2003-04-01

Sculpting: An Improved Inside-out Scheme for All Hexahedral Meshing

Kirk S. Walton
Brigham Young University - Provo

BYU ScholarsArchive Citation
Walton, Kirk S., "Sculpting: An Improved Inside-out Scheme for All Hexahedral Meshing" (2003). *Theses and Dissertations*. 3451.
https://scholarsarchive.byu.edu/etd/3451

SCULPTING: AN IMPROVED INSIDE-OUT SCHEME FOR ALL HEXAHEDRAL MESHING

KIRK S. WALTON

SCULPTING: AN IMPROVED INSIDE-OUT SCHEME FOR ALL HEXAHEDRAL MESHING

by

Kirk S. Walton

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Civil and Environmental Engineering
Brigham Young University
April 2003

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Kirk S. Walton

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

03-17-03
Date

Steven E. Benzley, Chair

17 MARCH 2003
Date

Richard J. Balling

17 March 2003
Date

Ray J. Meyers

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Kirk S. Walton in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

03-17-03
Date

S E Benzley
Steven E. Benzley
Chair, Graduate Committee

Accepted for the Department

Wayne C. Downs
Wayne C. Downs
Graduate Coördinator

Accepted for the College

Douglas M Chabries
Douglas M. Chabries
Dean, College of Engineering and Technology

ABSTRACT

SCULPTING: AN IMPROVED INSIDE-OUT SCHEME FOR ALL HEXAHEDRAL MESHING

Kirk S. Walton

Department of Civil and Environmental Engineering

Master of Science

Generating all hexahedral meshes on arbitrary geometries has been an area of important research in recent history. Hexahedral meshes have advantages over tetrahedral meshes in structural mechanics because they provide more accurate results with fewer degrees of freedom. Many different approaches have been used to create all-hexahedral meshes. Grid-based, inside-out, or superposition meshing all refer to a similar meshing approach that is very a common mesh generation technique.

Grid-based algorithms provide the ability to generate all hexahedral meshes by introducing a structured mesh that bounds the complete body modeled, marking hexahedra to define an interior and exterior mesh, manipulating the boundary region between interior and exterior regions of the structured mesh to fit the specific boundary of the body, and finally, discarding the exterior hexahedra from the given body.

Such algorithms generally provide high quality meshes on the interior of the body yet distort elements at the boundary in order to fill voids and match surfaces along these regions. The sculpting algorithm as presented here, addresses the difficulty in forming quality elements near boundary regions in two ways. The algorithm first finds more

intelligent methods to define a structured mesh that conforms to the body to lessen large distortions to the boundary elements. Second, the algorithm uses collapsing templates to adjust the position of boundary elements to mimic the topology of the body prior to capturing the geometric boundary.

ACKNOWLEDGMENTS

In writing this thesis I have had the aid of many special people who have not only influenced the direction of this work but also given me support and encouragement.

I would first like to thank Dr. Steven Benzley for offering me this job at the end of my sophomore year. Without his help and direction these past three years would have been much different.

I would also like to thank Jason Shepherd at Sandia National Laboratories for providing me with many of the ideas used in this thesis and for helping me troubleshoot programming problems.

Lastly, I would like to thank my family who has always given me their support and encouragement. Specifically, I would like to thank my sister Terri and her family, who not only helped me find this research position, but also invited me into their home for summer while I worked in Albuquerque on this project.

Contents

List of Figures

1 INTRODUCTION

Over the past several decades the finite element method has been used widely in the fields of structural mechanics, computational flow dynamics, and other fields because of its ability to break partial differential equations into a series of linear equations that can be easily solved using computers. As computers have become faster and more efficient, the finite element method has continued to gain momentum and support. A bottleneck in the analysis process has been and continues to be how to quickly and efficiently discretize geometric models into finite elements. For this reason, meshing, the process of discretizing a geometric domain, has been a focus of research over the last decade.

Meshing research has made significant advancements by providing the finite element community with fast and robust surface meshing algorithms for both triangular and quadrilateral elements as well as dependable tetrahedral meshing methods for volumes. Research continues though for an automatic hexahedral mesh generating algorithm for arbitrary geometries. Hexahedrons are preferred to tetrahedral elements because they can provide more accurate shape functions, directional sizing, and can decrease the overall element count[2]. While hexahedral element research has been the birthplace of many creative ideas only a few algorithms have found their way into mainstream use, such as mapping, submapping[24], grid-based approaches[18] and sweeping methods[11].

In a talk given at the 11th International Meshing Roundtable held at Cornell University in Ithaca, New York, Joe F. Thompson spoke on the need for art to influence the sciences and more specifically how "the problem of grid generation can still be as much an art form as it is a scientific discipline." He continued by stating, "creativity is the hallmark of the engineer."[23] In an effort to try new ideas the work described in this thesis investigates new directions in volumetric mesh generation. This work begins with a familiar path used by others, the method of superposition, grid-based, or inside-out meshing.

Grid-based meshing methods have been a form of mesh generation research for many years. In this method, a body is first overlaid with a structured grid. Elements are then removed from the structured grid to establish a set of elements that will be the basis for the specific meshing of the volume. While robust, these methods have been prone to provide poor quality elements near boundary regions.

Sculpting, the method presented here, is a new inside-out meshing algorithm that has been developed in an effort to increase the general quality of elements. Sculpting consists of the following steps:

1. Enclose a geometric model with a bounding box
2. Fill the bounding box with a structured mesh
3. Remove unwanted elements from the structured mesh
4. Collapse elements where appropriate
5. Match element nodes, edges, and faces to geometric vertices, curves, and surfaces
6. Add a boundary layer of elements when needed to improved mesh quality
7. Smooth the mesh

Steps taken to improve the quality of elements include aligning element layers with an axis of the element, collapsing elements onto neighbors to improve the initial mesh from which the volume mesh will be formed, and checking and adjusting elements in boundary regions for poor quality once the volume has been successfully meshed.

2 VOLUME MESHING TECHNOLOGIES

In the years that meshing has been a topic of research, many different methods have been studied. This chapter is provided as background to the myriad of three dimensional meshing algorithms currently available and also to suggest the limitations that these algorithms exhibit. This section also provides a background to the work others have done that directly has impacted the research presented in this thesis.

2.1 Mapping Algorithms

Mapping methods traditionally refer to a group of algorithms that create structured grids on a surface or on the interior of a volumetric body. A structured grid for a quadrilateral surface mesh is defined as having four element faces adjoined to each node. For a volumetric hexahedral mesh each interior node will have eight hexahedrons attached to it. For the purposes of this review, mapping algorithms will be defined as any meshing method that relies on the capability to create a structured grid on any of the volume's surfaces. These methods are bound to mapping and submapping.

2.1.1 Mapping

Volume mapping is a limited meshing method that only works on simple blocky or rounded elements that can be topologically modeled as a cube. This type of algorithm begins by identifying eight logical corners, which breaks the volume boundary into a shell of six mappable surfaces. Each of the six surfaces is then meshed by identifying four of the eight volume corners and constraining interval counts on opposite boundaries to be equal. Once the elements surfaces have been mapped, the algorithm uses the boundary shell to place interior nodes. An example of a mapped mesh is shown in the following figure.

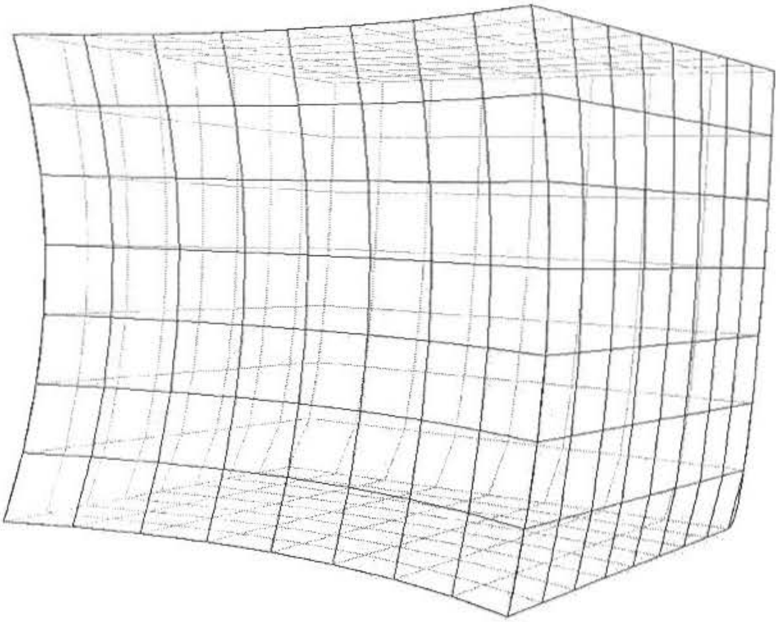


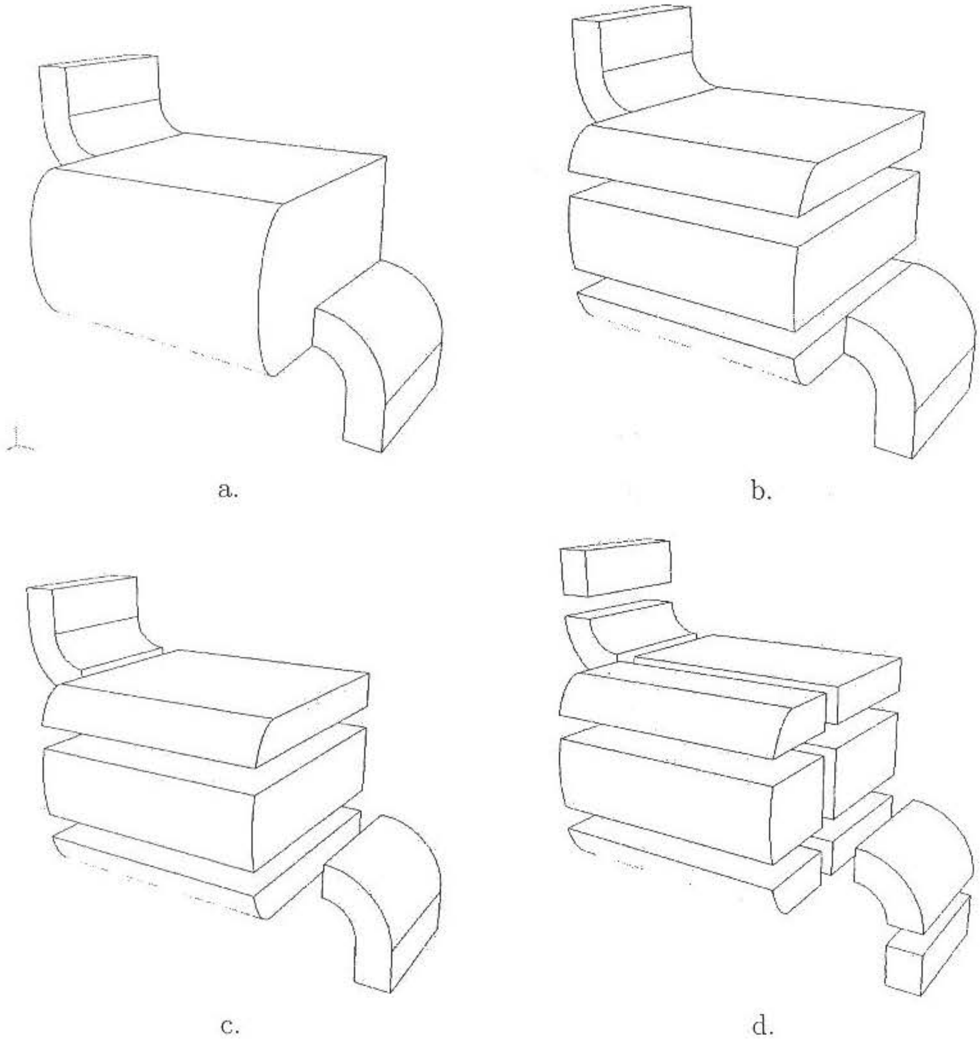
Figure 2.1: A mapped volume mesh.

Though successful on a limited number of geometries, mapping is a useful algorithm because it can quickly create a mesh with regular elements and no irregular nodes. Mapping also provides the following advantages[5]:

- Boundary Sensitivity: Well shaped elements that closely follow the shape of the boundary.
- Orientation Insensitivity: Repeatable and consistent results for all orientations of the underlying geometric body.

2.1.2 Submapping

Volumes that cannot be modeled as a rectangular shape can often be indirectly meshed using a mapping algorithm by first decomposing the geometry into rectangular, mappable regions. The decomposition of these geometries can be done manually or virtually. Submapping is an automated method to break complex models into a series of virtual sub-domains that mapping can then easily mesh[24].



a. b. c. d.

Figure 2.2: Subdividing a volume using submapping.

In order to divide a solid model into sub-domains, the algorithm needs to identify logical splitting planes. This is done by classifying geometric vertices by its interior angle as an end($\sim \pi/2$), sides($\sim \pi$), corners($\sim 3\pi/2$), or reversals($\sim 2\pi$). The model is then recursively split into sub-regions until all end or corner vertices have been eliminated. Figure 2.2 shows an example of a volume decomposition into mappable regions.

2.2 Unstructured Methods

Unstructured meshing algorithms refer to methods that often create nodes that are attached to more or less than four quadrilaterals for surface nodes and more or less than eight hexahedrons for interior nodes. The two methods that have shaped unstructured grid generation are advancing front algorithms and dual based meshing.

