{e_{ij}, e_{jk} \in E_M} |x_i + x_k - 2x_j|^2 / (l_{e_{ij}} + l_{e_{jk}})$$

Deformation energy E_d – Preventing degenerate or flipped tetrahedra and guarding the surface normal against flipping.

$$E_d = w_d \sum_{v_i \in V_S} \sum_{c_{ijkl} \in C} \log \frac{tr(J_t^T J_t)}{det(J_t)^{\frac{2}{3}}}$$

where J_t denotes the Jacobian of deforming a tetrahedron to the regular one. To prevent orientation flips of surface normals, for each boundary vertex v_i we add a fan of virtual tets by connecting all incident boundary triangles to a virtual vertex extruding v_i along its normal direction according to the local average edge length.

Repulsion energy E_r – Preventing singular arcs resulting from zipper node repair from getting geometrically too close to each other, acting on all vertices on the newly created arcs V_R

$$E_r = \frac{w_r}{s} \sum_{v_i \in V_R} E_{r_i}$$

$$E_{r_i} = \begin{cases} \frac{1}{2d_i} |x_i - \hat{p}_i|^2 & \text{if } d_i \leq l_{min} \\ 0 & \text{otherwise} \end{cases}$$

with $\hat{p}_i = \frac{p_i + p_i'}{2} + l_{min} \frac{p_i - p_i'}{|p_i - p_i'|}$ being the target point of the repulsion, computed from the point p_i' closest to v_i on another singular arc, d_i denoting its distance, and l_{min} being the minimal target edge length of singular vertices on the arc of v_i.

The default parameters are $w_a = w_s = w_c = 1$ and $w_r = 0.1, w_d = 0.01$ and $s = \sqrt[3]{|\mathcal{T}|}$ is chosen such that the overall objective is scale invariant. We use TinyAD [Schmidt et al. 2022] for automatic differentiation and the NewtonSolver of CoMISo [Bommes et al. 2012] for optimization. At the end of the singular vertex relocation, we employ Alg. 1 to smooth the frame field within a 2-ring neighborhood of singularities, aligning to features but not to singular arcs. The singularity relocation restricts modifications to the one-ring of the initial mesh and is thus often insufficient to obtain satisfactory results. However, alternating the singularity relocation with remeshing effectively unlocks the required degrees of freedom.

Mesh Improvement. We adopt an operator-based strategy similar to TetWild [Hu et al. 2018], which consists of three passes: edge splitting, edge collapsing, and edge swapping. Restricting the mesh improvement to the 2-ring neighborhood of singularities proved sufficient in our experiment. The target edge length for vertices – considered during edge collapses and splits – is chosen as the average length of all incident edges. Edge collapses are forbidden if they alter the topology of features, move a vertex away from a feature, or generate conflicting alignment constraints.

5.4 Practical Algorithm

Motivated by the large number of zipper nodes caused by noisy singularity graphs on tetrahedral meshes, our practical algorithm proceeds in two stages, illustrated in Fig. 18.

Stage I. alternates between the singularity repair of Sec. 5.2 and the singularity relocation of Sec. 5.3. However, to avoid an excessive number of singular arcs, the repair of isolated zipper nodes is suspended. In this stage, we do not enforce alignment of frames to singular arcs. Only in the singular vertex repair steps we temporarily align in order to identify sectors correctly. In Stage I we also do not repair edges of compound monodromy since these are typically resolved by the shrinkage term of the singularity re-location, getting smaller and smaller until being collapsed in the remeshing. While not important for local meshability, the collapse if often preferable for global meshability. After performing N_1 (default 12) iterations in Stage I, the algorithm proceeds with Stage II.

Stage II. also alternates between singularity repair and singularity relocation. However, this time includes the repair of zipper nodes and non-meshable edges of compound type. Since new singular arcs resulting from unzipping can be noisy, in Stage II we perform several iterations of singularity relocation in a row (default 6). Stage II terminates when all vertices and edges are locally meshable, aligning the field to all resulting arcs with Alg. 1 as a last step.

6 INTEGER-GRID MAP CONSTRUCTION

6.1 Seamless Map

Given a locally meshable frame field $\mathcal{F}$ on tetrahedral cells and corresponding matchings M on dual edges, the next step in our hexahedral meshing framework consists in generating a locally injective seamless map $f : \mathbb{R}^3 \to \mathbb{R}^3$, which is needed for integer-quantization. A seamless map can be obtained by solving a Poisson problem in the spirit of [Nieser et al. 2011], including all feature alignment and cut constraints but ignoring integer constraints. Frequently, the result is a degenerate map with inverted elements, and improved robustness can be achieved by adding a post-process that targets local injectivity [Garanzha et al. 2021]. In our experiments, we sometimes observed that [Garanzha et al. 2021] finds a locally injective map but does not obey the frame field topology. As discussed before, the matchings only constrain indices (and sector angles) up to additional integers inside the domain and half-integers on the boundary, allowing e.g. a singular arc of index $-\frac{1}{4}$ to change into one of index $\frac{3}{4}$, cf. Fig.13 of [Garanzha et al. 2022].

