Tusqh: Topological Control of Volume-Fraction Meshes Near Small Features and Dirty Geometry

Brian Shawcroft[*†] Kendrick M. Shepherd[*] Scott Mitchell[†]

Abstract

This work develops a framework to create meshes with user-specified homology from potentially dirty geometry by coupling background grids, persistent homology, and a generalization of volume fractions. For a mesh with fixed grid size, the topology of the output mesh changes predictably and monotonically as its volume-fraction threshold decreases. Topological anti-aliasing methods are introduced to resolve pinch points and disconnected regions that are artifacts of user choice of grid size and orientation, making the output meshes suitable for downstream processes including analysis. The methodology is demonstrated on geographical, mechanical, and graphics models in 2D and 3D using a custom-made software called Tusqh. The work demonstrates that the proposed framework is viable for generating meshes on topologically invalid geometries and for automatic defeaturing of small geometric artifacts. Finally, the work shows that although subdividing the background grid frequently improves the topological and geometrical fidelity of the output mesh, there are simple 2D examples for which the topology does not converge under refinement for volume-fraction codes.

1 Introduction

Getting geometry that is suitable for mesh generation is often more difficult and time consuming than creating a mesh from that geometry. "Ugly" geometry is ubiquitous "in the wild." Industrial and commercial CAD data are often hand-designed "blueprints" to guide assembly, and do not represent the as-built part. Data from imaging and segmentation may have topological inconsistencies. Even in cases with valid geometry and topology, analysts must carefully review, modify, and defeature models based on the intended purpose of the mesh because typical techniques generate meshes whose topology and geometry match the input models. Common issues in "ugly" geometry are gaps and overlaps, features smaller than the desired mesh size, topological complexities such as small holes, and small angles and thin regions that would produce poor-quality elements.

However, the mesh is a discrete approximation to the geometry. Why require a higher fidelity in the input than is aspired to in the output? Indeed, the community is developing tools to mesh ugly geometry, robustly producing meshes that are topologically correct and have high-quality elements despite topological and geometrical defects and small features in the input.

Sculpt [22, 24, 25] is one such tool, achieving a hexahedral mesh of reasonable quality, but reconstructing an approximation of the input geometry and topology. Inexact reconstruction is a *benefit* in the case of gaps, overlaps, and small features. The Sculpt algorithm starts with a background grid overlaying the input geometry. The fraction of each grid cell that lies inside the geometry of an input material is its *volume fraction.* Cells with volume fractions above a threshold (e.g., one-half) are retained; the rest are discarded. Heuristics remove undesirable topology such as pinch points and components consisting of only a few cells. Retained cells are then snapped to the geometry, and mesh quality is achieved through pillowing [18, 30, 31], smoothing [15], and other changes to mesh topology and node positions. Similarly, Morph [20, 29] is a parallel tet mesher using a background grid that snaps nodes to geometry based on dimension, proximity, and how other nodes are snapped. When no suitable node snap is found, Morph adds new nodes at the intersections with the geometry to produce nodes on the geometry boundary. In both Sculpt and Morph, the size of the background grid indirectly determines the geometric fidelity of the output to the input. TetWild [10, 11] uses a Delaunay triangulation rather than a background grid. Triangles representing the geometry are incrementally inserted and the mesh is refined. Edge length and geometric proximity parameters control the mesh resolution and geometric fidelity.

Though these tools always produce meshes with valid topology, there is no a priori knowledge of what the homology of the output will be, nor how it will compare to the input topology or the desired topology. For some downstream operations, small holes and features play a critical role in final results, while for others, these same holes and features are extraneous, may lead to overly dense meshes, and must be (manually) removed.

[*]Brigham Young University Department of Civil and Construction Engineering

[†]Sandia National Laboratories, Center for Computing Research.

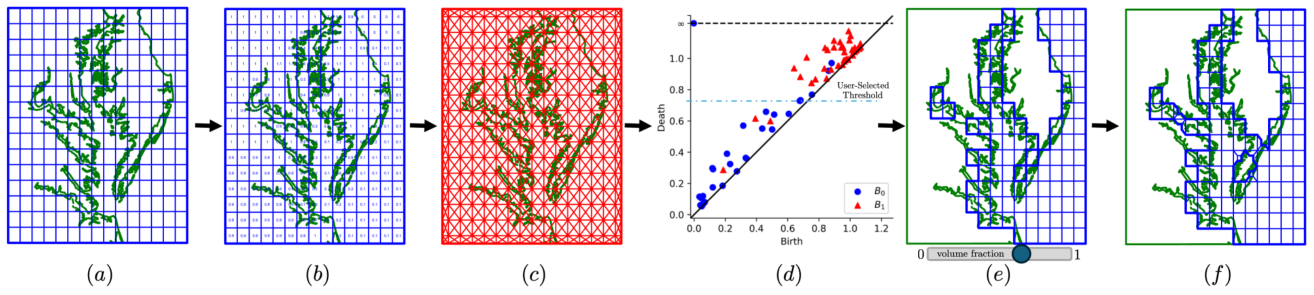


Figure 1: The key steps of Tusqh are illustrated over an example of the Chesapeake Bay. First (a), a background grid is prescribed, after which volume fractions are calculated (b). Next, a special subdivision of the background grid is computed (c), which transfers volumetric data for persistent homology computations (d). The user selects the desired mesh topology from (d), and finally the base (e) and anti-aliasing (f) mesh elements are generated.

In sum, a single model may require multiple representations depending on its intended purpose, each with varying mesh sizes and topological needs. However, tools with control over how to select the appropriate topology of the mesh for its intended purpose are in short supply, meaning that much of this work is deferred to time-consuming manual manipulation by engineers.

Herein, we explore how to robustly predict and achieve the desired mesh topology for algorithms based on background grids and volume fractions (including Sculpt) through the use of persistent homology and generalized winding numbers.

To accomplish this goal, Tusqh—a prototype mesher now available on GitHub [27]—was developed in Rhinoceros 3D. It is a testbed for research and demonstrates that our techniques are effective. Tusqh mimics the initial steps of Sculpt, using a background grid and volume fractions to decide which grid cells to retain. As with TetWild, it uses generalized winding numbers [2, 12] to define the "interior" for valid and invalid geometries. We explore the topological structure of potential meshes under different volume-fraction thresholds using persistent homology [7, 21]. Local connectivity decisions based on sub-sampling volume-fractions mitigate the effects of the arbitrary orientation and offset of the background grid. This enables the analyst to measure and select the desired mesh topology, which then informs which volume-fraction threshold to select. The user may also adjust the volume-fraction threshold on a sliding scale and visualize the choice of meshes. These meshes can serve as input for subsequent steps to improve geometric fidelity, such as Sculpt's snapping, pillowing, and smoothing. A schematic illustrating the key steps in the Tusqh framework is shown in fig. 1. Concluding theoretical results demonstrate that obvious applications of grid-based volume-fraction methods cannot guarantee consistent topological output.

2 Background Material

The proposed method, which selects which cells of a background grid to retain, is related to the computer graphics problem of rasterization [14]. Consequently, we first introduce background information about rasterization and anti-aliasing, after which fundamentals about tools employed in this work are introduced, including homology, persistent homology, and winding numbers.

Rasterization. *Rasterization* is the process of converting an arbitrary geometry into a grid-based representation. In traditional computer graphics, the background grid is screen pixels, and the objects are triangles embedded in floating point $\mathbb{R}^2$. The problem is to select which color and intensity to display in each pixel. *Aliasing* [5, 17] is a significant problem in rasterization: pixel values are sensitive to the offset, rotation, and size of the objects, as well as which locations within a pixel are sampled. Consider a non axis-aligned edge shared by a blue and a red triangle. For a given pixel, if we choose red or blue we get increased contrast but also stair-step patterns called "jaggies." If we choose a purple mixture the image appears smoother, but shading can produce Moiré patterns. Both patterns are glaring to human eyes. For small triangles, the pixel topology may not match the triangle topology; see fig. 2. Such topological errors may be visually insignificant, but they can lead to significant errors in simulation results.

Topological Anti-aliasing. As in graphics, volume-fraction meshing is sensitive to the offset and rotation of objects, as well as the grid size (analogous to pixel density in graphics). An axis-aligned gap is closed or open depending on its size and position relative to the grid; see fig. 3. A gap smaller than half the grid size is always closed. A gap larger than the grid size is always open. Between these, shifting the grid left will cause the mesh to alternate between closed and open.

