Structure-Informed Hex-dominant Mesh Simplification

Lei Si, Qixin Deng, Aobo Jin, Guoning Chen *Senior Member, IEEE*

***Abstract*—Hex-dominant mesh generation has recently received increasing attention from researchers and the simulation community due to its robustness compared to pure hex-mesh generation techniques. In this work, we present a first structure-informed simplification framework, aiming to reduce the number of non-hex cells from the hex-dominant meshes. Our framework extracts individual sub-structures via parallel relations from the input hex-dominant mesh, decomposes self-tangent and self-intersecting sub-structures for structure complexity control, and collapses certain sets of edges that are adjacent to particular non-hex cells to remove them. We design a filtering and ranking strategy to select edges and sub-structures for collapsing. To better understand the complexity of sub-structure configurations, we introduced a novel relation graph that captures the connections between edges and between edges and sub-structures. Additionally, we designed a smoothing algorithm for hex-dominant meshes that enhances hex cell quality, even in meshes containing various cell types. We evaluate the effectiveness of our framework by applying it to many hex-dominant meshes produced by three state-of-the-art hex-dominant meshing techniques. Our results achieve various levels of reduction in the number of non-hex cells of the input meshes, affirming the applicability of our framework for improving the hex-dominant meshes.**

***Index Terms*—Hex-dominant meshes, structure-informed, simplification**

I. Introduction

Volumetric meshes are employed in many scientific and engineering applications. Due to a number of desired numerical properties, hexahedral (hex-) meshes are preferred by many of these applications. Despite numerous efforts in the past three decades, automated generation of high-quality and feature-aligned pure hex-meshes for arbitrary 3D models remains an unsolved problem [1]. An alternative is to generate hexahedral (hex-) dominant meshes for those complex models, which has gained increasing attention in the past decade [2]–[5]. Many FEM and CFD tools, such as Ansys [6], MEFM [7], accept hex-dominant meshes with limited non-hex element types (e.g., tetrahedra, prisms, and wedges) as input [5]. While not as ideal as pure hex meshes, some simulations for architecture, mechanics, and fluid dynamics prefer a hex-dominant mesh that can have fewer elements with less accuracy loss, when compared with tetrahedral meshes [8]–[12].

Therefore, reducing non-hex elements (or cells) and allowing more volume to be occupied by hex elements is desirable, as fewer elements are needed in pure-hex regions. Furthermore, removing some non-hex cells may improve the mesh structure quality in certain regions (Figure 1 (d)(e)).

One way to achieve a higher percentage of hex elements in the hex-dominant meshes is to procedurally remove the non-hex cells. However, such local post-processing is challenging because (1) the non-hex elements can be arbitrarily complex (e.g., the non-hex cells in the output meshes from work by Gao et al. [4] can have up to 30 faces), not all of them can be easily removed; and (2) the removal of a non-hex cell locally may lead to a sub-optimal or even non-manifold configuration. As shown in Figure 1a, the example hex-dominant mesh contains a group of triangle prisms. Collapsing one cell will result in a hex-dominant mesh with more complex non-hex cells (top row of Figure 1, left example). That being said, removing non-hex cells with arbitrary configurations is as challenging as removing T-junctions from a T-mesh [13]. Considering a global strategy for removing these non-hex cells may lead to a better result. However, few studies have been done on this.

Similar to the base complex of pure hex meshes [14], [15], which decomposes a mesh into blocks with consistent topological configurations, we propose a sub-structure extraction approach for hex-dominant meshes. The approach extracts two types of sub-structures: *sheets* (adapted from pure hex sheets in base complex [14]–[16]) and *edge groups* (defined by sets of parallel edges found only in non-hex cells). Our observations show that collapsing sub-structures adjacent to certain non-hex cells leads to improved configurations (as demonstrated in the bottom row of Figure 1). However, a single non-hex cell can be adjacent to multiple sub-structures with different orientations. Thus, arbitrary selection of a sub-structure to collapse is insufficient for eliminating the target non-hex cell.

To address this challenge, we conduct a comprehensive analysis of non-hex cell configurations. Specifically, we introduce a novel neighborhood relation graph that systematically captures both edge-to-edge connections and edge-to-sub-structure relationships. This graph representation effectively reveals complex topological configurations that are typically hidden within volumetric meshes. Based on this graph representation, we develop a ranking strategy to select sub-structures for collapsing to produce optimal results, considering both non-hex reduction and quality constraints. To help maintain mesh quality, we adapt Laplacian smoothing by expanding one-ring neighbors to one-ring cell regions in hex-dominant meshes.

We evaluated our framework on diverse hex-dominant meshes generated by three state-of-the-art techniques. Our method consistently reduces non-hex cells, achieving more than 50% reduction for meshes with simple sheet configurations and an average reduction of 25% across all tested models.

Lei Si and Guoning Chen are with the University of Houston. Qixin Deng is with the Wabash College. Aobo Jin is with the University of Houston-Victoria. E-mail: lsi@uh.edu, gchen22@central.uh.edu, dengq@wabash.edu, jina@uhv.edu.

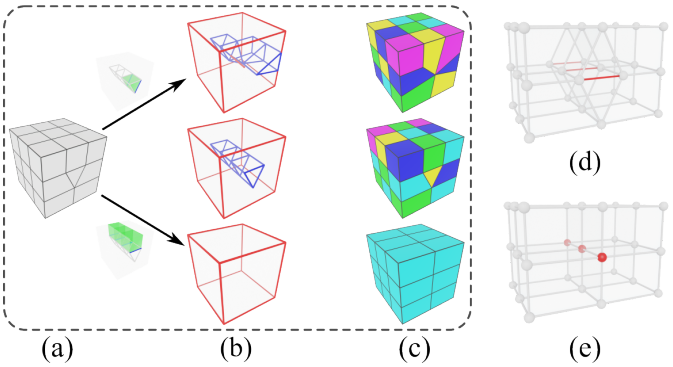


Fig. 1: Given a hex-dominant mesh (a), collapsing a triangle prism (in green) results in a mesh shown in the top row, while collapsing a group of prisms and a group of hexes (in green) yields a pure hex-mesh shown in the bottom row. The second column shows the hybrid singularity graphs [17] of the three meshes. The red lines are the conventional singularities, while the blue lines are the pseudo singular edges. The right column shows the hybrid base complexes of the meshes. Different colored blocks correspond to different base complex components [17]. (d) A parallel edge group with all red edges exclusively adjacent to non-hex cells. (e) Collapsing this edge group produces an ideal mesh configuration.

In summary, we introduce a first structure-informed simplification framework to procedurally remove the non-hex cells from hex-dominant meshes with the following contributions:

- We define sub-structures in hex-dominant meshes and propose a sub-structure decomposition strategy to handle self-tangent and self-intersecting configurations.
- We introduce a novel neighborhood relation graph to encode edge-to-edge and edge-to-sub-structure relationships for selecting edges and sub-structures to collapse.
- We incorporate advanced smoothing techniques for hex-dominant meshes into our framework to optimize mesh quality throughout the simplification process.
- We present a comprehensive simplification framework that procedurally eliminates non-hex cells through systematic identification and strategic ranking of candidate sub-structures for collapse.

II. Related Works

A. Pure Hex and Hex-dominant Meshing

Hex-dominant meshing is an alternative to all- (or pure) hex mesh generation. There are numerous efforts to generate all-hex meshes [1], including the sweeping techniques [18]–[20], grid or octree-based methods [21]–[23], polycubes mapping techniques [24]–[28], the frame (or octahedral) field based methods [29]–[33], block decomposition techniques [34], [35], and the recent motorcycle complex based methods [13], [36]. However, due to the challenges posed by the arbitrary geometry and topology configurations of different 3D models [37], robustly and automatically producing high-quality and feature-aligned all-hex meshes remains an unsolved problem [1], [38].

Alternatively, hex-dominant meshing does not require all-hex elements when tiling the volume. Instead, it tiles as many hex elements as possible while accepting a small number of non-hex elements [1]. Two groups of techniques have been introduced to achieve this goal. The first group starts with tetrahedral meshes and procedurally aggregates the adjacent tetrahedra to form hexahedra. The technique introduced by Meshkat and Talmor [39], the H-morph proposed by Owen [40], and a recent vertex-based strategy introduced by Pellerin et al. [41] belong to this group.

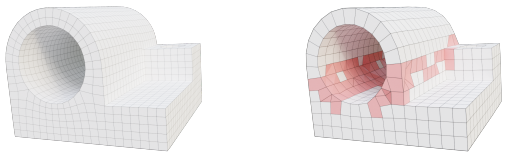
The second group of methods relies on certain 3D guidance fields (e.g., octahedral fields) to guide the aggregation of tetrahedra or the computation of a 3D volumetric parameterization for the generation of hex-dominant meshes. For example, Huang et al. [29] introduced the boundary conformal 3D cross field that can be used to guide the generation of hex-dominant meshes. Sokolov et al. [42] extended the periodic global parameterization for surfaces to 3D and proposed a field-aligned parameterization method to guide the agglomeration of tetrahedra to produce hex-dominant meshes. Gao et al. [4] improved Sokolov et al.'s method by addressing the non-conformality issue. However, their methods cannot control the complexity of the non-hex elements.

Other methods for generating hex-dominant meshes have been proposed. Lévy and Liu [43] introduced L_p centroidal Voronoi tessellation that can produce hex-dominant meshes to some extent when $p \to \infty$. Yu et al. [44] created hex-dominant meshes by employing a multi-model polycube-based algorithm that requires manual intervention. Livesu et al. [3] proposed an automatic process to produce hex-dominant meshes by mimicking manual block decomposition. More recently, Bukenberger et al. [5] generated at-most-hex meshes based on a 3D Lloyd relaxation under the L_∞ norm for a harmonious hex cell layout.

B. Hex-dominant Mesh Optimization

While the above methods can produce hex-dominant meshes with a small number of non-hex polyhedra, pure (or all) hex meshes are still preferred by the solvers of different PDEs. Yamakawa and Shimada [45] introduced HEXHOOP to convert a hex-dominant mesh to an all-hex mesh. The proposed method is capable of automatically converting a hex-dominant mesh to an all-hex mesh by subdividing a prism/pyramid/tetrahedral element into a set of smaller hex elements while ensuring topological conformity with neighboring elements. However, it cannot handle other more complex non-hex element types (e.g., polyhedra with arbitrary numbers of faces).

