Towards tetrahedral meshing with decoupled element and boundary constraints

Jonathan R. Bronson[1], Shankar P. Sastry[1], Mark B. Kim[1],
Joshua A. Levine[2], and Ross T. Whitaker[1]

[1] Scientific Computing and Imaging Institute, Salt Lake City, UT, U.S.A.
{bronson,sastry,mkim,whitaker}@sci.utah.edu
[2] Clemson University, Clemson, SC, U.S.A. levinej@clemson.edu

Summary. In tetrahedral mesh generation, the constraints imposed by adaptive element size, good tetrahedral quality (shape measured by some local metric), and material boundaries are often in conflict. Attempts to satisfy these conditions simultaneously frustrate many conventional approaches. We propose a new strategy for boundary conforming meshing that decouples the problem of building tetrahedra of proper size and shape from the problem of conforming to geometric boundaries. The proposed strategy is to first build a background mesh with the appropriate tetrahedral properties, and then to use a stenciling method to divide or cleave these elements to get a set of conforming tetrahedra, while strictly limiting the impacts cleaving has on element shape. Our contributions includes a new method for building graded, unstructured meshes and a generalization of the isosurface stuffing and lattice cleaving algorithms to unstructured background meshes.

1 Introduction

The generation of meshes that conform to several constraints, such as anisotropy and boundary conformation, is still a challenging problem requiring great computational effort. Many of these constraints are often in conflict with one another, preventing successful automatic mesh generation. We take the approach of explicitly decoupling the problem of conforming to boundaries from all other aspects of meshing. In the absence of the need to conform, other aspects of meshing become much easier to achieve.

Our approach begins by building a nonconforming tetrahedral background mesh. The elements of this mesh are created with the desired properties in terms of size and shape, but with no regard for adhering to boundaries. Next, we apply a single *cleaving* step to conform the mesh to material boundaries without greatly disturbing the characteristics of elements in the initial mesh. Near where we conform, we do expect some degradation of these properties to occur; however, we minimize its effects through a carefully designed set of operations.

Our proposed method is a generalization of those reported by the well known isosurface stuffing [3] and lattice cleaving algorithms [2]. Both algorithms show exactly the same behavior, albeit for a more limited class of boundaries (in the case of isosurface stuffing) and input background meshes (both algorithms require body-centered cubic (BCC) lattice inputs). Both algorithms also only target element quality, as measured by bounds on the dihedral angles, whereas, in this work, we also focus on element size and adaptivity (in terms of edge lengths) and anisotropy (in terms of edge orientations).

This work introduces a new strategy for building boundary conforming meshes out of nonconforming volumetric meshes. Specifically, we make the following contributions to the literature: (1) a new meshing strategy for boundary conforming tetrahedral meshes that works in the presence of nonmanifold boundaries, (2) a method for computing sizing fields of multimaterial volumetric data, and (3) a variational system for distributing mesh vertices (in the absence of boundaries) relative to these sizing fields

2 Background and Related Work

When meshing to conform to a boundary, the majority of algorithms first try to capture the boundary constraint. Typically, such meshes are produced by meshing boundary features in increasing dimensionality. But as dimension increases, the collection of lower dimensional elements impose an increasingly complex set of constraints for the next stage of meshing. By comparison, for approaches that start with a background lattice, proving one can still capture the boundary becomes more complex. Typically, these approaches use a structured lattice to construct meshes which are self-similar, such as an octree [5]. Labelle and Shewchuk used BCC lattices as background meshes to build tetrahedral meshes that conforming to a smooth boundary while maintaining dihedral angle bounds [3]. Bronson et al. [2] generalize the results of Labelle and Shewchuk with their lattice cleaving approach and were also able to generalize a proof that in the case of multimaterial boundaries, a bound for the dihedral angles of the resulting elements exists.

We remark that an interesting conclusion, parallel to our own, in the domain of hexahedral meshing by advancing front (paving and plastering) is that relaxing the boundary constraint by delaying boundary meshing leads to improved results [7].

3 Methodology

We provide algorithms which are separately capable of handling the specific pieces of the following pipeline (Fig.1).

Sizing Field. Mesh elements must be adaptively sized for the purpose of geometric fidelity and PDE solution accuracy while simultaneously reducing the number of elements needed for an accurate numerical simulation. A sizing field for the purpose of mesh generation should posses the following properties:

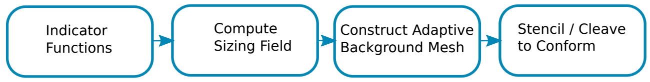

Fig. 1: Proposed meshing pipeline for conforming volumetric meshing

a) it should be small near thin features and high-curvature regions; b) it should progressively increase for larger features and lower-curvature regions; and c) it should satisfy Lipschitz continuity conditions.

The algorithm to compute a sizing field is borrowed from [1] and [4], and it computes the distance transform (DT) three times. First, a DT is computed starting from the material boundary surface. Since the DT is discontinuous at the media axis (where the wave front collide), these points of discontinuity are determined. The DT is computed again from the medial axis. The values of the DT at the boundary locations indicate the feature size at the boundary. The DT is computed again from the boundary, but this time, the initial distance at the boundary is assumed to be the feature size and the gradient of the DT is limited to a desired value. This gives us the sizing field over the whole domain.

Particle Systems for Adaptive Background Meshes. In our approach to adaptive mesh generation, we first distribute particles over the whole domain and then use delaunay tetrahedralization algorithm to construct a background mesh. We use an electrostatic particle simulation technique to distribute particles over the whole domain. The particles are assumed to be negatively charged and domain is assumed to posses a static charge density that attracts the particles to regions of interest. We construct a charge density that is a function of the sizing field in order to "attract" the particle towards the desired regions. When a stable distribution is reached, a tetrahedral mesh is generated using Tetgen [6]. The electrostatic potential due to the background charge density is computed by solving the Poisson equation, $\nabla^2 u(x) = f(x)$, where u is the electrostatic potential and f is the charge density. For the electrostatic simulation, the number of particles are seeded in the various part of the domain is proportional to the local charge density for quick convergence.