2.2.1 Sweeping

Sweeping is a very common and useful method to mesh volumes that have two topologically similar surfaces that are connected by mappable or submappable sides, or linking surfaces. The algorithm works by placing a surface mesh on one of the two similar surfaces as a source mesh and then propagating the surface mesh through the interior of the volume layer by layer until the second similar surface, the target surface, is reached[1][11]. Because the interior placement of nodes is only a projection of the source surface nodes' position, sweeping is considered a two and one-half dimensional meshing scheme. Figure 2.3 provides an example of a swept volume.

Continued research on sweeping algorithms has provided added capabilities to mesh volumes with multiple source or target surfaces, volumes of varying cross sectional area, and some multi-axis volumes[14][20][21][10][16].

2.2.2 Advancing Front

Advancing front algorithms were theoretically developed to mesh any three-dimensional model. Paving, a surface-meshing algorithm, is perhaps the best-known two-dimensional advancing front meshing technique and will be used to describe the advancing front process. Given a meshable domain as shown in Figure 2.4, paving starts at a corner of

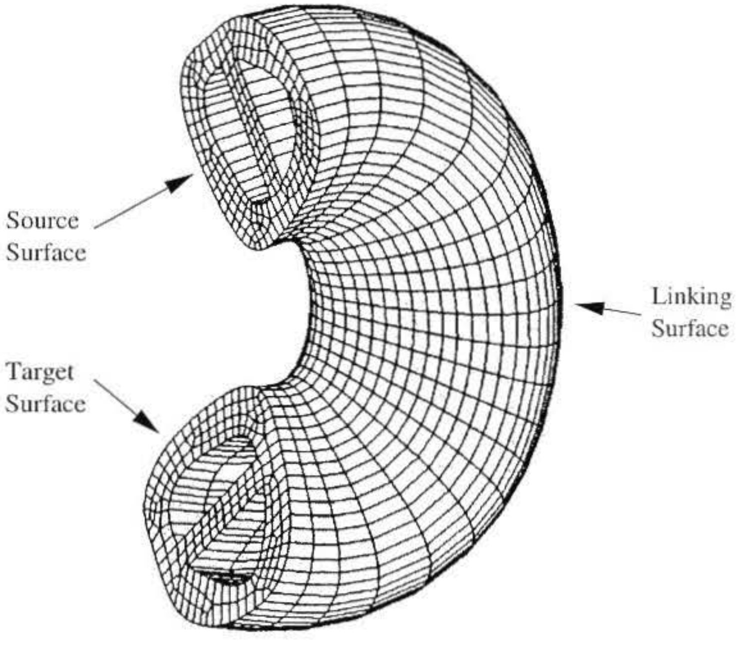


Figure 2.3: A swept volume.

the surface and uses mesh edges on the curves to build a quadrilateral face through the addition of an interior node.

The algorithm continues to add elements, one at a time, starting at the boundary layer and working its way to the interior of the volume[4]. Plastering, the volumetric extension of paving, similarly starts with a meshed boundary and begins placing hexahedra to the interior of the model until the volume is filled[7][3].

The benefits of advancing front algorithms include: being general enough to mesh any model, providing high quality elements near the boundary where quality is critical, and not being restricted by mesh interaction with multiple adjoining models. Unfortunately, plastering methods have not solved the volume-meshing problem. These algorithms have difficulty keeping track of element intersection and may end the meshing routine with coincident elements or voids. To eliminate voids or coincident elements some research has gone into methods that use an initial tetrahedral mesh and joins tetrahedrons together in order to form hexahedrons[17]. This technology eliminates interior voids, yet often is unable to combine all the tetrahedral elements to form hexahedrons,

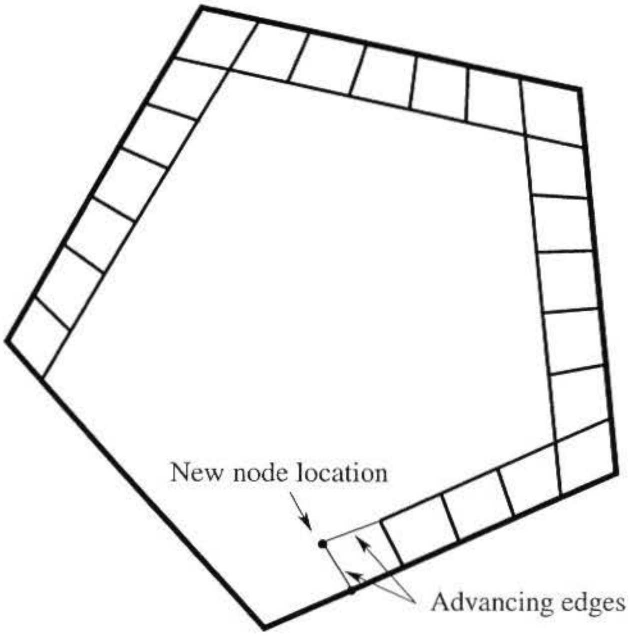


Figure 2.4: Paving a general surface.

leaving tetrahedrons in parts of the final mesh. Other methods have been proposed that use the dual of the mesh to keep track of the volume's interior. Whisker weaving, the most general of these methods, advances to the interior of the volume by intersecting dual sheets and places an element at the intersection of three sheets[22]. While being able to guarantee a mesh on any model that is topologically a ball, it is prone to creating inverted elements.

2.3 Grid-Based Methods

Grid-based meshing is also known as superposition or inside-out meshing. The premise behind this method is simple enough, build a structured grid sufficiently big around a geometric model and then force the elements to fit the model. The force fit method often requires eliminating elements to make this process easier. Successful grid-based methods require little user interaction, usually only the desired element size, and also provide a robust and general algorithm for any geometric model.

2.3.1 Schneiders Technique

Perhaps the most recognized work on grid-based algorithms is Schneiders advanced grid-based meshing, developed in the mid nineteen-nineties while working at MAGMA corporation[18]. By placing an axis aligned structured grid around the meshing volume, the algorithm proceeded to eliminate all the elements from the structured grid that did not fit entirely inside the model. As shown in Figure 2.5, the quadrilaterals from the initial interior mesh are then used as faces for hexahedra in the boundary layer. A projection from each face on the initial mesh to the boundary then creates the boundary hexahedral elements.

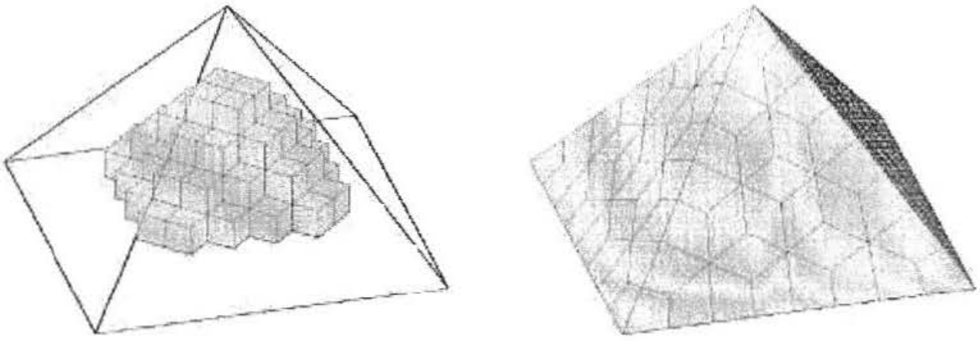
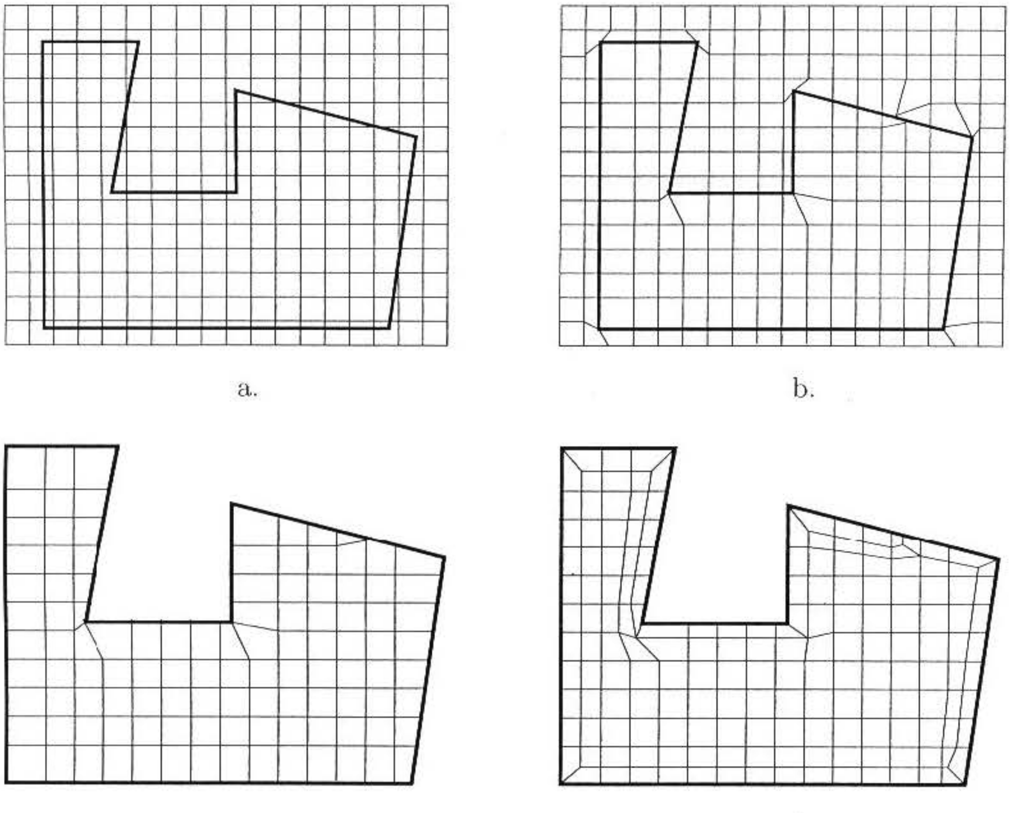


Figure 2.5: Initial mesh and isomorphic surface mesh.

Though general and robust, Schneiders' algorithm often distorts elements when mapping quadrilateral faces to areas containing geometric curves and vertices on the boundary. Local refinement to the initial mesh near geometric features can lessen the distortion of elements, but is not a guarantee for high quality elements. Additional distortion to boundary elements often occurs because of poor element layering. Depending of the orientation of the volume, the structured grid placed over the model may not share a common plane with any of the volume's surfaces, resulting in awkward element shapes at the boundary.

2.3.2 Additional grid-based research

Additional grid-based methods have been developed that vary from the basic approach that Schneiders implemented. The most common of these variants is a cutting type algorithm that begins by not removing elements for the superimposed grid, but local node movement around geometric features. Individual nodes are moved in order to align the faces of hexahedral elements with the boundary surface of the model. Once no element intersects the boundary, the mesh can be partitioned and the undesired portion of the mesh is discarded[9][13][6]. As with Schneiders' inside-out meshing algorithm, cutting methods suffer from distorted elements near boundary regions and poor element layering, though the insertion of an element layer near the boundary can often lessen the elements distortion. Figure 2.6 provides an example of the mesh cutting process.



a. b.

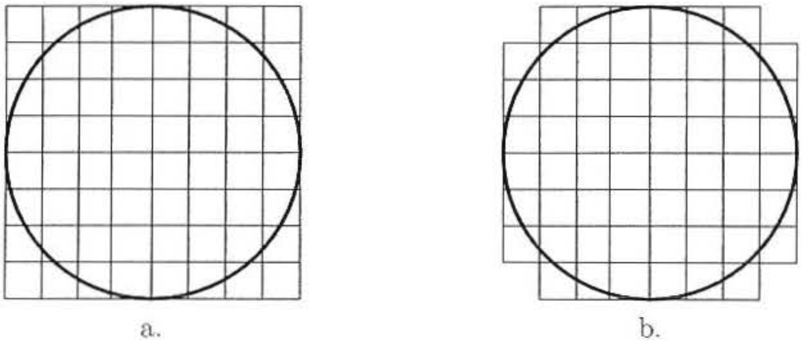
c. d.

Figure 2.6: Grid-based cutting method process.

3 OVERVIEW TO SCULPTING

Sculpting is a new inside-out or grid-based meshing method developed with the goal of improving element quality near boundary regions. Sculpting is comprised of several steps that will be developed in the following chapters of this thesis. The first of these steps is to enclose a geometric model with a bounding box. A bounding box is rectangular region that is sufficiently large to encompass a model. Bounding boxes are used as a guide to build a mapped grid, from which the initial mesh is formed. Bounding box selection is important because it affects the element layering of the initial mesh and when element are not properly layered, element quality decreases around planar geometric surfaces.

The second step of sculpting is to fill a bounding box with a mapped mesh. Figure 3.1a provides a two-dimensional example of a mapped bounding box around a circle. The



a. b.

Figure 3.1: The first three steps in sculpting, a. creating a bounding box and filling it with elements, b. removing elements to form an initial element set.

third step of sculpting is to remove unwanted elements from the mapped mesh to form an initial element set. Figure 3.1b is an example of an initial element set around a circle.

The fourth step in sculpting is to collapse elements where appropriate. While removing elements from the mapped mesh, element layers are shortened and when two consecutive layers are no longer the same length a stair-step, or jagged element layering is created as seen at the corners of Figure 3.1b. Collapsing elements is the process off joining edges in two-dimensions, or quadrilateral faces in three-dimensions in stair-stepped regions. The results of collapsing elements is seen in Figure 3.2.

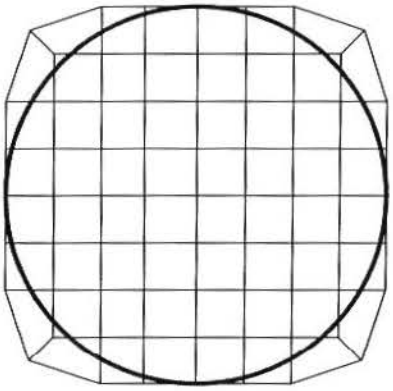


Figure 3.2: The fourth step in sculpting, collapsing elements.