To handle more complex configurations of non-hex elements, this paper proposes a more global method by resorting to the structure of the hex-dominant meshes [17]. This structure is closely related to the *base complex* of all-hex meshes [14], [46]. A base complex of an all-hex mesh is a partition of the mesh and its elements into larger hex blocks. Its complexity (usually characterized by the number of hex blocks) affects not only the quality of the corresponding mesh but also the subsequent fitting of high-order basis functions for computation [18], [47], [48]. To simplify the base complex, singularity alignment technique [14] and sheet removal method [16], [46], [49] have been proposed.



(a) (b)

Fig. 2: A pure hex-mesh [27] (a) and a hex-dominant mesh [4] (b). Non-hex cells are shown in red.

Brückler et al. [36] introduced the 3D motorcycle complex as an alternative to the base complex. The main difference between motorcycle complex and base complex of an all-hex mesh is that motorcycle complex allows the existence of T-junctions, allowing it to have fewer hex blocks than the conventional base complex. They later used the motorcycle complex to aid the generation of 3D T-meshes that can be converted to all-hex meshes [50]. Note that Schertler et al. [51] used quad-dominant motorcycle as a structure representation for quad-dominant meshes.

In summary, there is little work for the effective reduction of the non-hex elements in the given hex-dominant meshes with the aid of the structure information of the meshes.

III. Hex-dominant Mesh Basics

A. Hex-dominant Mesh

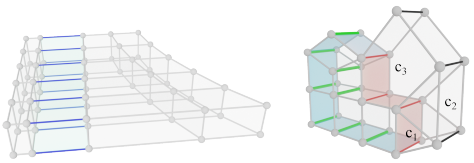
A hex-dominant mesh is a mixed-cell type mesh that primarily consists of hex cells but also contains a certain number of non-hex cells, as shown in Figure 2. It can be formally defined as $\mathcal{M} = (\mathcal{C}, \mathcal{F}, \mathcal{E}, \mathcal{V})$ within a volume Ω, where $\mathcal{V}$ defines the set of vertices, $\mathcal{E}$ represents the set of edges, $\mathcal{F}$ describes the set of faces, and $\mathcal{C}$ encompasses the set of three-dimensional polyhedral cells.

Edge set $\mathcal{E}$ is comprised of three subsets: $\mathcal{E} = (\bar{\mathcal{E}} \cup \mathbb{E} \cup \hat{\mathcal{E}})$. The subset $\bar{\mathcal{E}}$ contains edges exclusively associated with hex cells. The subset $\hat{\mathcal{E}}$, denoted as *non-hex edges*, consists of edges involved in non-hex cells. $\mathbb{E} = (\bar{\mathcal{E}} \cap \hat{\mathcal{E}})$, referred to as the transit edge set, marked in red in Figure 3, comprises edges that neighboring with both hex and non-hex cells. Face set $\mathcal{F}$ is defined as the union of three subsets: $\mathcal{F} = (\bar{\mathcal{F}} \cup \mathbb{F} \cup \hat{\mathcal{F}})$. The subset $\bar{\mathcal{F}}$ contains faces exclusively associated with hex cells. $\hat{\mathcal{F}}$, includes faces involved in non-hex cells, and $\mathbb{F} = (\bar{\mathcal{F}} \cap \hat{\mathcal{F}})$ consists of faces that interface between hex and non-hex cells, as illustrated by the red faces in Figure 3. Cell set $\mathcal{C}$ contains two subsets $(\mathcal{H}, \hat{\mathcal{H}})$ where $\mathcal{H}$ is a set of hex cells, and $\hat{\mathcal{H}}$ is a set of non-hex cells.

The transit elements ($\mathbb{E}$ and $\mathbb{F}$) play a crucial role in maintaining mesh connectivity across cell types, and the above element classification scheme facilitates the analysis of mesh topology and is particularly useful in developing algorithms for subsequent simplification.

B. Hybrid Base Complex

Referring to the existing method of structure analysis of hex-dominant meshes [17], we adopt the hybrid singularity graph and hybrid base complex for structure evaluation. A hybrid singularity graph is comprised of singularity edges and edges in non-hex cells. The hybrid base complex is the volume components that are constructed by separation surfaces traced out from edges in the corresponding hybrid singularity graph.



(a) (b)

Fig. 3: (a) A set of parallel edges (highlighted in blue) identified through the parallel relation definitions and propagation rules. (b) A group of parallel edges (green and red) defines a sheet bounded by blue and red patches. Due to adjacent non-hex cells (c_1, c_2, and c_3), the black parallel edges are excluded from the sheet.

C. Mesh Element Neighborhood Relations

In the context of mesh structures, we have two fundamental types of neighborhood relations between elements. These relations are fundamental for understanding the topological properties of the mesh and form the basis for hex-dominant mesh analyses and simplification algorithms.

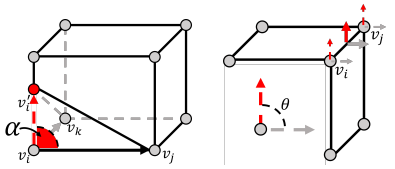
Element Adjacency Relation describes the adjacency between elements of the same type as follows:

- Two cells are adjacent if they share at least a face.
- Two faces are adjacent if they share at least an edge.
- Two edges are adjacent if they share a vertex.
- Two vertices are adjacent if they share an edge.

Element Inclusion Relation describes the adjacency between two elements of different types if one is a component of the other. It is important to note that this relation is bidirectional. For instance, vertices, edges, and faces that compose a cell are all considered to be adjacent to that cell, and conversely, the cell is adjacent to its component elements.

In the remainder of this paper, we collectively refer to both the Element Adjacency Relation and the Element Inclusion Relation as neighborhood relations. Elements that satisfy either of these relations are described as neighboring elements.

D. Element Corner



(a) (b)

Fig. 4: (a) Corner illustration: repositioning v_i' to v_i creates an ideal cell corner configuration. Vectors from v_i to v_i' and v_i to v_j define a face corner, forming angle α. (b) Sharp edge identification: an edge is classified as sharp when the angle $\theta > \beta_{sharp}$.

Cell Corner: A corner in a 3D cell is formed by all connected edges meeting at a vertex in a manifold schema. In hex cells,

the quality of the corner should approach orthogonality. The deviation of a corner from an ideal orthogonal configuration can be quantified using the Scaled Jacobian metric [52].

Face Corner: A face corner is formed by two edges (belonging to the same face) intersecting at a vertex when the face is manifold. In hex cells, each corner of the faces should ideally have a 90-degree angle.

E. Parallel Edge

Parallel Relation: Let $\mathcal{E}_q \subseteq \mathcal{E}$ be the subset of edges that are included in quadrilateral faces. Two edges $e_i, e_j \in \mathcal{E}_q$ are defined as *parallel* if and only if they belong to the same quadrilateral face and do not share any vertices [14]. We denote this relation as $e_i \parallel e_j$.

Propagation of Parallel Relation: The parallel relation exhibits a transitive property. For edges $e_i, e_j, e_k \in \mathcal{E}_q$, if $e_i \parallel e_j$ and $e_j \parallel e_k$, then $e_i \parallel e_k$, even e_i and e_k are in different faces.

Parallel Edge Set: Given an edge $e_i \in \mathcal{E}_q$, we define its parallel edge set as the set of all edges parallel to e_i, as illustrated in Figure 3a.

F. Definitions of Edges

Sharp Edges: An edge shared by two adjacent faces is classified as sharp when the angle θ between their face normals exceeds a user-specified threshold β_{sharp}.

Sharp edges play a critical role in defining the shape of a 3D volume. Accurate detection of these edges is essential for preserving boundary features during simplification. However, in hex-dominant meshes, faces can contain multiple edges (more than three), which may result in non-planar surfaces. To address this geometric complexity, we compute vertex normals at each face corner using local vector information. The face normal along an edge is then calculated by averaging the corresponding vertex normals, following the right-hand rule convention. As shown in Figure 4b, the sharpness of an edge is determined by analyzing the two directional vectors (highlighted in color) along that edge.

Local Edge: An edge surrounded by only non-quadrilateral faces is considered a local edge, as it has no parallel edges.

Edge Quality Score: The quality metric for an edge e, denoted as $q(e)$, is computed based on the absolute cosine values of its adjacent face angles. This metric is formally defined as:

$$q(e) = \min_{i \in K} 1 - |\cos(\alpha_i)| \tag{1}$$

where α_i represents the corner angle at a vertex of edge e in the surviving adjacent faces after the collapse operation. Notably, faces that are eliminated during the collapse operation are excluded from this computation.

The ranking of edges can be based on either the maximum or minimum cosine value. Using the maximum value prioritizes the best-case corner angle, while the minimum value considers the worst-case corner angle. In this study, we employ the minimum value approach for edge ranking to minimize potential impact of mesh quality.

IV. Mesh Sub-structure Configuration

A. Sub-structure from Hex-dominant Mesh

Sub-structures: We represent a parallel edge set and its associated elements as a *sub-structures* $\mathcal{S} = (\mathcal{C}_\mathcal{S}, \mathcal{F}_\mathcal{S}, \mathcal{E}^{\parallel}_\mathcal{S}, \mathcal{E}_\mathcal{S}, \mathcal{V}_\mathcal{S})$, where $\mathcal{E}^{\parallel}_\mathcal{S} \subseteq \mathcal{E}_q$ is the set of parallel edges in the group, for any $e_i, e_j \in \mathcal{E}^{\parallel}_\mathcal{S}$, we have $e_i \parallel e_j$. $\mathcal{C}_\mathcal{S}, \mathcal{F}_\mathcal{S}, \mathcal{E}_\mathcal{S}$, and $\mathcal{V}_\mathcal{S}$ are cells, faces, edges, and vertices adjacent to the edges in $\mathcal{E}^{\parallel}_\mathcal{S}$.

As these sub-structures $\mathcal{S}$ may span different regions and create complex configurations that challenge mesh simplification, we partition them into two subsets based on edge types.

