The Whisker Weaving Algorithm: A Connectivity-Based Method for Constructing All–Hexahedral Finite Element Meshes

Timothy J. Tautges

Comp Mechanics and Visualization Dept, Sandia National Laboratories, Albuquerque, NM 87185

Ted Blacker

Fluid Dynamics International, 500 Davis St., Suite 600, Evanston, IL 60201

Scott A. Mitchell

Comp Mechanics and Visualization Dept, Sandia National Laboratories, Albuquerque, NM 87185

Submitted to the International Journal of Numerical Methods in Engineering, March 12, 1996.

This work was supported by the U.S. Department of Energy under contract DE-AC04-94AL85000, by the Applied Mathematical Sciences program, U.S. Department of Energy Research, and by Sandia National Laboratories Cooperative Research and Development Agreement SC93/01247 (USCAR Scientific Computing for Automotive Applications Partnership).

Abstract

This paper introduces a new algorithm called whisker weaving for constructing unstructured, all-hexahedral finite element meshes. Whisker weaving is based on the Spatial Twist Continuum (STC), a global interpretation of the geometric dual of an all-hexahedral mesh. Whisker weaving begins with a closed, all-quadrilateral surface mesh bounding a solid geometry, then constructs hexahedral element connectivity advancing into the solid. The result of the whisker weaving algorithm is a complete representation of hex mesh connectivity only: Actual mesh node locations are determined afterwards.

The basic step of whisker weaving is to form a hexahedral element by crossing or intersecting dual entities. This operation, combined with seaming or joining operations in dual space, is sufficient to mesh simple block problems. When meshing more complex geometries, certain other dual entities appear such as blind chords, merged sheets, and self-intersecting chords. Occasionally specific types of invalid connectivity arise. These are detected by a general method based on repeated STC edges. This leads into a strategy for resolving some cases of invalidities immediately.

The whisker weaving implementation has so far been successful at generating meshes for simple block-type geometries and for some non-block geometries. Mesh sizes are currently limited to a few hundred elements. While the size and complexity of meshes generated by whisker weaving are currently limited, the algorithm shows promise for extension to much more general problems.

1. Introduction

Finite element analysis is used to model physical phenomena in a wide variety of disciplines, including structural mechanics and dynamics, heat transfer, and computational fluid dynamics (CFD). To perform any of these analyses, the problem domain must first be discretized into a mesh of regular one-, two- or three-dimensional elements. Traditionally, generating a mesh for a given problem has been very tedious, time consuming and error prone. Furthermore, increasing computer speeds have made the non-automated mesh generation process even more of a bottleneck in the design and analysis process. Automated meshing algorithms reduce this bottleneck and free the analyst to concentrate on more important design and analysis issues.

Automated meshing algorithms for general three-dimensional volumes traditionally give tetrahedral- or hexahedral-shaped elements, or a combination of the two types. A mesh is constrained in terms of how elements share subfacets; this problem is much less constrained for tetrahedral or mixed element meshes, hence tetrahedral and mixed element meshing algorithms have received the most attention in the past. However, due to increased accuracy and efficiency of 8-node hexahedral elements for non-linear structural mechanics and other applications, there is a growing demand for all-hexahedral meshing algorithms.

Currently, there are no perfect all-hexahedral meshing algorithms. Typical undesirable features include a lack of automation, boundary insensitivity (i.e. placing poorly shaped elements close to geometric boundaries), input-orientation sensitivity, and large mesh size (i.e. too many elements). One method used for generating all-hexahedral meshes is isoparametric (ijk) mapping [1]. Although this algorithm is very robust for block-type geometries, it is difficult to automate for general three-dimensional geometries[2],[3]. Also, refinements required in a particular region propagate along all three principal directions of the map. This results in an unnecessarily large number of elements, which increases analysis time. The mapping method is available today in most mesh generation tools [4],[5].

Another well-known meshing algorithm is based on the finite octree approach [6]. Octrees have been widely used to generate tetrahedral meshes, and are now being used to generate all-hexahedral meshes. While these algorithms produce structured elements on the interior of the volume, they generally produce poorly-shaped elements on the boundary. Boundary fitting algorithms must also grapple with numerous special cases, contributing to a lack of robustness. Octree-based algorithms are also orientation sensitive, meaning that a different mesh is generated if the volume is rotated and re-meshed.

The plastering algorithm [7] is the three-dimensional analogue of the paving algorithm[9]. That is, plastering builds an all-hexahedral mesh by starting with an all-quadrilateral surface mesh, and then iteratively projecting mesh faces into the un-meshed volume to create hexahedra. The algorithm is orientation insensitive. However, combining fronts can be quite difficult because of how subfacets are shared between elements. Development of the plastering algorithm has recently been discontinued in favor of the whisker weaving algorithm.

The Spatial Twist Continuum (STC) is a global construct built on the dual of an all-hexahedral mesh [10]. In the STC, the mesh dual is represented by an arrangement of intersecting surfaces which bisect hexahedral elements in each direction, forming curves and vertices of intersection.

This paper introduces a new algorithm, called whisker weaving, which is based on information encapsulated in the STC. Whisker weaving builds the dual representation of an all-hexahedral mesh using an advancing front method, starting with a geometry and an all-quadrilateral surface mesh. Whisker weaving simplifies the all-hexahedral meshing problem by first determining the connectivity of an all-hexahedral mesh without regard to its geometric embedding; thus, the most constrained part of the problem is solved first. Hexahedral mesh entities (nodes, edges and hexes) are constructed by converting the dual and then iteratively smoothing to generate the geometric position of the mesh nodes. The algorithm to convert the dual to a normal mesh, referred to as primal construction, is not described in detail here.

