SCI INSTITUTE
TECHNICAL REPORT

Quality Improvement and Feature Capture in Hexahedral Meshes

Jason F. Shepherd[†], *Claurissa J. Tuttle*[†], *Cláudio T. Silva*[†], *Yongjie Zhang*[×]
[†]University of Utah [×]University of Texas

UUSCI-2006-029

Scientific Computing and Imaging Institute
University of Utah
Salt Lake City, UT 84112 USA

August 10, 2006

Abstract:

Building high-quality quadrilateral/hexahedral meshes directly from volumetric data is hard. Existing algorithms for generating meshes from volumetric data are based on primal and dual isocontouring algorithms, and current research focuses on improving the quality of such meshes. Most techniques are based on isocontouring techniques, and work by generating a grid of hexahedra on the interior/exterior of an isosurface, and then adjusting the elements that lie on the boundary of the grid to fit the surface. As a result of the element adjustment, many of these elements lose their convexity, as measured by the scaled Jacobian metric. Recovering the convexity of these elements is difficult since the position of boundary vertices is restricted to the domain of the isosurface.

In this paper, we propose a solution to this problem using insights obtained from the structure of the dual of a hexahedral mesh. Our solution is to add a sheet of hexes along the boundary, composed of well-shaped elements. The additional degrees of freedom provided by this sheet enables the optimization of the original poorly-shaped elements that are no longer on the boundary allowing for the creation of a high-quality mesh. An extra benefit of our technique is being able to capture sharp features, which is done by inserting multiple sheets. Our experimental results demonstrate the successful removal of all bad elements (i.e. those elements of the mesh that have a scaled Jacobian measure of less than 0.2) in a number of complex examples.

THE UNIVERSITY OF UTAH

Quality Improvement and Feature Capture in Hexahedral Meshes

Jason F. Shepherd
University of Utah

Claurissa J. Tuttle
University of Utah

Cláudio T. Silva
University of Utah

Yongjie Zhang
University of Texas

August 10, 2006

Abstract

Building high-quality quadrilateral/hexahedral meshes directly from volumetric data is hard. Existing algorithms for generating meshes from volumetric data are based on primal and dual isocontouring algorithms, and current research focuses on improving the quality of such meshes. Most techniques are based on isocontouring techniques, and work by generating a grid of hexahedra on the interior/exterior of an isosurface, and then adjusting the elements that lie on the boundary of the grid to fit the surface. As a result of the element adjustment, many of these elements lose their convexity, as measured by the scaled Jacobian metric. Recovering the convexity of these elements is difficult since the position of boundary vertices is restricted to the domain of the isosurface.

In this paper, we propose a solution to this problem using insights obtained from the structure of the dual of a hexahedral mesh. Our solution is to add a sheet of hexes along the boundary, composed of well-shaped elements. The additional degrees of freedom provided by this sheet enables the optimization of the original poorly-shaped elements that are no longer on the boundary allowing for the creation of a high-quality mesh. An extra benefit of our technique is being able to capture sharp features, which is done by inserting multiple sheets. Our experimental results demonstrate the successful removal of all bad elements (*i.e.* those elements of the mesh that have a scaled Jacobian measure of less than 0.2) in a number of complex examples.

1 Introduction

This paper deals with the problem of improving quality and capturing features in hexahedral meshes. The problem we address is as follows: Given a 3D volume V, and an isosurface S defined inside this volume, compute a hexahedral mesh H that has S as its external boundary. Furthermore, we require H to be composed completely of high-quality elements suitable for use in computational simulations, in particular, H should not contain any inverted or void elements.

Creating hexahedral decompositions for general geometries is an important *mesh generation* problem. Mesh generation deals with the problem of decomposing complex geometry into discrete elements (meshes), which can be used for modeling, simulation, and visualization. These meshes play a significant role in computationally-based science and engineering.

Originally, mesh generation techniques required substantial manual intervention, which made the process error prone, and very expensive. In the last several years, tetrahedral mesh generation has become highly automated, and at this point in time, it is widely accepted that given good boundary constraints (i.e., a high-quality triangulation of the surface for the external boundary of a volume), it is highly probable that a high-quality tetrahedral mesh of the inside volume can be computed with minimal, if any, user input [1, 29].

Unfortunately, the relative ease with which tetrahedral meshes can be generated does not address many important computational applications. In fact, for many applications, quadrilateral and hexahedral discretizations are preferred for many reasons. One reason is efficiency. Tetrahedral meshes typically require 4-10 times more elements than a hexahedral mesh to obtain the same level of accuracy [7, 38]. Accuracy also plays a role. In some types of numerical approximations (i.e., high deformation structural finite element analysis), tetrahedral elements are mathematically *stiffer* due to a reduced number of degrees of freedom associated with a tetrahedral element [3, 6]. This problem is also known as *tet-locking*.

A more *intangible* reason is that hexahedral meshes have been used for decades for certain high-end application domains, and there is substantial evidence of the quality and accuracy of its applicability to these domains. Even if techniques based on high-order tetrahedral meshes were to emerge, it would take a substantial amount of time for all these applications to fully move to another technology.

Despite the importance of hexahedral meshing, its state of the art lags tetrahedral mesh generation. While, for tet meshing, there are a number of highly successful automated methods, for hexahedral meshing, most practical techniques still rely on considerable user input. In fact, most commercial tools are based on the relatively straightforward *mapping* [9, 10] or *sweeping* [4, 15, 19, 20, 30, 35, 36] techniques, which require manual partitioning of geometries.

Although labor intensive, these simple techniques have been applied to many simulations where the input comes from a CAD model. However, it is difficult to see how to apply these techniques to general geometry.

Our work in this paper is in the direction of quality improvement and feature capture in hexahedral meshes of complex geometries. We build on the work of Zhang et al. [41, 42], which computes *topological* hexahedral meshes from volumetric isosurfaces. A key result of our paper is to show how to capture boundary information, and how to optimize the quality of the elements generated, while keeping hexahedral meshes throughout the space occupied by volumetric data. In short, instead of *topological* hexahedral meshes (i.e., the connectivity of the cells respect hexahedral topology), our meshes also respect geometric and quality constraints.

Our specific contributions are:

- A method for quality improvement and feature capture in hexahedral meshes of an arbitrarily complicated domain.
- A theory of hexahedral boundary capturing to optimize the potential quality of the elements generated within the space, and the described method allows conformal hexahedral meshes throughout the space occupied by volumetric data.

- We describe the necessary topolgic structures needed within a hexahedral mesh to capture boundary discontinuities (i.e. sharp features).
- Finally, we demonstrate our technique on several models indicating the realized improvement in the overall quality of the final meshes on these models over previous algorithms.

2 Background

In this section, we provide a background on generating models from volumetric data, including the Marching Cubes family of algorithms. We offer a brief discussion of the history of octree-based methods for generating discrete geometries and highlight the relationship of these methods with Marching Cubes and dual contouring methods. We also describe the implementation of these methods for generating tetrahedral and hexahedral meshes.

2.1 Generating Models from Volumetric Data

Before meshing can be performed, a boundary surface is extracted that can serve as an initial constraint for hexahedral meshing. One of the most general and powerful techniques for volumetric boundary reconstructions is isosurface extraction. An *isosurface* $\mathscr{S}_a$ is defined as the *preimage* of a function $f : \mathbf{R}^n \to \mathbf{R}$ and value a; it is the set of values in the domain that map to a, *i.e.* $\mathscr{S}_a = \{x \in \mathbf{R}^n : f(x) = a\}$. We then say f is the *implicit function* and a is the *isovalue*. Most often, f is given as a sampled 3D volume.