The fifth step in sculpting is matching element nodes, edges, and faces to geometric vertices, curves, and surfaces. The sixth step is adding boundary layer elements around poor quality elements. Adding boundary layer elements is not a fully automated process and may require user interaction. The seventh and final step in sculpting is to rearrange element node locations by smoothing the mesh to improve the interaction between elements adjusted to fit the boundary and the underlying unchanged mesh. Figure 3.3 shows a smoothed sculpted mesh on a circle.

The following chapters will deal with these steps in more detail, using at times two-dimensional examples to help the reader visualize what sculpting is trying to do. In

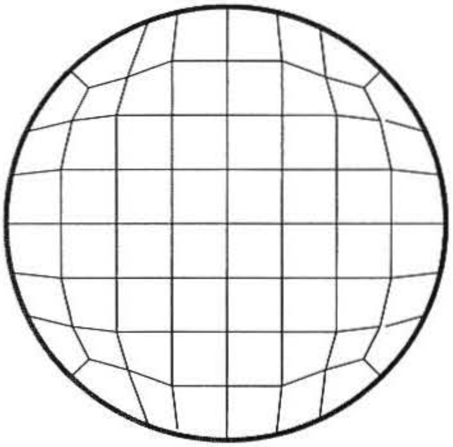


Figure 3.3: The fifth and sixth steps to sculpting, matching elements with geometric features and smoothing the mesh.

practice though, sculpting is a three-dimensional hexahedral meshing scheme and cannot be used on two-dimensional models.

4 INITIAL MESH GENERATION

The initial mesh used in a grid-based or overlay grid method is defined as the set of elements that remain when boundary elements are formed. For some algorithms this consists of a set of elements that lie entirely within the volume of the geometric model. Other methods may include elements that intersect the volume boundary and others may simply use an unchanged overlay grid. Regardless of what set of elements are selected as the initial mesh, the initial mesh often affects the quality of the final mesh near boundary areas. Distorted elements are common in boundary areas when geometric features do not line up properly with the alignment of elements. The alignment of most elements is dictated in the creation of the overlain structured grid. Within the grid, elements form layers that are parallel to three mutually orthogonal planes. These layers, if not selected properly, can cause jagged element-to-boundary intersections. Throughout this chapter consideration will be given to the possibilities that exist to create an initial mesh that will align elements more closely to the geometric boundary.

4.1 Bounding Box Creation

A bounding box, as shown in Figure 4.1, is a rectangular block that entirely encompasses the geometric model to be meshed. Bounding boxes have two useful qualities when creating an initial mesh. First, the sides of the bounding box are parallel to the layers that elements form in the grid and can be quickly used to decide the parallelism of element layers to the volume's surfaces. Second, bounding boxes quickly provide a minimal volume needed to form a structured grid large enough to surround the analyzed model. Because more than one kind of bounding box exists, sculpting has incorporated multiple bounding box choices to generate an initial mesh. These bounding box choices are named coordinate axes-aligned, tight-fitting, and user-generated boxes.

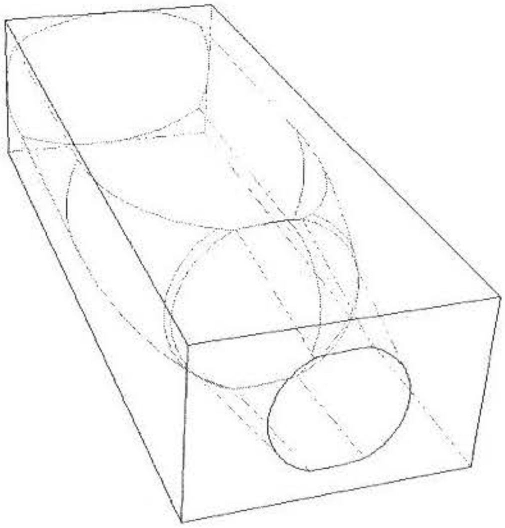
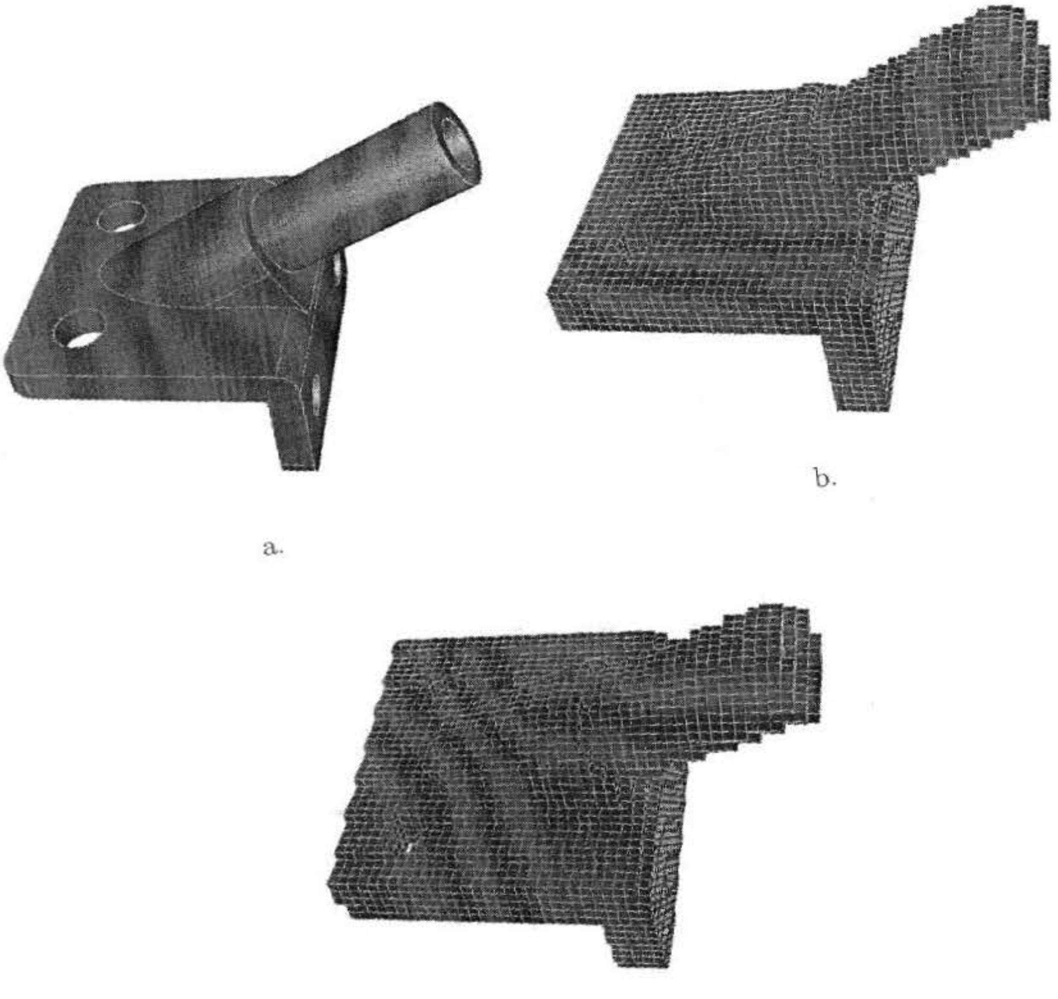


Figure 4.1: Bounding box for a typical model.

4.1.1 Coordinate Axis-Aligned Bounding Boxes

The easiest bounding box to create is a coordinate axes-aligned box that will leave element layers parallel to the global x, y, and z-axes. Because of its ease in creation, most grid-based algorithms have implemented coordinate axes-aligned grids as a basis for the initial mesh. Coordinate axes-aligned grids are not sensitive to the orientation of the object; often creating distorted boundary and unacceptable interior elements when surfaces of the meshing volume are not orthogonal to the global axes. The model in Figure 4.2a will be used to demonstrate the limitations of coordinate axes-aligned grids. In Figure 4.2b a coordinate axes grid has been placed around the model. Around the lower flange region of the model there is a smooth element to model boundary intersection and the mesh reasonably estimates the shape of the model. Around the upper cylindrical portion though, the mesh is rough around the boundary regions and poorly estimates the cylindrical shape. Alternatively, if the model is rotated slightly as seen in Figure 4.2c, the entire mesh becomes rough around the boundary, poorly estimating the entire model's shape.



a.

b.

c.

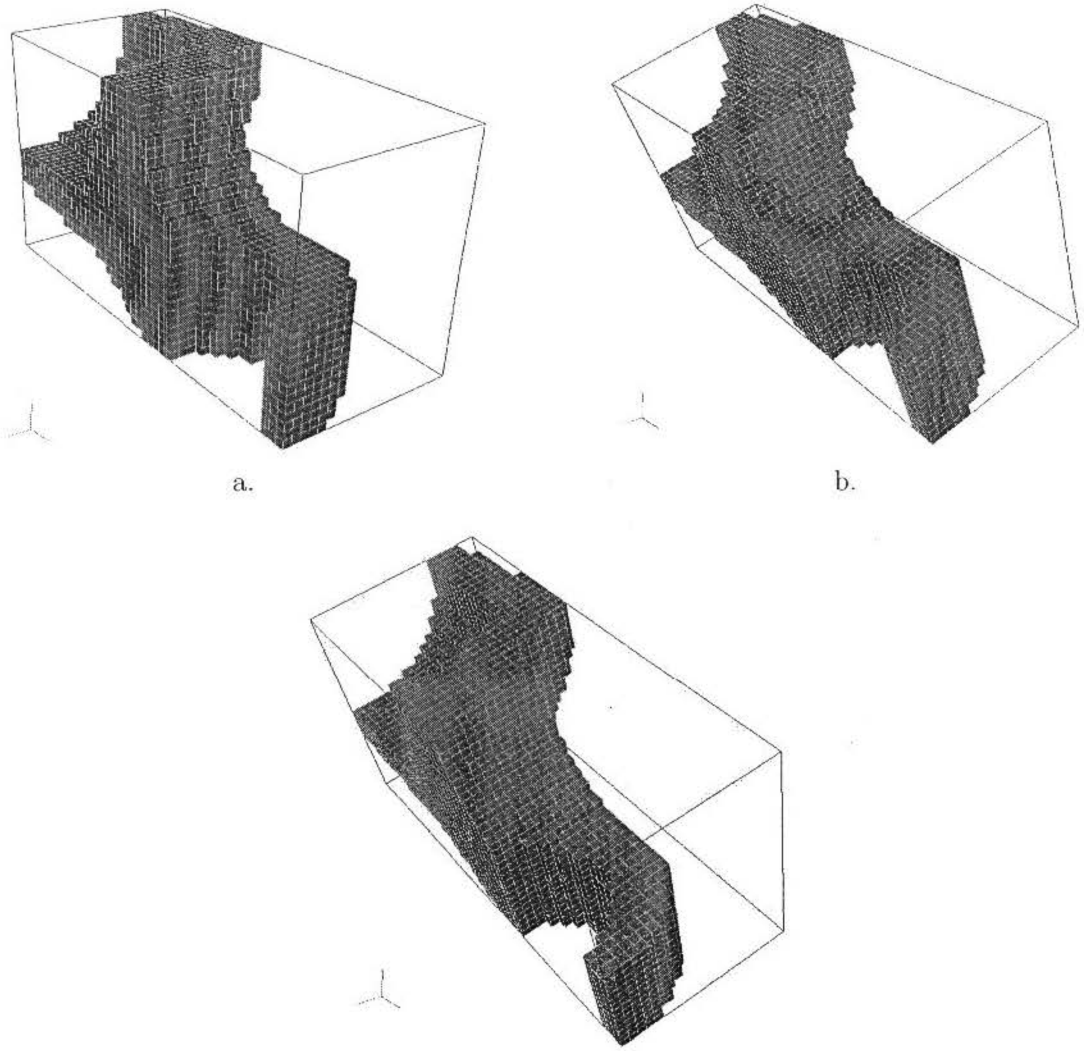
Figure 4.2: Results of a coordinate axes-aligned grid on a bracket.

Rough element boundary intersections poorly affect the quality of elements in boundary regions because trapezoidal shaped elements will be created to fill the void between the initial base elements and the boundary. Alternatively, pyramidal elements, elements that have two faces on the boundary, will be created if hexahedral elements are pushed to the boundary.

4.1.2 Tight Bounding Boxes

A possible solution to the orientation problem found with coordinate axes-aligned bounding boxes is to use a tight-fitting bounding box. A tight bounding box is the smallest rectangular volume that fits around any given model. Grid based methods can be benefited by tight-fitting boxes as the base for an initial mesh because the tight-fitting box will always provide the same structured grid for the model, regardless of orientation. Tight boxes also will decrease the size of the structured grid needed to surround the model, decreasing computation time on containment checks. Figure 4.3 is a good example of how a tight-fitting grid will produce a consistent grid for any orientation of the volume.