In figs. 4 and 5 we see the aliasing effects of

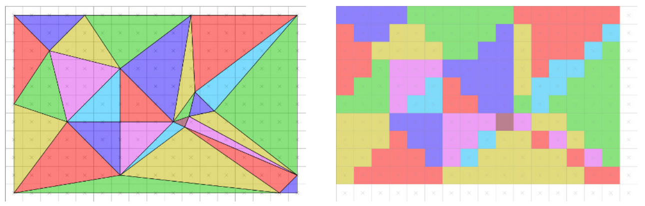


Figure 2: Rasterization of triangles into pixels for computer graphics. Note the pinches from the two left cyan triangles, the archipelago from the lower right pink triangle, and the multitude of additional topological errors in the lower right. Image courtesy Wikipedia `https://en.wikipedia.org/wiki/Rasterisation`

rotations, where the feature is not aligned with the grid. In fig. 4 the gap is a fixed width, but about the size of the grid cells, leading to the gap being inconsistently resolved as open or closed, with separate sides connected by pinch points. In fig. 5 we see a similar inconsistency, but exacerbated because the gap width varies. The boundary lines meet at a sharp angle, so fewer cells are retained as the apex is approached, leading to a chain of small disjoint mesh islands which we call an *archipelago*.

To address these undesirable local topological features, a topological anti-aliasing method is defined and coupled with introducing/removing various templated cells, as described in section 3. The first undesirable feature is pinches, where exactly two grid cells meet at a vertex with no shared edge, or exactly two 3D grid cells meet at an edge with no shared faces. These must be removed because the mesh is required to be locally connected face-to-face or disjoint. (The complement is also connected face-to-face or disjoint.) All other ways in which a mesh can be non-manifold do not occur, because we form the mesh from the union of some cells of a structured grid. Pinches are either separated into different components by removing small elements, or thickened into meeting face-to-face by adding small elements using data from the anti-aliasing framework. These small elements come from templates that split background grid

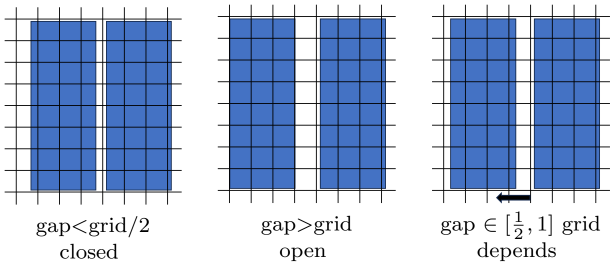


Figure 3: Small grid-aligned gaps are closed, large gaps are open, and intermediate gaps depend on their offset.

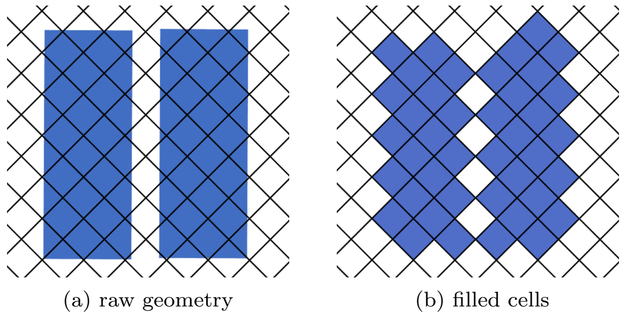


Figure 4: This unaligned gap is resolved inconsistently.

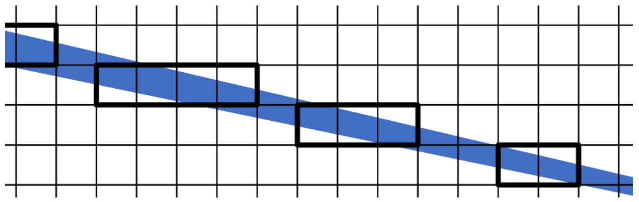


Figure 5: Rotational aliasing may cause stair-step patterns, and archipelagos of isolated islands near where two lines meet at a sharp angle. Bold-outlined cells are filled, thin are open.

cells and perform swaps. The second undesirable feature is archipelagos. Our anti-aliasing technique joins some islands with template elements around connecting edges and quads, and removes any small islands that remain. For comparison, Sculpt resolves pinches by adding or removing entire grid cells, and resolves small components by removing them [23].

For simplicity we only discuss domains with a single material and only discuss the retained cells. In principle our anti-aliasing could be extended and applied to multi-material volume-fraction meshing [31, 23].

Homology. The topology of a mesh should contain the significant features of a domain for its intended computational analysis. Herein, we shall study mesh topology using a cellular complex: nodes are zero-cells, edges are one-cells, faces are two-cells, and volumes are three-cells. Specifically, we will make use of simplicial and cubical complexes, in which two-cells are triangles and quadrilaterals, and three-cells are tetrahedra and hexahedra, respectively. Homology [8] is a mathematical tool that distinguishes cell complexes using certain algebraic quotient groups. The *Betti numbers* B_i count the rank of these groups. Specifically, B_0 equals the number of connected components, B_1 is the number of holes, and B_2 is the number of cavities or voids. For planar domains B_2 will always be zero.

Persistent homology [7, 21] describes homology changes as objects are added and connections are made.

A *filtration* has a "persistence parameter" which defines when a cell enters the complex. A filtration is monotonic, so no cell may ever leave the complex after entering. However, the homology has both additions and removals because adding a cell could, e.g., create a new connected component, or combine two components into one. The parameter value at which a group generator is created is called its "birth," while the value it disappears is called its "death." Birth and death coordinates are plotted in a persistence diagram such as fig. 1 (d). This not only counts Betti numbers, but tracks individual components and holes.

This work studies the persistent homology of cubical background grids using volume fractions as the persistence parameter. (In other work the signed distance to a domain boundary was the parameter [19].) Alternatively, zigzag persistence, which does not require a monotonic filtration [4, 6], could be used, but doing so would increase complexity and computational expense.

Winding Numbers. Volume fractions may be estimated by sampling points and counting the fraction of them inside the geometry. However, for "ugly" geometry, what is "inside" may be poorly defined. The generalized winding number [2, 12] overcomes this obstacle; see fig. 6. It gives answers identical to ray shooting for watertight domains, and gives answers that humans find both reasonable and intuitive for other domains. In its traditional form, the winding number at a point with respect to a closed curve describes the net number of times the curve encircles the point in the counter-clockwise direction, with negative numbers indicating clockwise encirclement. The winding number is the integral of the angle of the ray from the point to the curve as it is traversed. The generalized winding number extends this definition to sets of open curves. It yields a continuous value where 0 indicates outside and 1 is inside. For geometry with gaps, the winding number near a gap is typically between 0 and 1. In extreme cases, such as overlapping domain boundaries, invalid geometries may give values beyond $[0, 1]$. In 3D, the winding number integrates the solid angles seen from a point, e.g., for a volume defined by a triangle soup. Which normal direction is outward-facing determines the sign of the solid-angle contribution.

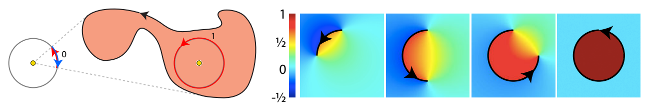


Figure 6: Winding number point and field values, courtesy Jacobson et al. [12] Figures 4 and 6. Used with permission of the Association for Computing Machinery, conveyed through Copyright Clearance Center, Inc.

3 Methodology

Volume Fractions. Herein we study both 2D and 3D domains. We define a regular background grid, e.g. by subdividing an axis-aligned bounding box. This grid is a cubical quadrilateral or hexahedral complex, depending on the domain dimension. The volume fraction of each maximal-dimension cell is computed as the average of the winding numbers of its sample points. Sample points lie in an s^d array, as shown in fig. 8. (Recall we calculate persistent homology based on volume fractions, with the goal that the user may select the volume fraction that achieves their desired mesh topology.)

Volume-Fraction Persistence Parameter. The persistence parameter used herein is the volume-fraction threshold, ordered from 1 down to 0 (i.e. by decreasing value). By defining volume fractions only for cells of maximal dimension, a mesh is defined by including all cells with volume fraction greater than or equal to the chosen threshold, and removing all others. The order that cells are added to the mesh as a function of the persistence parameter is demonstrated on a topologically invalid representation of the Chesapeake Bay in fig. 7. The persistent homology diagram is displayed in fig. 1 (d). These images illustrate significant topological aliasing, which limits the utility of these meshes. In what follows, we aim to mitigate these rasterization effects.