Integrable Frame Field Optimization. To prevent such failure cases, we propose a novel seamless map optimization based on integrable frame fields and a smoothness regularizer. The main difference compared to the approach above is that we do not allow any inverted element throughout the optimization and effectively prevent index changes that are typically caused by untangling inverted elements of the initial map. Hence, instead of starting the optimization from a conforming but potentially degenerate map, we do the opposite and start from a locally injective but non-conforming map. Dropping conformity is equivalent to giving up the integrability of the frame field. Hence, initialization with any target frame field satisfying $\det F > 0$ is possible. More precisely, for each tetrahedron t_i, we optimize a Jacobian matrix $J_i \in \mathbb{R}^{3\times3}$. Observing that we are targeting a map, which sends the tangent vectors of a frame onto coordinate axes, i.e., $J_i F_i = I$, the per-element initialization is simply $J_i = F_i^{-1}$. This initial map is locally injective since $\det F_i > 0$ and $\det J_i = 1/\det F_i$. Then we optimize a deformation objective with barrier behavior, tending to ∞ for degenerating elements. In all our experiment we use the symmetric Dirichlet energy $E_{SD} = \int_\Omega S_D(J)dV$ with

$$S_D(J) = \begin{cases} ||J||_2^2 + ||J^{-1}||_2^2 & \det J > 0 \\ \infty & \det J \leq 0 \end{cases} \tag{5}$$

Integrability of a piecewise linear map requires that the (matched) gradients of two neighboring tetrahedra c_i and c_j are identical when projected onto their common triangle t_{ij}, leading to linear constraints

$$P_{ij}(J_i^T - M_{ij}^{-T} J_j^T) = \bar{0} \tag{6}$$

with $P_{ij} \in \mathbb{R}^{2\times3}$ projecting gradients onto a basis of the plane of t_{ij}, matchings $M_{ij} \in O$, and the matrix of zeros $\bar{0} \in \mathbb{R}^{2\times3}$. Optimizing E_{SD} subject to linear integrability constraints and feature alignment constraints, often results in a locally injective seamless map. However, there is neither a guarantee on obtaining a locally injective map, nor on correctly re-producing the frame field topology. Due to the non-convex objective function, the optimization might converge to an infeasible point, never reaching an integrable frame field. Index changes, as discussed above, can also happen in this formulation. To further improve correct re-production of the frame field topology, we add (i) a smoothness regularizer $E_S = \int_\Omega ||\nabla J||_2^2 dV$, and (ii) ensure in the line search of the optimizer that we never pass through an inverted state – which is necessary to change the index assuming conformity. A candidate step $x + t\Delta x$ is truncated to a step length t such that $J + s\Delta J > 0$ for $s \in [0, t]$ is valid for all tetrahedra.

Implementation Details. The integrable frame field is optimized with a Truncated Newton Method [Wright et al. 1999] using a projected PCG linear solver to enable scalability to large tetrahedral meshes and accurate constraint satisfaction. All constraints feasible at the start are handled by the projected PCG, while integrability constraints, which are infeasible at the start, are included with quadratic penalty terms E_I, leading to the overall objective $E = E_{SD} + w_S E_S + w_I E_I$. As a final heuristic, which proved valuable in our experiments, we optimize a sequence of optimization problems, where we successively increase w_I and decrease w_S. All derivatives are computed algorithmically with [Schmidt et al. 2022], using Eigen [Guennebaud et al. 2010].

6.2 Integer-Grid Map

Given a valid seamless map, we first obtain a valid quantization with the robust algorithm of [Brückler et al. 2022a] using the motorcycle complex. The quantization constraints are then added to the optimization of the seamless map to compute a valid integer-grid map. Since the seamless map of the previous step serves as a locally injective initialization, we can immediately optimize a conforming piecewise linear map with the barrier energy E_{SD}, expressed with vertex coordinates. The linearly constrained non-convex problem is optimized with IPOPT [Wächter and Biegler 2006]. In our experiments, the integer-grid map construction succeeded in all cases, where a valid seamless map was obtained. However, for extremely coarse quantizations, which require large geometric distortions, failures can be observed since the tetrahedral mesh might not even have sufficient degrees of freedom to admit a locally injective map subject to the quantization constraints.

7 EVALUATION

7.1 HexMe Dataset

We evaluate the robustness of our hexahedral meshing framework by means of the HexMe dataset [Beaufort et al. 2022], which contains 189 tetrahedral meshes with feature tags inherited from CAD models, and is specifically designed for consistent and practically meaningful evaluation of hex meshing algorithms. We compare our algorithm with two state-of-the-art frame field based methods [Li et al. 2012] and [Jiang et al. 2014], both performing singularity repair targeting edge meshability. For each combination of model and algorithm, we analyze four aspects, i.e. (i) edge meshability, (ii) vertex meshability, (iii) validity of seamless map, and (iv) validity of integer-grid map, as depicted in Fig. 19 and described in more detail below. Our statistics are binary in the sense that all edges of a model need to be meshable to qualify as an edge-meshability success. Binary success/failure counting is justified by the fact that a single non-meshable local neighborhood will immediately prevent the generation of an integer-grid map. Please note that our evaluation criteria are sorted in a way that the success rates can only decrease since prior ones are a necessary condition for latter ones.