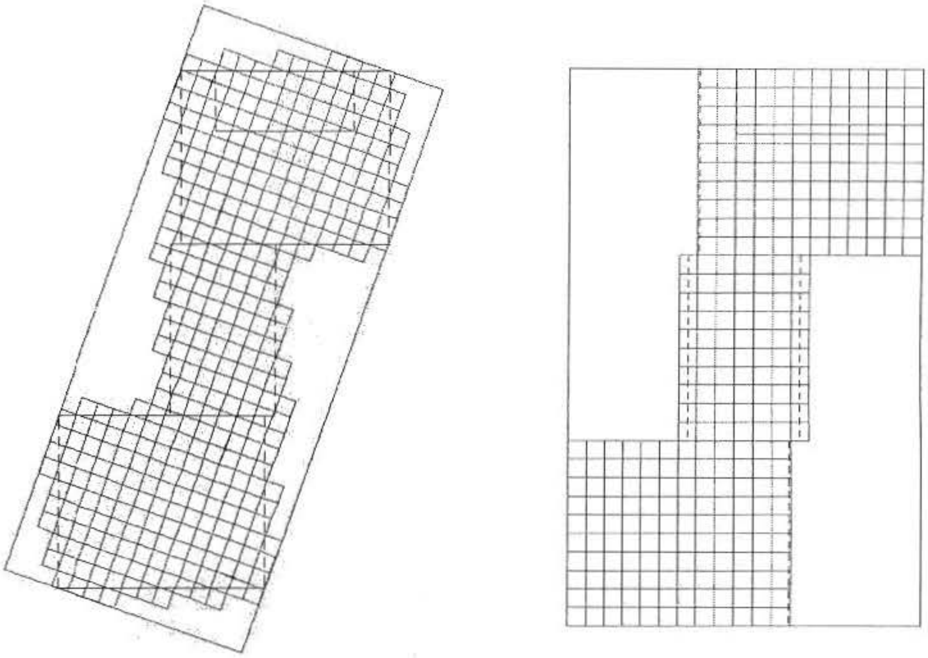
While tight bounding boxes are able to provide the same structured grid for any rotation or orientation of a model, they cannot guarantee the box orientation will be aligned with any of the planar surfaces of the model. The ability to recognize planar surfaces and find a box orientation aligned with at least one of these surfaces will be referred to as model sensitivity. For models as seen in Figure 4.3 model sensitivity is not a obvious problem because the general shape of the model is rectangular. The rectangular shape of the model will make the tight box creation simple and result in element layers being parallel to the model's surfaces. Figure 4.4a, however, provides an example where the smallest rectangular area for the volume shares no common surfaces with the tight box. The tight bounding box's ability to create the same grid for any orientation is not useful because the tight grid creates a rough element to boundary intersection, the same result a coordinate axes-aligned box would have given for a volume at an odd orientation. Figure 4.4b shows that for the models current orientation, a coordinate axes-aligned grid would have proved much more effective to line element layers up with the surfaces of the model, something a tight grid would never accomplish.



a. b.

c.

Figure 4.3: Results of tight bounding boxes on a hook model.



a. b.

Figure 4.4: Comparison between coordinate axes-aligned and tight bounding boxes.

Even though tight grids lack model sensitivity, they can still provide a consistent grid for any rotation, and for models as shown in Figure 4.3 the tight grid is very useful. Coordinate axes-aligned grids also can be useful in the right circumstances despite their shortcomings as seen in Figure 4.4b. The ultimate goal would be to have an algorithm that could create a model sensitive bounding region for initial meshes. This topic is discussed in section 4.1.4 in order to have a "smart" bounding box selection.

4.1.3 User Defined Bounding Boxes

The main goal of a grid-based algorithm is to provide the user with an all-encompassing algorithm that can mesh any geometry at the push of a button. A user defined bounding region would then be a contradiction to the general goal of said algorithms, requiring the user to identify multiple small boundary areas. Nevertheless, the user defined region method is presented as an additional option to the conventional use of one all-encompassing grid because of three main benefits. First, the user will generally select a set of boxes that will align element layers with planar geometric surfaces, effectively eliminating the guesswork done when dealing with coordinate axes-aligned or tight-fitting grids. Second, computational time to remove elements from the grid can decrease dramatically. Figure 4.5a shows a geometry that requires a fine grid to recover small geometric features, yet the main volume of a rectangular grid does not intersect the model. Computational time to remove the unneeded elements from this void is costly and may take longer than the time needed to mesh the boundary regions. Figure 4.5b is an example of a grid created by a series of small boxes that could be easily used to decrease the computational effort by concentrating elements in to areas the model actually is. Third, mesh clean up can be simplified. When using the option to define a bounding grid, it is assumed that the user has selected an element layer orientation that is advantageous to promote higher quality elements near the boundary. If this is done properly, there will be less need to remove or add elements in trouble regions.

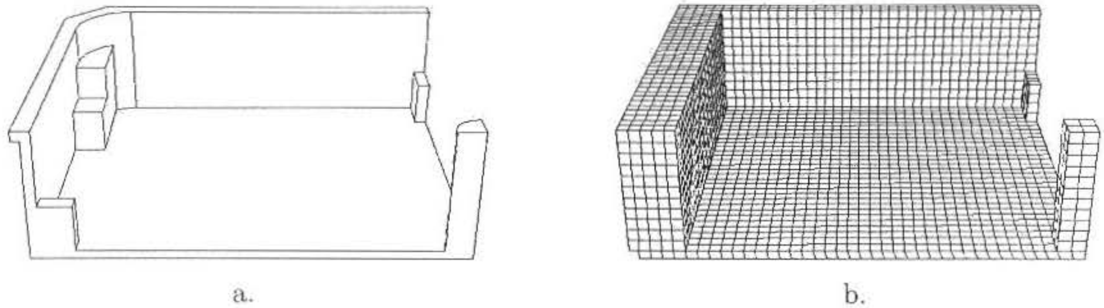


Figure 4.5: User defined bounding boxes for a model with a large void area.

4.1.4 Bounding Box Selection Criteria

As discussed previously, there needs to be a way to incorporate model sensitivity into the choice of bounding grids. To accomplish this goal, the sculpting algorithm has incorporated each of the previously mentioned bounding box options. A user can specifically choose to define a grid, or can allow sculpting to make the choice between a tight-fitting or axis-aligned grid. The selection to use an axis aligned box or a tight-fitting one uses a few simple checks to ensure that element layers are aligned with at least one of the geometric surfaces of the model.

When no user-defined grid is available, sculpting defaults to build a tight bounding box and then proceeds to check normals from planar faces of the volume for parallelism or perpendicularity with the principal axis of the box. A tight box is selected first because of orientation sensitivity of the tight-fitting grid. If there is parallelism between the volume's surface and the potential element layers, or if there are no available planar faces to check, a grid is constructed from the axes of the tight box and sculpting continues to the element removal from the grid. If the potential element layers are not aligned properly, sculpting will try building an axis-aligned box around the model and the same parallelism check is preformed. If the axis-aligned box fails, additional steps need to be taken to align element layers with the volume's surfaces. Currently sculpting will analyze the model for planar faces, preferably at least two orthogonal surfaces, and uses the axes defined by the surface's normals to rotate the these surfaces parallel to the global coordinate axes. Sculpting then rebuilds an axis-aligned block around the model, accomplishing the requirement that at least one of the model's surfaces will be parallel to element layers. Because rotations are needed to place the model into an orientation parallel with the global axes, the model should be rotated back into its original position once the grid is created.

4.2 Grid Reduction

Once a grid has been selected for a model the grid needs to be reduced to a set of elements that will define the initial mesh of the volume. The most commonly used initial mesh for any grid-based algorithm is a set of elements that residing entirely within the interior of the model. By restricting the initial mesh to only elements that are entirely within the volume other meshing methods need to recreate elements in these regions. Sculpting addresses grid reduction by discarding only elements that lie completely outside of the model. The decision to keep the intersecting elements was made to decrease computation time needed to recreate elements in these regions.

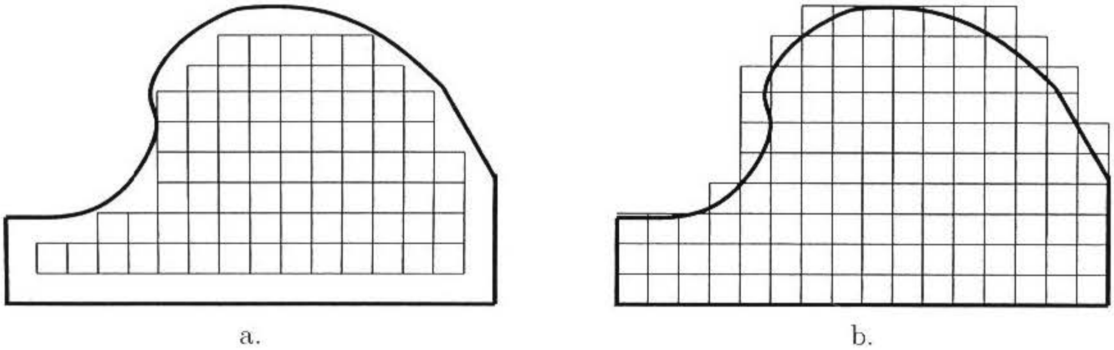

a. b.

Figure 4.6: Different sets of elements available for grid-based meshing.

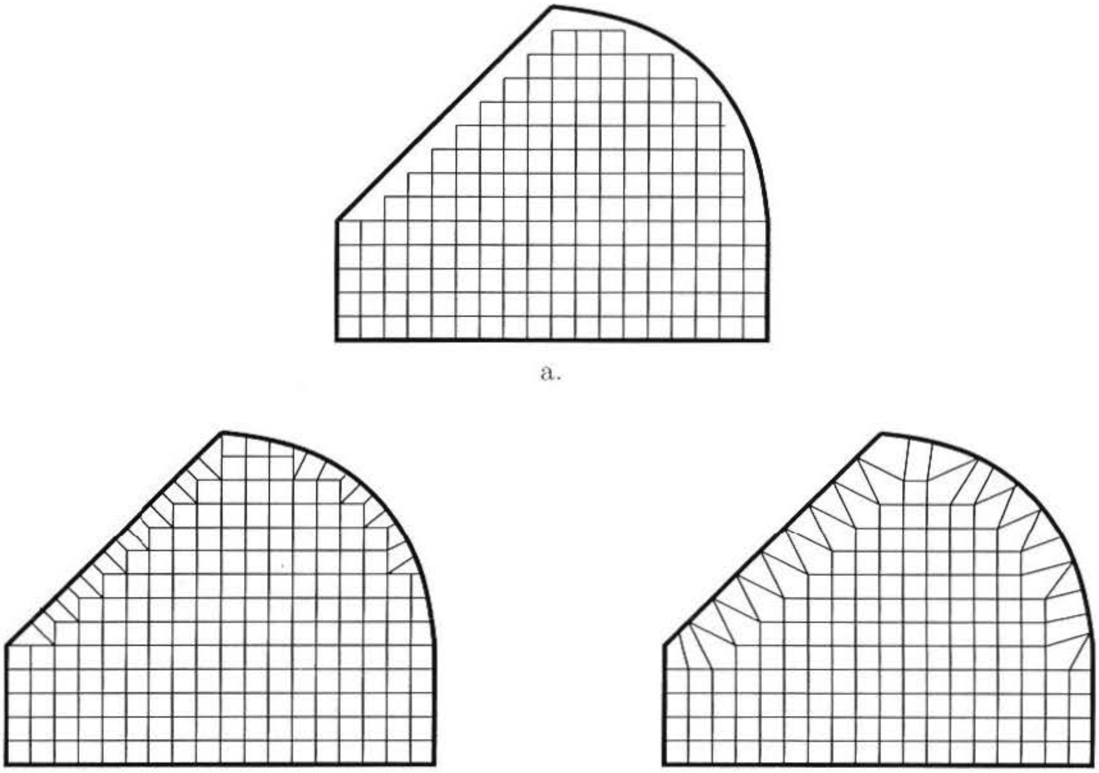
The selection process of what is inside and what is outside of the model can be done in many different methods. A complete yet inefficient method is to cycle through every element in the grid and identify each as an interior or exterior element. This method has a major benefit in that it can identify voids inside of the model. Sculpting has opted to restrict users from meshing models with interior voids in order to use a much more efficient method of identifying elements. The algorithm used to identify element location currently begins at the outer faces of the structured grid and looks for elements that are completely outside of the volume. Once one element is found it is marked as an exterior element and its neighbors are then recursively searched until intersecting elements are reached. If the recursive search completes without exploring all sides of the grid, new regions of the grid are searched until all the intersecting elements are found. By not identifying the interior elements of the grid, fewer containment checks are needed, speeding up the reduction process dramatically, especially for very large meshes. Once all the exterior elements are found, they are removed from the grid and sculpting is ready to manipulate the initial mesh to capture geometric features.

5 HEX COLLAPSING

The use of different bounding regions, as presented previously, can be used to improve the quality of element layers near planar surfaces. When bodies have non-planar surfaces, or planar surfaces that could not be aligned with a best-fit grid, no alteration of grid orientation can improve the element layering at the boundary. Sculpting however, introduces hex collapsing as a method to improve element alignment along these troublesome boundary areas. Hex collapsing is the process of removing the stair-steps or jagged edges out of an initial mesh. Hexes are considered collapsed when one of their exterior quadrilateral faces has been merged with another quad face about an adjoining edge. Collapsing elements connects orthogonal layers, forming a bent layer that can more closely approximate the curvature of a model. This chapter will consider the benefits of hex collapsing as well as the general process needed to identify collapsible areas and to perform the element collapses.

5.1 Benefits of Hex Collapsing

When curved surfaces are introduced into a model, structured grids loose their ability to accurately approximate the general shape of the model. Voids are created around curved features due to mutual orthogonality of element layers. The result is a stair-step formation of elements along the curved boundaries. Stair-step formation can also form if a model contains planar surfaces that were not orthogonally aligned with the orientation of the overlain-grid. When creating elements to fill voids or moving elements to boundary regions, these jagged element-boundary intersections often promote distorted element shapes. Figure 5.1 provides two, two-dimensional examples where elements have been distorted to match curved boundaries and in certain circumstances elements must be inverted in order to capture the boundary in the stair-stepped regions.



a.

b.

c.