Non-hex Sub-structure (Groups): A set of elements, denoted as $\hat{\mathcal{S}}$, is defined by a set of parallel edges $\mathcal{E}^{\parallel}_{\hat{\mathcal{S}}}$ that neighboring with non-hex cells. As illustrated in Figure 3b, the edges marked in red and black are parallel edge set of $\hat{\mathcal{S}}$.

Hex Sub-structure (Sheets): A sheet, denoted as $\bar{\mathcal{S}}$, is defined by a set of parallel edges $\mathcal{E}^{\parallel}_{\bar{\mathcal{S}}}$ neighboring with hex cells. These sub-structures are analogous to sheets in all-hex meshes [15], [46]. Figure 3b shows a group of parallel edges (marked in green and red) and their neighboring elements within hex cells, together defining a sheet.

Sub-structure Continuity: A sub-structure $\mathcal{S}_i$ is *continuous* or *complete* if and only if all parallel edges in the group are connected by a set of quadrilateral faces in the same structure without interruption. In this paper, we enforce the constraint that all groups must be continuous.

B. Sub-structure Configurations

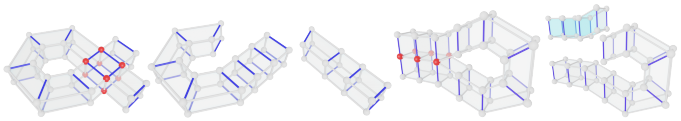


Fig. 5: (a) is a self-intersecting sheet. Red spheres highlight places where self-intersection occurs. This sheet can be decomposed into two sub-sheets, as shown in (b) and (c), without self-intersections. (d) shows a self-tangent sheet. Red spheres highlight places where self-tangent arises. This sheet can be decomposed into two sub-sheets, as shown in (e).

Two configurations in sub-structures are important for the subsequent simplification framework, i.e., the self-tangent and self-intersecting configurations.

Self-intersecting Sub-structure: a sub-structure is *self-intersecting* if it possesses at least two parallel edges that share a vertex and are from the same cell (i.e. the sheet returns to the same hex cell). Figure 5a illustrates a self-intersecting sheet defined by the blue edges. The red dots highlight the hex cell where the sheet self-intersects.

Self-tangent Sub-structure: a sub-structure is *self-tangent* if it has at least two parallel edges that share a vertex and are in two different but adjacent cells. Figure 5d illustrates a self-tangent sheet, defined by the blue edges, that is neighboring to itself. The red dots highlight the locations where the self-tangent configuration arises.

C. Neighborhood Relation between Sub-structures

As illustrated in Figure 1 (b-c, middle row), when a sub-structure has a simpler configuration, such as not being connected to other sub-structures, or being connected to other sub-structures but without loops, collapsing all edges that define that sub-structure can prevent the creation of new non-hex elements (Figure 1 (b-c, bottom row)). Based on this characteristic, in this paper, we focus on the sub-structure propagation direction at the sub-structure level, which differs from the neighborhood relations at the mesh level.

We define two sub-structures as *connected* when they share an edge that maintains parallel relationships with edges in either sub-structure. These connections arise either from separation at regions between hex cells and non-hex cells or from the subdivision of larger groups based on previously described configuration criteria.

Formally, let $\mathcal{S}_i$ and $\mathcal{S}_j$ be two sub-structures. They are considered connected if:

$$\exists e \in E: \quad e \in \mathcal{E}^{\parallel}_{\mathcal{S}_i} \wedge e \in \mathcal{E}^{\parallel}_{\mathcal{S}_j}$$

As illustrated in Figure 3b, non-hex cells c_1, c_2, and c_3 are adjacent to the sheet because they are sharing at least one edge (marked in red). The sheets neighboring non-hex cells are of particular interest in our subsequent simplification framework. For connections between non-hex sub-structures and sheets, the connecting edge e is always an element of $\mathbb{E}$, which represents the set of edges at the interface between hex and non-hex regions.

An edge is considered as *adjacent* to a sub-structure $\mathcal{S}$ when it belongs to $\mathcal{E}^{\parallel}_{\mathcal{S}}$. While other edges may be included in the sub-structure, they are not considered as neighboring elements since they do not play a crucial role in the sub-structure configuration.

D. Neighborhood Relation Graph

To describe the important entities involved in simplification, we construct an undirected graph $G_R = (N, R)$ to capture undiscovered patterns within sub-structures, which may affect the outcome of simplification. In this graph, nodes (N) represent mesh edges and sub-structures, while the graph edges (R) encode the neighborhood relations:

- Edge-to-edge adjacency $\mathcal{E} \sim \mathcal{E}$
- Edge-to-sub-structure adjacency relations (include both group $\hat{\mathcal{S}} \sim \mathcal{E}$ and sheet $\bar{\mathcal{S}} \sim \mathcal{E}$)
- sub-structure-to-sub-structure neighborhood relations (include both group-group $\hat{\mathcal{S}} \sim \hat{\mathcal{S}}$, sheet-group $\hat{\mathcal{S}} \sim \bar{\mathcal{S}}$ and sheet-sheet $\bar{\mathcal{S}} \sim \bar{\mathcal{S}}$)

We distinguish between sheet and non-hex sub-structure as separate entities in the graph, reflecting their construction from distinct cell types in hex-dominant meshes. This classification facilitates a more detailed analysis of relations between different sub-structures.

The graph representation does not help us intuitively understand some complex configurations between sub-structures (e.g., the configurations shown in Figure 6c and Figure 6f), for the development of an effective strategy for sub-structure selection during simplification (subsection V-E).

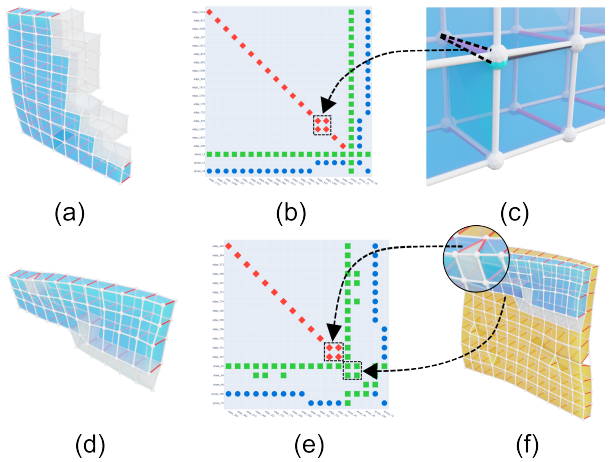


Fig. 6: (a) A sheet exhibiting an apparently optimal connectivity configuration. (b) The proposed NRG reveals two adjacent edges in the sheet's parallel edge set. (c) A magnified view of the irregular configuration. (d) A sheet with basic configuration. (e) The NRG demonstrates the interconnection between two sheets. (f) Two connected sheets, distinguished by different colors, are separated due to self-tangent configuration.

To address that, we represent the relation graph as an adjacency matrix and visualize this adjacency matrix by visually encoding different relations (Figure 6 b, e). This strategy is particularly effective for analyzing dense networks with complex topological structures, especially when geometric properties such as distance or size vary significantly across the graph [53]. Following the principles introduced in [15], [54], we define the adjacency matrix $\mathbf{M}$ of graph G_R, where:

$$m_{ij} = \begin{cases} 1 & \text{if nodes } n_i \text{ and } n_j \text{ are connected} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The matrix can be further decomposed to six sub matrices.

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{\mathcal{E}\mathcal{E}} & \mathbf{M}_{\bar{\mathcal{S}}\mathcal{E}} & \mathbf{M}_{\hat{\mathcal{S}}\mathcal{E}} \\ \mathbf{M}_{\mathcal{E}\bar{\mathcal{S}}} & \mathbf{M}_{\bar{\mathcal{S}}\bar{\mathcal{S}}} & \mathbf{M}_{\hat{\mathcal{S}}\bar{\mathcal{S}}} \\ \mathbf{M}_{\mathcal{E}\hat{\mathcal{S}}} & \mathbf{M}_{\bar{\mathcal{S}}\hat{\mathcal{S}}} & \mathbf{M}_{\hat{\mathcal{S}}\hat{\mathcal{S}}} \end{bmatrix}$$

For weighted graphs, m_{ij} represents the edge weights. In our visual matrix representation, we encode node types using a color scheme, Red: edges, Green: sheets, Blue: non-hex sub-structures. Edge weights are represented by varying opacity in the visual matrix.

We refer to this graph and its corresponding visual representation as the Neighborhood Relation Graph (NRG). To enhance the visualization of relations, we order the rows and columns of the matrix based on connections across entities, similar to the block-diagonal forms introduced by Behrisch et al. [54] and Xu et al. [15]. However, we maintain entities of the same type in a contiguous sequence to reveal patterns within each entity category.

With the above definition, we can design the NRG representations shown in Figure 6, where different entities are visualized with distinct colors and communities in the matrices. For the configuration illustrated in Figure 6a-c, two overlapping adjacent edges that are challenging to distinguish through direct visualization are clearly represented in the NRG (Figure 6b), while the matrix reveals that edges adjacent to

the sheet belong to distinct non-hex sub-structures without mutual connectivity, indicating that these non-hexagonal edges are either distributed in different locations or have complex configurations between them. The case presented in Figure 6d-f demonstrates sheet-to-sheet connectivity patterns visualized through the matrix representation (Figure 6e), which provides more intuitive identification of abnormal configurations compared to direct inspection of the mesh shown in Figure 6f.

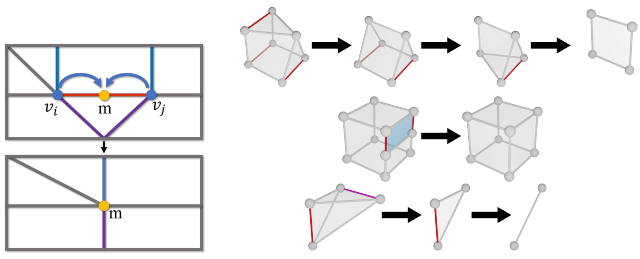
V. Hex-dominant Mesh Simplification

Building upon the above foundations, we propose a hex-dominant mesh simplification algorithm that leverages edge collapse operations, configuration validation, and selection ranking in 3 different operation levels. The detail is described in the following subsections.