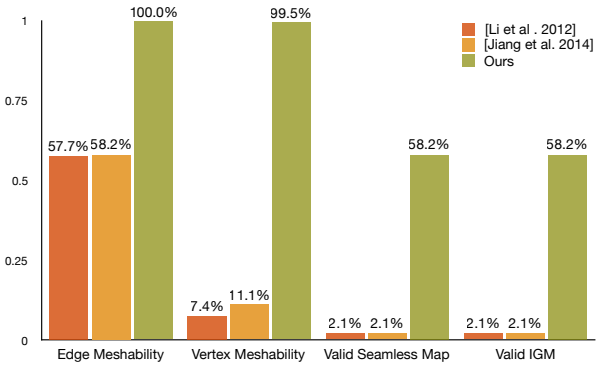


Fig. 19. Comparison with [Li et al. 2012] and [Jiang et al. 2014] on the HexMe dataset. The success rate w.r.t. edge meshability, vertex meshability, valid seamless map and valid IGM on a per model basis is depicted.

Initialization. For fairness of the comparison, all three methods are initialized with identical per-vertex octahedral fields, generated by [Ray et al. 2016], with field alignment constraints for all feature edges/faces. Singular tetrahedra are split before re-sampling the field on tetrahedra by interpolating and projecting SPH coefficients.

Local Edge Meshability. Local meshability of edge one-rings is verified according to the index and sector computations of Sec. 5.2, requiring $I < 1$ and that all sectors are of quad type. Our algorithm ensures edge meshability for all 189 HexMe models, while [Li et al. 2012] and [Jiang et al. 2014] succeed for 109 and 110 models respectively. While repairing all compound singular arcs, polar sectors enforced by feature constraints are not handled.

Local Vertex Meshability. Local meshability of a vertex is verified by explicitly constructing a seamless map with the technique of Sec. 6.1 but restricted to the one-ring neighborhood of tetrahedra.

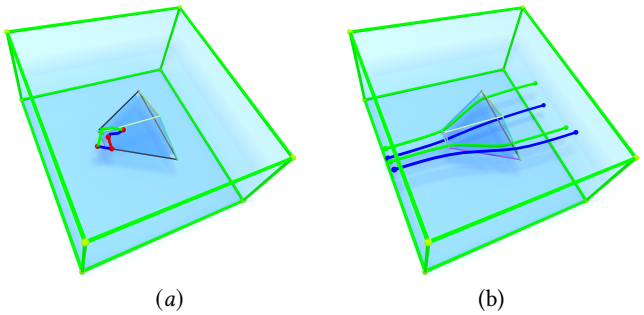


Fig. 20. (a) Singularity graph of `n04b_transition_prism` after correction with [Li et al. 2012] and [Jiang et al. 2014]. (b) Our singularity graph. Red and black edges are not locally meshable. Green and blue edges represent singular edges of valence -1 and +1. Other colored edges are features and zipper nodes are colored in red.

While [Li et al. 2012] and [Jiang et al. 2014] obtain 14 and 21 successful results, respectively, 188 out of 189 models pass the vertex meshability test for our algorithm. The large improvement is not surprising since none of the former methods explicitly repairs non-meshable vertices. Only `i28b_gc_tire_1218` did not succeed since the huge number of zipper nodes caused by the highly detailed profile of the tire caused a runtime beyond our time limit of 48h. Consequently, the sole failure case is not related to the theoretical gap of potentially being unable to resolve a zipper node.

Global Meshability. Necessary and sufficient conditions for global meshability are unknown. However, global meshability can be verified by the generation of a locally injective seamless map and requiring that the frame field topology and feature alignment are preserved. Both state-of-the-art methods generate seamless maps with the non-robust method of [Nieser et al. 2011], further decreasing the success rate to only 4 valid seamless maps. Our robust seamless map generation of Sec. 6.1 verifies that 110 out of 188 locally meshable fields are actually globally meshable. Fig.20(a)/(b) shows the singularity graphs of `n04b_transition_prism` after the correction with our method in comparison to the repair of prior methods, not resolving zipper configurations. Fig.21 shows `i29u_bracket`, where our field is locally meshable but not globally meshable. Our integer-grid map construction succeeds for all models with valid seamless map, enabling topologically valid hexahedral meshes, preserving all features of the input. Fig. 24 shows a subset of the output hex meshes, while all 110 are available as supplemental material. We improve the geometric quality of hexahedral meshes using Mesquite [Brewer et al. 2003]. Statistics of the illustrated examples including input mesh complexity, minimum/average scaled Jacobian, and runtime of different parts of the algorithm can be found in Table 1. Timings have been measured for a single thread on an AMD EPYC 7742 CPU. Our local meshability ensuring algorithm is time-efficient. It mostly involves local repair operations, and the continuous singularity optimization step including remeshing is restricted to the 2−ring neighborhood of the singularity graph. However, downstream steps of the hex meshing framework, e.g. the generation of seamless and integer-gird maps are significantly more expensive. The time costs to ensure local meshability ranges from 10 to 451 seconds, while