One of the most widely used techniques for isosurface generation is the Marching Cubes (MC) algorithm [22, 27]. Marching Cubes samples the function f at a grid of fixed resolution, and then uses a table of possible configurations of range signs to create a triangulated surface out of those samples. The main strengths of MC are its generality, simplicity and robustness. Following the discovery of the Marching Cubes algorithm, the last two decades have seen a constant stream of work on effective techniques for the computation and visualization of isosurfaces (e.g. see [2, 8, 21, 27, 39]).

Surfaces generated by MC have inherent bias caused by placing vertices on all intersections between grid edges and the surface. Because of this, meshes generated by MC (and variants) are typically over-tessellated, and contain many bad elements. Furthermore, as we will see later, for hexahedral meshing, we need a boundary surface that is composed solely of quads, (instead of triangles). Dual contouring methods [12, 14] reduce many of these problems. They generate surfaces that are the topological dual of the MC surfaces, and can also be used to reproduce sharp features by extrapolating the sampled normals. Under certain circumstances (described next), these techniques can be used to generate decent starting points for hexahedral meshing.

2.2 Octree Methods

In the 1980s, the octree technique was developed to generate tetrahedral meshes [33, 40]. Cubes containing the geometric model are recursively subdivided until the desired resolution is reached. Tetrahedral meshes are constructed from the irregular cells on the boundary and the internal regular cells. Quality is improved by restricting the octree construction; in particular, the octree subdivision is constrained to make the subdivision level of neighboring cells to only differ by one.

A grid-based approach was also proposed to generate a fitted 3D grid of hexahedral elements on the interior of the volume [31]. The regular grid of hexes is generated for the interior volume and hexahedral elements are added at the boundaries to fill gaps. The resulting mesh does not preserve a predefined input surface mesh, and will change as the orientation of the cubes in the octree structure is changed. This method is robust, but it tends to generate poor quality elements along the boundary. Therefore, quality improvement is required. Modifications were introduced to allow for significant transitions in element sizes utilizing an octree decomposition of the domain [32, 37].

2.3 LBIE-Mesher

Isocontouring methods, primal contouring (or Marching Cubes) and dual contouring, have been extended to generate finite element meshes from volumetric data. The Marching Cubes algorithm (MC) [22] was extended to extract tetrahedral meshes between two isosurfaces directly from volume data [11]. A different algorithm, Marching Tetrahedra (MT), was proposed for interval volume tetrahedralization [28].

Dual Contouring [14] was also extended to finite element mesh generation. A software package, namely LBIE-Mesher (Level Set Boundary Interior and Exterior Mesher), has been developed to construct adaptive and quality 2D (triangular/quadrilateral) and 3D (tetrahedral/hexahedral) finite element meshes [41, 42].

We base our work on LBIE-Mesher, which is used to generate the initial hexahedral meshes from the volumetric data [41, 42]. The meshing algorithm in LBIE-Mesher works as follows. First, a bottom-up surface topology preserving octree-based algorithm is applied to select a starting octree level. Then the dual contouring method is used to extract a preliminary uniform hexahedral mesh by analyzing each interior grid point. The uniform mesh is decomposed into finer hexes adaptively using predefined templates without introducing any hanging nodes. The positions of all boundary vertices are recalculated to approximate the boundary surface more accurately.

To improve mesh quality, LBIE-Mesher performs a final step, where geometric flows are used to smooth the boundary surface, and improve the quality of the extracted hexahedral meshes [33]. There are three main steps in LBIE-Mesher quality improvement scheme: (1) Denoising the surface mesh; which consists of vertex adjustment in the normal direction with volume preservation. Here the surface diffusion flow is chosen to smooth the surface, and the discretized Laplacian-Beltrami operator is used to solve the geometric partial differential equation (GPDE). (2) Improving the aspect ratio of the surface mesh; which consists of vertex adjustment in the tangent direction with feature preservation. Surface features are preserved since the movement in the tangent plane does not change the surface shape. (3) Improving the aspect ratio of the hexahedral mesh; which consists of adjustment inside the volume.

3 Theory of Hexahedral Mesh Structures

3.1 The Dual of Hexahedral Meshes

Many of the concepts that will be described in this paper utilize a *dual* representation of a hexahedral mesh. We will utilize the dual representation of a hexahedral mesh as defined by Mitchell [24] in his hexahedral mesh existence proof, but also use some concepts from Murdoch [26]. We also utilize the hexahedral constraints for maintaining a hexahedral topology and sufficient quality as outlined in [34].

Each hexahedron consists of three pairs of opposing quadrilaterals. This pairing of opposing quadrilaterals can be dualized to a single quadrilateral *patch* located medially between the two original quadrilaterals. Thus, for each hexahedron, there are three, intersecting, patches which are dual to the original hexahedron. The intersection point is known as a *centroid* in the dual, and the centroid is dual to the hexahedron.

Interior to a hexahedral mesh, an interior quadrilateral will be shared by two hexahedra. As a result, the boundary of each of the patches created by the dualization of each of the hexahedron matches with the patches of adjacent elements creating a collection of manifolds (known as *sheets*), that uniquely define the original hexahedral mesh. Additionally, these sheets interact according to specific rules (a more detailed, and complete, enumeration of these rules can be found in [24]):

1. Only three sheets can intersect at any given centroid.

2. A sheet cannot be tangent to any other sheet.

3. Sheets will either span the space, or will be closed, within the space.

In terms of quality, hexahedra that are more perfectly cubical in shape are desired [16]. In terms of the dual, we can append the following considerations to the rules above to enhance the quality of a hexahedral mesh:

1. For each hexahedron, the three sheets should be locally orthogonal (this prevents element *skewing*).

2. Each sheet should maintain low curvature (planar sheets are preferred for ideal quality).

3.2 Surface Capture

For any hexahedral mesh, we make the following (self-evident) assertion:

Lemma 1. *The boundary of any hexahedral mesh is a quadrilateral mesh.*

Utilizing Lemma 1 and the method of constructing the dual of a hexahedral mesh described in the previous section, we assert the following:

Lemma 2. *Each quadrilateral on the boundary is contained in one hexahedral element of the mesh, and there is an opposing quadrilateral in this hexahedron to the quadrilateral on the boundary of the mesh.*

The boundary quadrilateral and the opposing quadrilateral will dualize to a single patch of one of the sheets within the hexahedral mesh. The collection of patches created from each of the boundary quadrilaterals bears some similarity to the geometric boundary of the model. If the patches belong to a single sheet, we can reduce the skew and limit the curvature of the sheet to the curvature of the geometric boundary; however, if the patches belong to multiple sheets, then the intersection between two sheets near the boundary force element skewing or increased

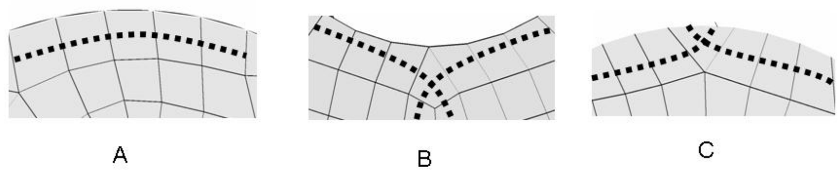

Figure 1: When partial sheets capture boundaries, the quality and regularity of the mesh is affected. Image A shows elements from a single sheet capturing the upper boundary of the solid. Image B and C use patches from two sheets to capture the upper boundary of the solid. In image B and C, note how the regularity of the mesh is affected, and the resulting skew in the transition element due to the sheet curvature away from the boundary. In image C, a near doublet element is formed due to the low curvature of the boundary being captured.

sheet curvature where the sheets diverge from each other. A cross section of a few possible mesh configurations highlighting the sheets near the boundary of the mesh is shown in Figure 1.