A. Edge Collapse Operations

The fundamental atomic operation in a mesh simplification algorithm is the edge collapse operation. This operation eliminates an edge by merging two adjacent vertices and thereby changes adjacent faces, as shown in Figure 7a. In the context of hex-dominant meshes, depending on the face configuration of a non-hex cell, the non-hex cell can be eliminated by collapsing a set of edges in a designed sequence. Figure 7b demonstrates various non-hex cells that can be transformed into elements valid for a pure hex mesh through the careful application of edge collapses in the specified order.

In this paper, aiming to simplify hex-dominant meshes, we employ the same atomic operation as other simplification algorithms, but the edge collapse strategy in our approach is structure-informed. The goals of the structure-informed edge collapse strategy are to (a) reduce the number of non-hex cells, (b) improve mesh configuration, (c) reduce the complexity of individual non-hex cell configurations, and (d) avoid creating cells with negative scaled Jacobian values as much as possible. To achieve these goals, the collapse strategy is constructed using an edge validation process and a set of candidate edge selection methods, which are discussed in detail in the following sections.



(a) (b)

Fig. 7: (a) Collapsing two adjacent vertices to their midpoint eliminates non-quadrilateral faces. (b) Strategic edge collapse operations can eliminate or convert a subset of non-hex elements into hex cells.

B. Edge Validation in Sandbox

In hex-dominant meshes, where cells have different types and varied mesh structures, edge collapse operations may create invalid configurations like non-manifold edges or elements with negative Scaled Jacobian values. Since it is a local operation that primarily affects the one-ring neighborhood of a target edge, we validate mesh quality by testing each collapse operation in an isolated region, called a *sandbox*, before applying changes to the mesh.

Sandbox: A sandbox for collapse operation validation is constructed from the one-ring neighboring elements of a vertex set $\mathcal{V}_{sd}$, which is derived from a given edge set $\mathcal{E}_{sd}$.

Validation Criteria: The following conditions must be satisfied in the sandbox for edge collapse approval:

- Manifold preservation: The resulting mesh topology must maintain manifold properties
- Jacobian validity: No newly created elements may exhibit negative Jacobian determinants
- Angular deviation: Corner angle modifications must remain within user-specified thresholds

The angular deviation threshold affects both boundary and internal elements. For all edges, after an edge collapse operation, If any face incident to the resulting vertex exhibits a corner angle that deteriorates by more than a user-specified threshold β_{corner} compared to its original state, or if the collapse operation for a boundary edge creates a new sharp edge, the edge is marked invalid.

C. Local Edge Selection

Edge Ranking Strategy: In a hex-dominant mesh, multiple edges may be eligible for collapsing. To select appropriate edges in each local region while aiming to produce desired outputs, as shown in Figure 1d-e, we designed a two-phase ranking strategy. First, we evaluate and rank all edges within their associated cells, selecting only those with the lowest quality score (Equation 1) in their local neighborhood. Second, for multiple candidate edges sharing a vertex, we randomly select and retain only one edge among those connected edges.

D. Graph-based Sub-structure Analysis

Based on the matrix representation introduced in subsection IV-D, we utilize matrix-based metrics to evaluate sub-structures. Specifically, we quantify the structural quality by measuring the distance between the edge region matrix ($\mathbf{M}_{\mathcal{EE}}$) and an identity matrix. This comparison is particularly meaningful because an identity matrix represents the ideal configuration: a perfect sub-structure would consist exclusively of parallel edges with no shared vertices between parallel edges within the same sub-structure.

Hamming Distance $d_H(\mathbf{M}_i, \mathbf{I})$: The Hamming distance provides an unweighted measure between the edges in NRG matrix and the identity matrix. A larger distance indicates more neighboring connections within the same sub-structure, suggesting a more complex mesh configuration. This metric helps identify potentially problematic sub-structures where:

- Higher values indicate increased topological complexity
- Lower values suggest better-aligned sub-structure

We employ hamming distance to evaluate sub-structure configurations. The distance is not only useful for measuring differences between sub-structures but also valuable for

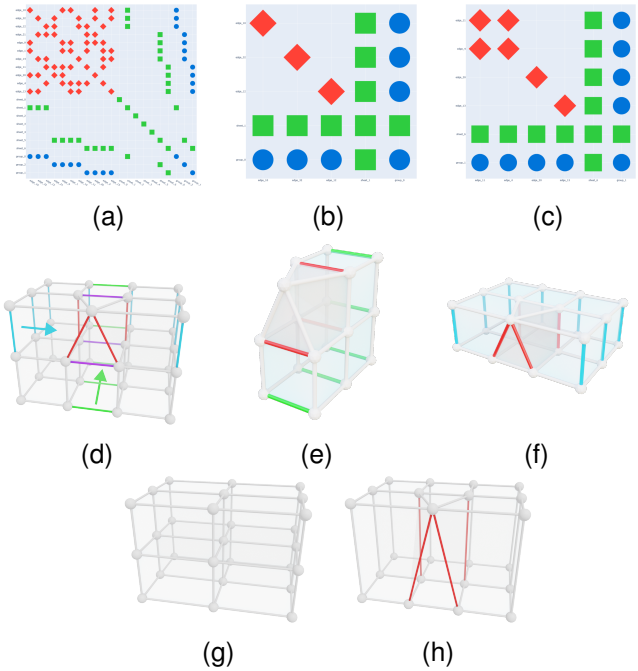


Fig. 8: The Neighborhood Relation Graph (NRG) provides direct visualization for analysis and guides the simplification process of a mesh. (a) Visualized NRG matrix for the input mesh shown in (d). (b,c) NRG matrices for sheets shown in (e) and (f), respectively. (g) Result of collapsing the sheet shown in (e). (h) Final output mesh obtained by collapsing the sheet in (f). The quality difference between (g) and (h) demonstrates that sheets with simpler NRG patterns produce better configurations.

sub-structure selection during simplification, as sub-structures with non-zero distances may not effectively reduce non-hex cells after collapsing. As illustrated in Figure 8, collapsing a sheet (Figure 8e) with zero Hamming distance (Figure 8b) successfully transforms the input mesh (Figure 8d) into the configuration shown in Figure 8g. In contrast, when attempting to collapse a sheet (Figure 8f) with a Hamming distance of 2 (Figure 8c), the operation fails to reduce the number of non-hex cells, as demonstrated in Figure 8h.

E. Sub-structure Selection Strategy

A mesh contains a large number of sub-structures with varying neighborhood connectivity patterns within a parallel edge set, creating many challenges. To systematically simplify hex-dominant meshes, we implement a three-phase ranking strategy for sub-structure selection:

Phase 1: Parallel Edge Connectivity Filter. We restrict our simplification to sub-structures with a Hamming distance of zero. sub-structures with non-zero distance are filtered out, as this indicates the presence of neighboring edges within the same parallel edge set. This filtering ensures topological consistency within each sub-structure.

Phase 2: Configuration Validation. Each sub-structure is validated through two aspects. First, we evaluate individual edge collapse operations and compute edge scores. Second, we verify the mesh configuration after collapsing all parallel edges within the sub-structure. The sub-structure is excluded from simplification if it creates negative scaled Jacobians, non-manifold configurations, non-hex cells. Users can control whether to allow the creation of non-hex cells and vertices with higher valence.

Phase 3: Sub-structure Ranking. For each valid sub-structure $\mathcal{S}_i$, we compute a composite quality metric $q(\mathcal{S}_i)$ that combines multiple quality measures:

$$q(\mathcal{S}_i) = min(q(e)) + \Delta\hat{\mathcal{H}}(\mathcal{S}_i) + \Delta R_{\mathcal{H}}(\mathcal{S}_i) + \Delta\mathcal{J}_{min} \quad (3)$$

where $min(q(e))$ represents the minimum edge quality score among all edges in $\mathcal{E}^{\parallel}_{\mathcal{S}_i}$. The relative change in non-hex elements, denoted as $\Delta\hat{\mathcal{H}}(\mathcal{S}_i)$, is defined as:

$$\Delta\hat{\mathcal{H}}(\mathcal{S}_i) = \frac{|\hat{\mathcal{H}}|^{after} - |\hat{\mathcal{H}}|^{before}}{|\hat{\mathcal{H}}|^{before}}$$

where $|\hat{\mathcal{H}}|^{before}$ and $|\hat{\mathcal{H}}|^{after}$ represent the count of non-hex elements before and after the operation. The change in hex element ratio, denoted as $\Delta R_{\mathcal{H}}(\mathcal{S}_i)$, is defined as:

$$\Delta R_{\mathcal{H}}(\mathcal{S}_i) = R^{after}_{\mathcal{H}}(\mathcal{S}_i) - R^{before}_{\mathcal{H}}(\mathcal{S}_i)$$

where $R^{after}_{\mathcal{H}}(\mathcal{S}_i)$ and $R^{before}_{\mathcal{H}}(\mathcal{S}_i)$ represent the hex element ratios before and after the collapse operation. $\Delta\mathcal{J}_{min}$ represents the change in the minimum scaled Jacobian of hex elements. All these metrics are evaluated within the local sandbox region rather than the entire mesh domain. The terms $\Delta R_{\mathcal{H}}(\mathcal{S}_i)$ and $\Delta\mathcal{J}_{min}$ have ranges of $[-1, 1]$ and $[-2, 1]$ respectively, in order to increase the sensitivity when collapse operation creates lower hex ratios or poorer hex cell quality in their domain.

Sub-structures are ranked in descending order of their quality metrics $q(\mathcal{S}_i)$, with the sub-structure having the highest $q(\mathcal{S}_i)$ selected as the optimal candidate for collapse operations, leading to optimal mesh quality outcomes, as illustrated in Figure 1, and Figure 8.

When collapsing a group (parallel edges in non-hex cell region), as illustrated in Figure 9, the edge collapse operations are restricted to edges $e \in \hat{\mathcal{E}} - \mathbb{E}$, since collapsing transition edges would convert adjacent hex cells into non-hex ones.