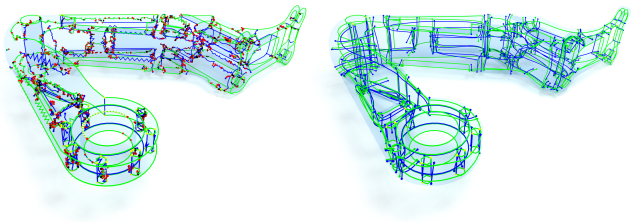


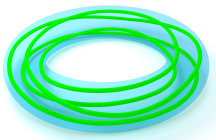
Fig. 21. Our algorithm is able to ensure local meshability for complex objects like `i29u_bracket`. However, a hexahedral mesh could not be generated due to remaining global meshability defects.

the seamless map generation takes up to 3373s, and the IGM generation up to 41475s. The dominating time cost of map construction is sometimes caused by sub-optimally positioned singularities that cause large distortions in the map.

Table 1. Statistics and Timings.

Mesh	#Tet	Scaled Jacobian	LM-Repair (s)	S-Map (s)	IGM (s)
i06u_m6	189547	0.218/0.977	73	1992	5716
i12u_s5	107154	0.038/0.963	52	974	1085
i14b_s7	70968	0.133/0.954	327	1440	4097
i15b_s8	132749	0.268/0.973	278	779	5785
i18b_s22	295584	0.132/0.969	451	3373	41475
i25c_s40	24305	0.126/0.958	16	195	434
n08c_pentapyr	6144	0.269/0.926	31	166	101
n09c_pyramid	7015	0.264/0.948	10	39	30
n10u_qtorus_cyl	67154	0.307/0.974	279	829	536
s04b_tetrahedron	36329	0.123/0.973	23	93	1059
s08c_cross_cyls_dr	19747	0.231/0.982	36	91	84

HexMe Categories. We further analyze global meshability of our algorithm w.r.t. different categories of the HexMe dataset, depicted in Fig. 22. While global meshability can be obtained for many models from the *simple* and *nasty* categories, only for roughly one third of the *industrial* models valid IGMs can be constructed. When excluding *box-embedded* models with interior feature constraints, the success rate increases significantly, revealing that those are the most challenging in terms of global meshability. The success rates of *curvature-adapted* and *uniform* tetrahedral meshes are identical, verifying that our algorithm does not significantly depend on the input tetrahedral mesh. This is not surprising since remeshing is performed whenever necessary to enable local singularity repair operations, or to improve the geometry of singular arcs. However, there are still sometimes differences caused by the initial tetrahedral mesh. The initial frame fields might be different, e.g. being non-globally-meshable for `s16u_torus` shown on the right, while the corresponding curvature-adapted tet mesh `s16c_torus` offers an initial field without twisted singular curves, enabling a globally meshable field.



Parameters. For benchmarking local meshability, we use default parameters for all experiments. However, for benchmarking global meshability, parameter tuning enables 20 additional success cases. A major factor is the dual path search to unzip isolated zipper nodes. While the concrete path is irrelevant for local meshability it might

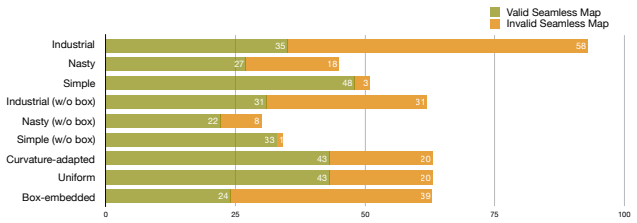


Fig. 22. Number of valid vs.invalid seamless maps obtained in each category.

have an impact on global meshability. Sometimes, altering the number of iterations of Stage I, and/or changing the field alignment weight enables a globally meshable field. During quantization we target $\#hexa = 100k$ for all models except for `i17u_s20`, which required $\#hexa = 600k$.

7.2 Additional Experiments

Stress Test. We tested robustness of our pipeline on the tetrahedral mesh of a sphere with a random frame field, generated by sampling random unit quaternions. Fig. 23a depicts the singularity graphs and streamlines of the input random frame field, while Fig. 23(b) shows the locally meshable result of our algorithm, successfully resolving all defects.