Sub-cell Volume Fractions. To assist in topological anti-aliasing, we also define volume fractions for all lower-dimensional grid cells. For any such n-cell, its sample points are those in a (fictitious) grid cell centered at that n-cell; see fig. 8. We use only even numbers s of sample points because these samples do not lie on cell boundaries. Volume fractions for these lower-dimensional cells are computed as averages of winding numbers associated with sample points contained in the fictitious grid cell, in a manner analogous to cells of maximal dimension. As before, only cells with volume fraction greater than or equal to a prescribed threshold will remain in the cell complex. All others are omitted. Omitted cells are called "exterior" cells, while remaining cells are called "interior" cells. We call this sampling process "*subgrid sampling.*"

Anti-aliasing. To address the undesirable rasterization effects introduced by our background grid (including pinch and archipelago removal), we employ subgrid sampling as an anti-aliasing method.

In 2D, the only possible pinch is two quads meeting at a pinch vertex, whereas in 3D there are 11 possible configurations of pinched edges and vertices. These are shown in fig. 9. To find pinches, we consider each vertex and the neighborhood of cells containing it, i.e.,

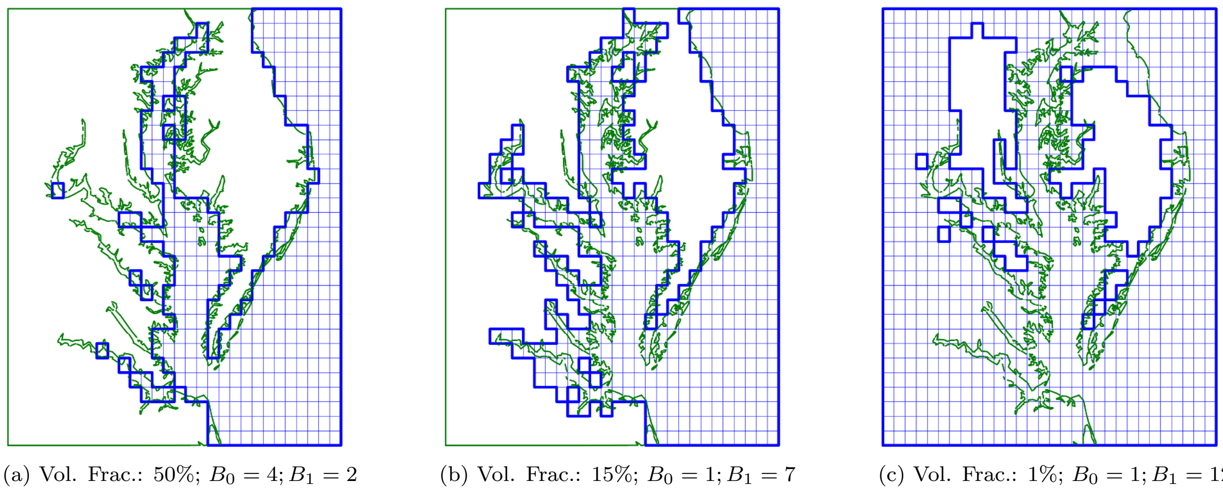

(a) Vol. Frac.: 50%; $B_0 = 4; B_1 = 2$ (b) Vol. Frac.: 15%; $B_0 = 1; B_1 = 7$ (c) Vol. Frac.: 1%; $B_0 = 1; B_1 = 12$

Figure 7: Chesapeake Bay meshes and their Betti numbers change as the volume-fraction threshold is lowered. The model contains deliberate errors: gaps, overlaps, and offsets. As a result of these errors, the winding numbers of sampled points may be positive in regions that are clearly inland, as in the lower-left region of the rightmost figure. Also note how rasterization affects the thin bays and peninsulas, where here we have skipped anti-aliasing.

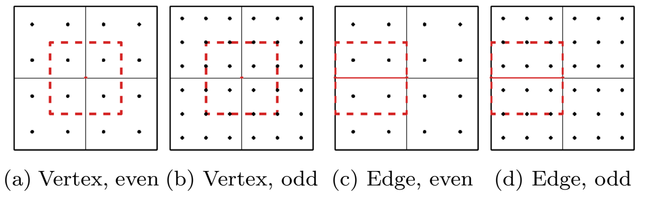

(a) Vertex, even (b) Vertex, odd (c) Edge, even (d) Edge, odd

Figure 8: Sample $s \times s$ arrays are shown. Samples contained by the red dashed lines define vertex and edge volume fractions in 2D. We use only even sample arrays.

2×2 quads in 2D and $2 \times 2 \times 2$ hexes in 3D. If the neighborhood corresponds to a pinch case, the pinch vertices and edges are queued. Each pinch in the queue is processed in a way that is compatible with processing nearby pinches. The pinches are connected if the subcells (vertices or edges) are interior, and disconnected if they are exterior. Each pinch is repaired by splitting cells (either mesh cells or their complement) using predefined splits, and discarding or adding some of the split cells. The splits are shown in fig. 10. In 2D, the template is a one-to-five split. In 3D, pinch edges are repaired before vertices. The edge-repair template is a one-to-seven split. For pinch vertices, we follow with a two-to-six split of any pairs of hexes from two different one-to-seven splits that share a face; see fig. 10c.

To separate cells, splits are performed on the cells of the mesh itself, and child cells that contain pinch vertices or edges are removed, as shown in fig. 11. To connect cells, splits are performed on the complement of the mesh, and child cells that contain pinches are added to the mesh, as illustrated in fig. 12. A single mesh can use both separations and connections in different regions. However, we require that all adjacent pinches must be resolved in the same way to ensure validity of the resulting mesh. Two sets of pinches separated by cells without pinches can be resolved in opposite ways. Our rules occasionally indicate that adjacent pinches should be resolved in opposite ways. We pre-select whether we connect or separate these cases, see fig. 13.

When separating pinches, the configuration in fig. 9d is the only exception to the rule of removing all the child cells that share the pinch edge. The one-to-seven split is performed on the hexahedra sharing the pinch edge, and all the child hexahedra that contain that edge are removed. This turns the central vertex into a pinch. To prevent that, one additional child hexahedron is removed. In the orientation of fig. 9d, the removed hex is the rightmost child of the top hex, which contains the central vertex; see fig. 11d.

For resolving archipelagos, all the connected components of the mesh are identified. For any pair of connected components, if the edges that connect them are interior to the geometry, the components are joined using templates along those edges. The remaining connected components that contain fewer than a user-defined number of highest-dimensional cells are removed. This process is demonstrated in fig. 14. In figs. 15 and 16 the utility of the anti-aliasing algorithm is demonstrated on the unaligned gap and sharp angle

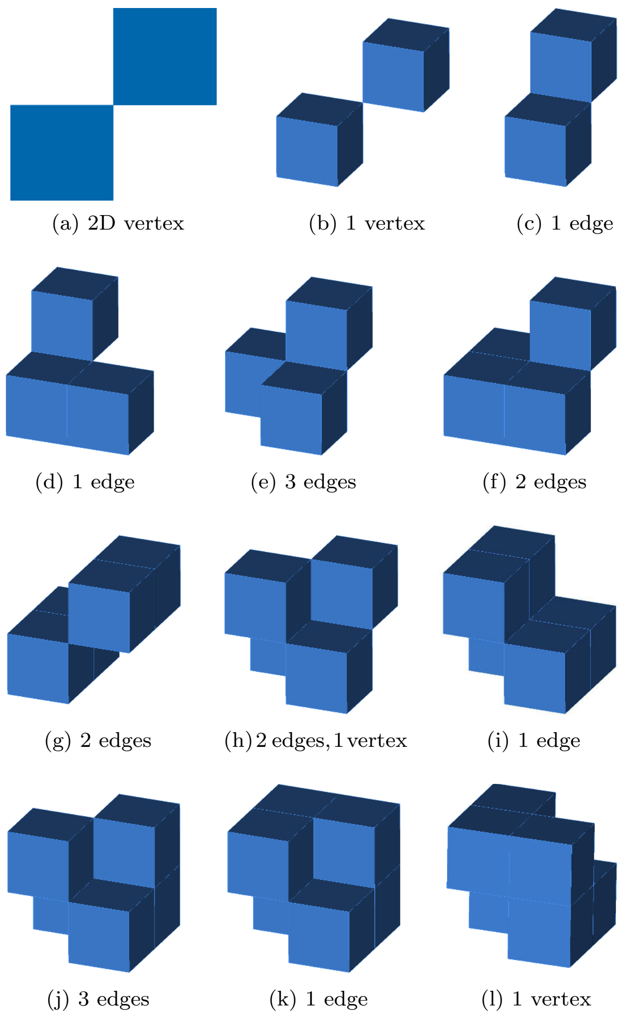

(a) 2D vertex (b) 1 vertex (c) 1 edge
(d) 1 edge (e) 3 edges (f) 2 edges
(g) 2 edges (h) 2 edges, 1 vertex (i) 1 edge
(j) 3 edges (k) 1 edge (l) 1 vertex

Figure 9: All possible pinches in 2D and 3D are shown.