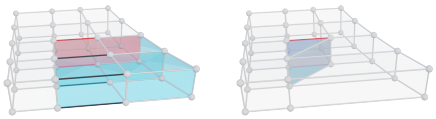
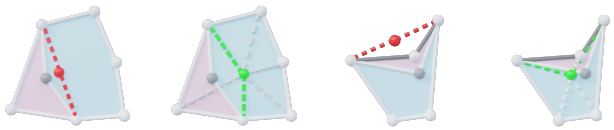


Fig. 9: (a) Parallel edges (shown in black and red) within non-hex cells (blue). Red faces indicate interfaces between hex and non-hex cells. (b) During edge collapse, only black edges are processed while red edges are excluded from the group to prevent conversion of existing hex cells into non-hex cells.

F. Mesh Smoothing

Since hex-dominant mesh has different types of cells, we can only use smoothing methods that don't need specific cell type information, like Laplacian smoothing. Based on this demand, we developed a new volumetric mesh smoothing algorithm by extending the Laplacian smoothing method to work better with hex-dominant meshes.

Our method is different from the traditional Laplacian smoothing approach in how it handles vertex neighborhoods. The traditional method only looks at vertices that directly connect to the target vertex (called one-ring neighbor vertices). Instead, we first find all the cells that contain the target vertex (one-ring cells) and use all vertices from these cells to calculate where the target vertex should move. For example, consider a vertex located at the junction of three cells: one side has two hex cells, while the other side has a large non-hex cell. Adjusting the middle vertex can cause the hex cells to develop low-quality corners. In some cases, a negative scaled Jacobian value may be introduced after smoothing by the traditional approach Figure 10. The proposed approach helps keep vertices within a reasonable volume space by considering more neighboring information. While this can help reduce problems with negative scaled Jacobian values that may occur in traditional Laplacian smoothing when vertices move too far, it cannot guarantee their elimination.



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

Fig. 10: (a) The traditional approach creates a flat corner due to the lack of edge connectivity to the red node. (b) By considering all one-ring neighbor vertices, the vertex moves toward the centroid, which creates a convex quad face. (c) On the boundary, the traditional approach creates inverted cells since the vertex position moves outside the boundary. (d) With the adjusted approach, the vertex remains inside the boundary.

G. Pipeline of Hex-dominant Mesh Simplification

By integrating the previously described techniques, we present a comprehensive simplification algorithm (Algorithm 1) for hex-dominant meshes. The algorithm consists of four main stages executed iteratively:

Stage 1: Local Edge Collapse. The algorithm begins with local edge as detailed in Section V-C. We collapse a batch of selected local edges in each iteration. This stage continues until no additional local edges meet the selection criteria.

Stage 2: Non-hex Group Collapse. Following local edge operations, we iteratively process ranked parallel edge groups associated with non-hex elements. In each iteration, we collapse one group and recompute the rankings of all remaining groups. This stage precedes sheet operations because these groups are typically adjacent to non-hex elements, and their collapse may create more favorable configurations for subsequent sheet operations, like illustration in Figure 9. The process iterates until no valid groups remain.

Stage 3: Sheet Collapse. Similar to stage 2, we apply the ranking strategy to identify and collapse optimal sheet candidates. After each sheet collapse, the algorithm re-evaluates and selects the next highest-ranked sheet. This process continues until no valid sheets remain for consideration.

Stage 4: Mesh Quality Optimization. Although we do smoothing after each operation, an additional final optimization stage is still incorporated. This stage specifically focuses on ensuring that the resulting mesh maintains high-quality elements after all structural modifications.

The pipeline iterates through these four stages sequentially until the algorithm reaches a state where no further valid modifications can be identified in any stage, ensuring maximum simplification while maintaining mesh quality constraints.

Algorithm 1 Hex-Dominant Mesh Simplification

Require: Input hex-dominant mesh $\mathcal{M}$
Ensure: Simplified hex-dominant mesh $\mathcal{M}'$

```
do
    ▷ Stage 1: Local Edge Collapse
    while valid local edge exist do
        Select and collapse batch of local edges
        Quick smoothing
    end while
    ▷ Stage 2: Non-hex Group Collapse
    while valid groups exist do
        Collapse highest-ranked group
        Quick smoothing
        Re-evaluate remaining groups
    end while
    ▷ Stage 3: Sheet Collapse
    while valid sheets exist do
        Collapse highest-ranked sheet
        Quick smoothing
        Re-evaluate remaining sheets
    end while
    ▷ Stage 4: Mesh Quality Optimization
while Collapsed an edge
return M'
```

VI. Results and Evaluation

We have applied the proposed simplification framework to a number of hex-dominant meshes produced by three state-of-the-art approaches, i.e., the Robust Hex-dominant meshing [4], the LoopyCuts [3], and the At-Most-Hex meshing [5]. 49 meshes are tested, among them 27 are Robust-Hex-Dominant meshing, 15 from LoopyCuts, and 7 from At-Most-Hex. Table I provides the statistics of the meshes and their resulting meshes shown in the paper. The complete statistics are provided in the supplemental material.

A. Improvement on Robust Hex-dominant Meshing

The robust hex-dominant mesh generation algorithm uses an optimized frame field to guide the mesh generation process, which yields isotropic hex cells aligned with the boundary features. While the cells and singularities visually aligned well, some cells contain more than 30 faces, and arbitrary connectivity creates challenges. Nevertheless, our proposed simplification method still reduces more than 20% of non-hex cells on average for those test cases, and the maximum numbers of the remaining non-hex cells also decrease.

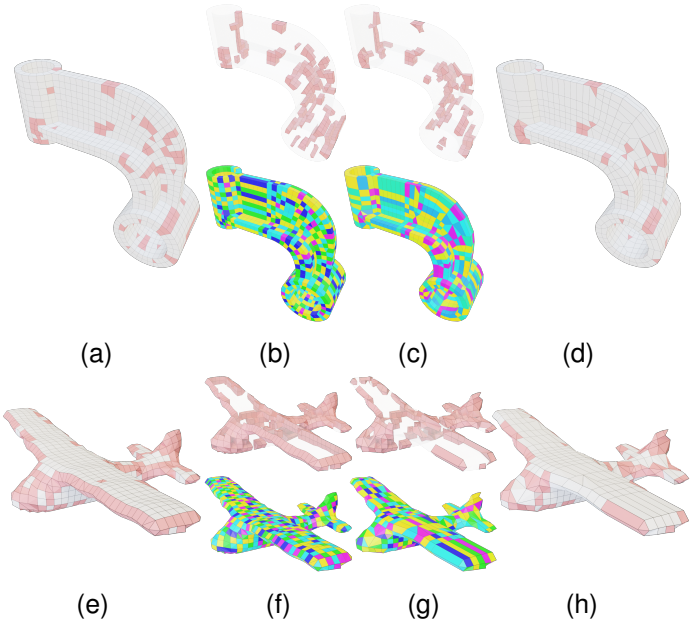


Fig. 11: Two meshes from the robust hex-dominant approach [4]. The top row shows the $hanger$ model (a), its non-hex cells (b, top), hybrid base complex (b, bottom), and the simplified mesh (d), its non-hex cells (c, top), and hybrid base complex (c, bottom). The bottom row shows the $airplane1$ model (e), its non-hex cells (f, top), and hybrid base complex (f, bottom). After simplification (h), the remaining non-hex cells (g, top) result in large components in the left wing of the hybrid base complex (g, bottom).

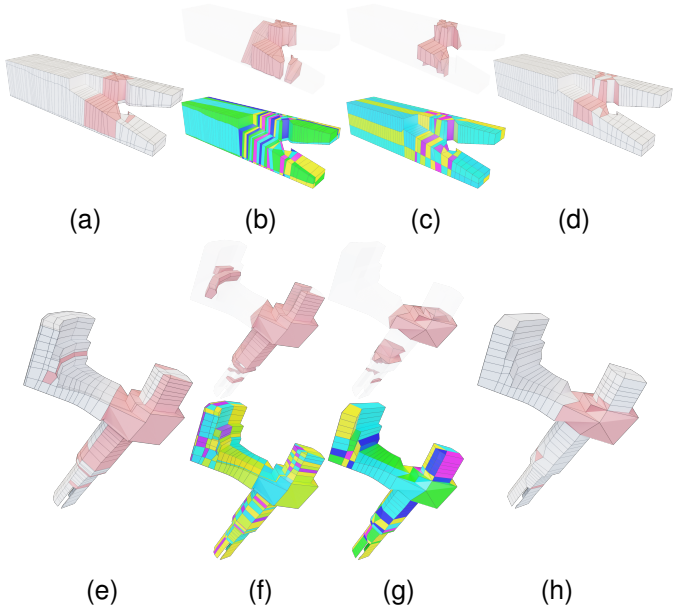


Fig. 12: Two meshes with the LoopyCuts approach [3]. The top row shows the $halved\_oblique\_scarf\_2\_mm$ model (a), its non-hex cells (b, top), and hybrid base complex (b, bottom). After simplification (d), the remaining non-hex cells (c, top) lead to a simpler hybrid base complex (c, bottom). The bottom row shows the $nugear\_mm$ model (e), its non-hex cells (f, top), and hybrid base complex (f, bottom). After simplification (h), non-hex cells in the L-shaped region are fully removed (g, top), yielding a much simpler hybrid base complex (g, bottom).

Figure 11 presents two models from robust hex-dominant meshing. The first model, a $hanger$ shown in Figure 11a, contains 240 non-hex cells, with the maximum number of faces among all cells being 13. As illustrated in Figure 11b, the majority of non-hex cells are concentrated in the lower right region, resulting in small components in the extracted hybrid base complex. After 74 iterations of simplification, the number of non-hex cells reduced to 141, representing a 41% reduction. This reduction leads to larger components in the extracted hybrid base complex, as shown in Figure 11c.

The second model, an $airplane$ shown in Figure 11e, contains 383 non-hex cells, with the maximum number of faces among all cells being 23. After 161 iterations of simplification, 157 non-hex cells were eliminated, corresponding to a 41% reduction. The maximum number of faces among all cells dropped to 18. While some non-hex cells remain at the wing locations, they are no longer continuously connected, resulting in significantly larger components in the hybrid base complex (shown in Figure 11g) compared to the extracted hybrid base complex components from input mesh.