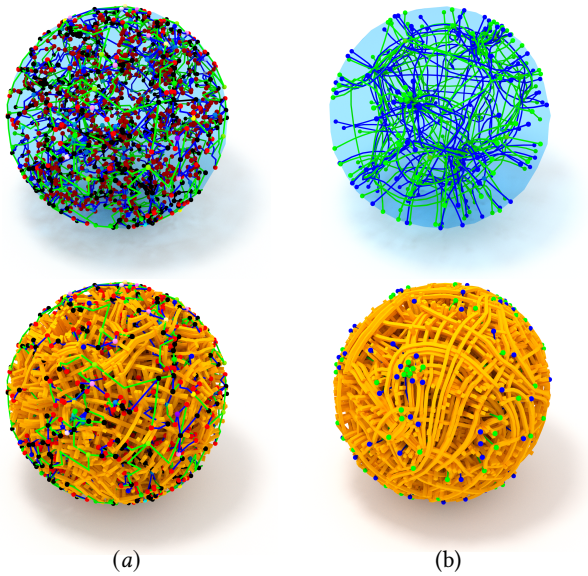


Fig. 23. Stress test. (a) A random input field and its singularity graph. Black edges are compound singular edges and black vertices are invalid singular nodes, while red vertices are zipper nodes. (b) The locally meshable output field and its singularity graph.

Coarse Block Decomposition. Frame field based methods offer singularities suitable for coarse block decompositions. Our algorithm minimizes the number of additionally generated singular arcs, and combined robust map construction and robust quantization of [Brückler et al. 2022b] is thus capable of generating coarse block decompositions, as demonstrated in Fig. 25.

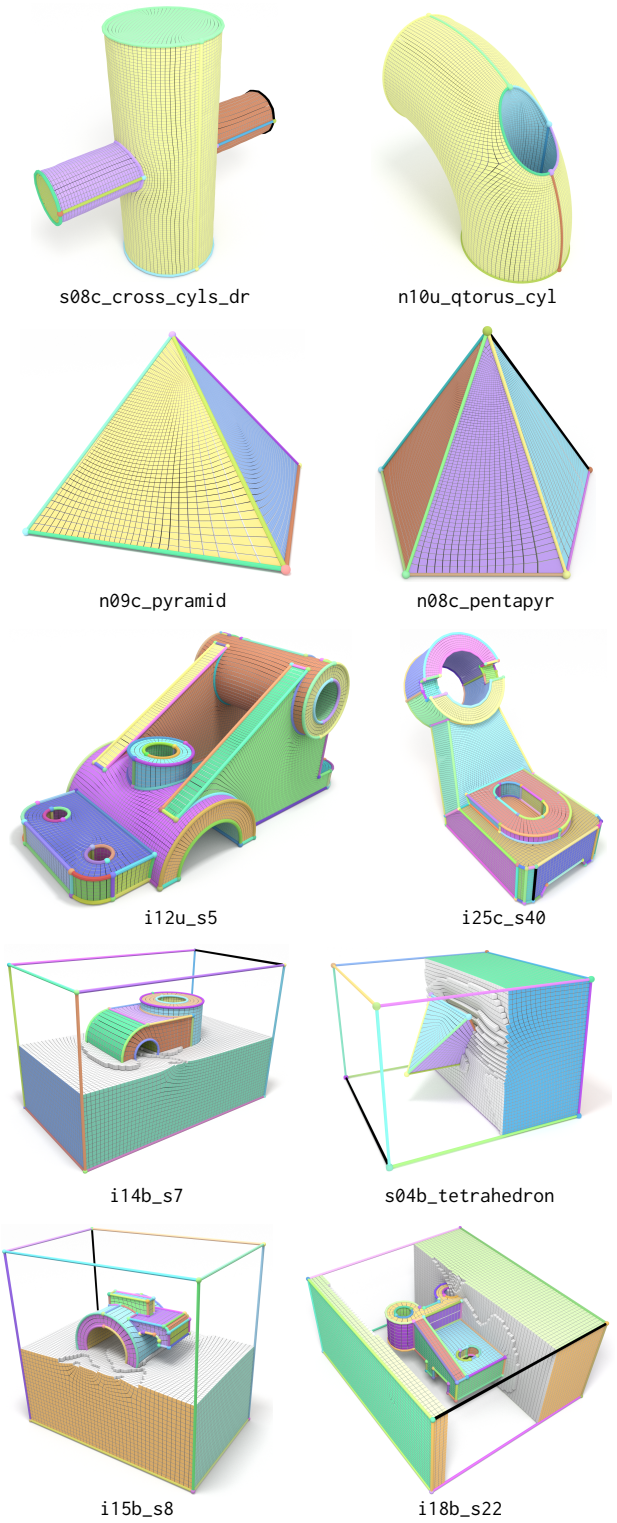


Fig. 24. Several hex meshes automatically generated by our method. Feature nodes/arcs/surfaces are colored and preserved as in the input tet meshes.