This paper is organized as follows. Section 2 gives a description of the dual information encapsulated in the STC and how it can be used to derive connectivity information about a mesh. Section 3 describes the basic whisker weaving algorithm, which is sufficient to derive mesh connectivity for basic block-type geometries and surface meshes. Section 4 introduces several STC entities appearing when meshing more complicated geometries. Section 5 discusses the detection and resolution of some invalid connectivity cases arising in whisker weaving. Section 6 contains a number of examples which show both the generality and some of the limitations of the current whisker weaving algorithm. Finally, Section 7 contains a summary and discusses directions for future research.

2. The Spatial Twist Continuum

A fundamental difficulty of automated meshing is that a mesh is constrained in terms of how elements can share subfacets. This problem is much more difficult and highly constrained for all-hexahedral meshes than for all-tetrahedral or mixed-element meshes. These constraints are encapsulated for all-hexahedral meshes by viewing the geometric dual of the mesh as an arrangement of surfaces; this arrangement is called the Spatial Twist Continuum[10], abbreviated STC. The STC is the fundamental data structure used by whisker weaving.

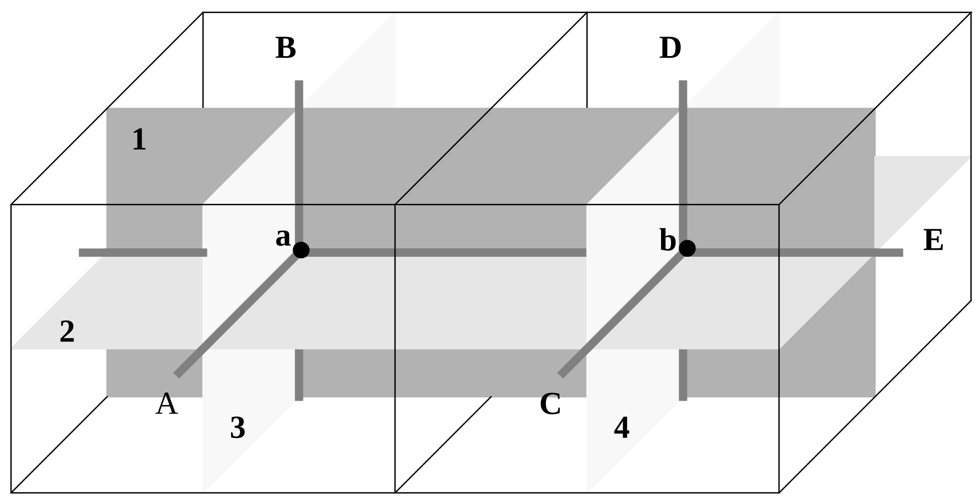

Figure 1. A solid composed of two hexahedra, showing whisker sheets (1-4), chords (A-E) and STC vertices (a-b).

The mesh composed of two hexahedral elements in Figure 1 is used to introduce the dual information encapsulated by the STC. There are three planes bisecting each element; these planes are referred to as *whisker sheets* or simply sheets. Whisker sheets can be thought of as general surfaces which propagate through a hexahedral mesh, bisecting each hexahedron between pairs of faces. The mesh in Figure 1 contains four sheets, numbered 1-4. The arrangement of these surfaces is referred to as the STC. The cell complex induced by the intersection of the surfaces is precisely the mesh dual. Present even in unstructured meshes, a sheet represents a layer of hexahedral elements which pairwise share faces.

The curve formed by the intersection of two sheets is called a *whisker chord*; there are five chords in Figure 1, A-E. Chords represent a column of elements which propagates through the mesh. For example, chord E in Figure 1 defines the column formed by the two elements in this mesh. Each hexahedral element corresponds to a dual entity, called an STC vertex, which is the intersection of three sheets (also the intersection of three chords). So, in Figure 1, STC vertex **a** is formed by the intersection of sheets 1, 2 and 3, (at chords A, B and E); STC vertex **b** is formed by the intersection of sheets 1, 2 and 4 (at chords C, D and E).

The intersection of a sheet with the geometric surface is a closed curve called a loop, representing the dual of a cycle of surface quadrilaterals which pairwise share opposite edges. Each boundary face is part of two loops, and is the starting point of a chord. Note that a sheet can contain many (or zero) loops, but in practice most sheets have a single loop. Also, a loop can intersect itself, resulting in a self-intersecting sheet. Sheets with more than one loop and with self-intersections are discussed in more detail in Section 4.

The one-to-one correspondence between dual (STC) entities and mesh entities is shown in Table 1. A mesh facet of dimension d corresponds to a STC facet of dimension $3 - d$.

Table 1. Correspondence between STC and mesh entities.

STC Entity	Dimension	Mesh Entity	Dimension
STC vertex	0	Hex	3
STC edge	1	Face	2
STC 2-cell	2	Edge	1
STC 3-cell	3	Node	0

Sheet diagrams

Whisker sheets are 2-dimensional surfaces twisting and crossing in space. Their combinatorial crossings with other sheets is explicitly represented, but their geometric position is only implied. In the case of whisker weaving, sheets can always be drawn in the plane as *sheet diagrams*. Intuitively, a sheet is flattened so that its loop forms a polygon in the plane and its chords are line segments inside this polygon.* Whisker weaving builds the STC in an advancing front manner. Hence at most stages of the algorithm a sheet diagram will only define the outermost intersections with other sheets. An example sheet diagram is shown in Figure 2.

Each chord on a sheet, formed by the intersection with another sheet, is represented by a sequence of edges on the sheet diagram. Since a chord is the intersection of two sheets, it must be represented on two sheet diagrams. To distinguish between the two representations of a chord, each representation is referred to as a sheetchord. Thus, a chord has two corresponding sheetchords. Most chords start at a quadrilateral face of the given surface mesh, shown as a vertex of intersection with the loop. This intersection is labeled outside the polygon with the id of the corresponding mesh face, and inside with the id of the other sheet passing through the mesh face (the *other sheet number*). The portion of a sheetchord lying between two vertices is an edge of the dual, called an *STC edge*. An STC edge is dual to a mesh face. The entities on the advancing front have special importance: The last edge

*Sheet diagrams for self-intersecting sheets and sheets with more than one surface loop are more complicated, and are discussed in Section 4.