B. Improvement on LoopyCuts

Loopycuts method mimics manual cutting scenarios by cutting meshes along a set of loops to generate pure hexahedral or hex-dominant meshes. In the hex-dominant meshes generated by this method, visual results reveal complex configurations near shape features, leading to arbitrary non-hex cells. Our proposed simplification method reduces approximately 35% of the non-hex cells on average and improves cell shapes using an enhanced Laplacian smoothing method.

Figure 12 presents two CAD models where more than 50% of non-hex cells are removed, resulting in fewer hybrid base complex components. The first model, $halved\_oblique\_scarf\_2\_mm$ (Figure 12d), initially contained 105 non-hex cells which were reduced to 47, with the maximum number of faces among all cells decreasing from 10 to 9. The distribution of non-hex cells shows no significant difference between the input (Figure 12b) and output (Figure 12c), as this model's non-hex cells are clustered together. However, visualization of the extracted hybrid base complex reveals that the output mesh components exhibit improved size and shape compared to the input, which demonstrates the effectiveness of our simplification and smoothing approaches.

In the second model, $nugear\_mm$(Figure 12e), the number of non-hex cells is reduced from 136 to 26, representing an 81% reduction. The complete removal of non-hex cells in the upper left region changes the extracted components from 15 blocks to 2 blocks, indicating a simpler structure there (Figure 12f). This leads to significant visual differences compared to the hybrid base complex extracted from the input mesh (Figure 12g). A similar pattern can be observed in the lower region as well.

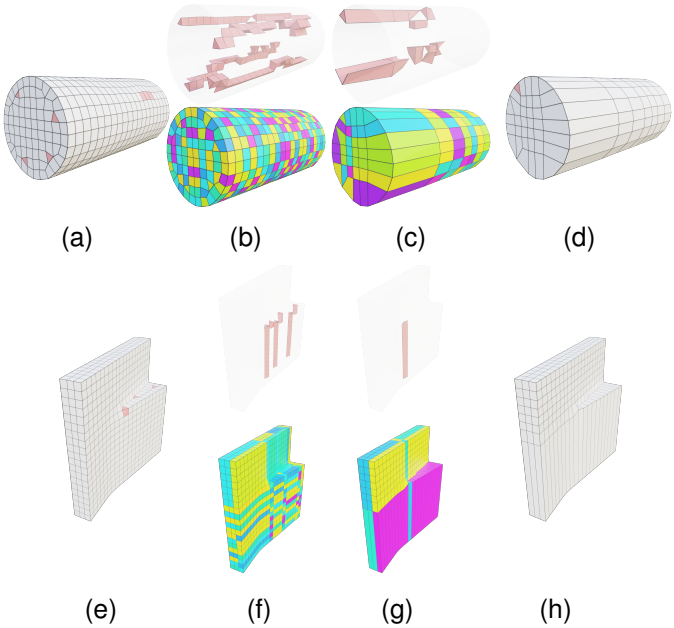


Fig. 13: Two meshes from the At-most-hex approach [5]. The top row shows the $cylinder\_g$ model (a), its non-hex cells (b, top), hybrid base complex (b, bottom), the simplified mesh (d), remaining non-hex cells (c, top), and the simpler hybrid base complex (c, bottom). The bottom row shows the $jump\_ramp$ model (e), its non-hex cells (f, top), and hybrid base complex (f, bottom). After simplification (h), most non-hex cells are removed (g, top), thus, a simpler hybrid base complex (g, bottom).

C. Improvement on At-Most-Hex meshing

The At-most-hex meshing method constructs hex-dominant meshes by extracting cells based on a 3D Lloyd relaxation under the L_∞ norm. Due to the method's limited cell types, our dataset shows significant reductions in non-hex cells and notable improvements in mesh structure.

The $cylinder\_g$ model, shown in Figure 13a, initially contains 110 non-hex cells. After simplification, the number of non-hex cells was reduced to 5, representing an 86% reduction. The number of components in the extracted hybrid base complex is reduced from 1275 to 114. The remaining non-hex cells could not be fully eliminated due to feature preservation and configuration constraints, as discussed in subsection V-B.

The $jump\_ramp$ model (Figure 13e) shows remarkable improvement, with non-hex cells reduced from 36 to just one. This single non-hex cell plays a crucial role in maintaining both the object's shape and connecting different regions, ensuring each region contains hex cells with the desired quality. The hybrid base complex demonstrates improved mesh structure quality after the removal of non-hex cells.

D. Discussion

Our experimental evaluation demonstrates the effectiveness of our approach through comparative analysis with three different mesh generation methods. By employing a systematic strategy that combines single-edge and parallel-edge collapse operations, our simplification framework successfully reduced 25% of non-hex elements on average across all test datasets.

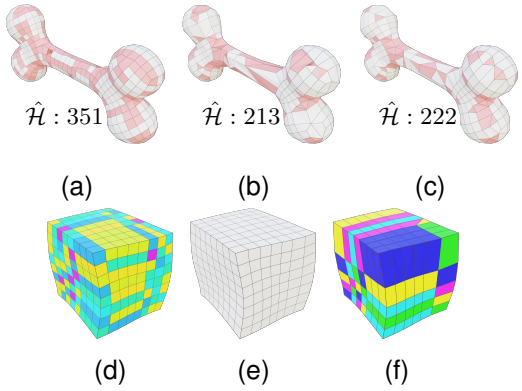


Fig. 14: (a) A bone model containing 351 non-hex cells, with a maximum of 23 faces in a single cell. (b) Simplified mesh without valence control: despite reducing the number of non-hex cells, certain vertices exhibit notably higher valence, resulting in complex connectivity configurations. (c) Simplified mesh with valence control: maintaining vertex valence during simplification preserves mesh quality but achieves less reduction in non-hex cells. (d) A twisted cube model [4] contains 18 non-hex cells. (e) Simplified to a pure hex mesh using the proposed ranking strategy. (f) Random selection strategy leaves 3 non-hex cells.

During simplification, we observed that in certain cases, vertex valence significantly increased, as our default configuration allows increased valence complexity when no inverted hex cell is created. However, our framework also provides an option to monitor and control vertex valence changes. As shown in Figure 14a-c, while vertex valence control results in a lower reduction of non-hex cells compared to the approach without valence control, it achieves better overall valence quality. Figure 14d-f indicates that the proposed ranking strategy produces better results than randomly selecting edges or sub-structures for collapse during mesh simplification.

The proposed algorithm has two user-specified thresholds, β_{sharp} for sharp edge detection and β_{corner} for edge collapse validation. While fine-tuning these thresholds have potential to produce better results on different models, adjusting them for each model is not the main focus of our study. We set both thresholds to 30° for all test cases, which is sufficient to obtain the desired results. A more detailed parameter study can be found in the supplemental document.

VII. Conclusion and Future Work

We present a new framework for reducing non-hex cells in hex-dominant meshes by using structural information. Our approach begins with single-edge collapse operations that only modify non-hex cells, followed by the extraction and analysis of sub-structures from the input meshes. The framework then decomposes self-intersecting and self-tangent sub-structures into separate sub-structures. Subsequently, sub-structures adjacent to non-hex cells are filtered and ranked. The highest-ranked sub-structures are collapsed to eliminate adjacent non-hex cells. This process continues iteratively until no further edge or sub-structure collapses are possible. We demonstrated the preliminary effectiveness of our framework by applying

Mesh	Iteration	$\|\mathcal{H}\|$	$\|\hat{\mathcal{H}}\|$	$R_{\Delta\hat{\mathcal{H}}}$	Max Poly Face	$\mathcal{J}_{min/max/avg}$	$Time(s)$
$halved\_oblique\_scarf\_2\_mm$‡	72	535 494	105 **47**	55.24%	10 **9**	0.28/1.00/0.95 0.00/1.00/0.97	145.02
$mechanical05\_mm$‡	281	502 193	575 **246**	57.22%	13 **11**	0.29/1.00/0.94 0.00/1.00/0.86	680.39
$nugear\_mm$‡	55	370 191	136 **26**	80.88%	12 12	0.10/1.00/0.92 0.09/1.00/0.81	172.24
$airplane1*$	161	824 572	383 **226**	40.99%	23 **18**	0.03/1.00/0.98 0.01/1.00/0.93	384.77
$bone*$	123	672 563	351 **213**	39.32%	23 **15**	0.08/1.00/0.97 0.03/1.00/0.91	300.13
$bone\_valence\_control*$	113	672 590	351 **222**	36.75%	23 **17**	0.08/1.00/0.97 0.02/1.00/0.92	321.34
$hanger*$	74	1404 1171	240 **141**	41.25%	13 **12**	0.69/1.00/0.99 0.00/1.00/0.97	480.73
$cylinder\_g$†	27	1545 253	110 **15**	86.36%	6 6	0.37/1.00/0.97 0.17/1.00/0.95	322.21
$jump\_ramp$†	17	1344 490	36 **1**	97.22%	6 6	0.78/1.00/0.99 0.65/1.00/1.00	566.69

TABLE I: Statistics information of our method applied to hex-dominant meshes. $*$ indicate meshes shown in the paper from the Robust-Hex-Dominant meshing [4], † from At-Most-Hex method [5], and ‡ from LoopyCuts [3]. The column $|\mathcal{H}|$ shows the numbers of hex cells, $|\hat{\mathcal{H}}|$ are numbers of non-hex cells, $R_{\Delta\hat{\mathcal{H}}}$ shows the percentage of reduction of non-hex cells. $Time$ is the average time of each iteration, the unit is second. $\mathcal{J}_{min/max/avg}$ displayds the scaled Jacobian values in a minimum/maximum/average format.

it to hex-dominant meshes generated using three different meshing techniques, showing its value to the community.

Despite the preliminary effectiveness, our approach has several limitations that impact its performance and usability. First, to minimize degenerate elements or topological inconsistencies in the mesh, the quality scores of affected edges and neighboring structures must be re-evaluated after each edge collapse operation before selecting the next collapse candidate. This creates strong data dependencies that prevent straightforward parallelization. This restricts our evaluation to relatively coarse meshes. To accelerate computation, identifying which operations can be safely performed in parallel without conflicts is a promising direction.