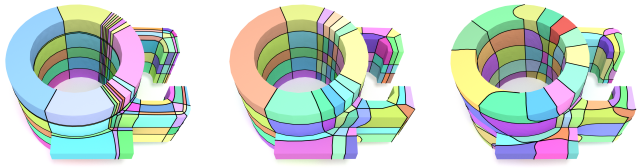


Fig. 25. Base complex with number of blocks $|B|$ of meshes obtained through quantization with different target complexity $|C|$. A good compromise is offered by $|C| = 1k$, while the finer one has excessively many blocks and the coarser one is severely distorted.

8 CONCLUSION & FUTURE WORK

We have presented the theory of locally meshable frame fields and a novel algorithm to convert a given frame field into a locally meshable one. While it significantly improves the robustness of frame field based hexahedral mesh generation algorithms, additional effort will be required to obtain full robustness.

Local Meshability. We plan to establish strict termination guarantees for the resolution of zipper nodes, and include anti-quad configurations into the theory and the algorithm.

Global Meshability. Since local meshability is necessary but not sufficient for global meshability, the obvious next step will be to target global meshability. A direct generalization of [Myles et al. 2014] is impossible due to the non-integrability of surfaces of 3D frame fields. However, attempting to generate the motorcycle complex [Brückler et al. 2022b] directly from a frame field is nevertheless a promising research direction. Such a method would inevitably require additional repair mechanisms since in contrast to the 2D setting, singularity and feature constraints can prevent existence of a valid motorcycle complex, even for locally meshable frame field.

Locally Injective Maps. For a given globally meshable field, none of the existing techniques can guarantee to find a seamless map of identical topology. Consequently, future work is required to obtain stronger guarantees, also regarding preservation of frame field topology.

ACKNOWLEDGMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (AlgoHex, grant agreement No 853343).

REFERENCES

Cecil G Armstrong, Harold J Fogg, Christopher M Tierney, and Trevor T Robinson. 2015. Common themes in multi-block structured quad/hex mesh generation. *Procedia Engineering* 124 (2015), 70–82.

Daniel Asimov. 1993. *Notes on the topology of vector fields and flows.* Technical Report. Technical report, NASA Ames Research Center, 1993. RNR-93-003.

Pierre-Alexandre Beaufort, Maxence Reberol, Heng Liu, Franck Ledoux, and David Bommes. 2022. Hex Me If You Can. *Computer Graphics Forum* 41, 5 (2022), 125–134.

David Bommes, Marcel Campen, Hans-Christian Ebke, Pierre Alliez, and Leif Kobbelt. 2013a. Integer-grid maps for reliable quad meshing. *ACM Trans. Graph.* 32, 4 (2013), 1–12.

David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. 2013b. Quad-mesh generation and processing: A survey. In *Computer graphics forum*, Vol. 32. Wiley Online Library, 51–76.

David Bommes, Henrik Zimmer, and Leif Kobbelt. 2009. Mixed-integer quadrangulation. *ACM Trans. Graph.* 28, 3 (2009), 1–10.

David Bommes, Henrik Zimmer, and Leif Kobbelt. 2012. Practical mixed-integer optimization for geometry processing. In *Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7*. Springer, 193–206.

Michael L Brewer, Lori Freitag Diachin, Patrick M Knupp, Thomas Leurent, and Darryl J Melander. 2003. The Mesquite Mesh Quality Improvement Toolkit. In *IMR*.

Hendrik Brückler, David Bommes, and Marcel Campen. 2022a. Volume parametrization quantization for hexahedral meshing. *ACM Trans. Graph.* 41, 4 (2022), 1–19.

Hendrik Brückler, Ojaswi Gupta, Manish Mandad, and Marcel Campen. 2022b. The 3D Motorcycle Complex for Structured Volume Decomposition. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 221–235.

Etienne Corman and Keenan Crane. 2019. Symmetric moving frames. *ACM Trans. Graph.* 38, 4 (2019), 1–16.

J Austin Cottrell, Alessandro Reali, Yuri Bazilevs, and Thomas JR Hughes. 2006. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering* 195, 41-43 (2006), 5257–5296.

Xingyi Du, Noam Aigerman, Qingnan Zhou, Shahar Z Kovalsky, Yajie Yan, Danny M Kaufman, and Tao Ju. 2020. Lifting simplices to find injectivity. *ACM Trans. Graph.* 39, 4 (2020).

Xianzhong Fang, Weiwei Xu, Hujun Bao, and Jin Huang. 2016. All-hex meshing using closed-form induced polycube. *ACM Trans. Graph.* 35, 4 (2016), 1–9.

Theodore Frankel. 2011. *The Geometry of Physics: An Introduction* (3 ed.). Cambridge University Press.

Xifeng Gao, Zhigang Deng, and Guoning Chen. 2015. Hexahedral mesh re-parameterization from aligned base-complex. *ACM Trans. Graph.* 34, 4 (2015), 1–10.

Xifeng Gao, Wenzel Jakob, Marco Tarini, and Daniele Panozzo. 2017a. Robust Hex-Dominant Mesh Generation Using Field-Guided Polyhedral Agglomeration. *ACM Trans. Graph.* 36, 4, Article 114 (jul 2017).