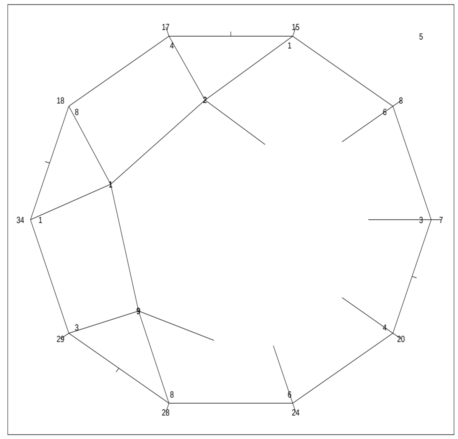

Figure 2. A sheet is represented with its loop as an *n*-sided polygon, where *n* is the number of quadrilateral faces on the loop. The rest of the sheet diagram lies inside the loop.

in the sequence for a chord represents a quadrilateral face on the advancing front, and is called a *whisker*.

An STC 2-cell is a polygon bounded by two or more STC edges. An *uncompleted* 2-cell is the collection of STC edges between two adjacent whiskers, inclusive. An STC 2-cells is dual to a mesh edge.

An STC vertex is dual to a mesh hex. These vertices are sometimes called *whisker hexes* to emphasize the fact that they represent only the connectivity of a normal hex element and not its geometric position. Since each whisker hex is formed by the intersection of three sheets, it appears as a vertex on three sheet diagrams. Furthermore, these vertices are the pairwise crossings of three chords.

Figure 2 shows the sheet diagram for sheet 5 in a new example. Recall that the chord with label 16 outside the loop and 1 inside the loop for sheet 5 begins at face number 16 and is formed by the intersection of sheets 5 and 1. This chord is represented by a sheetchord on sheet 5 (uniquely labelled the 16-1 sheetchord) and also by a sheetchord on sheet 1 (the 16-5 sheetchord). The first STC edge on this sheetchord (between the loop and STC vertex 2) represents face 16, while the first STC vertex on this sheetchord represents hex element number 2. The STC 2-cell bounded by STC vertex 2 and the loop intersections of chords 16-1 and 17-4 represents the mesh edge shared by faces 16 and 17.

An STC 3-cell is dual to a mesh node. An STC 3-cell is a polyhedron whose bounding surfaces are STC 2-cells. For simplicity, STC 3-cells are not explicitly represented: they are only needed during primal construction, where they are generated on the fly.

STC Edge Traversal

An important whisker weaving procedure is the traversal of consecutive STC edges bounding a 2-cell of a whisker sheet. For example, adjacent whiskers are found in this manner. Even before a sheet is completely woven, the STC contains valuable mesh connectivity information. To see this, consider the following example. Figure 3 shows a

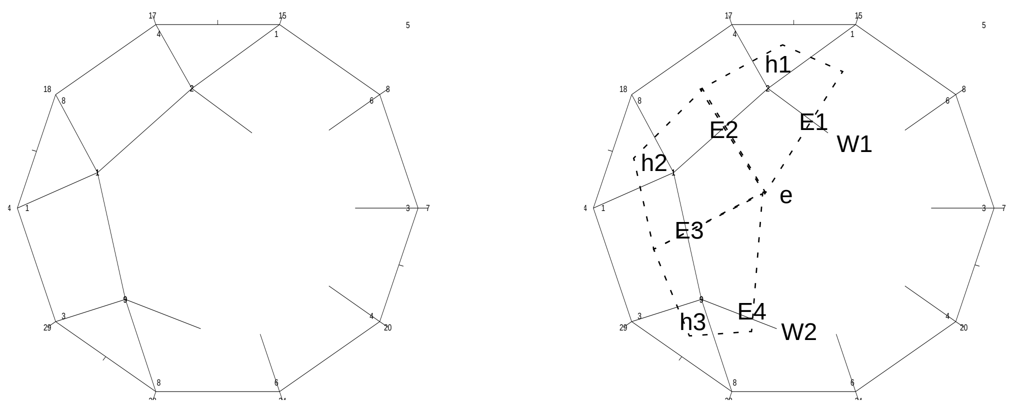

Figure 3. Partially completed sheet diagram (left); top view of three hexes superimposed on some of the chords making up those hexes (right).

sheet diagram with the top faces of three hexes superimposed with dotted lines. Whiskers **W1** and **W2** are on the current meshing boundary, and bound a STC 2-cell. The STC edges **E1-E4** between **W1** and **W2** are dual to faces which share a common mesh edge **e**. The STC vertices on that STC 2-cell represent hexes **h1-h3** which also share edge **e**.

Edge traversal can be used to find important features of the STC. Whisker weaving occasionally creates an STC locally dual to two faces sharing two edges. This is the most basic invalidity created: most other invalidities contain one or more pairs of faces which share two edges. The detection of this invalid connectivity is performed in STC space, and is based on “repeated STC edges”. In the previous paragraph it was shown that a pair of STC edges in the same 2-cell represents a pair of faces sharing an edge. Likewise, if two STC edges are part of two distinct 2-cells (i.e. they are “repeated” in both 2-cells), they represent a pair of faces which share two edges. Repeated STC edges are found by traversing the 2-cells next to both whiskers of a chord, searching for another STC edge that appears twice in these 2-cells. If such an STC edge is found, there are two repeated STC edges, and an invalidity has been found.