Figure 5.1: Poor element quality near curves due to stair-stepped element layers, a. initial mesh, b. adding elements to fill boundary voids, c. moving nodes to boundary to fill voids.

To improve element quality in stair-stepped regions, sculpting introduces hex collapsing as a means of turning element layers onto themselves, creating bent element layers that more closely mimic curved features. The two-dimensional model in Figure 5.1 is shown below in Figure 5.2 with collapsed elements in stair step regions. The results of these element collapses, as shown in Figure 5.2a, are trapezoidal shapes whose exterior edge have a slope that is closer to the tangential component of the curved boundary. As seen in Figure 5.2b, the trapezoidal elements ensure that while some elements in the final mesh may be stretched in order to meet the boundary, none of the elements will be inverted to do so.

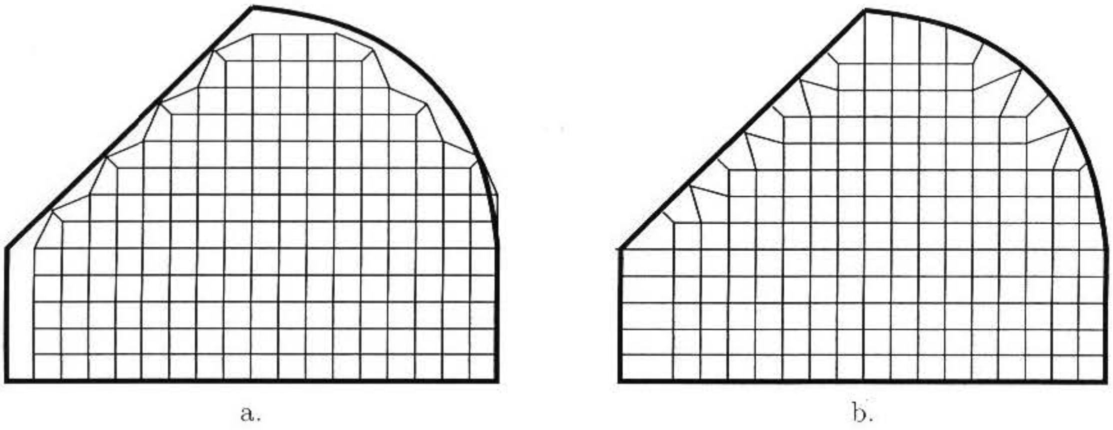


Figure 5.2: Collapsed elements around stair-stepped regions.

Collapsing hex elements onto their neighbors simplifies the boundary recovery process because elements have been adjusted to match geometric features and will need to be distorted less to conform to the boundary. This new approach, while helpful, does introduce the possibility of corrupting the integrity of the initial mesh because not all available stair-steps need to be, or should be collapsed. The following sections are presented to develop a smart approach to identifying which stair-step regions should or should not be collapsed.

5.2 Collapsing Process

Removing stair-steps from an initial mesh is a relatively simple process; exterior, orthogonal faces of neighboring hexahedral elements are joined about an exposed edge. The actual process of merging element faces is a local change, but to effectively remove stair-steps and actually improve the boundary matching capabilities of the initial mesh, each altered element needs to recognize the global effects to its action. The following will discuss the methods used to identify where potential collapses can occur and which of the identified selections will be useful.

5.2.1 Identifying Collapsible Edges

The first step taken in hex collapsing is to identify collapsible edges. A collapsible edge is defined as a series of boundary edges not included in the same element and that has three hexahedral elements and two boundary faces attached to each edge. The collapsible edges are found by searching boundary faces for an acceptable edge, as described above, and then recursively checking neighbor edges for an advancing edge. The collapsible edge grows in either direction from the beginning edge until suitable end points are found. Suitable end points are defined as points where the advancing collapsible edge cannot find a continuing edge that meets the above criteria or where the next available edge remains part of the same element as the current edge. Figure 5.3 illustrates examples of three types of edge points: free end points, where no advancing edge is available, and both open and closed intersecting end points, where an advancing edge is found to be owned by the same element as the current edge.

Ideally, all collapsible edges would find free end points. For the case of simple models this usually is the case. Free end points require little computational effort and the decision to collapse or not to collapse edges is usually based on edge interaction with neighbors rather than model constraints. As models become more complex, it is unrealistic to expect all free end points and often, intersecting end points out number free end points. Intersecting end points generally occur around special features of the model when two or three collapsible edges are connected to a common hexahedral element and usually require more model information to determine the usefulness of a potential collapse. The identification of intersecting end point type requires the edge's quadrilateral

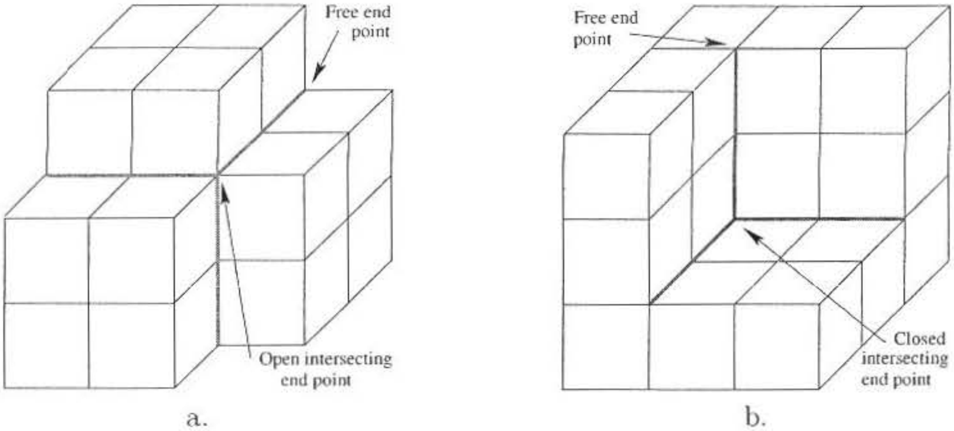
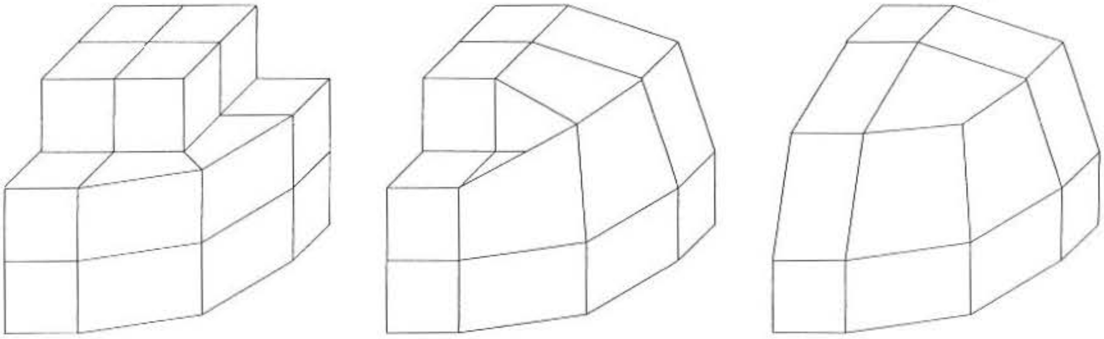


Figure 5.3: Example endpoints for collapsible edges.

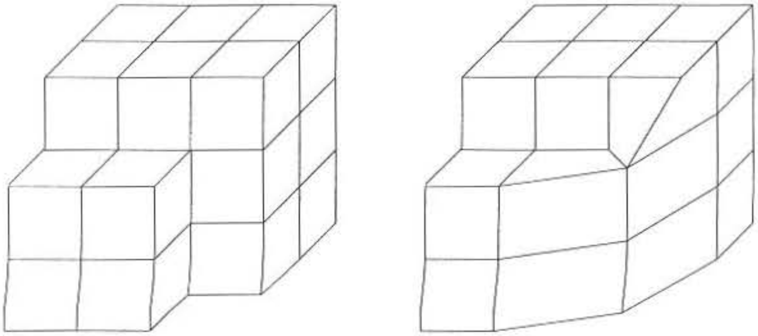
face connectivity and affects the collapsible edge's use. Open intersecting end points, as shown in Figure 5.3a, connect an advancing edge to the next available edge through the same hexahedral element. If both edges are connected to the same hex element and a common quadrilateral face, as seen in Figure 5.3b, the end point is marked a closed end point.

As end points of collapsible edge are being identified, the usefulness of certain edges is also being ascertained. While open end points created by three separate edges are retained and allowed to act independent of each other, being collapsed based on the needs of the geometry, an open end point created by only two edges will group both edges together and remove both from the list of possibilities. This occurs because collapsing either of the two edges would alter the mesh in an undesirable manner. Figure 5.4 provides examples of collapsing a single edge or multiple edges on a three-edge open end point and Figure 5.5 shows the undesirable effects of collapsing one edge on a two-edge open end point. When closed end points are encountered, the collapsible edges are also grouped together and removed as possible candidates. Unlike a two-point open intersection not one of the collapsible edges can be removed from this configuration without forming undesirable knife elements[8]. It would be possible to collapse all three edges, but this would form a void in the middle of the collapsed area where elements would not be conformal.



a. b. c.

Figure 5.4: Edge collapse combinations on open intersecting end points.



a. b.

Figure 5.5: Collapsing one edge on a two-edge open intersecting end point.

5.2.2 Collapsing Edges

Once all the useful collapsible edges have been identified, i.e. all edges with either free or three-edge open end points, sculpting can evaluate the need to collapse elements around geometric features. A mechanical part is used in the next set of figures in order to demonstrate this process. In Figure 5.6, possible collapsible edges have been highlighted. Notice that around the front of the model there are two-edge open end points as well as closed end points that have not been highlighted because they were eliminated from the process previously. The protrusion from the front of the model also demonstrates the types of geometric features where troublesome end points are often found.

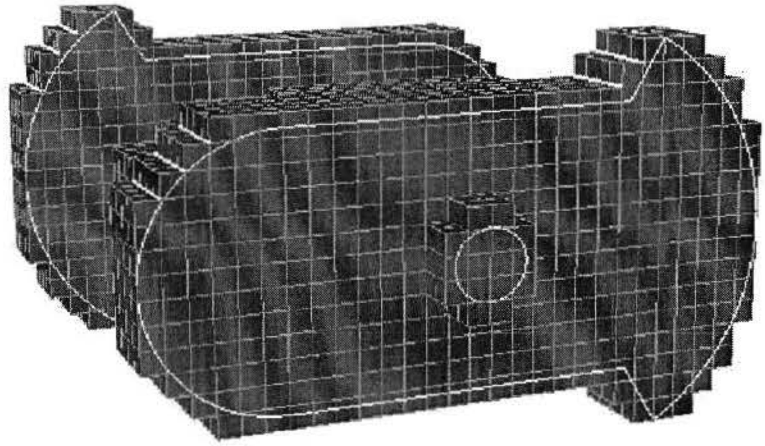


Figure 5.6: Possible edges available for hex collapsing on a mechanical part.

As seen in Figure 5.6 collapsible stair-steps are usually found around curved surfaces, coinciding with the geometric regions where collapsing helps the most. It is possible that a collapsing edge would alter elements that are currently parallel to the boundary of the volume. Figure 5.7a provides an enhanced view where two collapsible edges lie along a slanted surface. If the lower of the two edges were to be collapsed, it would bend elements parallel to a planar surface, adversely affecting the initial mesh.

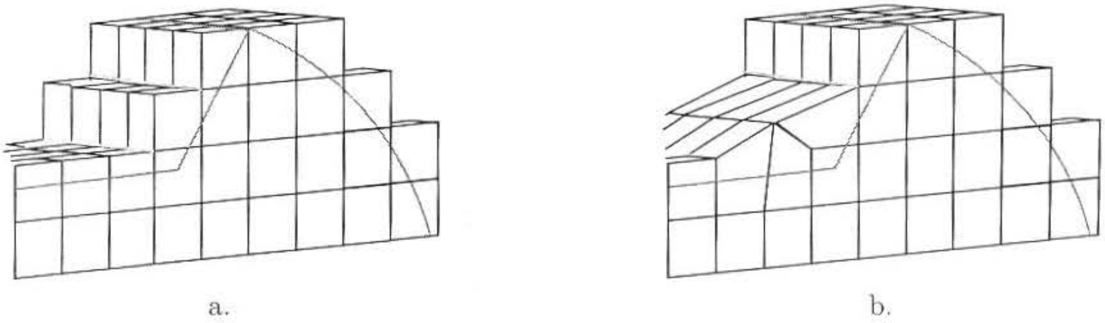

a. b.

Figure 5.7: Enhanced region of a mechanical part where hex collapsing is not desirable.

To avoid collapsing hexes that currently lie parallel to a boundary, sculpting uses a method of checking angles between normals from quadrilateral faces and underlying surfaces. If the angle between the normals is small, sculpting assumes that it is better to leave the elements as is and the collapsible edge is removed from the list. Once all parallel conflicts are removed, sculpting begins collapsing individual edge groups. A rule of collapsing is that no two consecutive edges can be collapsed. Figure 5.8a provides an example where three collapsible edges are consecutively connected by edges that will be merged when collapsed. If all three were collapsed then the elements would be twisted in an un-desirable manner as seen in Figure 5.8b. It is more desirable that two of the three edges act independent of each other as seen in 5.8c.