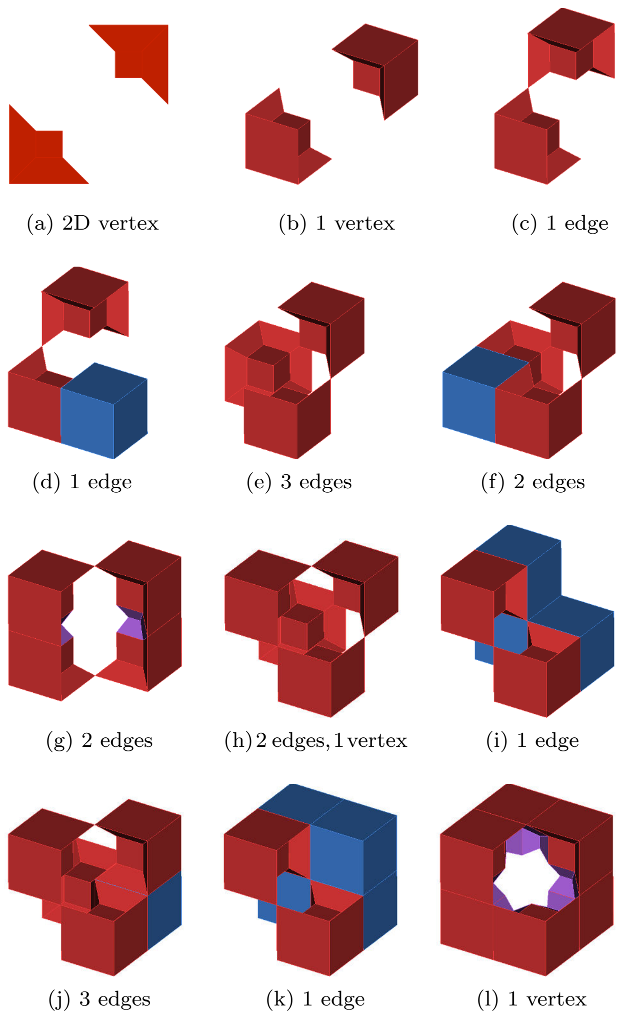

(a) 2D vertex (b) 1 vertex (c) 1 edge
(d) 1 edge (e) 3 edges (f) 2 edges
(g) 2 edges (h) 2 edges, 1 vertex (i) 1 edge
(j) 3 edges (k) 1 edge (l) 1 vertex

Figure 11: Pinch shrinking templates are shown. Pinches on the boundary of the neighborhoods (non-centered vertices) are resolved by neighborhoods centered on them.

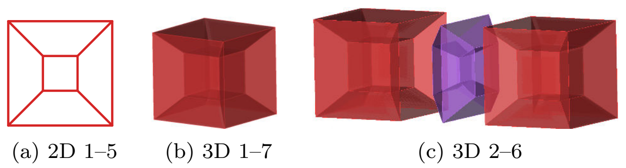

(a) 2D 1–5 (b) 3D 1–7 (c) 3D 2–6

Figure 10: Templates for fixing pinches are depicted.

of figs. 4 and 5 respectively.

Transferring Persistent Parameters to Simplices. Having a framework by which topological anti-aliasing can be performed on the mesh by use of subgrid sampling and templates, we now turn our attention to ensuring that the topological anti-aliasing defined above is accurately represented in persistent homology calculations. Because most open-source persistent homology software employs simplicial complexes, we first transform the cubical filtration into a topologically-equivalent simplicial filtration. In what follows, we prioritize consistent pinch resolution, over archipelago resolution. As a result, we make the assumption in 2D that an edge shared by two interior quadrilaterals will also be interior, and that an edge shared by two interior vertices must also be interior. Similarly, in 3D, a face shared by two interior hexahedra will be interior, as will a face bounded by four interior edges and vertices. We first focus on the 2D framework, followed by 3D.

In 2D, a primal vertex induces a dual 2-cell, and a primal edge induces a dual edge, and a primal quad induces a dual vertex. Each dual vertex is assigned the filtration value of its corresponding primal face's volume fraction. The dual mesh is then further subdivided into

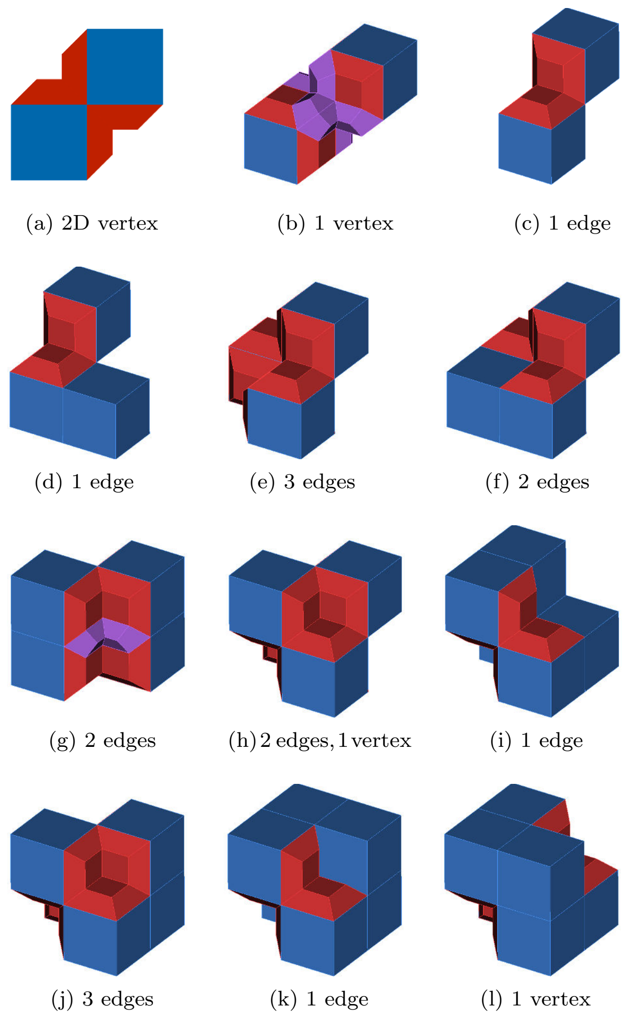


Figure 12: Pinch growing templates are shown.

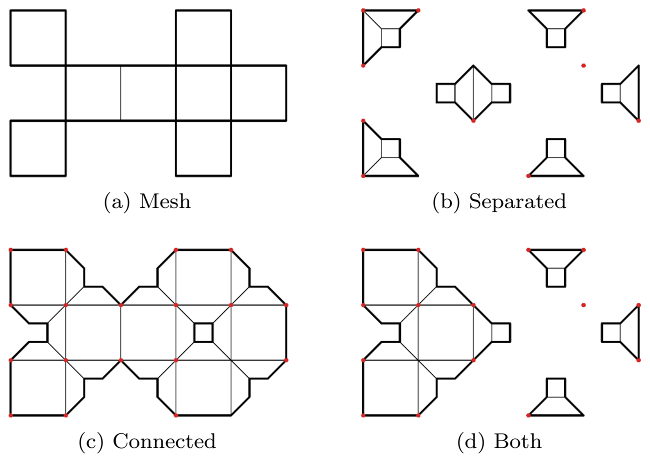


Figure 13: Adjacent pinches must be resolved the same way. Here the resolved mesh is shown for different configurations of “inside” vertices (red).