For example, consider the two sheet diagrams shown in Figure 4. The ends of the 7-2 and 12-2 sheetchords are part of the same STC 2-cell on sheet 1, which means they share the edge represented by that 2-cell. The same whiskers (sheetchords 7-1 and 12-1) are part of the same 2-cell on sheet 2, meaning that they share another edge represented by a different

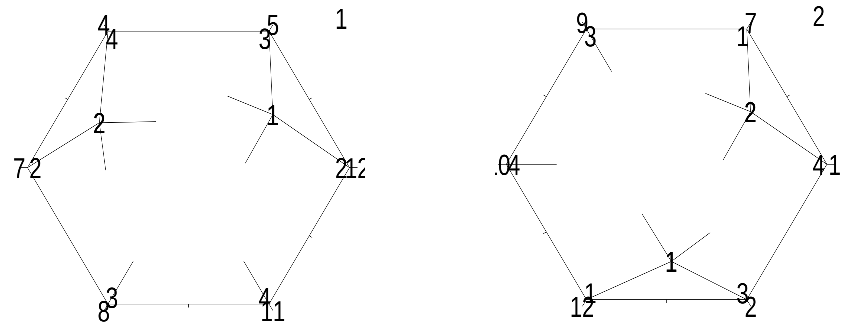

Figure 4. Two sheets showing a repeated STC edge.

2-cell. This signifies that the faces dual to the whiskers share two edges.

After the detection of a repeated STC edge pair, the invalid connectivity must either be resolved immediately or left in place in the form of an invalid element. The resolution of invalid connectivity is described in Section 5.

3. The Whisker Weaving Algorithm

One of the primary uses of the STC is to visualize the three-dimensional meshing problem as a series of inter-dependent two-dimensional problems. The STC arrangement, which consists of a collection of intersecting three-dimensional surfaces (whisker sheets) in space, is initially defined by a collection of incomplete, two-dimensional sheet diagrams. The whisker weaving algorithm starts with this arrangement and builds the connectivity of an all-hexahedral or mostly-hexahedral mesh, starting from the boundary mesh and moving inward. The algorithm seeks to build mesh connectivity through the application of a simple set of rules which govern the formation of individual hex elements (hexes), and the resolution of local connectivity problems. An important feature of the algorithm is that the set of rules is kept very simple; in particular, every effort is made to avoid special case rules that may not apply to geometries in general.

The starting point for whisker weaving is a set of quadrilaterals defining a surface that encloses a 3-dimensional volume. The surface mesh can be generated with any suitable algorithm, for example mapping [1] or paving [9]. While in general the volume can have arbitrary geometry and topology, in practice topological balls have been easier to mesh.

In this section, the whisker weaving algorithm will be described as it applies to a very simple example problem. Issues arising when applying the algorithm to more complicated problems are discussed in later sections. The geometry and surface mesh for this example, referred to here as the “small corner hex” problem, are shown in Figure 5, where the two

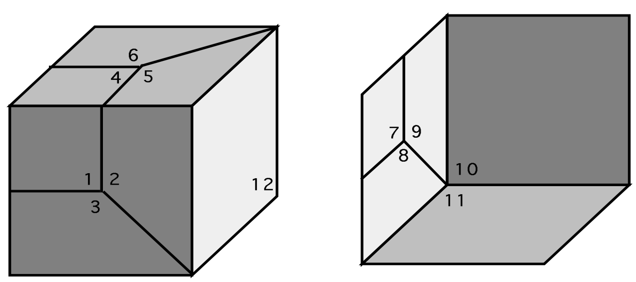

Figure 5. Geometry and surface mesh for the small corner hex problem.

views show the front faces (left), and a cutaway view of the back faces (right). Faces are also numbered, for reference.

In its simplest form, the whisker weaving algorithm consists of four steps. After an initialization step, the next three steps create a hex and resolve some connectivity invalidities. The same three steps are repeated until the entire hexahedral mesh is completed. Block-type solids with regular surface meshes can be meshed using just these steps, while more complicated examples require additional steps, which are discussed in Section 4. The four steps are.

1. Construct initial loops, sheets, chords and sheetchords.

2. Find three pairwise adjacent chords.
3. Construct a whisker hex by crossing the 3 chords.
4. Check for invalid connectivity on all sheets involved in the hex just formed.

These steps are now described in detail.

1. Construct initial loops, sheets, chords and sheetchords.

Given a surface mesh, a loop is initialized by picking any untraversed quadrilateral edge as the starting edge, and traversing to the opposite quadrilateral edge. The opposite edge is shared with another quadrilateral, so the traversal is repeated until the starting edge is reached from the other side. The loop puts the mesh faces on an ordered list as it passes through them.

All loops have been formed when all quadrilaterals are crossed by two loops. Initially, a unique whisker sheet is defined for each loop. (Later, some sheets might be merged to form one sheet with multiple loops.) There are four loops for the example problem (see Figure 6). After the loops are created, a chord and its corresponding sheetchords are constructed for each mesh face. Figure 7 shows the corresponding four sheet diagrams for this example.

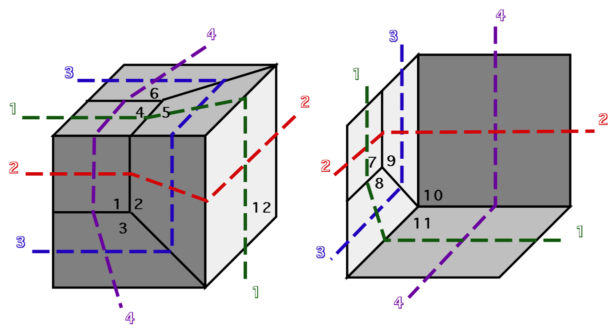

Figure 6. Whisker sheet loops for the small corner hex problem.