Xifeng Gao, Daniele Panozzo, Wenping Wang, Zhigang Deng, and Guoning Chen. 2017b. Robust structure simplification for hex re-meshing. *ACM Trans. Graph.* 36, 6 (2017), 1–13.

Xifeng Gao, Hanxiao Shen, and Daniele Panozzo. 2019. Feature Preserving Octree-Based Hexahedral Meshing. In *Computer graphics forum*, Vol. 38. Wiley Online Library, 135–149.

Vladimir Garanzha, Igor Kaporin, Liudmila Kudryavtseva, François Protais, David Desobry, and Dmitry Sokolov. 2022. Practical lowest distortion mapping. *arXiv preprint arXiv:2201.12112* (2022).

Vladimir Garanzha, Igor Kaporin, Liudmila Kudryavtseva, François Protais, Nicolas Ray, and Dmitry Sokolov. 2021. Foldover-free maps in 50 lines of code. *ACM Trans. Graph.* 40, 4 (2021), 1–16.

Gaël Guennebaud, Benoît Jacob, et al. 2010. Eigen v3. http://eigen.tuxfamily.org.

Tobias Günther and Irene Baeza Rojo. 2021. Introduction to vector field topology. In *Topological Methods in Data Analysis and Visualization VI*. Springer, 289–326.

Hao-Xiang Guo, Xiaohan Liu, Dong-Ming Yan, and Yang Liu. 2020. Cut-enhanced PolyCube-maps for feature-aware all-hex meshing. *ACM Trans. Graph.* 39, 4 (2020).

Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2018. Tetrahedral meshing in the wild. *ACM Trans. Graph.* 37, 4 (2018).

Jin Huang, Tengfei Jiang, Zeyun Shi, Yiying Tong, Hujun Bao, and Mathieu Desbrun. 2014. $l1$-based construction of polycube maps from complex shapes. *ACM Trans. Graph.* 33, 3 (2014), 1–11.

Jin Huang, Yiying Tong, Hongyu Wei, and Hujun Bao. 2011. Boundary aligned smooth 3D cross-frame field. *ACM Trans. Graph.* 30, 6 (2011), 1–8.

Tengfei Jiang, Jin Huang, Yuanzhen Wang, Yiying Tong, and Hujun Bao. 2014. Frame Field Singularity Correction for Automatic Hexahedralization. *IEEE Transactions on Visualization and Computer Graphics* 20, 8 (2014), 1189–1199.

David A. Kopriva. 2009. *Implementing spectral methods for partial differential equations: Algorithms for scientists and engineers*. Springer Science & Business Media.

Felix Kälberer, Matthias Nieser, and Konrad Polthier. 2007. QuadCover - Surface Parameterization using Branched Coverings. *Computer Graphics Forum* 26, 3 (2007), 375–384.

Lingxiao Li, Paul Zhang, Dmitriy Smirnov, S Mazdak Abulnaga, and Justin Solomon. 2021. Interactive all-hex meshing via cuboid decomposition. *ACM Trans. Graph.* 40, 6 (2021), 1–17.

Yufei Li, Yang Liu, Weiwei Xu, Wenping Wang, and Baining Guo. 2012. All-hex meshing using singularity-restricted field. *ACM Trans. Graph.* 31, 6 (2012), 1–11.

Heng Liu, Paul Zhang, Edward Chien, Justin Solomon, and David Bommes. 2018. Singularity-constrained octahedral fields for hexahedral meshing. *ACM Trans. Graph.* 37, 4 (2018).

Marco Livesu, Nico Pietroni, Enrico Puppo, Alla Sheffer, and Paolo Cignoni. 2020. Loopycuts: Practical feature-preserving block decomposition for strongly hex-dominant meshing. *ACM Trans. Graph.* 39, 4 (2020).

Marco Livesu, Luca Pitzalis, and Gianmarco Cherchi. 2021. Optimal dual schemes for adaptive grid based hexmeshing. *ACM Trans. Graph.* 41, 2 (2021), 1–14.

Marco Livesu, Alla Sheffer, Nicholas Vining, and Marco Tarini. 2015. Practical hex-mesh optimization via edge-cone rectification. *ACM Trans. Graph.* 34, 4 (2015), 1–11.

Max Lyon, David Bommes, and Leif Kobbelt. 2016. HexEx: Robust hexahedral mesh extraction. *ACM Trans. Graph.* 35, 4 (2016), 1–11.

Manish Mandad, Ruizhi Chen, David Bommes, and Marcel Campen. 2022. Intrinsic mixed-integer polycubes for hexahedral meshing. *Computer aided geometric design* 94 (2022).

Loïc Maréchal. 2009. Advances in octree-based all-hexahedral mesh generation: handling sharp features. In *Proceedings of the 18th international meshing roundtable*. Springer, 65–84.