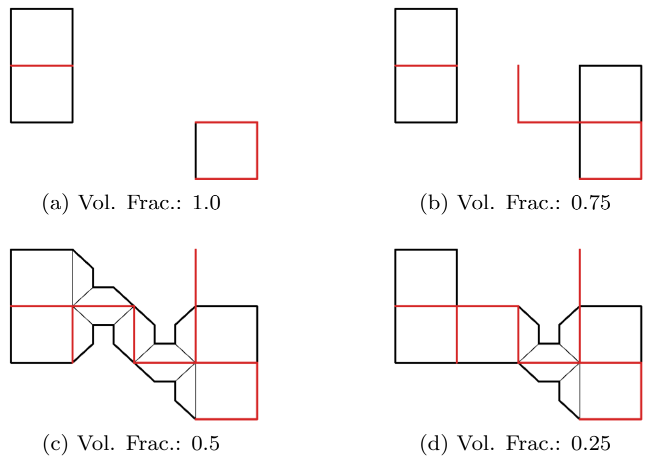
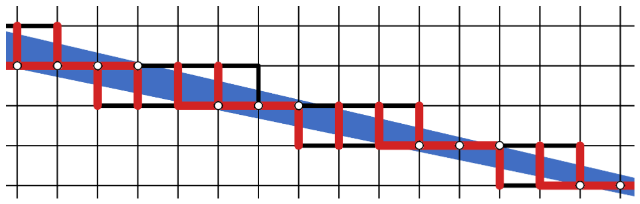


Figure 14: Faces are connected using templates when “inside” edges (red) create a continuous bridge between components.

a simplicial mesh. To create the simplicial mesh, an additional simplicial vertex is introduced at the centroid of each dual 2-cell. Simplices are then formed as the join of each dual face’s edge with the simplicial centroid vertex. This simplicial vertex corresponds to a vertex on the primal mesh, and takes the filtration value of the corresponding primal vertex’s volume fraction. However, to preclude the introduction of spurious topological artifacts (and consistent with previous computations), we also require that this volume fraction be between the maximal and minimal volume fraction of the surrounding four vertices. The filtration value of this simplicial vertex then informs whether two primal faces connected with a pinch should be topologically separated or connected. Having thus defined filtration values for all vertices of the induced simplicial complex, we then use a Vietoris–Rips complex to calculate persistent homology, meaning that at persistence value $k \in \mathbb{R}$, all vertices with persistence parameter value greater than or equal to k are added to the filtration, then all edges between already-added vertices, then all triangles formed by already-added edges.

In 3D, a primal vertex induces a dual volume cell, and a primal edge induces a dual face, a primal face induces a dual edge, and a primal volume induces a dual vertex. As in 2D, each dual vertex is assigned the filtration value of its primal volume. To generate a simplicial mesh from the dual mesh, each dual face is subdivided into four triangles with a new vertex introduced, as in 2D. Here, the additional vertex corresponds to a primal edge and will take the filtration value of the volume fraction of this primal edge, subject to constraints


(a) Subcells

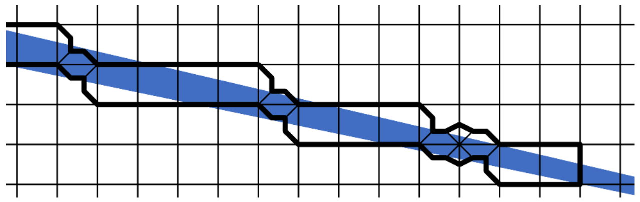
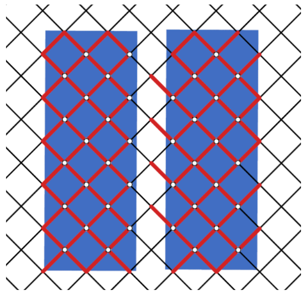

(b) Anti-aliased

Figure 15: Subgrid sampling and anti-aliasing performed on fig. 5.


(a) Subcells

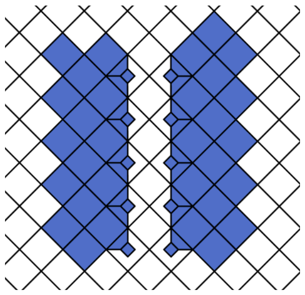

(b) Anti-aliased

Figure 16: Subgrid sampling and anti-aliasing are performed on fig. 4.

keeping the filtration value between the maximal and minimal values of the surrounding four simplices of the dual face. Each dual volume is then subdivided into 24 tetrahedra by introducing a single simplicial vertex at the centroid of the dual volume and taking the join of this vertex with each of the 24 triangles defined on the (subdivided) faces of the dual volume. Again, this new simplicial vertex will correspond to a vertex on the primal mesh, and consequently takes a filtration value of the volume fraction of this primal mesh vertex (again subject to the constraint that the filtration value must be between the maximal and minimal values of the 26 surrounding vertices). The filtration values of the simplicial vertices corresponding to primal vertices and primal edges is then locally consistent with the procedures performed resolve pinch points, and will have identical persistent homology. This conversion process, from a cubical primal cell complex into a simplicial one with an identical filtration is illustrated in figs. 17 and 18.

For the sake of completeness, we also note that similar primal-to-dual-to-simplicial operations could be performed on unstructured background meshes. In the 2D case, each dual cell of maximal dimension and with k sides would be subdivided into k triangles. In 3D, each dual cell of maximal dimension and with ℓ faces, with the ith face having k_i sides, would subdivide into $\sum_{i=1}^{\ell} k_i$ tetrahedra.

Finally, we note that for a truly general framework, a vertex in the above-defined simplicial complexes would need to be defined for each cell in the primal complex. Particularly, in 2D we currently introduce vertices corresponding to primal faces and vertices, but not for primal edges. This would require splitting every dual face into 8 simplices, rather than 4. In 3D, we introduce vertices for all cells except for primal faces, and would require splitting every dual volume into 48 tetrahedra, rather than 24. Given the challenges of navigating this topological space in a meaningful way (as well as the additional computational expense incurred by such a navigation), we leave this for future work.

4 Results

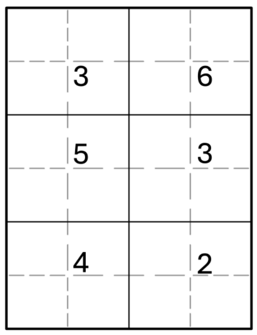
4.1 Computational Results. Tusqh was developed and evaluated using a custom plugin to Rhinoceros 3D and Grasshopper. Winding numbers were computed using libigl [13]. Persistent homology was computed using Aleph [26], which is based on PHAT [3].

The framework is tested on a 2D model of Chesapeake Bay[1] with boundary errors, a 3D model of mechanical bearings [2], and a 3D graphics model of *The Bronco Buster*[3]. Snapshots of meshes given computed volume fractions are shown in figs. 7, 19 and 20. Model errors for the Chesapeake Bay include overlapping edges, repeated/offset edges, and numerous gaps. Despite the "interior" of the bay being ill-defined, the proposed method still captures the intended geographic domain with respect to both the continent and to islands. A complete view of the homological structure based on varying the volume fraction for the Chesapeake Bay is shown in fig. 1 (d), while similar persistence diagrams for the volumetric models are shown in fig. 21. Results demonstrate that a mesh with the desired homological structure could be extracted from the background grid by selecting the correct threshold. These figures are primarily for illustrative purposes: we purposely chose a coarse grid size to generate the topological issues we are addressing. In practice, a finer grid

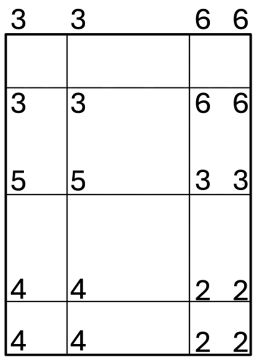
[1]Model derived from `https://vecta.io/symbols/281/ecosystems-maps/93/usa-md-va-chesapeake-bay-line-map`

[2]Model provided at `https://ten-thousand-models.appspot.com/detail.html?file_id=1716283`

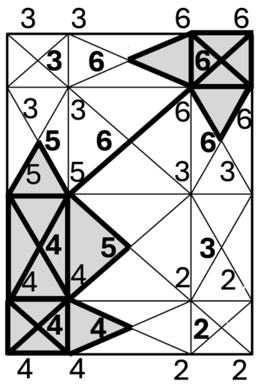
[3]Model provided at `https://tinyurl.com/4cmrptev`



(a) Primal volume fractions



(b) Dual volume fractions


(c) Connected simplices

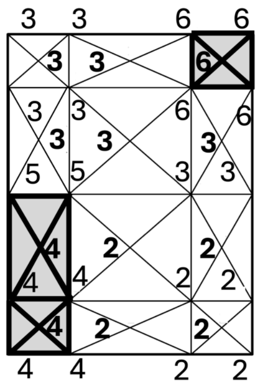

(d) Disconnected simplices

Figure 17: An example of converting 2D cell volume fractions into a filtration of a simplicial complex. Numbers are proportional to volume fraction. Figure 17a shows the ordering of grid cells from high to low. Figure 17b shows the dual grid with vertex values transferred from grid cells. In figs. 17c and 17d the dual complex is subdivided into a simplicial complex and the threshold is set so that all cells with value 4 or more are added. The choice of volume fraction for the introduced vertices (in bold) determines the connectivity near pinches.