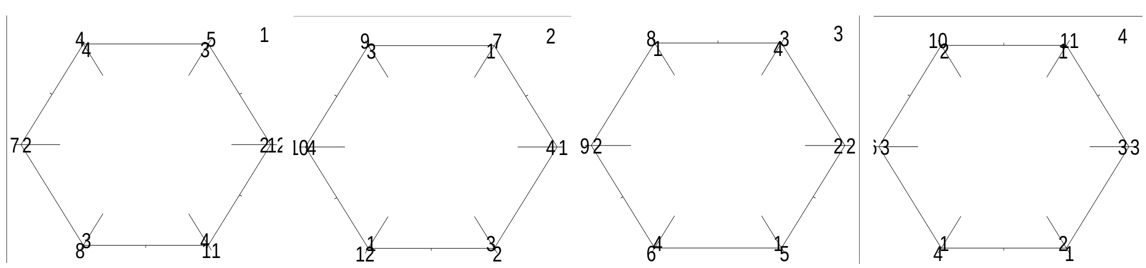

Figure 7. Whisker sheet diagrams for the small corner hex problem. Sheet numbers are shown in upper corner of each sheet diagram; for each sheetchord, the loop face id is written outside the loop, and the other sheet number for the sheetchord is written inside the loop.

2. Find three pairwise adjacent chords.

In the whisker weaving algorithm, hexes are formed by crossing three chords to form three STC vertices. This is shown on sheet diagrams by the crossing of three pairs of sheetchords on three sheets. In mesh space, this represents the formation of a hex using three faces which pairwise share three edges.

To find the three chords required to form a hex, a pair of adjacent whiskers on any sheet is chosen and a local search of the corresponding whiskers on the two “other” sheets is conducted to see if a proper third chord can be found. For example, in Figure 7 the 12-2 and 5-3 whiskers on sheet 1 are chosen as candidates; this completely defines the three sheets used to form the next hex (sheets 1, 2 and 3). A proper third chord must be the intersection of sheets 2 and 3. The third chord is found by examining the adjacent whiskers on the other two sheets (sheets 2 and 3): weaving checks for the existence of a chord corresponding to face **f**, such that whisker **f**-3 is adjacent to whisker 12-1 on sheet 2, and whisker **f**-2 is adjacent to whisker 5-1 on sheet 3. In Figure 7 the chord with whiskers 2-3 and 2-2 on sheets 2 and 3, respectively, is chosen to be the third chord. Figure 5 shows that this corresponds to forming a hex using faces 2, 5 and 12; the figure shows that these faces pairwise share three edges.

Sometimes there is no third chord with whiskers adjacent to those for the first two chords on the other two sheets. In hex space this corresponds to there being no third face that shares edges with the first two faces. This case is described in Section 4.

3. Construct a whisker hex by crossing the 3 chords.

The three chords are crossed to form a new whisker hex; the three vertices are labeled with the whisker hex id. The three sheets involved in the first crossing are shown in Figure 8.

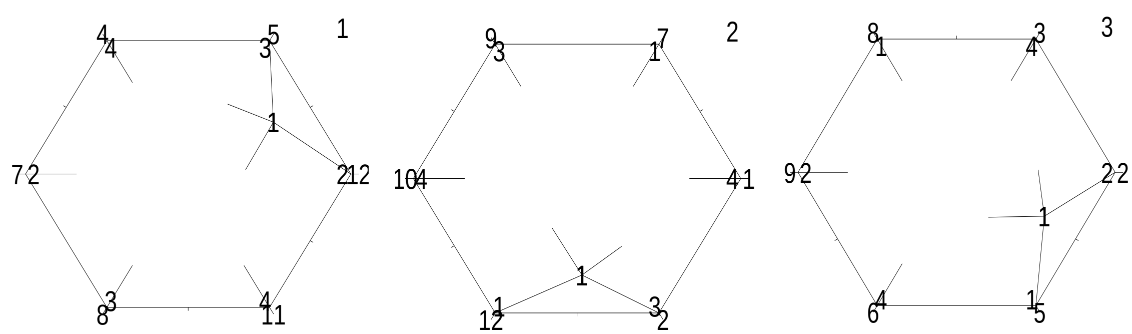

Figure 8. Whisker sheet diagrams for the example problem after the formation of the first whisker hex.

There are some caveats as to which chords can be chosen in step 2. In particular, the formation of hexes with very large or very small angles between face pairs on the exterior geometry must be prevented. In terms of STC entities, this restriction is stated as follows:

Rule 1: Do not cross bare whiskers except around geometric corners.

Bare whiskers are defined as sheetchords which contain no STC vertices, and correspond in hex space to faces on the surface mesh which have yet to be attached to a hex element. Rule 1 prevents the use of two faces in the same hex whose interior angle is not close to $\pi/2$. Figure 8 shows that all whiskers involved in the first crossing are bare, and that there is indeed a geometric corner (indicated by a hash mark on the outer loop) between each pair that crosses. Faces in the interior of the volume have some flexibility in their geometric embedding, so this rule is not tested for non-bare whiskers. However, there are other rules to prevent the formation of poorly-connected hexes (e.g. two hexes which share two faces). These rules are introduced later in this section.