When sculpting encounters consecutive collapsible edges, like in 5.8a, sculpting arbitrarily collapses one of the edges and removes the closest consecutive edge from the list of possible collapses. This means that if two consecutive edges are encountered, only one of the edges will be collapsed. Figure 5.9 shows a mechanical part after hex collapsing has been preformed. Notice that sculpting has not collapsed all of the possible edges shown in Figure 5.6. At the base of the model where three consecutive edges were found, only two of the edges were collapsed as desired. Also at the top of the model,

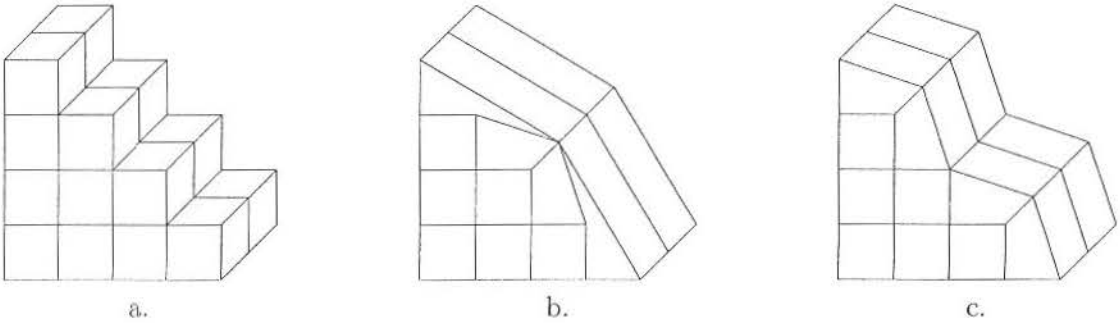


Figure 5.8: Collapsing consecutive edges.

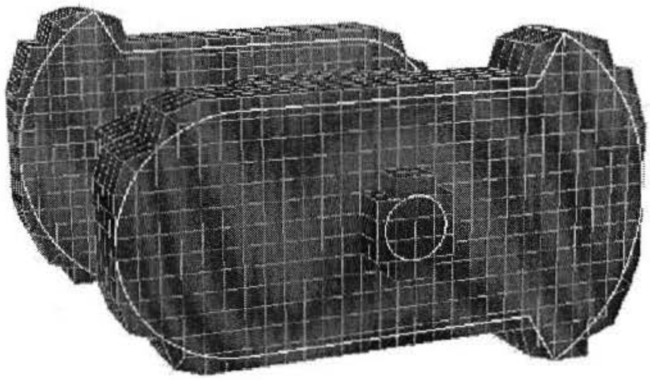
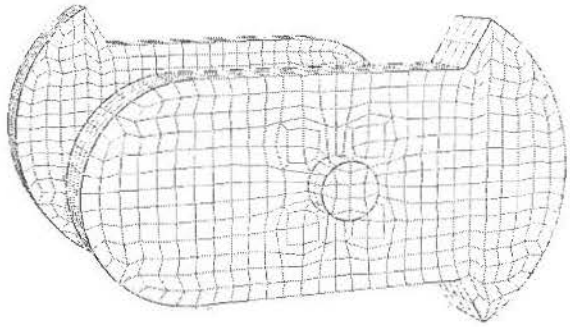


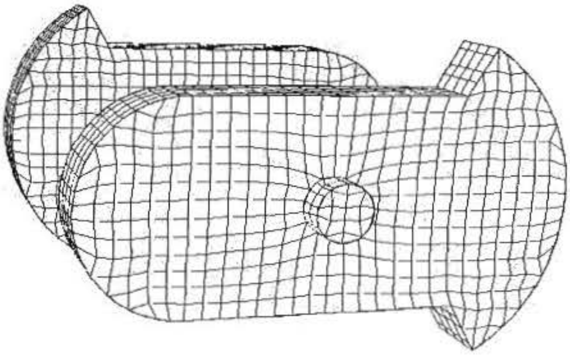
Figure 5.9: Collapsed hexes on a mechanical part.

near the sharp edge, two collapsible edges were found yet only one of the collapses was preformed. Due to the arbitrary nature of hex collapsing, the collapsing pattern is not equal for the top and bottom of the model near the sharp edges. In each case one of the possible collapsible edges was left unchanged, leaving stair-steps in these regions.

Hex collapsing does not correct all of the problems around the exterior of the initial mesh as was mentioned previously when two adjacent edges were not allowed to collapse. Regardless, sculpting can increase the overall quality of elements around the boundary. Figure 5.10 shows two meshes, one completed with hex collapsing and the addition of element layers or pillows of elements around poor quality elements and the other without. It is obvious that hex collapsing improved the quality of elements in most stair-stepped regions.



a. Sculpted mesh using hex collapsing



b. Sculpted mesh without hex collapsing

Figure 5.10: Comparison of sculpted all hex meshes on a mechanical part using hex collapsing.

6 BOUNDARY RECOVERY

Adapting a structured grid to match the boundaries of a volume is the most complicated step in grid-based meshing. Other methods often project quadrilateral faces to the geometric boundary, thus adjusting elements as they are created to fit geometric features. Sculpting uses an approach similar to that used by mesh cutting techniques that repositions element nodes, edges, and faces to match geometric vertex, curve, and surface locations[13][6]. Sculpting does have an advantage to mesh cutting techniques because the elements used to match the boundary have already been designated in previous steps. Boundary meshing by repositioning node, edge, and quad face locations can be done using two different methods. One is a surface to vertex approach that first considers quadrilateral face assignment to the correct surface, then matching edges to the surface's curves, and finally ensuring that each vertex has been assigned a node. The second approach is directly opposite to the first in that, nodes and edges are first assigned to geometric vertices and curves forming closed sets of quadrilateral faces. The sets of faces can then be quickly assigned to an underlying surface. The later of the two approaches has proved to be the more promising, though less robust approach at this time.

6.1 Surface To Vertex Meshing

Both boundary-meshing methods are based on breaking an initial mesh into logical sections that match the underlying geometry. When building the boundary mesh by first matching quadrilateral faces with geometric surfaces and then proceeding to match nodes to vertex locations, the concept is to quickly move nodes to the closest geometric surfaces and assign quad faces to an owning surface if all its nodes are currently located on the same surface. Quadrilateral faces that are not assigned an owning surface then must be repositioned to match curve and vertex locations. While this approach will quickly assign the interior quadrilateral elements to a geometric surface, multiple iterations are needed to correctly move elements straddling curves and vertices to a single surface.

A two dimensional model provided in Figure 6.1 is used to outline this process. This graphic shows a relatively simple surface that has initially been overlain with a structured grid and next the exterior nodes are moved to the boundary. Because the surface is easily contained in a box, the majority of the nodes currently are aligned with the boundary and do not require any adjustment. Along the cutout section however, there exists one layer of elements (i.e. the horizontal row) that matches the surface boundary, whereas a vertical layer intersects the boundary edge. The nodes along the vertical layer are directly moved to the closest surface, aligning the mesh edges with the geometric curve until the final edge straddles the corner vertex because the opposite node currently has a boundary position. Once a mesh has advanced to this point sculpting locates the elements with conflicting node owners and then determines which of the two elements needs to be corrected. This example demonstrates how node movement cannot

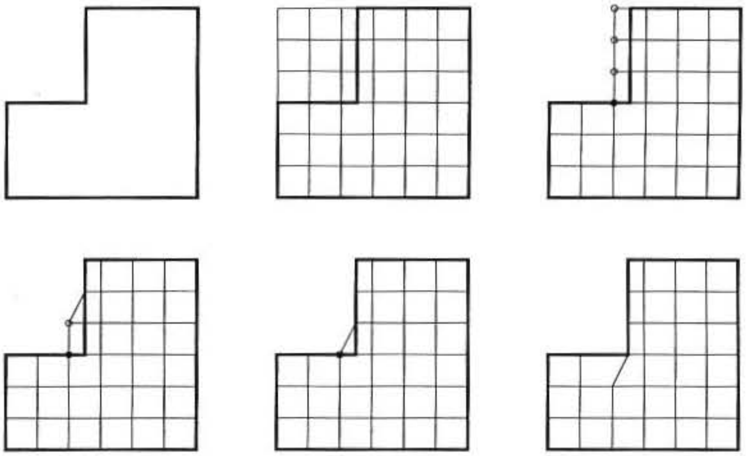


Figure 6.1: A two-dimensional example of boundary node movements around special features.

be simply based on placement to the nearest surface. The node at the interior corner must lie on both boundary edges.

Moving nodes to the boundary becomes more complicated in three-dimensional situations than what was shown previously in the two-dimensional example. The most difficult three-dimensional problem deals with node coincidence. Figure 6.2 shows a cylindrical surface extruded from a planar surface. Due to the selected element size, a box of hexahedral elements is being used to match the cylindrical volume protruding from the planar surface. Away from the curve joining the two surfaces, nodes are directly assigned to the curved surface. Close to the curved edge, corner nodes on the element block will try to move to the planar surface, which is the closer but incorrect surface. Moving the corner node to the planar surface would also place the node on top of an

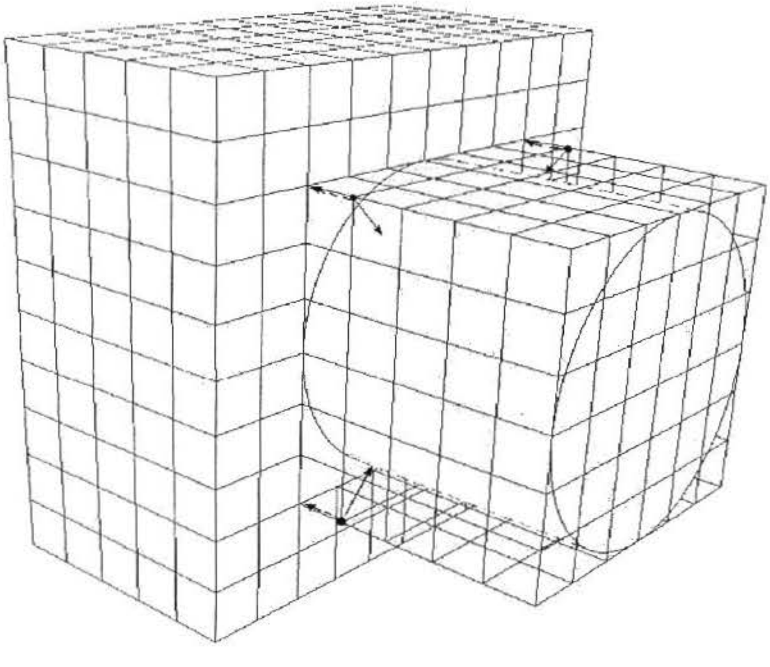


Figure 6.2: Extruded cylinder overlain with an initial mesh.

existing node. This is illustrated in the image as the dashed arrow. The correct node movement would be to follow the solid arrow shown. To avoid node coincidence, sculpting resolves boundary moves as follows.

1. An optimum (allowable) element size, h, is determined.
2. The distance from each node to all nearby surfaces is computed. The closest (i.e. minimum) distance is stored as d, and all other distances are saved an ordered array, t.
3. The distance d is compared to h. If it is roughly equal to h, the distances in array t are used to find a more suitable placement for the node.
4. Alternative moves are assessed by determining if the distance to a surface, d, is less than $\sqrt{3}$h, (i.e. $\sqrt{3}$h is the maximum distance a node could possibly move if a portion of the element is contained inside the volume). If no such move is found, then the node is moved as in the previous case.
5. If there is such a move, the neighboring nodes are checked to see which surface they have moved to. If none of these neighbors has been moved to the boundary, nodes are selected to find one that can move to a surface with a d less than h.
6. When this fails the original move is used

Figure 6.3 shows a volume after one iteration of nodal movements. There still are quad faces lie on two surfaces that need to be fixed for the mesh to be valid. These remaining quadrilateral faces are moved to curves and vertices based on distance and previous node movements. Edges that straddle the geometric curve on the face are identified and the distance to the other node's surface is calculated. If one of the nodes has been moved previously, it is given precedence and the other node is moved. If neither of the nodes has been moved previously then the closest to the other's surface is moved to the curve. A final check is made to ensure that all nodes on a quad face are connected to the same geometric surface. This check is needed because in certain circumstances a quadrilateral face may begin with as many as three nodes on a surface. It is important to notice that because more of the quad's nodes begin on one surface does not signify that

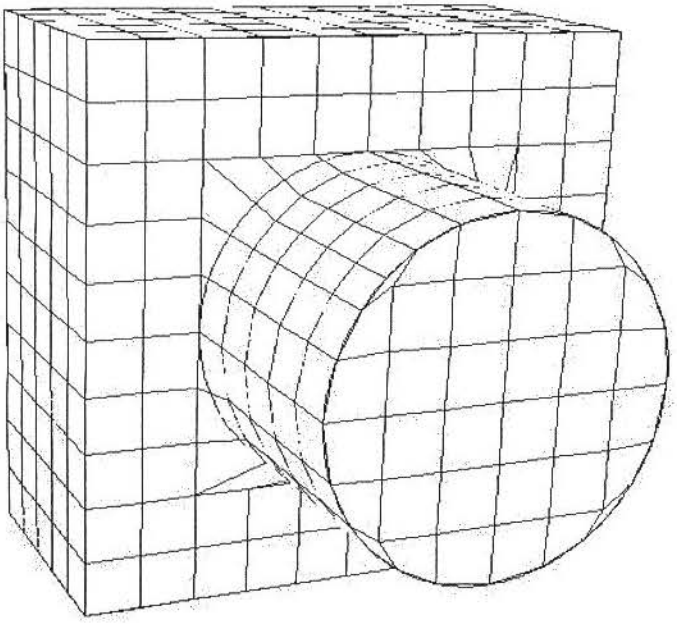


Figure 6.3: A three-dimensional example of node movements after one iteration in sculpting.

the surface with three nodes is the proper owner of the face. The edge checking routine will decide which face is most appropriate for the face, yet will fail to catch the middle of the three nodes. This occurs because when the straddling edges were first identified, the two edges pointing to the middle node where not straddling a curve. Usually in these circumstances the three nodes will be assigned to the curve that separates the two surfaces in question.