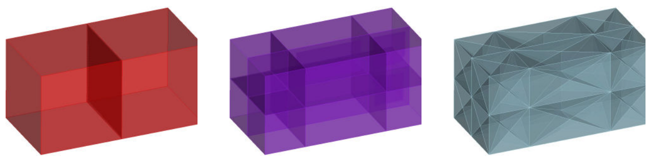
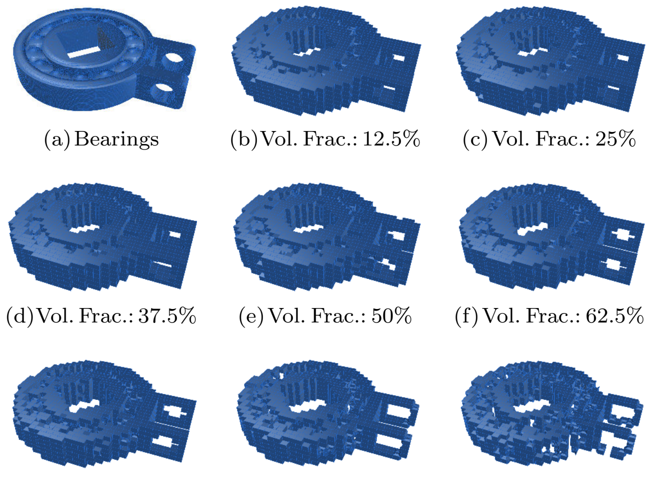


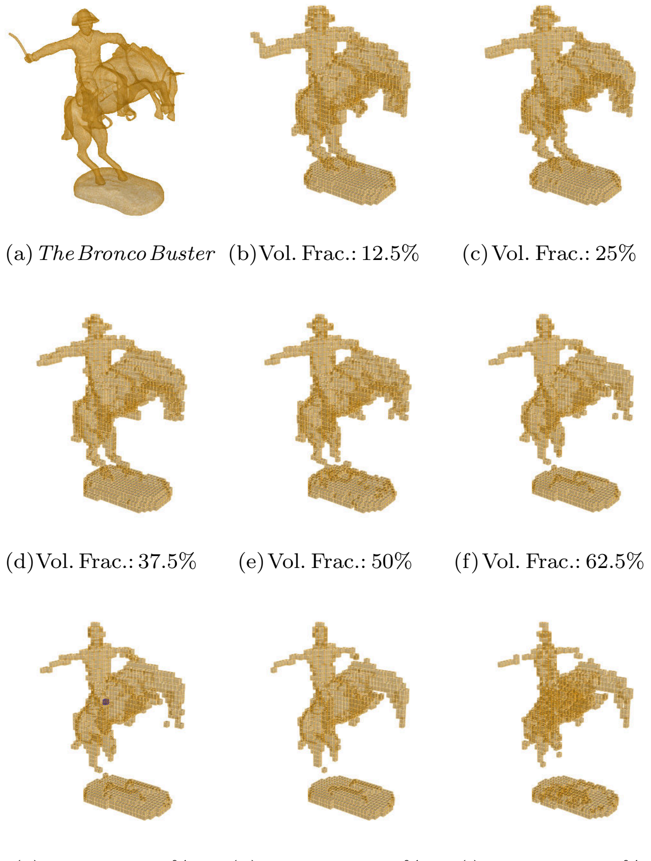
Figure 18: The conversion process taking two adjacent 3D hexes into a tetrahedral simplicial complex with an equivalent filtration is shown.


(a) Bearings (b) Vol. Frac.: 12.5% (c) Vol. Frac.: 25%

(d) Vol. Frac.: 37.5% (e) Vol. Frac.: 50% (f) Vol. Frac.: 62.5%

(g) Vol. Frac.: 75% (h) Vol. Frac.: 87.5% (i) Vol. Frac.: 100%

Figure 19: Meshes of bearings are shown.


(a) *The Bronco Buster* (b) Vol. Frac.: 12.5% (c) Vol. Frac.: 25%

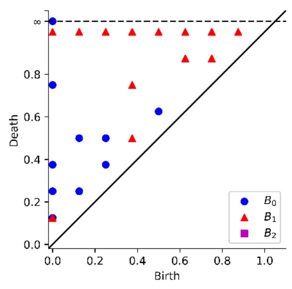
(d) Vol. Frac.: 37.5% (e) Vol. Frac.: 50% (f) Vol. Frac.: 62.5%

(g) Vol. Frac.: 75% (h) Vol. Frac.: 87.5% (i) Vol. Frac.: 100%

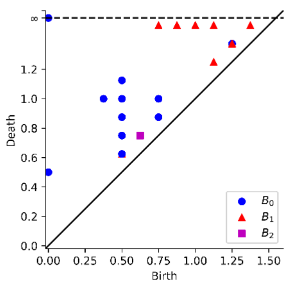
Figure 20: Meshes of *The Bronco Buster* are shown.

would better capture local behavior.

Anti-aliasing. Figure 22 demonstrates the anti-aliasing technique on a mesh of the Chesapeake Bay to resolve both pinches and archipelagos. The anti-aliased mesh is analysis suitable, which is illustrated through a



(a) Bearings



(b) *The Bronco Buster*

Figure 21: Persistence diagrams for the volumetric meshes are shown, with paramerter 1 minus the volume fraction.

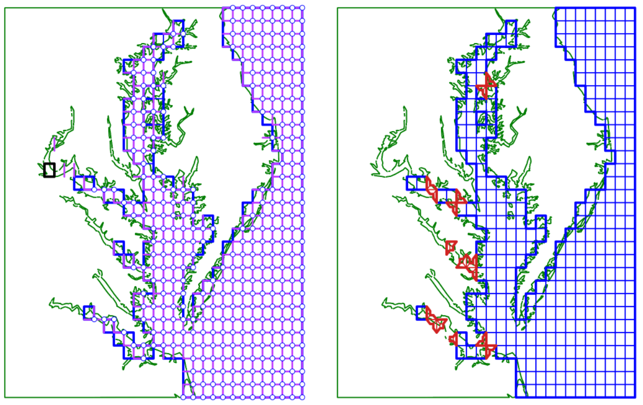
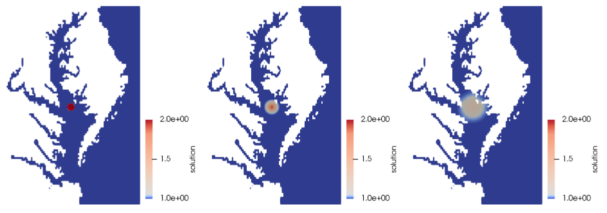


Figure 22: Meshes of Chesapeake bay with subcells and anti-aliasing are shown. Interior vertices and edges are shown as blue outlined circles and purple lines respectively, while removed faces are shown in black (left). Connecting and separating templates are in red (right).

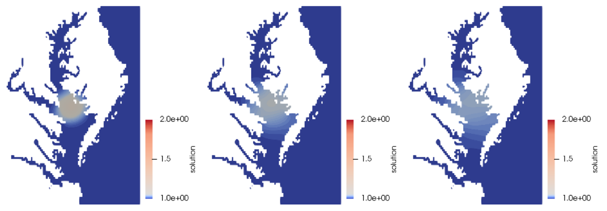
simulation of contaminant propagation, e.g. an oil spill, on a refined version of the Chesapeake Bay in fig. 23. Simulations were performed using MFEM [1, 16].