Following the formation of the first whisker hex, a second whisker hex can be formed using steps 2 and 3. Figure 9 shows the sheets involved in the formation of the second whisker

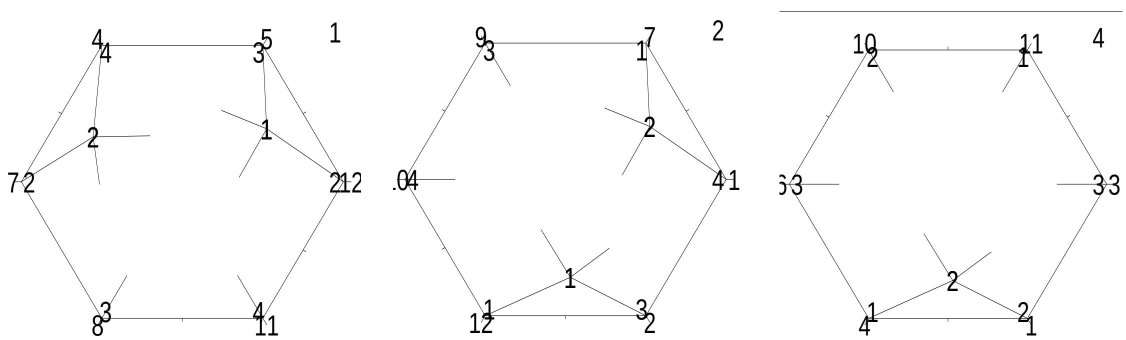

Figure 9. Whisker sheet diagrams for the example problem after the formation of the second whisker hex.

hex (sheets 1, 2 and 4).

4. Check for invalid connectivity on all sheets involved in the hex just formed.

It is possible that forming a crossing introduces invalid connectivity in the mesh. For example, in Figure 9 the whiskers of the chords beginning at faces 7 and 12 are adjacent on sheet 1 and also on sheet 2. This is an example of a repeated STC edge pair, discussed earlier, which represents invalid mesh connectivity. The resolution of this problem is to join the two chords. Since the dual representation of a chord is a column of hexes, in hex space a chord join corresponds to seaming two faces together, which joins two columns of hexes. A chord join always appears on two sheet diagrams, because chords are formed by the intersection of two sheets.

Joining two chords removes four dangling whiskers from two sheet diagrams. Since this prevents the formation of any more hexes on those chords, chord joins tend to minimize the number of elements formed in a mesh. In most mesh generation algorithms, the twin goals of small mesh size and good element shape compete. However, in this phase of the whisker weaving algorithm there is no geometric information, so element shape is not a consideration. In practice, the quality of elements formed by whisker weving depends heavily on the density of the surface mesh; for coarse surface meshes, the resulting hex quality is rather poor, since it is more constrained by the geometry. The converse is true for weaves derived from finer surface meshes. Hence the algorithm simply tries to add as few elements as possible by joining chords whenever possible:

Rule 2: Join chords whenever it is allowed.

By repeating steps 2-4, the remaining parts of the example problem can be meshed. The final whisker sheet diagrams for the example are shown in Figure 10.The completion of the

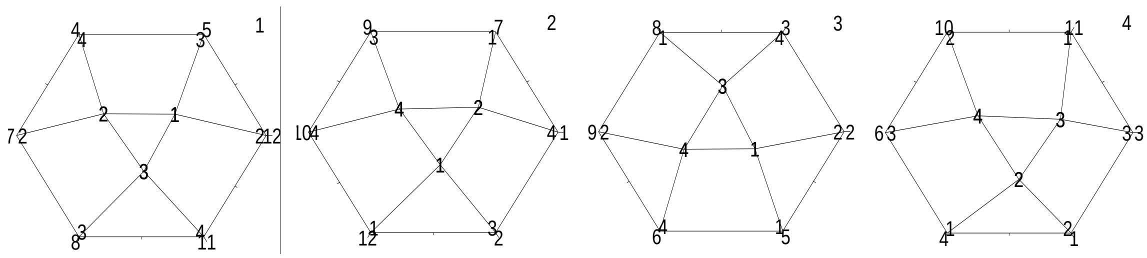

Figure 10. Final whisker sheet diagrams for small corner hex example.

weave is indicated by the absence of any dangling whiskers. A completed weave represents the entire connectivity of the all-hexahedral mesh. Note that whisker weaving does not yield any information about the geometric position of hex elements, and does not determine actual element or node positions in the interior of the volume. However, given a valid connectivity, these positions can be determined in a relatively straightforward manner using the primal construction algorithm (this algorithm will be described in an upcoming paper).

In its simplest form, the whisker weaving algorithm can be thought of as a heuristic for making connectivity crossings in dual space, with the goal of completely specifying the connectivity of an all-hexahedral mesh. The algorithm is local in nature; that is, it performs no global search to determine the “best” crossing to make. Because of the local nature of the algorithm, whisker weaving has very low complexity, of O(n), where n is the number of hexes formed. However, also because of the local nature, whisker weaving may not produce the optimal mesh connectivity, e.g. it may produce more than the optimal number of irregular nodes (interior nodes in 3d which are connected to more or less than 6 edges). Also, the heuristic nature of the algorithm sometimes causes it to form hexes which subsequently need to be removed. These problems have proven to be tractable, though, and are a small price to pay for the speed of an O(n) algorithm.

4. Special STC Constructs

This section discusses special types of chords and sheets which arise when meshing more general geometries and surface meshes. These entities include blind chords, merged sheets, and self-intersecting loops, sheets, and chords.

Blind Chords

In Section 3, all whisker chords begin and end on the geometric surface. However, for some geometries and surface meshes, there are chords which never emerge on the geometric surface. These types of chords are referred to as blind chords. An example of a geometry and surface mesh which contain blind chords is shown in Figure 11, referred to as the

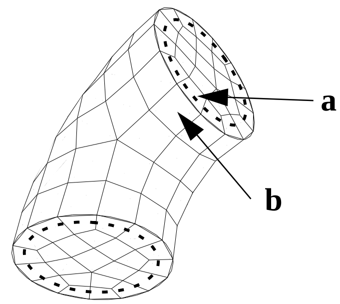

Figure 11. Geometry and surface mesh for the macaroni problem, where meshing results in blind chords and merged sheets.

“macaroni” problem for obvious reasons. A natural place to start meshing this problem would be to place a ring of elements around the end cap of the geometry. This ring of elements has a blind chord running through it which never emerges on the geometric surface.