If we assume no discontinuities in the boundary surface description, meshes which utilize a single sheet to capture all of the boundary quadrilaterals will invariably admit elements whose potential quality is higher than meshes which utilize multiple sheets to capture the boundary surface.

3.3 Curve Capture

Where discontinuities exist in the geometry, it is often desirable to place quadrilaterals such that the edges of the quadrilaterals align themselves with the discontinuity. These discontinuities are often called *sharp features*, and in solid models are often the trimmed boundaries of pairs of surfaces within the solid model.

Given an existing hexahedral mesh with a sharp feature, a few observations regarding the sharp feature can be noted. Similar to an offset patch for a given quadrilateral on the boundary, there is a line segment formed from the intersection of two different sheet patches within one of the hexahedra containing the edge on the sharp feature. Figure 2 shows an example where the intersection of two sheets within a mesh form the necessary mesh topology to capture a sharp feature in a given geometry.

A similar methodology is incorporated into the grafting method proposed by Jankovich et al. [13] to capture new curves in existing meshes. The method starts with a mesh that captures the general shape of the surface, and then inserts a new sheet that intersects the boundary sheets. The intersection of the two sheets creates the mesh topology necessary to capture the new curve in the existing mesh.

4 The Algorithm

In this paper, we describe the sheet insertion algorithm for improving hexahedral mesh generation from volumetric data. Our work is based on the adaptive octree algorithm of Zhang et al. [41–43] (implemented in LBIE-Mesher) to construct conformal hexahedral meshes with enough geometric details to capture the desired isosurface and with various geometric quality metrics. As described

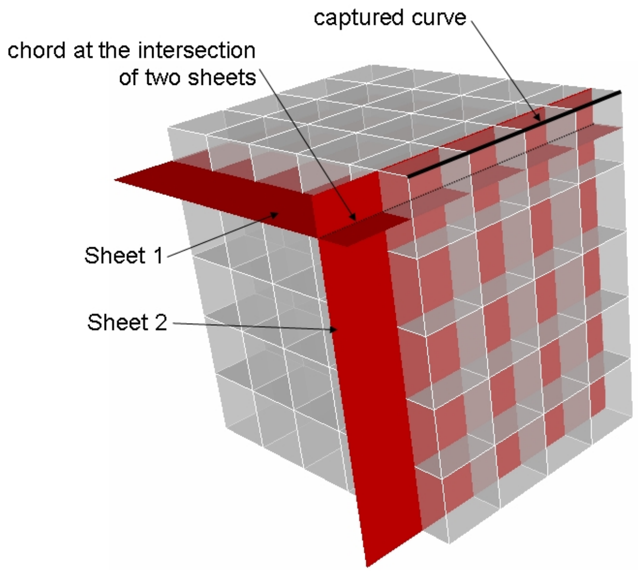

Figure 2: The intersection of two sheets produces a mesh topology that enables curves, or geometric discontinuities, to be captured in a hexahedral mesh.

before, this algorithm by itelf does not generate elements of sufficient quality. In our work, we further improve the quality of the hexahedral mesh near the isosurface by inserting a new sheet, which is geometrically similar to the isosurface, into the mesh. This new sheet has the effect of aligning the boundary hexes with the isosurface, and of providing extra degrees of freedom and flexibility for the generation of elements (hexahedra) with sufficient quality for computer-based simulations.

4.1 Adding a Boundary Sheet

In order to add a new sheet near the boundary of the geometry, we use a modified version of a pillowing algorithm as described by Mitchell [25]. The sheets can be inserted utilizing the primal elements of an existing mesh, and without explicitly creating a geometric definition for the sheet and calculating the intersections with the other local sheets in the space.

4.1.1 Sheet Insertion via *Pillowing*

The basic pillowing algorithm is as follows (also refer to Figure 3):

1. *Define a shrink set* - For our purposes, this step involves dividing the existing mesh into two sets of elements: one set for each of the half-spaces defined by the sheet to be inserted. One of these two sets of hexahedral elements comprises the shrink set. The choice of which one should be the shrink set is arbitrary, although the best algorithmic choice will be the set with the fewest number of elements.

2. *Shrink the shrink set* - Create a gap region between the two previous element sets (see Figure 3). The difficulty in this step involves splitting the shared nodes, edges, and quads in the

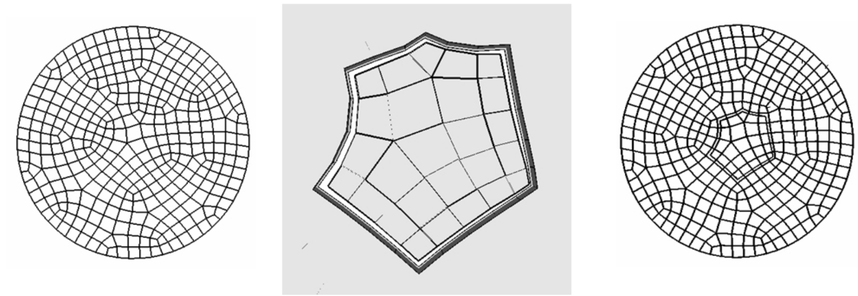

Figure 3: A basic pillowing operation starts with an initial mesh (A) from which a subset of elements is defined to create a shrink set. The shrink set is separated from the original mesh and 'shrunk' (B), and a new layer of elements (i.e. a dual sheet) is inserted (C) to fill the void left by the shrinking process.

existing mesh, while maintaining the appropriate correspondence of the mesh entities with the geometric topology.

3. *Connect with a layer of elements* - A new sheet of elements is conformally inserted between the two element sets. To complete this step, an edge is inserted between each node that was separated during the shrinking operation. Utilizing the quadrilaterals on the boundary between the two sets of hexes, along with these new edges, it is fairly straightforward to determine the connectivity of all of the hexes in this new layer.

When inserting a single sheet next to the boundary, we define our shrink set as all of the elements within the solid. We desire the original surface mesh to be undisturbed by any of our modification operations, so a copy of the surface mesh is made prior to shrinking the layer of elements near the boundary. Reconnecting the boundary of the shrink-set with the correct nodes in the previously copied surface mesh completes the inserted sheet. For each face on the boundary, one new hexahedral element is created as the boundary layer defining the newly inserted sheet in the improved mesh. Figure 4 demonstrates a mesh with a newly inserted sheet for both the interior and exterior boundary.

4.2 Sharp Feature Capture

Whenever a discontinuity in the geometry of the boundary occurs, it is advantageous to the quality of the mesh and to the fidelity of the geometry to have a string of mesh edges which align themselves with the discontinuity. A string of mesh edges results in a hexahedral mesh whenever two sheets intersect. We can control the placement of the edges resulting from the intersection of the two sheets by controlling the locations of the sheet intersections. A similar methodology is utilized by Borden et al. [5] to insert surfaces into existing meshes. In Figure 5 we place a planar sheet behind the face in the head model. The intersection of the planar sheet (which also captures a new planar surface) with the boundary sheet inserted earlier, to improve the quality of the mesh, produces a string of mesh edges that are nicely aligned to create the sharp *corner*. This allows the face to be *cut* from the head model.