Comparisons with Alternative Methods. A table comparing Tusqh against alternative meshing methods is presented in table 1. Here, an "X" indicates software capacity, while "(X)" indicates limited capacity. Particularly, TriWild allows for interior holes to be removed, but these must be explicitly specified and must be within appropriate input tolerance bounds to prevent unexpected results. In these regards, Tusqh is competitive with alternative meshing software.

However, Tusqh's capacity to select homology given persistence data makes it more apt to represent desired mesh topology. For instance, when *The Bronco Buster* is immersed in a background grid of 40 × 80 × 80 hexahedral elements, and meshes of comparable element



(a) Time step: 0 (b) Time step: 50 (c) Time step: 100



(d) Time step: 200 (e) Time step: 400 (f) Time step: 800

Figure 23: A finite element solution of contaminant propagation is displayed on a Tusqh mesh of the Chesapeake Bay.

sizes are requested for TetWild, Morph, and Sculpt, only Tusqh successfully defines a mesh of a single connected component, and that only with volume fraction selection of 12.5% (as opposed to alternative volume fractions). However, all other methods presented in table 1 have higher geometric fidelity than Tusqh because of Tusqh's ability to only yield voxel output currently. Furthermore, because Tusqh was developed as a prototype plug-in for Rhinoceros 3D primarily using C# (with a wrapper around portions of libigl's C++ code), it is currently much less competitive computationally.

4.2 Anti-aliasing Guarantees and Limitations.

As noted in Section 2, one of the primary difficulties with volume fraction-based meshing methods is mitigating the effects of rasterization (i.e. choice of orientation and sample size) through topological anti-aliasing. The following theoretical result holds regarding the success of the proposed anti-aliasing methods in mitigating topological rasterization. A proof of the result can be found in the appendix of the preprint version of this document on arXiv [28].

THEOREM 4.1. *Given a rectangular lattice in $\mathbb{R}^2$ with characteristic length ℓ overlaying two parallel half-spaces separated by a length of L, topological rasterization may occur when $\ell(\sqrt{2}-1) < L \leq \ell$. For the subgrid sampling scheme proposed in this text, topological rasterization may only occur when $\ell(\sqrt{2}-1) < L < \frac{\sqrt{2}\ell}{2}$. Finally, topological rasterization due to changes in orientation cannot be resolved for L such that $\ell(\sqrt{2}-1) < L < \frac{\ell}{2}$.*

Table 1: Comparisons Between Meshers Capable of Operating on Dirty Geometry

Method	2D	3D	Type	Invalid Geometry Input	Manifold Output (if Desired)	Homology Control
TriWild [9]	X		Simplicial	X		(X)
TetWild [10, 11]		X	Simplicial	X	X	
Morph [20, 29]		X	Simplicial	X		
Sculpt [22, 24, 25]		X	Cubical	X	X	
Tusqh	X	X	Cubical	X	X	X

4.3 Topological Effects of Grid Refinement. To conclude the results section, we demonstrate that topological pathologies may arise when naively using persistent homology to find an appropriate grid size to achieve a desired topology. When features are isolated or globally the same scale, grid refinement has intuitive and predictable topological effects. However, general inputs may *not* demonstrate monotonic filtration behavior with grid refinement. Counterexamples show non-monotonic filtration behavior by grid size. Discretization by grid cells and their alignment with input features strongly effects topological behavior. Thus algorithm parameters of when to refine the background grid may have unexpected effects on mesh topology.

Convergence. For many inputs, as the grid is refined, topological features of the input are resolved and the output mesh topology becomes stable. However, for some inputs, the topology never converges and no filtration is possible. For some inputs it may be possible to define a filtration, with simplices only appearing, never disappearing. If simplices appear and disappear, zig-zag persistence could computationally predict topology.

In figs. 4 and 5 a feature is inconsistently resolved due to aliasing effects of unaligned grids. For the constant-width gap in fig. 4, refining or coarsening the grid makes the gap resolved consistently as open or closed. However, for the variable-sized gap in fig. 5, global uniform refinement may merely move where the problem occurs. The example is a wedge of material bounded by two lines meeting at a small angle α at an apex. In locations where the grid size is about the same as the local width, whether a cell is included or excluded can change every few grid cells, leading to many separate connected grid components. For any small grid size, there will be some portion of the wedge where the lines are about that size apart, specifically in the range $[\frac{1}{2}, 1] \cot \alpha$ squares away from the apex. The geometry is undesirable because the islands move location. The grid topology may be constant over refinement, but that topology differs from the topology of the underlying object and is undesirable.

Non-convergence. In each of the examples in fig. 24 the output mesh topology does not converge under refinement. That is, there is no grid size below which the output mesh topology does not change. The background grid is uniform, but we only draw some of the relevant cells at each level of refinement. Blue (closure) cells are mostly material and thus included in the output mesh. Red (gap) cells are unfilled and excluded. Under refinement, the meshes alternate between one and two connected components ad infinitum. The grid squares containing the corner alternate between filled and open, because of the corner's relative position inside its square. The descriptions of the geometries are finite, just two triangular blocks. It is simple, plausible, and without sharp angles. The only unusual feature is the blocks touch at a single pinch point.

The upper and lower examples in fig. 24 have alternate sizes of when they are open and closed. If an input has both of these pinch features between two material blocks, then exactly one of the pinches will be closed, giving a mesh with the homology of a disk. It is converged in the sense that the homology does not change under refinement, but the local connectivity does change. Hence, even if we were to use zigzag homology it would not distinguish between this case and a single solid block of material.

The analytic description of the geometries in fig. 24 is two triangular blocks of material with slopes -3 and 1 meeting at a corner. In the upper example, the corner's coordinate is $(\frac{2}{3}, 0)$, and in the lower it is $(\frac{1}{3}, 0)$. If we start with a unit grid with a vertex at the origin, then under refinement the grid square containing the corner alternates between having the corner 2/3 of the way along the bottom edge (blue), and 1/3 of the way (red). Such blue squares have volume fraction 10/18 and the red squares 7/18. This construction is not tight. The slopes may be different. The corner may lie at some other coordinate, and a grid vertex will never lie on it if its x-coordinate is not $k/2^m$ for some $\{k, m\} \in \mathbb{Z}$. Thus more complicated sequences than alternating may be constructed.

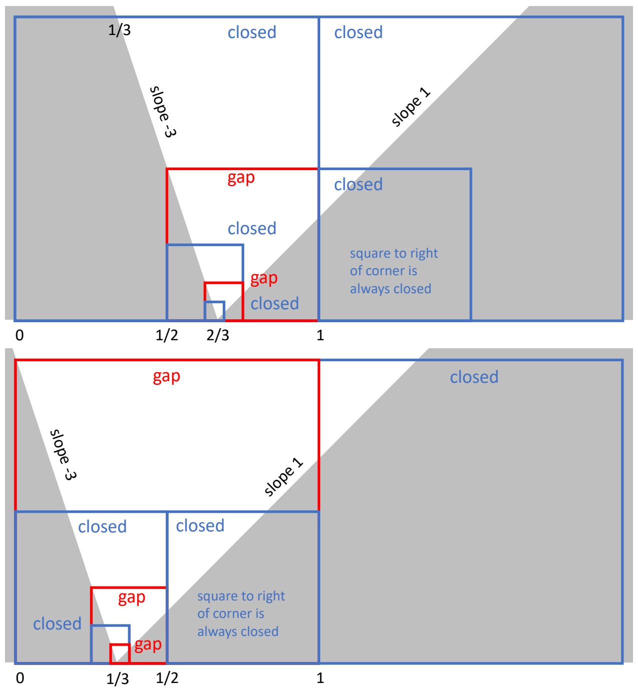


Figure 24: In these counterexamples to convergence, the grid topology alternates between one and two connected components ad infinitum under refinement. The alternations of the upper and lower examples are opposite.