3.1 Unstructured Cleaving

To produce surface conforming meshes from an unstructured background mesh we adapt the lattice cleaving[2] algorithm to unstructured meshes. Lattice cleaving, like isosurface stuffing[3], combines tetrahedral stenciling with a set of simple mesh modification rules to guarantee element quality.

Adapting these algorithms to unstructured meshes poses two main challenges. First, some form of parity rule is required to ensure neighboring tetrahedra share consistent face boundaries. Second, the α-parameter vector becomes as long as the number of edges in the mesh, with optimal values unique to each mesh.

Stencil Consistency

To ensure stencil consistency for unstructured meshes, we modify the strategy of stencil generalization used in the lattice cleaving algorithm. Any strategy that generalizes all possible stencils to the symmetric 6-cut case is effectively a parity rule. These generalizations can be achieved by placing *virtual cuts* along any edge with no material interface. These cuts can be interpreted as cuts that have already *snapped* to the incident vertex, collapsing the stencil elements that helped define the would-be interface. Similar roles exist for *virtual triples* and *virtual quadruples*.

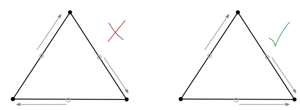

Fig. 2: Left: Cyclic virtual cuts lead to an unsatisfiable generalization. Right: Any ordered priority can lead to safe generalization.

We observe that as long as the direction of virtual cuts is acyclic, a consistent and nondegenerate stenciling is always possible. Enforcing this acyclic property only requires a strict total ordering on the vertices of the background mesh. This ordering can then be used to place virtual cuts, and subsequently determine ideal locations for virtual triples and virtual quadruples.

Alpha Selection

The quality preservation of the cleaving algorithms relies on the selection of violation parameter vector α. Unfortunately, the complexity of determining an optimal α is significant. Isosurface stuffing employed brute-force interval arithmetic to determine tight bounds on optimal α-parameters. This form of solution becomes intractible for the multimaterial case of lattice cleaving, due to the 2 and 3-dimensional violation regions, as well as the increase in number of stencil cases. As the length of the α-parameter vector becomes a function of the edge count in an unstructured mesh, a systematic optimization of these values is not an option. Thus, we aim first for a way to achieve *safe* α-parameters.

Although tight bounds for the lattice cleaving algorithm are unknown, a formal proof showed that such bounds do exist as a function of α. This proof directly applies to unstructured meshes, since it does not rely on the specific structure of the background lattice, merely that its element not be degenerate. Its premise also offers a convenient way to pick safe α-parameters. For any tetrahedron, a set of α-balls can be chosen to satisfy the non-overlapping condition. These α-balls must not overlap along edges, and they must not overlap along altitudes. Thus, a safe choice for any vertex can always be made by choosing the half the length of shortest edge or altitude.

4 Preliminary Results

Figure 3 shows a cut-away view and a surface view of two unstructured adaptive tetrahedral meshes. Since the background mesh adapts to feature size, the cleaving algorithm has sufficient resolution to capture the surfaces and their topology properly, while remaining coarse in areas where the fine resolution is unnecessary.

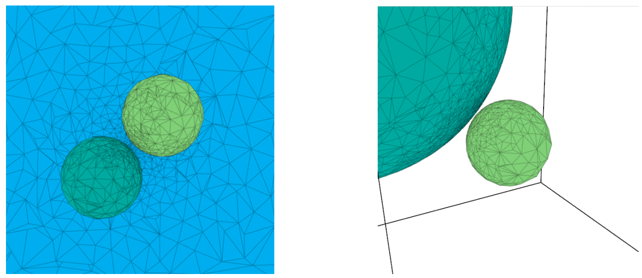

Fig. 3: Left: Cut away view of two spheres kissing. Right: Surface view of two spheres of different sizes kissing.

The particle system achieves its goals with simplicity due to the lack of interface surfaces. The lattice cleaving algorithm generalizes to unstructured meshes with only a few changes. Our α-parameter selection provides a starting point for more sophicated methods. Together, this early work suggests that the union of traditional and combinatoric meshing techniques promises to provide a fertile ground for new developments in high quality conforming mesh generation for unstructured meshes.

References

1. S. Bouix and K. Siddiqi. Divergence-based medial surfaces. In *In Sixth European Conference on Computer Vision*, pages 603–618, 2000.
2. J. R. Bronson, J. A. Levine, and R. T. Whitaker. Lattice cleaving: Conforming tetrahedral meshes of multimaterial domains with bounded quality. In *21st IMR*, pages 191–210, Oct. 2012.
3. F. Labelle and J. R. Shewchuk. Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. *ACM Trans. Graph.*, 26(3):57, 2007.
4. P.-O. Persson. Mesh size functions for implicit geometries and PDE-based gradient limiting. *Engineering with Computers*, 22(2):95–109, 2006.
5. M. S. Shephard and M. K. Georges. Automatic three-dimensional mesh generation by the finite octree technique. *Int. J. Num. Meth. Eng.*, 32(4):709–749, 1991.
6. H. Si. TetGen: A Quality Tetrahedral Mesh Generator and Three-Dimensional Delaunay Triangulator. http://tetgen.berlios.de/.
7. M. L. Staten, R. Kerr, S. J. Owen, and T. D. Blacker. Unconstrained paving and plastering: Progress update. In *15th IMR*, pages 469–486, 2006.