A blind chord is formed when a hex is created using only two pre-existing faces. For example, forming a hex using faces marked **a** and **b** in Figure 11 would form a blind chord. Since these two faces share one edge, there is one sheet common to them. The other two sheets of the hex are the other sheets of the two faces. The blind chord is an intersection of these non-shared sheets, and therefore its whiskers are drawn on those sheets.

An example sheet diagram containing a pair of newly created blind chords is shown in Figure 12. Theoretically, there are only a few differences between a blind chord and an

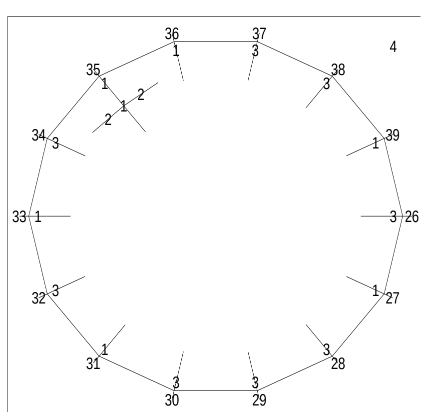

Figure 12. Loop and sheet diagram for sheet 4 of the macaroni problem.

ordinary chord. Since a blind chord does not begin on a face of a loop, it can advance in two directions. To maintain consistency, a blind chord is represented in the whisker weaving implementation as two chords, each of which can advance independently.

There are two ways in which a blind chord can be completed. First, it can join with itself, in which case it forms a closed curve completely inside the mesh. Second, it can join with an ordinary chord, in which case it is appended to that chord and the blind chord goes away.

Merged Sheets

The first step in the whisker weaving algorithm was to form loops on the surface mesh, where each loop was assumed to define a unique sheet. In the final mesh for the example problem in Section 3, each sheet contained exactly one loop on the geometric surface. However, this is not always the case for more general 3-dimensional meshes. In fact, whisker sheets can have two or more loops of faces (or zero) on the surface mesh. For example, the macaroni problem shown in Figure 11 contains two face loops (shown with dotted lines) which are eventually merged into the same sheet. Merged sheets often occur in annular and cylindrical solids.

Since whisker weaving starts by assuming each loop is on a unique sheet, at some point in the weaving process a decision must be made to merge sheets together. This decision is based on quality criteria and is made when two meshing fronts meet. The criteria currently used to motivate a sheet merge are the following:

- Two hexes share an edge.
- Two of the faces containing the shared edge, one in each hex, are on the geometric boundary, and are nearly co-planar.*
- The two adjacent faces, one in each hex, do not have the same two sheets.

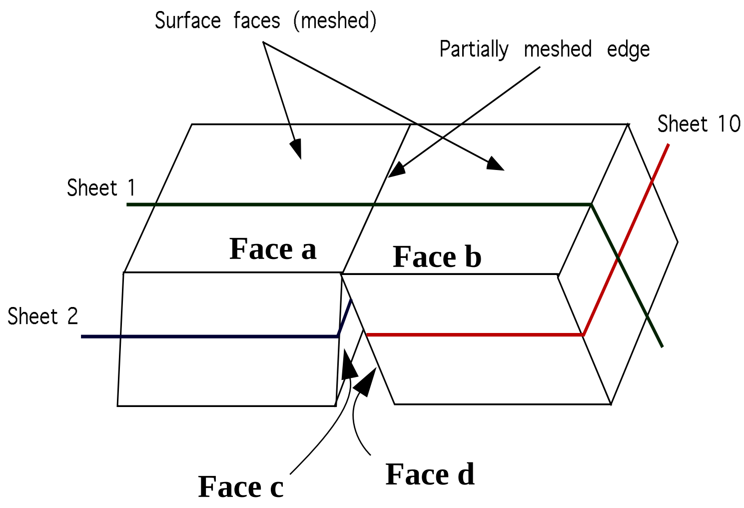

Figure 13. Hex connectivity resulting in a sheet merge. Faces **c** and **d** will be seamed, leading to the merging of sheets 2 and 10.

*A pair of faces are considered co-planar if their dihedral angle is between 150 and 225 degrees.

For example, Figure 13 shows the two hexes which share an edge. The two co-planar faces are labelled **a** and **b**, and the two faces to be seamed are labelled **c** and **d**. Since the criteria for a sheet merge are met here, the non-common sheets in faces **c** and **d** (sheets 2 and 10) must be merged.

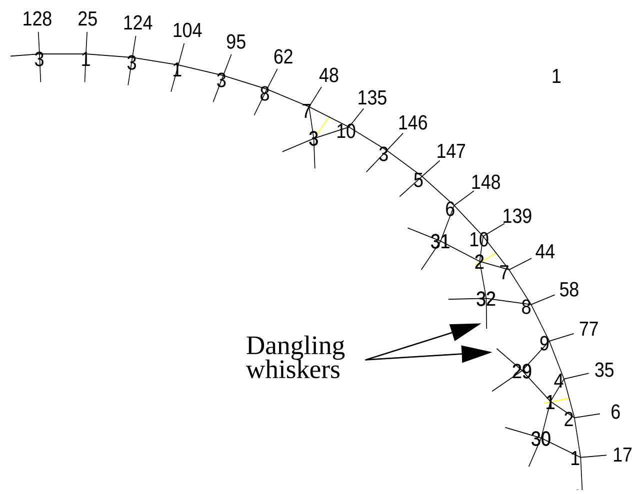

Figure 14. Whisker sheet diagram indicating the need for a sheet merge.