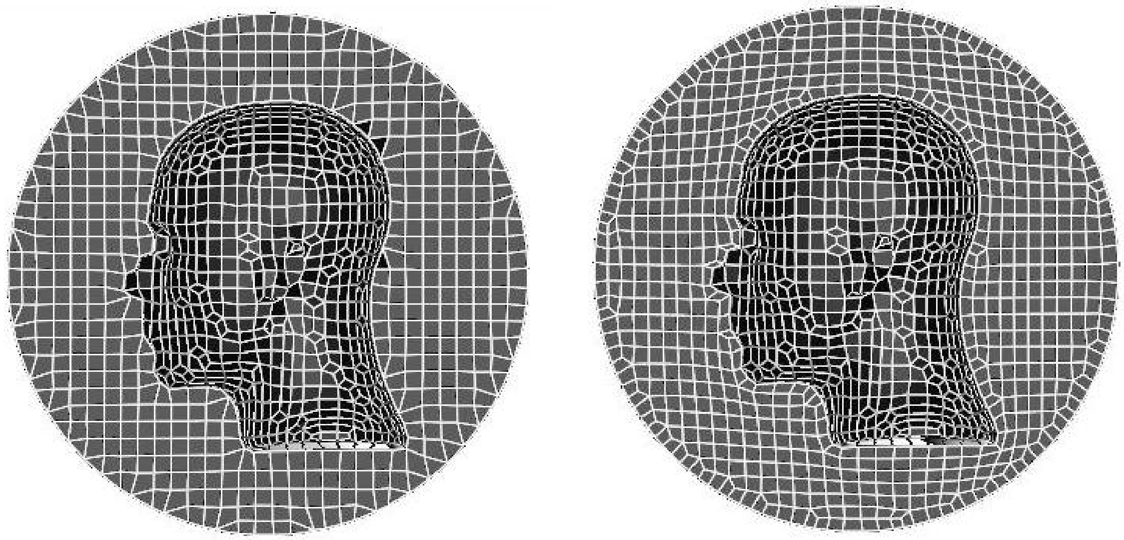

Figure 4: A hexahedral mesh of a sphere with a human head embedded in the center. The mesh on the left is the mesh resulting from an octree mesh refined to a level to capture major details of the geometry of the embedded head. The resulting mesh then has the nodes on the boundaries moved to better capture the geometric features of the data. The mesh topology on the right is identical to the mesh topology on the left, with the exception of two additional sheets of hexahedral elements for the interior and exterior boundary surfaces using a sheet insertion (pillowing) algorithm. The resulting mesh topology has a higher quality potential and flexibility due to the improved mesh topology capturing the geometric boundary. (Note the quadrilaterals in the left figure, which look more triangular in shape, and how the addition of the inserted sheet allows some additional flexibility to improve the shape of these elements.)

By inserting multiple sheets, we can perform *Boolean*-like CG operations in the hexahedral mesh while still maintaining conformity with all of the split-off pieces. In Figure 6 we demonstrate several successive spherical cuts from a single hexahedral mesh of a cubical geometry. Where two sheets intersect, the result is a string of mesh edges which align with the cut enabling the sharp features in the resulting model to be recognized. Figure 7 lists the resulting quality of each of the elements demonstrating the overall high quality of the resulting mesh.

4.3 Verification, Validation, Smoothing

In this section, we describe the methods utilized to verify the quality of the resulting meshes. We also describe the optimization methods used in validating the potential quality of the given geometry and mesh topology.

4.3.1 Element Quality Verification

Verification of element quality is accomplished using the scaled Jacobian metric for a hexahedral element. The Jacobian matrix is calculated for each node with respect to a hexahedral element. For

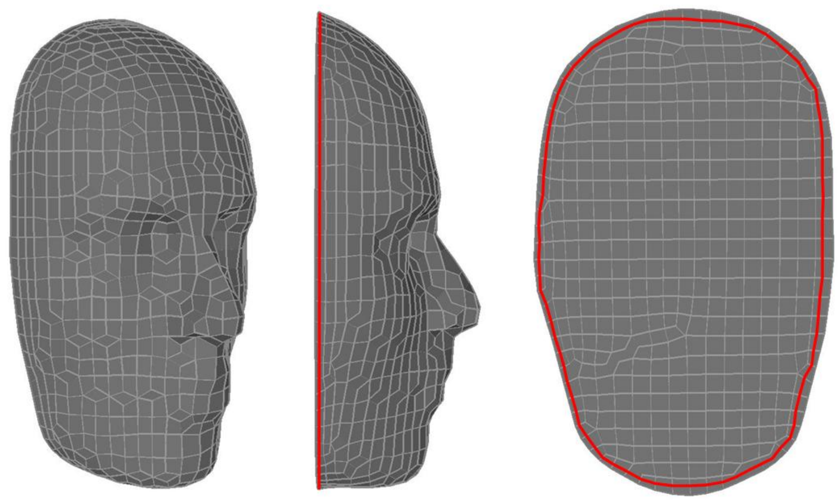

Figure 5: In locations where two sheets intersect, the resulting mesh topology contains a string of edges that can be aligned with sharp features, or boundary discontinuities. In this image, we 'cut' the face from the head model by inserting a planar sheet behind the face. The inserted boundary sheet (shown in the image on the right), along with the newly inserted planar sheet (middle image) behind the face, results in a mesh topology which contains a string of edges sufficient to produce a sharp boundary where the two sheets intersect, as shown in the middle image above.

each node in a hex, there are exactly three neighbor nodes connected via an edge in the hexahedra. For a single node of a hexahedron, the Jacobian matrix is defined as:

$$A_0 = \begin{vmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 \\ z_1 - z_0 & z_2 - z_0 & z_3 - z_0 \end{vmatrix}$$

For a single hex, there will be eight such matrices (for additional discussion on elements with multiple Jacobian matrices see [16]). The minimal determinant of these eight matrices is known as the *Jacobian* of the hexahedral element. A *scaled* Jacobian normalizes the vectors; therefore, the range of possible values for a scaled Jacobian is [-1.0, 1.0]. Values below zero of the scaled Jacobian are indicative of loss of convexity of the element.

Throughout the remainder of this paper, we will assume that any element with scaled Jacobian less than zero is unsuitable for analysis and any element with scaled Jacobian less than 0.2 is questionable for use in an analysis.

The element quality calculations shown in this paper were performed using the software library, VERDICT [18], which contains a comprehensive suite of mesh quality metrics for evaluating the quality of hex, tet, quad and triangle finite elements.

4.3.2 Mesh Optimization

Because correct and optimal placement of the new sheet can be difficult, it is often desirable to perform a smoothing, or mesh optimization, operation on the resulting mesh after the new sheet has been inserted to obtain better nodal placement and higher quality elements. We have utilized the TSTT Mesh Quality and Improvement Toolkit (MESQUITE) library of smoothing algorithms to accomplish the optimization of the meshes [23]. Within the MESQUITE class of mesh smoothers we have access to both *untangling* smoothers (a *tangled* mesh contains elements which may be inverted, or have a negative Jacobian value) and a wide range of optimization algorithms for untangled meshes. We have utilized a *mean ratio* smoother (as described by Knupp [17]), which incorporates an L2-norm template with guarantees that (1) the mesh will remain untangled if the initial mesh is untangled, and (2) the average value of the mean ratio will either stay the same, or be decreased.