This simple heuristic algorithm has worked for many cases but is obviously not valid for all cases. Additional work has gone into resolving edge ambiguities. The result of this work is provided in the next section.

6.2 Vertex To Surface Meshing

An alternate approach to the method just presented is to first assign nodes to vertices and connecting edges between vertex nodes to curves. The assignment of mesh edges to curves will create closed loops that enclose sets of quadrilateral faces. Once the initial mesh is partitioned into quad sets, the faces can be quickly moved to underlying geometric surfaces. Figure 6.4 shows an initial mesh partitioned by a set of desired closed loops. We will refer to this graphic to check if sculpting is moving elements properly to the geometry.

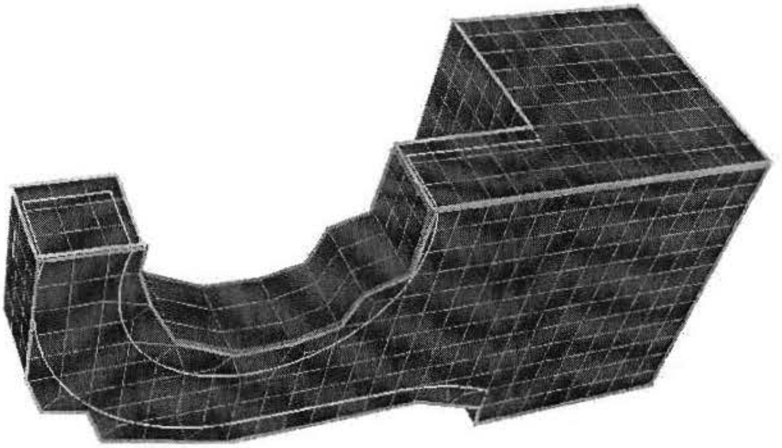


Figure 6.4: Partitioned initial mesh into desired surface mesh sections.

To move mesh entities to the boundary, sculpting begins by selecting a vertex from a surface. Around this vertex a group of boundary nodes is selected that lie within a maximum of 1.5h, where h is the element size and 1.5 is selected because it is just less than $\sqrt{3}$, the distance across the volumetric diagonal of a hexahedral element. Once the set of nodes is established, each of the nodes is checked and an ordered list is created based on distance to the vertex. The ordered list is then traversed comparing the number of hexahedral elements that a node is connected to. In most cases, vertices are located at convex corners or side locations along a model, so nodes with fewer attached hexahedral elements are given priority to nodes closer to the vertex. There are instances where a vertex is found at a concave corner. For these instances, sculpting tries to assign a node with five or seven attached hexes to the vertex. Figure 6.5 shows examples of the node selection process for convex and concave corner vertices.

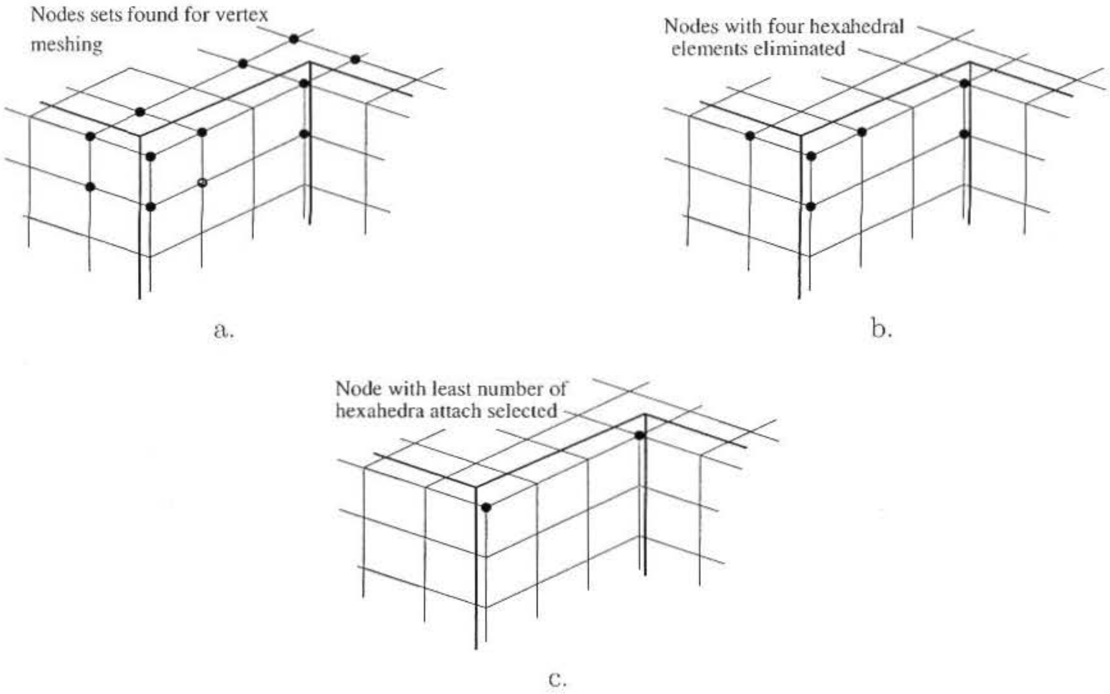


Figure 6.5: Meshing convex and concave vertices.

Once one vertex on the selected surface is meshed, sculpting begins meshing surface curves starting at the meshed vertex. Edges are added to the curve by looping through the edges attached to the current node and finding the closest opposite node to the edge, ensuring that the opposite node would not try to move to the current node location and that the opposite node is not attached to four hexahedral elements. Once a new edge is added to the curve, the process continues with the opposite node becoming the current node until a node satisfying the vertex meshing requirements described above is found. Curve meshing continues until all the curves on a surface are meshed and the beginning vertex is reached. Many of the specifications and requirements for curve meshing are based on assumptions that the mesh is structured, meaning four hexahedral elements are connected to interior surface nodes, and edge nodes will be connected to one to two elements. Interior curves present certain difficulties and there needs to be a method to define the angle between the two surfaces. In cases where two surfaces form an interior curve, the allowable number of hexahedral elements attached to a curve node can be increased to six. Figure 6.6 shows the hook model with the vertices and curves meshed. Sculpting has done a good job at following the desired loops define above as surface meshes.

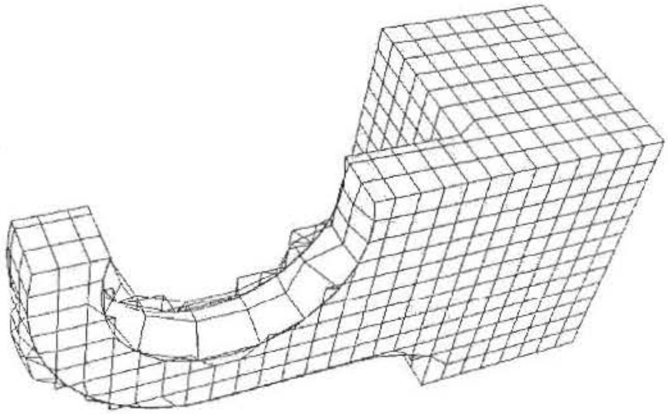


Figure 6.6: Hook model with curves and vertices meshed.

The last step required in sculpting to capture the geometric boundary is to assign enclosed quadrilateral faces to the underlying geometric surfaces. This process takes place after all the closed loops have been created. Each geometric surface is selected individually and a quadrilateral face attached to one of the surface's curves is selected as a starting point. From this starting face, a path across one of the closed loops is traversed until a face on an opposite curve on the surface is found. If a curve is found that does not lie on the surface, the opposite face on the starting curve is used to traverse the surface. Once a path has been found successfully across the mesh, all of the elements in the closed loop are assigned to the surface. This process continues until a path has been found across all the remaining quad sets and all the quad faces have been assigned to a surface. Figure 6.7 shows the final result of meshing from a closed loop method.

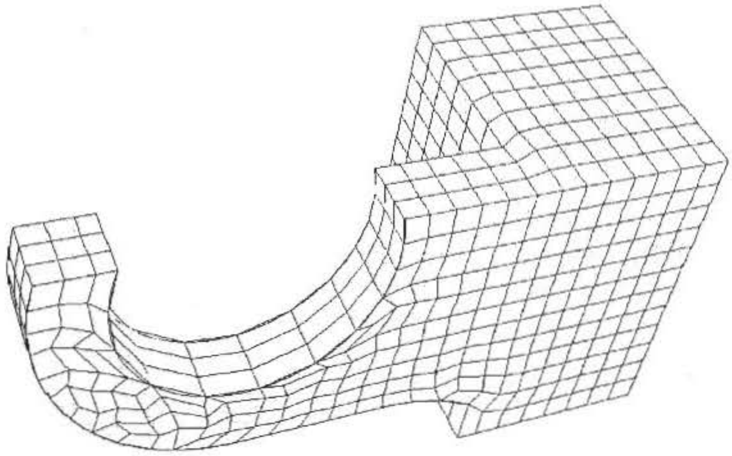


Figure 6.7: Final mesh of a hook element using a closed loop method.

7 EXAMPLES

Throughout this chapter the required steps sculpting uses to mesh geometric models will be demonstrated by showing the model to be meshed with a selected bounding box, the structured mesh with elements removed, the collapsed version of the stair-stepped mesh when applicable, and the final sculpted mesh.

7.1 Dumbbell Shape

This section shows the steps taken to mesh a dumbbell shaped model. The model was meshed using the surface to vertex approach of sculpting and hex collapsing was not preformed because no possible collapses would have improved the initial mesh quality. Figure 7.1 shows the volume surrounded by a specified bounding box. Figure 7.2 shows

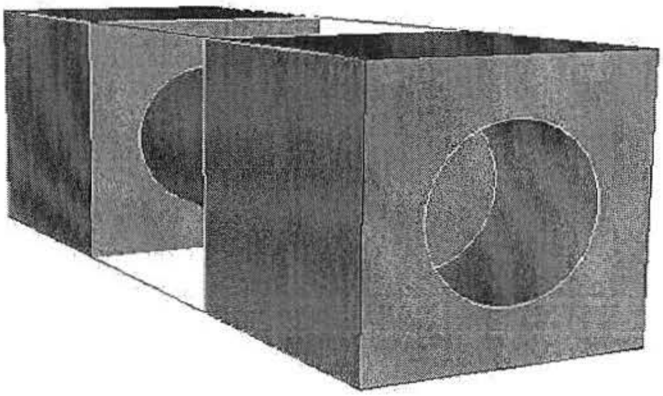


Figure 7.1: Dumbbell shape with selected bounding box.

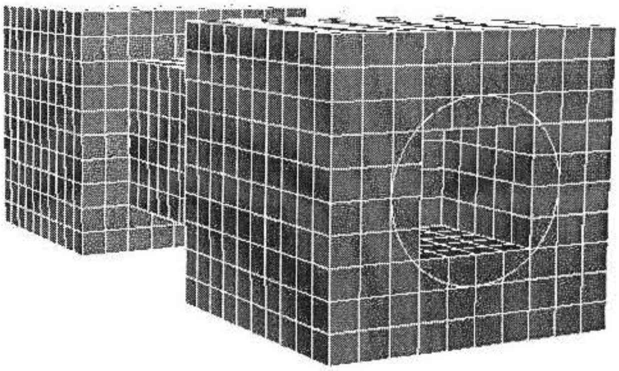


Figure 7.2: Structured grid surrounding a dumbbell shape with unwanted elements removed.

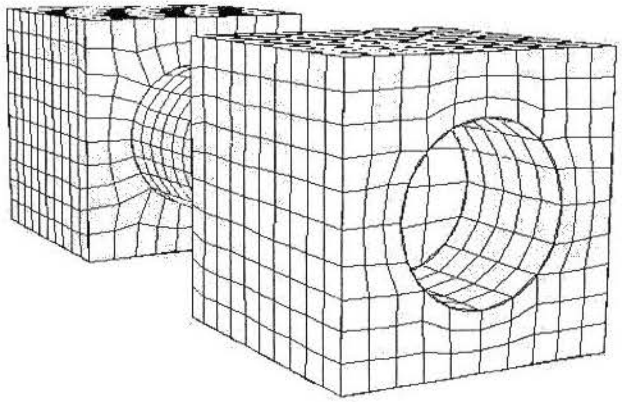


Figure 7.3: Sculpted all hexahedral mesh of a dumbbell shape.

the structured grid as seen after removing unwanted elements and Figure 7.3 provides the final sculpted mesh for the volume.

7.2 Half Torus

Meshing a half torus section required using the vertex to surface approach of sculpting. This section shows four steps of the sculpting process. The first image, Figure 7.4, shows the half section surrounded by a tight bounding box. The second image, Figure 7.5, shows the structured grid with unwanted elements removed. Hex collapsing is not preformed on this model because the interaction of collapsible edges is so great that nearly all have closed intersecting end points. The third image, Figure 7.6, shows the model after nodes and edges have been assigned to vertices and curves. The final image, Figure 7.7, provides the final sculpted mesh for the model. Note a boundary layer of elements has been inserted to eliminate hexahedral elements having more than one quadrilateral face on the boundary.