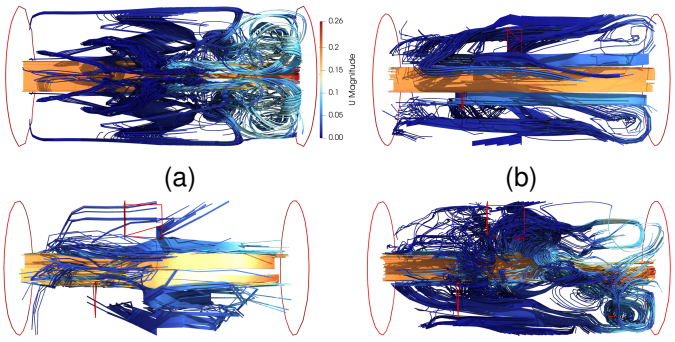


Fig. 15: (a) Reference simulation result obtained on a pure hex mesh with a cylindrical shape. (b) Simulation result obtained on a mesh from the at-most-hex approach [5]. (c) Simulation result obtained on the mesh produced by the proposed approach. (d) Simulation result obtained on our simplified mesh after refinement using OpenFOAM's built-in refinement tool.

Second, the usability of the simplified meshes is another limitation. Based on attempts to apply the simplified meshes in the simulation toolkit OpenFOAM [55], we found that our current results are not all simulation-ready. Since our approach cannot resolve all complex cell types and configurations present in the input mesh, most meshes still cause the simulation process to crash after simplification. Nevertheless, we successfully ran a simulation on a cylinder mesh simplified from the at-most-hex output. As illustrated in Figure 15c, the accuracy of the simulation result obtained on the simplified mesh is not comparable to the result obtained directly from the mesh produced by the at-most-hex approach (Figure 15b). However, our simplified mesh can be refined using OpenFOAM's built-in refinement tool, whereas the mesh from the at-most-hex approach cannot. With this refinement, our mesh produces more detailed results (Figure 15d). But, some undesired configurations remain in the simplified mesh (for example, at inner boundaries generated by the at-most-hex approach), causing the result to still show significant differences compared to the one computed on a mesh containing all hexahedral cells of good quality (Figure 15a). This example demonstrates that our approach still contributes to resolving certain invalid configurations and therefore provides benefits for simulation tasks. In the future, we will develop a method that simplifies arbitrary hex-dominant meshes into simpler configurations with fixed cell types.

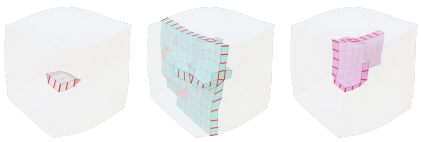


Fig. 16: (a) A simple sheet. (b,c) Two self-intersecting sheets obtained after the horizontal split.

Third, mesh refinement was not employed in our simplification pipeline as it may introduce complex configurations that are difficult to handle. For instance, as shown in Figure 16a, we started with a simple sheet from a twist cube mesh produced

by the at-most-hex approach. After refinement, this sheet split into two complex sheets (Figure 16b and Figure 16c) with self-intersecting configurations, which our method cannot simplify. We plan to address this in the future work.

Fourth, the current results can only reduce non-hex cells by around 25% on average. This limitation is primarily caused by the current approach only handling sub-structures that are not connected to other sub-structures in each iteration. More complex configurations, such as sets of sub-structures that are connected to each other, are difficult to handle without a proper sub-structure selection strategy. As future work, a more comprehensive ranking strategy could be developed by incorporating sub-structure connectivity information from the NRG to evaluate sequential collapse operations, since certain optimal configurations may arise after collapsing multiple edges and sub-structures. Additionally, a strategy for determining the optimal decomposition approach could improve the efficiency of the simplification process.

REFERENCES

[1] N. Pietroni, M. Campen, A. Sheffer, G. Cherchi, D. Bommes, X. Gao, R. Scateni, F. Ledoux, J. Remacle, and M. Livesu, "Hex-mesh generation and processing: A survey," *ACM Transactions on Graphics*, vol. 42, no. 2, Oct. 2022. [Online]. Available: https://doi.org/10.1145/3554920

[2] P.-E. Bernard, J.-F. Remacle, N. Kowalski, and C. Geuzaine, "Frame field smoothness-based approach for hex-dominant meshing," *Computer-Aided Design*, vol. 72, no. C, p. 78–86, Mar. 2016. [Online]. Available: https://doi.org/10.1016/j.cad.2015.10.003

[3] M. Livesu, N. Pietroni, E. Puppo, A. Sheffer, and P. Cignoni, "Loopycuts: practical feature-preserving block decomposition for strongly hex-dominant meshing," *ACM Transactions on Graphics*, vol. 39, no. 4, Aug. 2020. [Online]. Available: https://doi.org/10.1145/3386569.3392472

[4] X. Gao, W. Jakob, M. Tarini, and D. Panozzo, "Robust hex-dominant mesh generation using field-guided polyhedral agglomeration," *ACM Transactions on Graphics*, vol. 36, no. 4, Jul. 2017. [Online]. Available: https://doi.org/10.1145/3072959.3073676

[5] D. R. Bukenberger, M. Tarini, and H. P. A. Lensch, "At-most-hexa meshes," *Computer Graphics Forum*, vol. 41, no. 1, pp. 7–28, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14393

[6] Ansys, Inc. (2025) Ansys engineering simulation software. Online: Available: https://www.ansys.com/.

[7] R. Anderson, J. Andrej, A. Barker, J. Bramwell, J.-S. Camier, J. Cerveny, V. Dobrev, Y. Dudouit, A. Fisher, T. Kolev, W. Pazner, M. Stowell, V. Tomov, I. Akkerman, J. Dahm, D. Medina, and S. Zampini, "Mfem: A modular finite element methods library," *Computers & Mathematics with Applications*, vol. 81, pp. 42–74, 2021, development and Application of Open-source Software for Problems with Numerical PDEs. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0898122120302583

[8] R. Zhang, K. P. Lam, and Y. Zhang, "Conformal adaptive hexahedral-dominant mesh generation for cfd simulation in architectural design applications," in *Proceedings of the 2011 Winter Simulation Conference (WSC)*, 2011, pp. 928–942.

[9] S. Yamakawa and K. Shimada, "Increasing the number and volume of hexahedral and prism elements in a hex-dominant mesh by topological transformations." in *International Meshing Roundtable*, 2003, pp. 403–413.

[10] N. Sawant, S. Yamakawa, K. Shimada, and S. Singh, "Automatic hex-dominant mesh generation for complex flow configurations," *SAE International Journal of Engines*, vol. 11, no. 6, pp. 615–624, 2018. [Online]. Available: https://www.jstor.org/stable/26649119

[11] A. Ruggiero, R. D'Amato, and S. Affatato, "Comparison of meshing strategies in thr finite element modelling," *Materials*, vol. 12, no. 14, p. 2332, 2019.

[12] DHCAE Tools. (2025) Castnet: Gui-environment for openfoam. Online: Available: https://dhcae-tools.com/CastNet.html.

[13] H. Brückler and M. Campen, "Collapsing embedded cell complexes for safer hexahedral meshing," *ACM Transactions on Graphics*, vol. 42, no. 6, Dec. 2023. [Online]. Available: https://doi.org/10.1145/3618384

[14] X. Gao, Z. Deng, and G. Chen, "Hexahedral mesh re-parameterization from aligned base-complex," *ACM Transactions on Graphics*, vol. 34, no. 4, Jul. 2015. [Online]. Available: https://doi.org/10.1145/2766941

[15] K. Xu and G. Chen, "Hexahedral mesh structure visualization and evaluation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 1173–1182, 2019.

[16] G. Xu, R. Ling, Y. J. Zhang, Z. Xiao, Z. Ji, and T. Rabczuk, "Singularity structure simplification of hexahedral meshes via weighted ranking," *Computer-Aided Design*, vol. 130, p. 102946, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010448520301391

[17] L. Si, H. Cao, and G. Chen, "Hybrid base complex: Extract and visualize structure of hex-dominant meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 31, no. 3, p. 1818–1829, Mar. 2025. [Online]. Available: https://doi.org/10.1109/TVCG.2024.3372333

[18] X. Gao, T. Martin, S. Deng, E. Cohen, Z. Deng, and G. Chen, "Structured volume decomposition via generalized sweeping," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 7, p. 1899–1911, Jul. 2016. [Online]. Available: https://doi.org/10.1109/TVCG.2015.2473835

[19] H. Wu, S. Gao, R. Wang, and M. Ding, "A global approach to multi-axis swept mesh generation," *Procedia Engineering*, vol. 203, pp. 414–426, 2017, 26th International Meshing Roundtable, IMR26, 18-21 September 2017, Barcelona, Spain. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877705817343801

[20] S. Zhang, G. Xu, H. Wu, R. Gu, L. Qi, and Y. Pang, "Medial hex-meshing: high-quality all-hexahedral mesh generation based on medial mesh," *Eng. with Comput.*, vol. 40, no. 4, p. 2537–2557, Jan. 2024. [Online]. Available: https://doi.org/10.1007/s00366-023-01925-5

[21] X. Gao, H. Shen, and D. Panozzo, "Feature preserving octree-based hexahedral meshing," *Computer Graphics Forum*, vol. 38, no. 5, pp. 135–149, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13795

[22] L. Pitzalis, M. Livesu, G. Cherchi, E. Gobbetti, and R. Scateni, "Generalized adaptive refinement for grid-based hexahedral meshing," *ACM Transactions on Graphics*, vol. 40, no. 6, Dec. 2021. [Online]. Available: https://doi.org/10.1145/3478513.3480508

[23] L. Maréchal, "Advances in octree-based all-hexahedral mesh generation: Handling sharp features," in *Proceedings of the 18th International Meshing Roundtable*, B. W. Clark, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 65–84.