Utilizing the guarantees of the L2-norm template, we can assume that if the nodes on the boundary of the mesh are fixed in place while the interior nodes are free to move during optimization, then the smoothed mesh will have either the same or better quality upon completion of the optimization. We, therefore, can run the smoother until the optimization converges with the given geometry and mesh topology. The results reported in the next section have all been optimized using this mean-ratio smoother. While it may be possible to subsequently improve the quality of some of the individual elements, it would be done at the expense of the quality of the adjacent elements. Therefore, we have some confidence that the average element quality for the given mesh topology and geometry is optimal, and only modifications to the mesh topology (i.e. sheet insertions or extractions, refinements, etc.) can be utilized to gain additional average quality improvements in the reported meshes.

5 Results

In this section we display results detailing the difference in element quality after improving the method of capturing the original boundary. We will utilize the scaled Jacobian metric, described earlier, as the definitive measure for hexahedral quality.

In order to make appropriate comparisons, the boundary (surface) mesh is assumed to be fixed, thus only nodes on the interior of the mesh are free to move in an effort to optimize the quality of the reported mesh. The boundary meshes were originally optimized using the *geometric flow* smoothing algorithm as reported by Zhang et al. [43]. No additional optimization of the boundary mesh has been incorporated, although in many cases improved quality of the interior elements would result with improved boundary optimization. The results shown in the figures in this section are the difference in the quality of the meshes before insertion of the hexahedral sheets and after insertion of the hexahedral sheets.

Additionally, all reported hexahedral meshes have been smoothed with the MESQUITE smoother in an attempt to report maximal quality results for the given geometry and mesh topology. The distributions of element quality are reported using the *scaled Jacobian* metric as implemented in the VERDICT library.

There are four examples given: a human knee model (Figure 8), a human head (Figure 10), a meshed sphere around the human head (Figure 12), and a model of an mAChe biomolecule (Figure 14). The distribution of element quality is given for each of the meshes both before sheet insertion and after inserting a sheet to more appropriately capture the geometric features of the boundary surfaces (see Figures 9, 11, 13, and 15). In each of the quality distribution plots, there is significant movement of the element quality towards higher scaled Jacobian values. Additionally, the introduction of the boundary sheet enables the removal of all negative or questionable scaled Jacobian values from the displayed meshes.

6 Conclusion

In this paper, we have outlined a method utilizing a simple hexahedral sheet insertion process to improve the quality of the elements that were generated using an octree-based algorithm with methodologies similar to the dual contouring approach that captures isosurfaces within volumetric data. We provide background on the dual of a hexahedral mesh as the basis for outlining a theory for capturing boundaries in hexahedral discretizations. Using the boundary capturing theories and treating the resulting boundary quadrilateral meshes as a fixed discrete geometry and topology, we demonstrate how by strategically inserting sheets into the octree-produced hexahedral meshes, we can significantly improve the overall quality of the resulting meshes without changing the boundary quadrilaterals. We also show how insertion of multiple sheets can be utilized to capture discontinuities, or sharp features, in the meshes. We have demonstrated the quality improvement on several models. We have attempted to ensure that the comparisons are accurate by utilizing the MESQUITE mesh optimization algorithms on the meshes both before and after the sheet insertion. This has been done in an effort to realize the highest potential mesh quality for the given mesh topologies. The VERDICT library is also used to verify and report the final mesh quality.

Acknowledgments

C. Silva is funded by the National Science Foundation (grants CCF-0401498, EIA-0323604, OISE-0405402, IIS-0513692, CCF-0528201), the Department of Energy, Sandia National Laboratories, Lawrence Livermore National Laboratory, an IBM Faculty Award, and the National Institutes of Health.

References

[1] P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun. Variational tetrahedral meshing. *ACM Trans. Graph.*, 24(3):617–625, 2005.

[2] C. L. Bajaj, V. Pascucci, and D. R. Schikore. Fast isocontouring for improved interactivity. In *1996 Volume Visualization Symposium*, pages 39–46, 1996.

[3] S. E. Benzley, E. Perry, K. Merkley, and B. Clark. A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elasto-plastic analysis. In *Proceedings, 4th International Meshing Roundtable*, pages 179–191. Sandia National Laboratories, October 1995.

[4] T. D. Blacker. The cooper tool. In *Proceedings, 5th International Meshing Roundtable*, pages 13–30. Sandia National Laboratories, October 1996.

[5] M. J. Borden, J. F. Shepherd, and S. E. Benzley. Mesh cutting: Fitting simple all-hexahedral meshes to complex geometries. In *Proceedings, 8th International Society of Grid Generation Conference*, 2002.

[6] M. L. Bussler and A. Ramesh. The eight-node hexahedral elements in fea of part designs. *Foundry Management and Technology*, pages 26–28, November 1993.

[7] A. O. Cifuentes and A. Kalbag. A performance study of tetrahedral and hexahedral elements in 3-d finite element structural analysis. *Finite Elements in Analysis and Design*, 12(3-4):313–318, 1992.

[8] P. Cignoni, C. Montani, E. Puppo, and R. Scopigno. Optimal isosurface extraction from irregular volume data. *1996 Volume Visualization Symposium*, pages 31–38, 1996. ISBN 0-89791-741-3.

[9] W. A. Cook and W. R. Oakes. Mapping methods for generating three-dimensional meshes. *Computers In Mechanical Engineering*, CIME Research Supplement:67–72, August 1982.

[10] J. F. Dannenhoffer(III). A block-structuring technique for general geometries. In *29th Aerospace Sciences Meeting and Exhibit*, volume AIAA-91-0145, January 1991.

[11] I. Fujishiro, Y. Maeda, H. Sato, and Y. Takeshima. Volumetric data exploration using interval volume. *IEEE Transactions on Visualization and Computer Graphics*, 2(2):144–155, 1996.

[12] S. Gibson. Using distance maps for accurate surface representation in sampled volumes. In *1998 Volume Visualization Symposium*, pages 23–30, 1998.

[13] S. R. Jankovich, S. E. Benzley, J. F. Shepherd, and S. A. Mitchell. The graft tool: An all-hexahedral transition algorithm for creating multi-directional swept volume mesh. In *Proceedings, 8th International Meshing Roundtable*, pages 387–392. Sandia National Laboratories, October 1999.

[14] T. Ju, F. Losasso, S. Schaefer, and J. Warren. Dual contouring of hermite data. *ACM Transactions on Graphics*, 21(3):339–346, 2002.

[15] P. Knupp. Next-generation sweep tool: A method for generating all-hex meshes on two-and-one-half dimensional geometries. In *Proceedings, 7th International Meshing Roundtable*, pages 505–513. Sandia National Laboratories, October 1998.

[16] P. M. Knupp. Algebraic mesh quality metrics. *SIAM J. Sci. Comput.*, 23(1):193–218, 2001.

[17] P. M. Knupp. Hexahedral and tetrahedral mesh shape optimization. *International Journal for Numerical Methods in Engineering*, 58(1):319–332, 2003.

[18] S. N. Laboratories. Verdict mesh verification library (see http://cubit.sandia.gov/verdict.html), October 2005.

[19] M. Lai, S. E. Benzley, G. D. Sjaardema, and T. J. Tautges. A multiple source and target sweeping method for generating all-hexahedral finite element meshes. In *Proceedings, 5th International Meshing Roundtable*, pages 217–228. Sandia National Laboratories, October 1996.