Zoë Marschner, David Palmer, Paul Zhang, and Justin Solomon. 2020. Hexahedral Mesh Repair via Sum-of-Squares Relaxation. In *Computer Graphics Forum*, Vol. 39. Wiley Online Library, 133–147.

Ashish Myles, Nico Pietroni, and Denis Zorin. 2014. Robust Field-Aligned Global Parametrization. *ACM Trans. Graph.* 33, 4, Article 135 (jul 2014).

Matthias Nieser, Ulrich Reitebuch, and Konrad Polthier. 2011. Cubecover–parameterization of 3d volumes. In *Computer graphics forum*, Vol. 30. Wiley Online Library, 1397–1406.

David Palmer, David Bommes, and Justin Solomon. 2020. Algebraic representations for volumetric frame fields. *ACM Trans. Graph.* 39, 2 (2020), 1–17.

Nico Pietroni, Marcel Campen, Alla Sheffer, Gianmarco Cherchi, David Bommes, Xifeng Gao, Riccardo Scateni, Franck Ledoux, Jean Remacle, and Marco Livesu. 2022. Hex-mesh generation and processing: a survey. *ACM Trans. Graph.* 42, 2 (2022), 1–44.

Luca Pitzalis, Marco Livesu, Gianmarco Cherchi, Enrico Gobbetti, and Riccardo Scateni. 2021. Generalized adaptive refinement for grid-based hexahedral meshing. *ACM Trans. Graph.* 40, 6 (2021), 1–13.

Michael Rabinovich, Roi Poranne, Daniele Panozzo, and Olga Sorkine-Hornung. 2017. Scalable Locally Injective Mappings. *ACM Trans. Graph.* 36, 2 (2017).

Nicolas Ray, Dmitry Sokolov, and Bruno Lévy. 2016. Practical 3D frame field generation. *ACM Trans. Graph.* 35, 6 (2016), 1–9.

Maxence Reberol, Alexandre Chemin, and Jean-François Remacle. 2019. Multiple approaches to frame field correction for CAD models. *arXiv preprint arXiv:1912.01248* (2019).

Patrick Schmidt, Janis Born, David Bommes, Marcel Campen, and Leif Kobbelt. 2022. TinyAD: Automatic Differentiation in Geometry Processing Made Simple. In *Computer graphics forum*, Vol. 41. Wiley Online Library, 113–124.

Teseo Schneider, Yixin Hu, Xifeng Gao, Jeremie Dumas, Denis Zorin, and Daniele Panozzo. 2022. A large scale comparison of tetrahedral and hexahedral elements for finite element analysis. *ACM Trans. Graph* (2022).

Hanxiao Shen, Leyi Zhu, Ryan Capouellez, Daniele Panozzo, Marcel Campen, and Denis Zorin. 2022. Which Cross Fields Can Be Quadrangulated? Global Parameterization from Prescribed Holonomy Signatures. *ACM Trans. Graph.* 41, 4 (2022).

Dmitry Sokolov and Nicolas Ray. 2015. *Fixing normal constraints for generation of polycubes*. Technical Report.

Justin Solomon, Amir Vaxman, and David Bommes. 2017. Boundary element octahedral fields in volumes. *ACM Trans. Graph.* 36, 4 (2017).

Marco Tarini, Kai Hormann, Paolo Cignoni, and Claudio Montani. 2004. Polycube-maps. *ACM Trans. Graph.* 23, 3 (2004), 853–860.

Ryan Viertel and Braxton Osting. 2019. An Approach to Quad Meshing Based on Harmonic Cross-Valued Maps and the Ginzburg–Landau Theory. *SIAM Journal on Scientific Computing* 41, 1 (2019), A452–A479.

Ryan Viertel, Matthew L Staten, and Franck Ledoux. 2016. *Analysis of Non-Meshable Automatically Generated Frame Fields*. Technical Report. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States).

Andreas Wächter and Lorenz T Biegler. 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming* 106, 1 (2006), 25–57.

David White, Lai Mingwu, Steven Benzley, and Gregory Sjaardema. 1996. Automated Hexahedral Mesh Generation by Virtual Decomposition. *Proceedings of the 4th International Meshing Roundtable* (1996).

Stephen Wright, Jorge Nocedal, et al. 1999. Numerical optimization. *Springer Science* 35, 67-68 (1999).

Kaoji Xu, Xifeng Gao, and Guoning Chen. 2018. Hexahedral mesh quality improvement via edge-angle optimization. *Computers & Graphics* 70 (2018), 17–27.

Wuyi Yu, Kang Zhang, and Xin Li. 2015. Recent algorithms on automatic hexahedral mesh generation. In *2015 10th International Conference on Computer Science & Education (ICCSE)*. IEEE, 697–702.

Paul Zhang, Xinyi Cynthia Fan, and Klara Mundilova. 2023. Local Decomposition of Hexahedral Singular Nodes into Singular Curves. *Computer-Aided Design* 158 (2023).