[24] J. Gregson, A. Sheffer, and E. Zhang, "All-hex mesh generation via volumetric polycube deformation," *Computer Graphics Forum*, vol. 30, no. 5, pp. 1407–1416, 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2011.02015.x

[25] M. Livesu, N. Vining, A. Sheffer, J. Gregson, and R. Scateni, "Polycut: monotone graph-cuts for polycube base-complex construction," *ACM Transactions on Graphics*, vol. 32, no. 6, Nov. 2013. [Online]. Available: https://doi.org/10.1145/2508363.2508388

[26] J. Huang, T. Jiang, Z. Shi, Y. Tong, H. Bao, and M. Desbrun, "L1-based construction of polycube maps from complex shapes," *ACM Transactions on Graphics*, vol. 33, no. 3, Jun. 2014. [Online]. Available: https://doi.org/10.1145/2602141

[27] X. Fang, W. Xu, H. Bao, and J. Huang, "All-hex meshing using closed-form induced polycube," *ACM Transactions on Graphics*, vol. 35, no. 4, Jul. 2016. [Online]. Available: https://doi.org/10.1145/2897824.2925957

[28] H.-X. Guo, X. Liu, D.-M. Yan, and Y. Liu, "Cut-enhanced polycube-maps for feature-aware all-hex meshing," *ACM Transactions on Graphics*, vol. 39, no. 4, Aug. 2020. [Online]. Available: https://doi.org/10.1145/3386569.3392378

[29] J. Huang, Y. Tong, H. Wei, and H. Bao, "Boundary aligned smooth 3d cross-frame field," *ACM Transactions on Graphics*, vol. 30, no. 6, p. 1–8, Dec. 2011. [Online]. Available: https://doi.org/10.1145/2070781.2024177

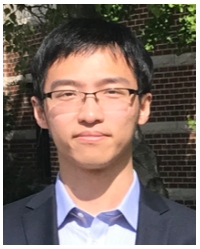
[30] M. Nieser, U. Reitebuch, and K. Polthier, "Cubecover– parameterization of 3d volumes," *Computer Graphics Forum*, vol. 30, no. 5, pp. 1397–1406, 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2011.02014.x

[31] Y. Li, Y. Liu, W. Xu, W. Wang, and B. Guo, "All-hex meshing using singularity-restricted field," *ACM Transactions on Graphics*, vol. 31, no. 6, Nov. 2012. [Online]. Available: https://doi.org/10.1145/2366145.2366196

[32] T. Jiang, J. Huang, Y. Wang, Y. Tong, and H. Bao, "Frame field singularity correctionfor automatic hexahedralization," *IEEE Transactions on*

Visualization and Computer Graphics, vol. 20, no. 8, pp. 1189–1199, 2014.
[33] H. Liu, P. Zhang, E. Chien, J. Solomon, and D. Bommes, "Singularity-constrained octahedral fields for hexahedral meshing," *ACM Transactions on Graphics*, vol. 37, no. 4, Jul. 2018. [Online]. Available: https://doi.org/10.1145/3197517.3201344
[34] L. Li, P. Zhang, D. Smirnov, S. M. Abulnaga, and J. Solomon, "Interactive all-hex meshing via cuboid decomposition," *ACM Transactions on Graphics*, vol. 40, no. 6, Dec. 2021. [Online]. Available: https://doi.org/10.1145/3478513.3480568
[35] J. Xie, J. Xu, Z. Dong, G. Xu, C. Deng, B. Mourrain, and Y. J. Zhang, "Interpolatory catmull-clark volumetric subdivision over unstructured hexahedral meshes for modeling and simulation applications," *Computer Aided Geometric Design*, vol. 80, p. 101867, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167839620300546
[36] H. Brückler, O. Gupta, M. Mandad, and M. Campen, "The 3d motorcycle complex for structured volume decomposition," *Computer Graphics Forum*, vol. 41, no. 2, pp. 221–235, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14470
[37] N. Ray, D. Sokolov, M. Reberol, F. Ledoux, and B. Lévy, "Hex-dominant meshing: Mind the gap!" *Computer-Aided Design*, vol. 102, no. C, p. 94–103, Sep. 2018. [Online]. Available: https://doi.org/10.1016/j.cad.2018.04.012
[38] T. Blacker, "Automated conformal hexahedral meshing constraints, challenges and opportunities," *Engineering With Computers*, vol. 17, pp. 201–210, 10 2001.
[39] S. Meshkat and D. Talmor, "Generating a mixed mesh of hexahedra, pentahedra and tetrahedra from an underlying tetrahedral mesh," *International Journal for Numerical Methods in Engineering*, vol. 49, no. 1-2, pp. 17–30, 2000. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/1097-0207%2820000910/20%2949%3A1/2%3C17%3A%3AAID-NME920%3E3.0.CO%3B2-U
[40] S. J. Owen, "Hex-dominant mesh generation using 3d constrained triangulation," *Computer-Aided Design*, vol. 33, no. 3, pp. 211–220, 2001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010448500001214
[41] J. Pellerin, A. Johnen, and J.-F. Remacle, "Identifying combinations of tetrahedra into hexahedra: a vertex based strategy," *Procedia Engineering*, vol. 203, pp. 2–13, 2017, 26th International Meshing Roundtable, IMR26, 18-21 September 2017, Barcelona, Spain. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877705817343370
[42] D. Sokolov, N. Ray, L. Untereiner, and B. Lévy, "Hexahedral-dominant meshing," *ACM Transactions on Graphics*, vol. 35, no. 5, Jun. 2016. [Online]. Available: https://doi.org/10.1145/2930662
[43] B. Lévy and Y. Liu, "Lp centroidal voronoi tessellation and its applications," *ACM Transactions on Graphics*, vol. 29, no. 4, Jul. 2010. [Online]. Available: https://doi.org/10.1145/1778765.1778856
[44] Y. Yu, J. G. Liu, and Y. J. Zhang, *HexDom: Polycube-Based Hexahedral-Dominant Mesh Generation*. Cham: Springer International Publishing, 2022, pp. 137–155. [Online]. Available: https://doi.org/10.1007/978-3-030-92540-6_7
[45] S. Yamakawa and K. Shimada, "Hexhoop: Modular templates for converting a hex-dominant mesh to an all-hex mesh," *Engineering with Computers*, vol. 18, pp. 211–228, 10 2002.
[46] X. Gao, D. Panozzo, W. Wang, Z. Deng, and G. Chen, "Robust structure simplification for hex re-meshing," *ACM Transactions on Graphics*, vol. 36, no. 6, Nov. 2017. [Online]. Available: https://doi.org/10.1145/3130800.3130848
[47] M. N. Akram, G. Chen, and Y. Zhang, "Impact of hex-mesh structure to simulation quality – a first study," in *28th International Meshing Roundtable, Research Abstract*, 2019.
[48] Q. Zhao, G. Xu, Z. Xiao, H. Wu, R. Gu, Y. Liu, and Y. Pang, "Bc-hexmatching: an improved hexahedral mesh matching approach based on base-complex structure," *Eng. with Comput.*, vol. 40, no. 4, p. 2209–2226, Oct. 2023. [Online]. Available: https://doi.org/10.1007/s00366-023-01908-6
[49] J. Gao, Z. Xiao, S. Shen, C. Xu, J. Cai, and G. Xu, "Improved hexahedral mesh generation from quadrilateral surface meshes," *Comput. Struct.*, vol. 307, no. C, Jan. 2025. [Online]. Available: https://doi.org/10.1016/j.compstruc.2024.107620
[50] H. Brückler, D. Bommes, and M. Campen, "Volume parametrization quantization for hexahedral meshing," *ACM Transactions on Graphics*, vol. 41, no. 4, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530123
[51] N. Schertler, D. Panozzo, S. Gumhold, and M. Tarini, "Generalized motorcycle graphs for imperfect quad-dominant meshes," *ACM Transactions on Graphics*, vol. 37, no. 4, Jul. 2018. [Online]. Available: https://doi.org/10.1145/3197517.3201389
[52] P. M. Knupp, C. Ernst, D. C. Thompson, C. Stimpson, and P. P. Pebay, "The verdict geometric quality library." Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA ..., Tech. Rep., 2006.
[53] X. Que, J. Huang, J. Ralph, J. Zhang, A. Prabhu, S. Morrison, R. Hazen, and X. Ma, "Using adjacency matrix to explore remarkable associations in big and small mineral data," *Geoscience Frontiers*, vol. 15, no. 5, p. 101823, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1674987124000471
[54] M. Behrisch, B. Bach, N. Henry Riche, T. Schreck, and J.-D. Fekete, "Matrix reordering methods for table and network visualization," *Computer Graphics Forum*, vol. 35, no. 3, pp. 693–716, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.12935
[55] OpenFOAM. (2025) Openfoam: The open source cfd toolbox. Online: Available: https://www.openfoam.com/.

Lei Si is a Ph.D. in Computer Science. He received a Ph.D degree in Computer Science from the University of Houston in 2025, an M.S. degree with honors in Computer Science from the University of Illinois at Springfield in 2020, and B.E. in Cybersecurity at North China University of Technology in 2018. His research focuses on geometric modeling, visualization, physically-based simulation, computer vision, cyber security, and artificial intelligence.



Qixin Deng is Anne Korb Shane and David N. Shane Professor in the Liberal Arts and BKT Assistant Professor of Computer Science at Wabash College. He received a BS degree in electronics and information engineering from Zhengzhou University in 2014, an MSc in electrical and computer engineering from Purdue University in 2017, and a Ph.D. in computer science from the University of Houston in 2023. His research interests include face modeling, animation, 3D reconstruction, and machine learning.

Aobo Jin is an Assistant Professor at the University of Houston-Victoria. He received a BS degree in electrical engineering from Dalian University of Technology in 2011, a Master of Science degree in electrical engineering from the Department of Electrical and Computer Engineering, a Ph.D. degree in computer science from the Department of Computer Science, University of Houston, Texas, in 2016 and 2021 respectively. His research interests include computer graphics, virtual human modeling and animation, and sketch-based modeling.

Guoning Chen is a Professor at the Department of Computer Science at the University of Houston. He earned a Ph.D. degree in Computer Science from Oregon State University in 2009. His research interests include visualization, data analytics, computational topology, geometric modeling, geometry processing, and physically-based simulation. He is a senior member of IEEE and a member of ACM.