[20] M. Lai, S. E. Benzley, and D. R. White. Automated hexahedral mesh generation by generalized multiple source to multiple target sweeping. *International Journal for Numerical Methods in Engineering*, 49(1):261–275, September 2000.

[21] Y. Livnat, H.-W. Shen, and C. R. Johnson. A near optimal isosurface extraction algorithm using the span space. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):73–84, 1996. ISSN 1077-2626.

[22] W. E. Lorenson and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics (Proceedings of SIGGRAPH '87)*, 21(4):163–169, 1987.

[23] MESQUITE: The Mesh Quality Improvement Toolkit, Terascale Simulation Tools and Technology Center (TSTT), http://www.tstt-scidac.org/research/mesquite.html, 2005.

[24] S. A. Mitchell. A characterization of the quadrilateral meshes of a surface which admit a compatible hexahedral mesh of the enclosed volumes. In *13th Annual Symposium on Theoretical Aspects of Computer Science*, volume Lecture Notes in Computer Science: 1046, pages 465–476, 1996.

[25] S. A. Mitchell and T. J. Tautges. Pillowing doublets: Refining a mesh to ensure that faces share at most one edge. In *Proceedings, 4th International Meshing Roundtable*, pages 231–240. Sandia National Laboratories, October 1995.

[26] P. J. Murdoch. *The Spatial Twist Continuum: A Dual Representation of the All Hexahedral Finite Element Mesh*. Published Doctoral Dissertation, Brigham Young University, December 1995.

[27] G. M. Nielson and B. Hamann. The asymptotic decider: Removing the ambiguity in marching cubes. In *IEEE Visualization*, pages 83–91, 1991.

[28] G. M. Nielson and J. Sung. Interval volume tetrahedrization. In *IEEE Visualization*, pages 221–228, 1997.

[29] S. Owen. A survey of unstructured mesh generation technology. http://www.andrew.cmu.edu/user/sowen/survey/index.html.

[30] X. Roca, J. Sarrate, and A. Huerta. Surface mesh projection for hexahedral mesh generation by sweeping. In *Proceedings, 13th International Meshing Roundtable*, volume SAND 2004-3765C, pages 169–180. Sandia National Laboratories, September 2004.

[31] R. Schneiders. A grid-based algorithm for the generation of hexahedral element meshes. *Engineering With Computers*, 12:168–177, 1996.

[32] R. Schneiders. An algorithm for the generation of hexahedral element meshes based on an octree technique. In *Proceedings, 6th International Meshing Roundtable*, pages 183–194. Sandia National Laboratories, October 1997.

[33] M. S. Shephard and M. K. Georges. Three-dimensional mesh generation by finite octree technique. *International Journal for Numerical Methods in Engineering*, 32:709–749, 1991.

[34] J. F. Shepherd. Hexahedral mesh generation constraints. *SCI Institute Technical Report*, UUSCI-2006-010, 2006.

[35] J. F. Shepherd, S. A. Mitchell, P. Knupp, and D. R. White. Methods for multisweep automation. In *Proceedings, 9th International Meshing Roundtable*, pages 77–87. Sandia National Laboratories, October 2000.

[36] M. L. Staten, S. A. Canann, and S. J. Owen. Bmsweep: Locating interior nodes during sweeping. In *Proceedings, 7th International Meshing Roundtable*, pages 7–18. Sandia National Laboratories, October 1998.

[37] F. R. S. Weiler and R. Schneiders. Automatic geometry-adaptive generation of quadrilateral and hexahedral element meshes for fem. In *Proceedings, 5th International Conference on Numerical Grid Generation in Computational Field Simulations*, pages 689–697. Mississippi State University, April 1996.

[38] V. I. Weingarten. The controversy over hex or tet meshing. *Machine Design*, pages 74–78, April 18, 1994.

[39] J. Wilhelms and A. V. Gelder. Octrees for faster isosurface generation. *ACM Transactions on Graphics*, 11(3):201–227, 1992.

[40] M. A. Yerry and M. S. Shephard. Three-dimensional mesh generation by modified octree technique. *International Journal for Numerical Methods in Engineering*, 20:1965–1990, 1984.

[41] Y. Zhang and C. Bajaj. Adaptive and quality quadrilateral/hexahedral meshing from volumetric data. *Computer Methods in Applied Mechanics and Engineering (CMAME)*, 195(9-12):942–960, 2006.

[42] Y. Zhang, C. Bajaj, and B.-S. Sohn. 3D finite element meshing from imaging data. *The special issue of Computer Methods in Applied Mechanics and Engineering (CMAME) on Unstructured Mesh Generation*, 194(48-49):5083–5106, 2005.

[43] Y. Zhang, C. Bajaj, and G. Xu. Surface smoothing and quality improvement of quadrilateral/hexahedral meshes with geometric flow. In *Proceedings, 14th International Meshing Roundtable*, pages 449–468. Sandia National Laboratories, September 2005.

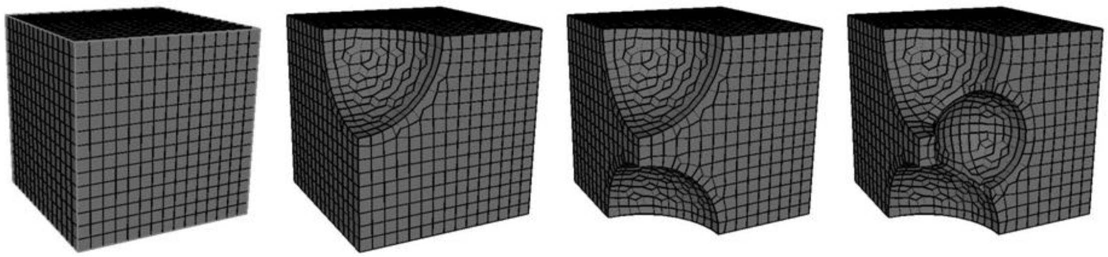

Figure 6: By inserting spherical sheets into the geometry, we can perform CG-type cutting operations in the mesh, while maintaining the integrity of the hexahedral mesh. At each of the boundary surfaces, the intersection of the spherical sheet with the original planar sheets in the cuboid mesh is sufficient to produce a hexahedral mesh with a string of mesh edges that can be utilized to capture the boundary discontinuities resulting from the spherical cuts.

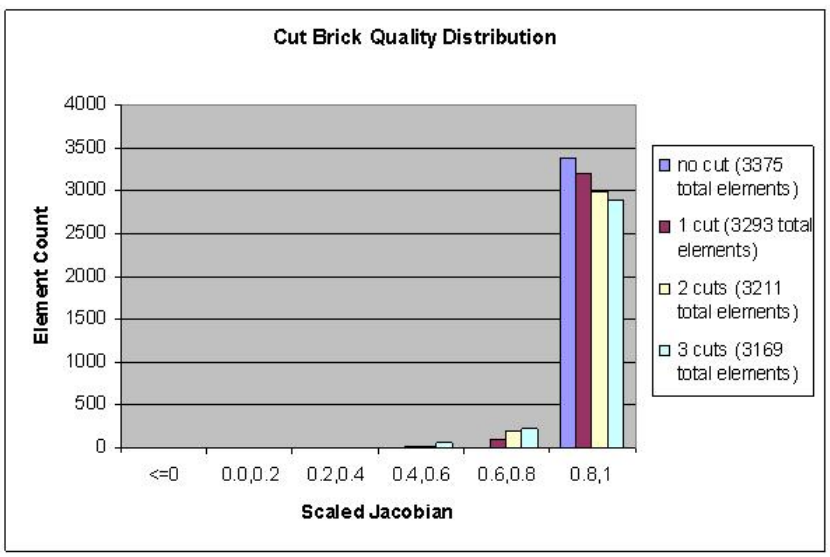

Figure 7: The distribution of scaled Jacobian values for the cuboid geometry with the spherical cutouts shown in Figure 6.