5 Conclusion

In this work, a mesh generation framework is developed to facilitate the creation of meshes on potentially dirty geometry based on user-specified needs through the use of persistent homology. The framework is built on a volume-fraction based meshing method, similar to Sculpt, in which mesh elements are considered “in” or “out” of the given geometry depending on whether they have volume fraction above a cutoff range; the most desirable mesh can then be selected based on the appropriate homological structure induced by this volume fraction. These volume-fraction based algorithms, including Sculpt, behave quite differently than boundary-fitted algorithms, such as Delaunay Refinement, when the local mesh size is decreased. For boundary-fitted algorithms, reducing the mesh size to the local feature size or less allows the mesh to have good quality and recover the input topology exactly. For the family of volume-fraction codes, the mesh quality is good regardless of the local feature size, but in some cases the topology does not converge as the mesh size decreases by subdividing the background grid. However, for a fixed grid, we may predict the resulting mesh topology, measure how that changes as we vary the volume-fraction threshold using persistent homology, and select an appropriate topology for the mesh’s intended purpose. When the background grid is about the same size as gaps and thin-region features, the alignment and orientation of the background grid with respect to the features can create aliasing artifacts in the mesh. There may be spurious pinches and archipelagos. A gap or thin region may be resolved inconsistently in different locations. Subgrid sampling provides guidance on whether to connect, separate, or remove components. These connections and separations can be achieved using templates of small elements. We have demonstrated theoretically and experimentally that subgrid sampling can mitigate the effects of aliasing, making connections more consistent. The software, Tusqh, demonstrates the potential of the meshing framework in both two and three dimensions, and is planned for open source release. As a counterpoint we have theoretical analysis showing that for any volume-fraction threshold we choose, there are cases where aliasing artifacts will still arise.

There are a number of avenues for future work to build on and improve this framework. First, it may be valuable to have adaptive, non-uniform background grids and volume fraction thresholds to mesh more interesting geometries at reduced expense. However, doing this may require the use of zigzag persistence to accurately capture mesh topology, particularly when refinement does not converge on separation or closure of local mesh features. Further work should also better utilize fitting algorithms, such as those present in Sculpt, to more accurately fit to the geometric input data that Tusqh approximates. To do this more flexibly, additional research should generalize these methods to unstructured cubical complexes. Finally, it is anticipated that for this framework to be usable in practice, it would need to be redeveloped using C++.

Acknowledgements

We thank Steve Owen for providing Sculpt and explaining its subtleties to us. We thank Daniele Panozzo for answering our questions about libigl. We thank Alec Jacobson and the other authors of libigl, especially the winding number component, for making it freely available to the community. We thank Darren Engwirda for discussing coastal climate modeling with us. We thank Anjul Patney for pointing out the connection between rasterization and volume fraction meshing.

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR), Applied Mathematics Program. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525.

References

[1] 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, 81 (2021), pp. 42–74.

[2] G. Barill, N. G. Dickson, R. Schmidt, D. I. Levin, and A. Jacobson, *Fast winding numbers for soups and clouds*, ACM Trans. Graph., 37 (2018), pp. 1–12.

[3] U. Bauer, M. Kerber, J. Reininghaus, and H. Wagner, *PHAT – persistent homology algorithms toolbox*, Journal of Symbolic Computation, 78 (2017), pp. 76–90. Algorithms and Software for Computational Topology.

[4] G. Carlsson and V. De Silva, *Zigzag persistence*, Foundations of Computational Mathematics, 10 (2010), pp. 367–405.

[5] F. C. Crow, *The aliasing problem in computer-generated shaded images*, Communications of the ACM, 20 (1977), pp. 799–805.

[6] T. K. Dey and T. Hou, *Computing zigzag vineyard efficiently including expansions and contractions*, in Symposium on Computational Geometry (SoCG), Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024.

[7] H. Edelsbrunner, D. Letscher, and A. Zomorodian, *Topological persistence and simplification*, Discrete Computational Geometry, 28 (2002), pp. 511–533.

[8] A. Hatcher, *Algebraic Topology*, Cambridge University Press, 2001.

[9] Y. Hu, T. Schneider, X. Gao, Q. Zhou, A. Jacobson, D. Zorin, and D. Panozzo, *Triwild: robust triangulation with curve constraints*, ACM Trans. Graph., 38 (2019).

[10] Y. Hu, T. Schneider, B. Wang, D. Zorin, and D. Panozzo, *Fast tetrahedral meshing in the wild*, ACM Trans. Graph., 39 (2020).

[11] Y. Hu, Q. Zhou, X. Gao, A. Jacobson, D. Zorin, and D. Panozzo, *Tetrahedral meshing in the wild*, ACM Trans. Graph., 37 (2018).

[12] A. Jacobson, L. Kavan, and O. Sorkine-Hornung, *Robust inside-outside segmentation using generalized winding numbers*, ACM Trans. Graph., 32 (2013), p. 33.

[13] A. Jacobson, D. Panozzo, et al., *libigl — a simple C++ geometry processing library*. `https://libigl.github.io/`, 2018.

[14] J. Jon Hasselgren, T. Akenine-M oller, and L. Ohlsson, *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation*, Addison-Wesley Professional, 2005, ch. 42, Conservative Rasterization. `https://developer.nvidia.com/gpugems/gpugems2/part-v-image-oriented-computing/chapter-42-conservative-rasterization`.

[15] P. Knupp, *Introducing the target-matrix paradigm for mesh optimization by node movement*, Eng. Comput., 28 (2012), pp. 419–429.

[16] *MFEM: Modular finite element methods [Software]*. `mfem.org`.

[17] D. P. Mitchell, *The antialiasing problem in ray tracing*, Advanced Topics in Ray Tracing, SIGGRAPH Course Notes, (1990).

[18] S. A. Mitchell and T. J. Tautges, *Pillowing doublets: Refining a mesh to ensure that faces share at most one edge*, in International Meshing Roundtable (IMR), 1995. Sandia National Laboratories tech. rep. SAND-95-2356C.

[19] C. Moon, S. A. Mitchell, J. E. Heath, and M. Andrew, *Statistical inference over persistent homology predicts fluid flow in porous media*, Water Resources Research, 55 (2019).

[20] D. Noble, M. Staten, and C. R. Wilson, *Using a faceted geometry representation to improve the performance of overlay grid meshing*, in International Meshing Roundtable (IMR), Research Note, 2024. Sandia National Laboratories tech. rep. SAND2024-08942A.

[21] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod, and H. A. Harrington, *A roadmap for the computation of persistent homology*, EPJ Data Science, 6 (2017), p. 17.

[22] S. J. Owen, *Parallel smoothing for grid-based methods*, in International Meshing Roundtable (IMR), Research Note, 2012, pp. 161–178.

[23] S. J. Owen, J. A. Brown, C. D. Ernst, H. Lim, and K. N. Long, *Hexahedral mesh generation for computational materials modeling*, Procedia Engineering, 203 (2017), pp. 167–179. In International Meshing Roundtable (IMR).

[24] S. J. Owen and T. R. Shelton, *Evaluation of grid-based hex meshes for solid mechanics*, Eng. Comput., 31 (2015), pp. 529–543.

[25] S. J. Owen, M. L. Staten, and M. C. Sorensen, *Parallel hex meshing from volume fractions*, in International Meshing Roundtable (IMR), Springer Berlin Heidelberg, 2012, pp. 161–178.

[26] B. Rieck et al., *Aleph — a library for exploring persistent homology*, 2016. `https://github.com/Pseudomanifold/Aleph`.

[27] B. Shawcroft, K. M. Shepherd, and S. A. Mitchell, *Tusqh*. `https://github.com/bshaw23/Tusqh`.

[28] ———, *Tusqh: Topological control of volume-fraction meshes near small features and dirty geometry*, CoRR, arXiv:2502.10609 (2025).

[29] M. Staten, D. Noble, and C. R. Wilson, *Constructing tetrahedral meshes no matter how ugly the CAD*, in International Meshing Roundtable (IMR), Research Note, 2024. Sandia National Laboratories tech. rep. SAND2024-03643C.

[30] M. L. Staten, J. F. Shepherd, and K. Shimada,

Mesh matching — creating conforming interfaces between hexahedral meshes, in International Meshing Roundtable (IMR), Berlin, Heidelberg, 2008, Springer Berlin Heidelberg, pp. 467–484.

[31] Y. Zhang, T. J. Hughes, and C. L. Bajaj, *An automatic 3D mesh generation method for domains with multiple materials*, Computer Methods in Applied Mechanics and Engineering, special issue on Computational Geometry and Analysis, 199 (2010), pp. 405–415. Extended from Proc. 16th International Meshing Roundtable (IMR), 2007.