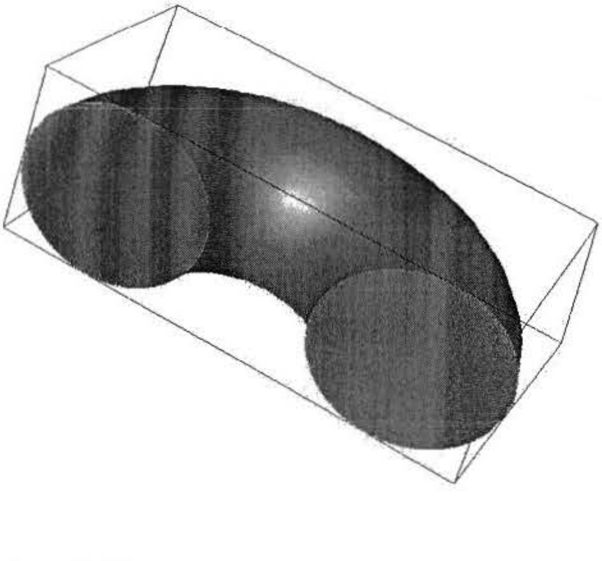


Figure 7.4: Half torus section with a selected bounding box.

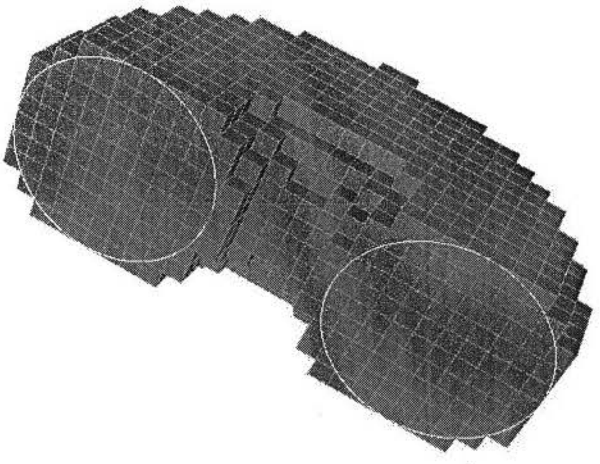


Figure 7.5: Half torus section surrounded by a structured grid with unwanted elements removed.

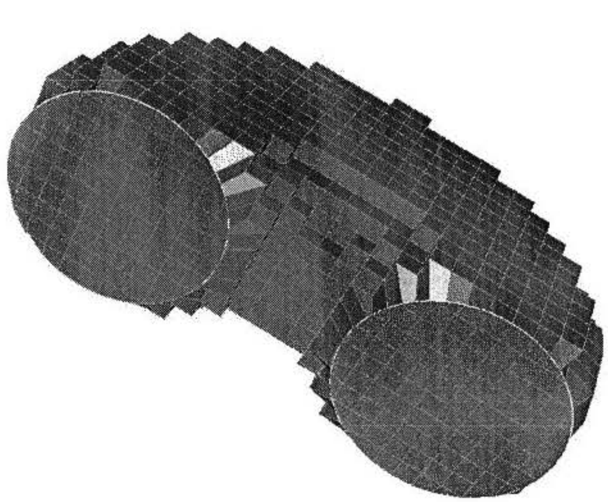


Figure 7.6: Half torus section with nodes and edges moved to match vertex and curve locations.

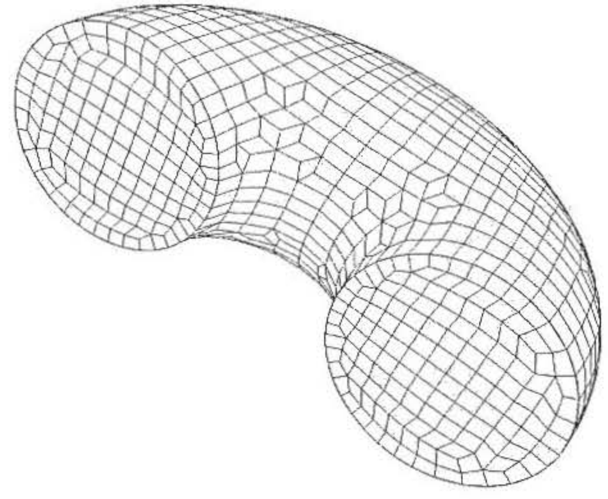


Figure 7.7: Sculpted mesh of a torodial section.

7.3 Block With Cutouts

The final example provided demonstrates sculptings' ability to mesh a block with torodial cutouts. The vertex to surface approach to sculpting was also required to mesh this model. Only two of the meshing steps are shown for this model. Obviously, the default bounding region for a box is the box and it is unnecessary to show an image of the selected bounding region. The first image, Figure 7.8, shows the structured grid with unwanted elements removed and the final image, Figure 7.9, provides the final sculpted mesh on the model.

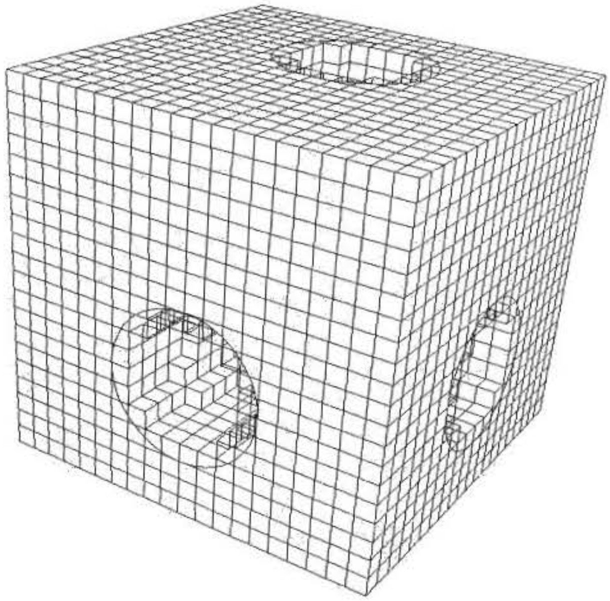


Figure 7.8: Block with torodial cutouts surrounded by a structured grid with unwanted elements removed.

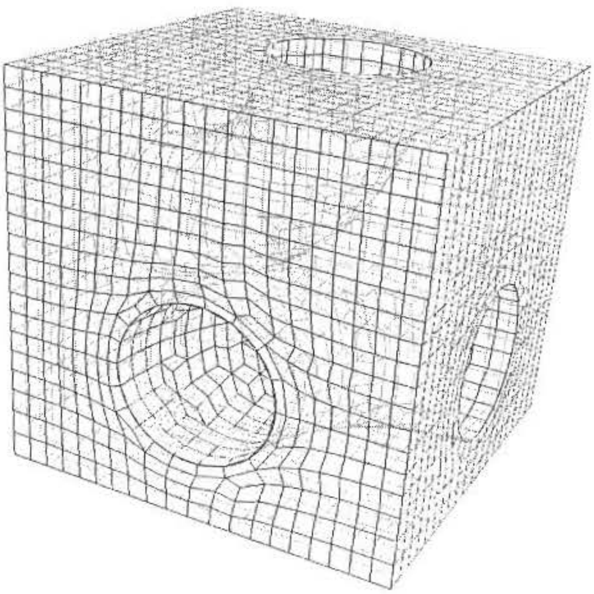


Figure 7.9: Sculpted mesh of a block with torodial cutouts.

8 CONCLUSION

Grid-based meshing algorithms are limited by the quality of elements in boundary regions. Element quality in these regions depends greatly on the superposition grid used to construct the mesh. Generally coordinate axis aligned bounding grids have been used, which distort elements and cannot guarantee the same mesh for different model orientations. This thesis has presented multiple alternatives to commonly used coordinate axis-aligned grids, namely tight-fitting bounding grids and user defined bounding grids, that can improve element layering around boundary regions and produce consistent meshes for different orientations. Providing the user with bounding box options is unique to sculpting and probably will be seen more in the future of mesh generation as the ability to totally automate algorithms decreases. For users that do not desire using a defined boundary, a generic algorithm has been presented that will select a grid that will align element layers with planar surfaces of a volume. Additional research could be preformed determine the likelihood of automating the decomposition of geometric models into small bounding regions, enabling sculpting to align the superposition grid more efficiently while decreasing the computational effort required to define an initial grid.

Boundary layer alignment, as mentioned above, is only one of the methods introduced by sculpting to improve the quality of boundary elements. Hex collapsing, the act of joining orthogonal element layers into a single bent layer by merging exposed quadrilateral faces about adjoining edges, has also been presented as a method to bend element layers near non-planar surfaces or diagonal surfaces in the structured grid. Bending element layers creates angled elements that more closely match the shape of these geometric features. While generally improving the initial meshes capability to fit geometries, hex collapsing can introduce poor elements if not done properly. Sculpting has worked to identify areas where hex collapsing would adversely affect the quality of elements and to consider the underlying geometric needs of a model.

Through the methods of smart superposition grid selection and collapsing stair-steps around curved and angled surfaces, sculpting has provided feasible methods to improve the general quality of elements in boundary regions. Sculpting has provided best results on geometric models whose collapsed stair-step mesh can be easily partitioned into sets of quadrilateral faces that match underlying geometric surfaces.

Bibliography

[1] BLACKER, T. The cooper tool. In *Proceedings, 5th International Meshing Roundtable '96* (1996), Sandia National Laboratories, pp. 13–29.

[2] BLACKER, T. Meeting the challenge for automated conformal hexahedral meshing. In *Proceedings, 9th International Meshing Roundtable '00* (2000), Sandia National Laboratories, pp. 11-19.

[3] BLACKER, T., MEYER, R. Seams and wedges in plastering: A 3D hexahedral mesh generation algorithm. *Engineering with Computers 2* (1993), 83–93.

[4] BLACKER, T., STEPHENSON, M.B. Paving: A new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering 32* (1991), 811-847.

[5] BLACKER, T., STEPHENSON, M.B., MITCHINER, J., PHILLIPS, L., AND LIN, Y. Automated quadrilateral mesh generation: a knowledge system approach, 1988. ASME Paper NO. 88-WA/CIE-4.

[6] BORDEN, M. J. Modification of all-hexahedral finite element meshes by dual sheet-insertion and extraction. Master's thesis, Brigham Young University, 2002.

[7] CANANN, S. Plastering and optismoothing: New approaches to automated, 3D hexahedral mesh generation and mesh smoothing. Ph.D. Dissertation, Brigham Young University, 1991.

[8] CLARK, B. W., BENZLEY, S. E. Development and evaluation of a degenerate seven-node hexahedron finite element. *Proceedings, 5th International Meshing Roundtable '96* 1996, Sandia National Laboratories, pp. 321-331.

[9] DHONDT, G. Unstructured 20-node brick element meshing. In *Proceedings, 8th International Meshing Roundtable '99* (1999), Sandia National Laboratories, pp. 369–376.

[10] JANKOVICK, S., BENZLEY, S., SHEPHERD, J., MITCHELL, S. An all-hexahedral transition algorithm for creating a multi- directional swept volume mesh. In *Proceedings, 8th International Meshing Roundtable '99* (1999), Sandia National Laboratories, pp. 387-392.

[11] KNUPP, P. Next-generation sweep tool: a method for generating all-hexahedral meshes on two-and-one-half dimensional geometries. In *Proceedings, 7th International Meshing Roundtable '98* (1998), Sandia National Laboratories, pp. 505–513.

[12] KNUPP, P. Applications of mesh smoothing: copy, morph, and sweep on unstructured quadrilateral meshes. *International Journal of Numerical Methods in Engineering 45* (1999), 37–45.

[13] MARECHAL, L. A new approach to octree-based hexahedral meshing. In *Proceedings, 10th International Meshing Roundtable '01* (2001), Sandia National Laboratories, pp. 209–221.

[14] MINGWU, L., AND BENZLEY, S. A multiple source and target sweeping method for generating all hexahedral finite element meshes. In *Proceedings, 5th International Meshing Roundtable '96* (1996), Sandia National Laboratories, pp. 217–225.

[15] MITCHELL, S., AND TAUTGES, T. Pillowing doublets: refining a mesh to ensure that faces share at most one edge. In *Proceedings, 4th International Meshing Roundtable* (1995), Sandia National Laboratories, pp. 231–240.

[16] MIYOSHI, K., AND BLACKER, T. Hexahedral mesh generation using multi-axis cooper algorithm. In *Proceedings, 9th International Meshing Roundtable '00* (2000), Sandia National Laboratories, pp. 89–97.

[17] OWEN, S. Constrained triangulation: Application to hex-dominant mesh generation. In *Proceedings, 8th International Meshing Roundtable '99* (1999), Sandia National Laboratories, pp. 31-41.

[18] SCHNEIDERS, R. A grid-based algorithm for the generation of hexahedral elements meshes. *Engineering with Computers, v 12, n 3-4* (1996), 168–177.

[19] SCHNEIDERS, R., SCHINDLER, R., AND WEILER, F. Octree-based generation of hexahedral element meshes. In *Proceedings, 5th International Meshing Roundtable '96* (1996), Sandia National Laboratories, pp. 205–215.

[20] SHEPHERD, J., MITCHELL, S., KNUPP, P., AND WHITE, D. Methods for multi-sweep automation. In *Proceedings, 9th International Meshing Roundtable '00* (2000), Sandia National Laboratories, pp. 77–87.

[21] STATEN, M., CANNAN, S., AND OWEN, S. Bmsweep: locating interior nodes during sweeping. In *Proceedings, 7th International Meshing Roundtable '98* (1998), Sandia National Laboratories, pp. 7–18.

[22] TAUTGES, T., BLACKER, T., AND MITCHELL, S. The whisker weaving algorithm: a connectivity-based method for all-hexahedral finite element meshes. *International Journal for Numerical Methods in Engineering 39* (1996), 3327–3349.

[23] THOMPSON, J. F. What's wanting to happen. Talk given at 11th International Meshing Roundtable '02, Ithaca, NY, September 2002.

[24] WHITE, D. R. Automatic, quadrilateral and hexahedral meshing of pseudo-cartesian geometries using virtual subdivision. Master's thesis, Brigham Young University, 1996.