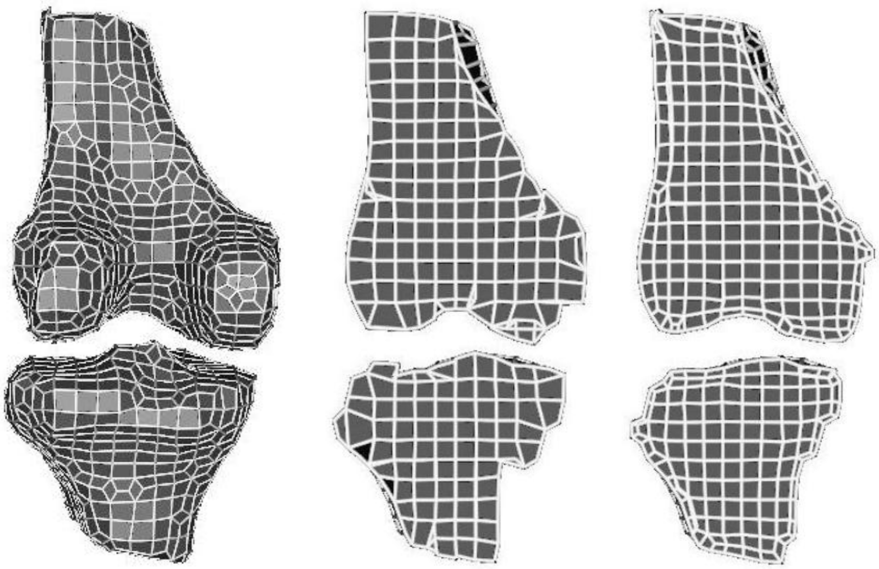

Figure 8: Human knee model. The original mesh contains 1338 total hexes whose surface boundary was optimized utilizing ‘geometric flow’ optimization. A cut-away view of the original mesh is shown in the center. An additional sheet of hexahedral elements was added to the boundary to improve potential quality resulting in a new mesh consisting of 2682 total elements, leaving the original surface mesh unchanged (shown on the far right).

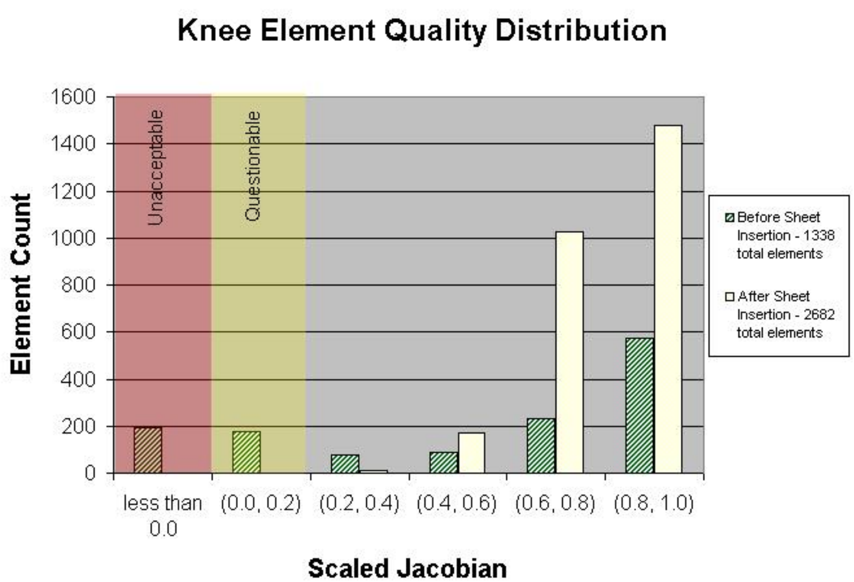

Figure 9: Distribution of element quality based on the scaled Jacobian measure of each element. The mesh optimized with ‘geometric flow’ smoothing contains 367 elements of questionable or unacceptable quality, while the mesh with improved boundary capture has all elements with scaled Jacobians greater than 0.2.

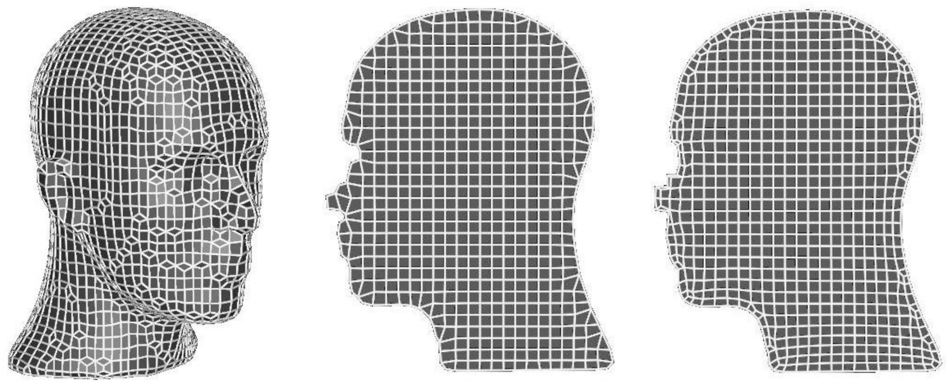

Figure 10: Human head model. The original mesh contains 6583 total hexes whose surface boundary was optimized utilizing 'geometric flow' optimization. A cut-away view of the original mesh is shown in the center. An additional sheet of hexahedral elements was added to the boundary to improve potential quality resulting in a new mesh consisting of 9487 total elements, leaving the original surface mesh unchanged (shown on the far right).

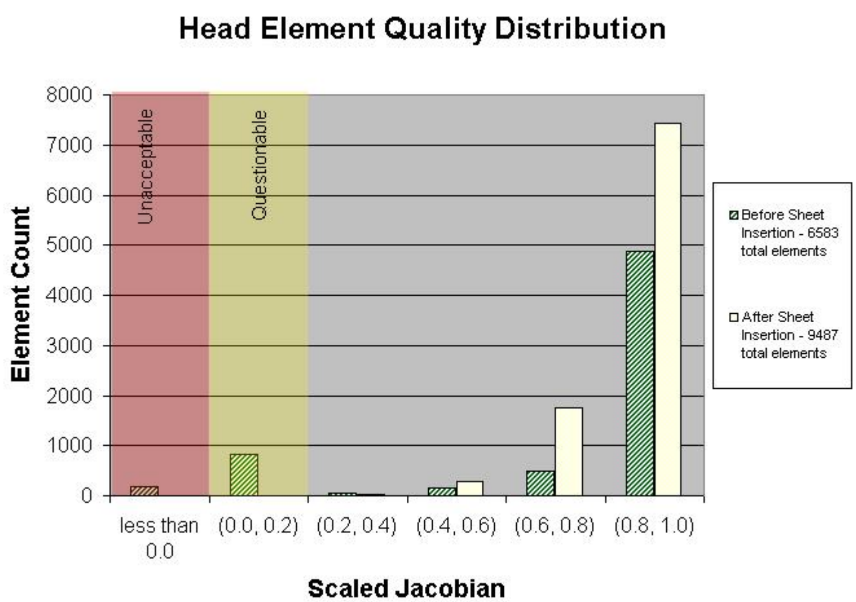

Figure 11: Distribution of element quality based on the scaled Jacobian measure of each element. The original octree mesh has 1025 questionable elements (i.e. scaled Jacobian greater than zero but less than 0.2) or unacceptable (scaled Jacobian less than zero). The mesh optimized with 'geometric flow' smoothing contains 1025 elements of questionable or unacceptable quality, while the mesh with improved boundary capture has only 1 element of questionable quality (due to poor quality of one of the boundary quadrilaterals).

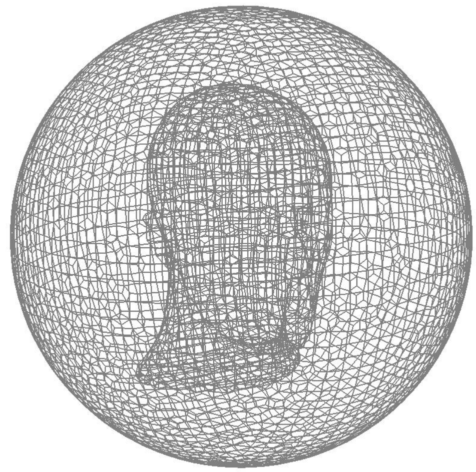

Figure 12: Human head model embedded in a spherical boundary mesh. The original mesh contains 13352 total hexes whose surface boundary was optimized utilizing ‘geometric flow’ optimization. An additional sheet of hexahedral elements was added to the boundary to improve potential quality resulting in a new mesh consisting of 19412 total elements, leaving the original surface mesh unchanged. (Refer to Figure 4 for cut-away views of the mesh.)

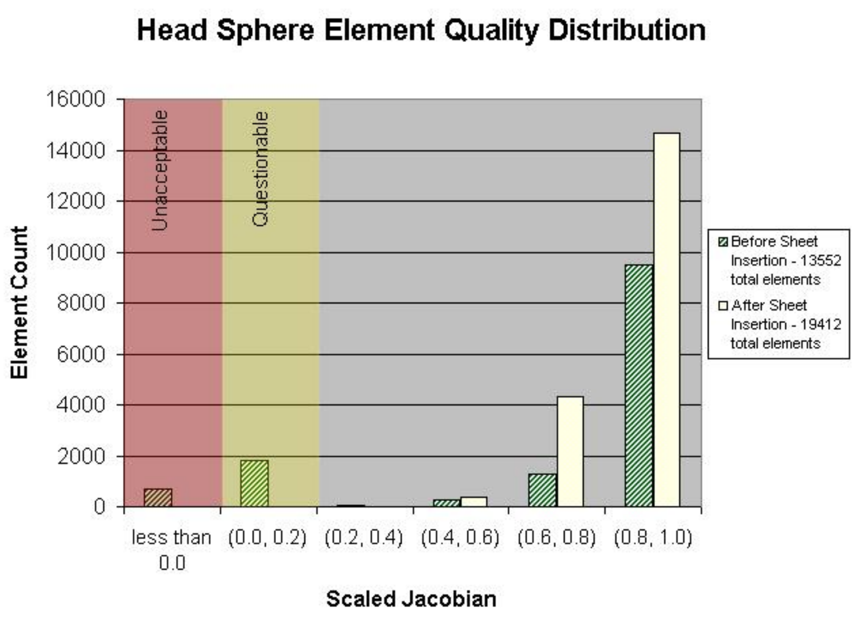

Figure 13: Distribution of element quality based on the scaled Jacobian measure of each element. The mesh optimized with ‘geometric flow’ smoothing contains 2485 elements of questionable or unacceptable quality, while the mesh with improved boundary capture has all elements with scaled Jacobians greater than 0.2.

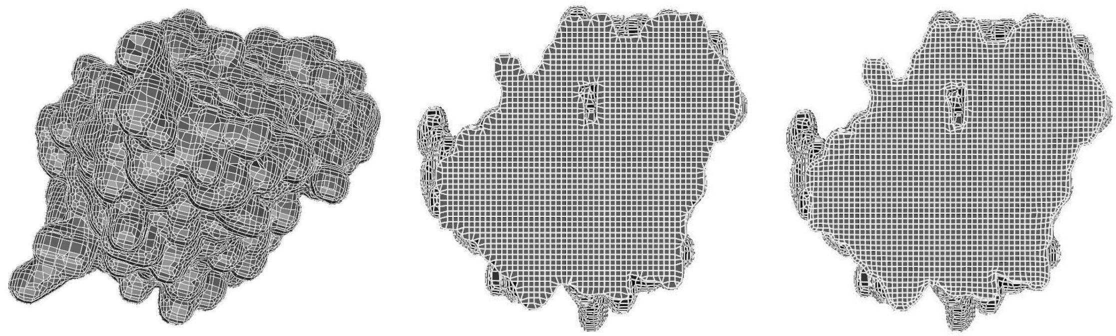

Figure 14: Biomolecule mAChE model. The original mesh contains 70913 total hexes. A cut-away view of the original mesh is shown in the center. An additional sheet of hexahedral elements was added to the boundary to improve potential quality resulting in a new mesh consisting of 90937 total elements, leaving the original surface mesh unchanged (shown on the far right).

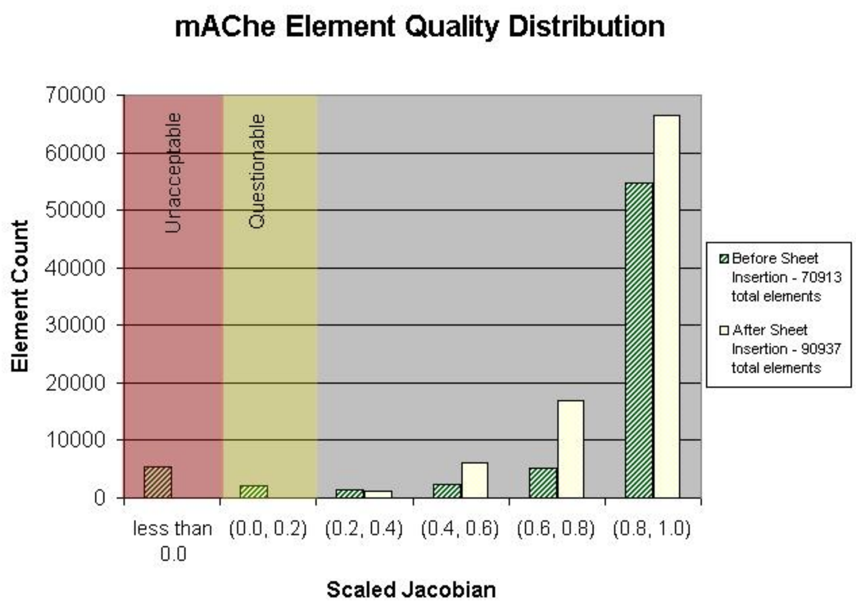

Figure 15: Distribution of element quality based on the scaled Jacobian measure of each element. The original octree mesh has 7298 questionable or unacceptable elements, while the mesh with improved boundary capture has 111 elements of questionable quality (due to poor quality of the boundary quadrilaterals).