Figure 14 shows the STC data for an example of this situation which occurs in the macaroni problem. Here, hexes 29 and 32 share a mesh edge (i.e. they are in the same 2-cell). The vertices representing these hexes are both directly connected to the loop by STC edges, indicating that they contain faces in hex space that are on the geometric boundary. These faces are co-planar (this can be tested using data for each boundary face). The other two faces sharing the mesh edge are indicated by the adjacent whiskers extending from hexes 29 and 32. Since these whiskers have different sheets (sheet 2 versus sheet 10), they cannot be joined directly. By the criteria listed above, sheets 2 and 10 can be merged and the two faces seamed.

When one sheet is merged into another, the resulting sheet contains two separate face loops. These loops are joined in STC space by a single chord. This is represented graphically by drawing an inner and an outer loop, joined by the chord connecting them. The resulting sheet merge from the example above is shown in Figure 15. Note that there is no topological distinction between the inner and outer face loops. After the sheets have been merged, but before weaving continues, there is a single chord joining the inner and outer loops. The sheet has a single contiguous loop of whiskers, and therefore can be woven like any un-merged sheet.

For certain geometries and surface meshes it is desirable to join more than two loops. For example, for a cylinder with a small cubical indentation (not on an endcap), it would be advantageous to join the two endcaps and the four-chord loop around the indentation. Indeed, arbitrarily many indentations could be added to the geometry, leading to arbitrarily many sheet merges.The case of a single sheet having three or more distinct face loops would be represented by more than one inner face loop.

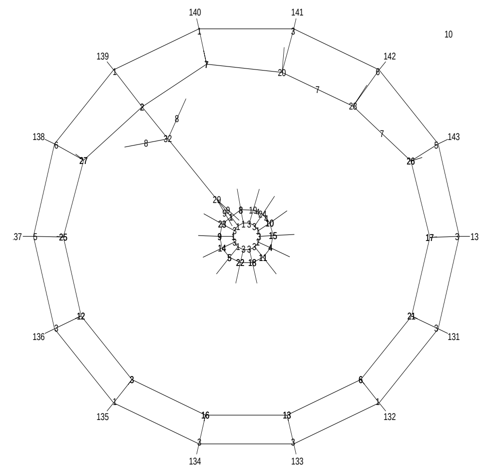

Figure 15. Sheet after merging; inner and outer loops represent two loops of faces on the surface mesh.

The sheet merge criteria is currently based on both geometric and connectivity information. Because whisker weaving generates only connectivity information on the interior of the mesh, the current sheet merge detection criteria will not work there. The development of sheet merge criteria based solely on connectivity information is a focus of current research.

Self-Intersections

When a loop passes through a quadrilateral twice, crossing itself, it is called a self-intersection. Self-intersecting sheets and loops are a complication in the STC that is quite common, arising from surface meshes with irregular nodes (e.g. nodes in the middle of a geometric surface with more or less than 4 edges connected to them). Each self-intersection of a loop implies a self-intersecting chord, and implies that the sheet defined by the loop is also self-intersecting. For example, in the macaroni problem in Figure 11 two of the eleven loops self-intersect a total of 20 times.

Certain STC vertices appear in a slightly different way on self-intersecting sheets than on normal sheets. If two of the three sheets defining an STC vertex are the same, then the recurring sheet is by definition a self-intersecting sheet. The vertex appears twice on the self-intersecting sheet and once on the third sheet. If all three sheets are the same, then the vertex appears three times on the self-intersecting sheet.

Self-crossing sheets are in practice more difficult to weave than sheets that do not cross themselves. The existence of a *valid* STC is equivalent to the existence of a mesh. For volumes that are topologically a ball, for there to exist a valid STC it is necessary and sufficient for there to be an even number of surface quadrilaterals where a loop self-intersects. This condition is equivalent to there being an even number of surface quadrilaterals total. Some additional conditions are necessary and sufficient for toroidal and other topologically complicated input. See [12] for the details.

5. Connectivity Resolution in Whisker Weaving

All-hexahedral meshes are highly constrained in terms of how facets can connect to one another. A common advancing front approach is to add a hexahedron that is of the right general shape, but temporarily violates these constraints, and then make small local changes to fix connectivity problems. For example, the plastering algorithm generates pairs of mesh faces which share two edges [7]. This is then fixed by a seam operation, which merges the two faces into one face. Similar situations arise in the whisker weaving algorithm. This section discusses the algorithms used in whisker weaving for detecting and resolving invalid connectivity.

The mesh invalidities currently created by whisker weaving are all based on the base case of two mesh faces sharing two edges. Because of this, all invalidity detection in whisker weaving is based on the detection of repeated STC edges, described earlier. After its detection, invalid connectivity must either be resolved immediately, or left in place in the form of a degenerate element. The seam and removed hex seam resolution techniques are described first, and afterwards the formation of degenerate knife elements are described.

Seam

Recall that the most common invalid connectivity problem arising in whisker weaving algorithm is the case of two mesh faces which share two edges. The obvious solution is to “seam” the two faces into one face. In STC space, this corresponds to joining two chords together. This case is described earlier, and for example motivates the joining of chords beginning at faces 12 and 7 on sheets 2 and 4 in Figure 8. Note that after one chord join is performed, the adjacent chords should also be checked for possible joins.

Removed Hex Seam

In the simple seam example, the repeated STC edges are both whiskers, that is dual to faces on the meshing front, and are joined together without affecting the rest of the STC. In a removed hex seam case, one of the repeated STC edges is not a whisker. This STC edge is dual to a face with a hex on either side of it, and not on the meshing front. This is resolved by removing hex elements on the meshing front from the chord through the edge until the edge is on the meshing front. Then the two whiskers are seamed normally.

Figure 16 shows an example of sheet diagrams with an unresolved removed hex seam case. On sheet 5 (right), traversing from whisker **a** and going clockwise, STC edges **a** and **b** fall on the same STC 2-cell. On the other sheet (sheet 3, left), traversing from the same STC edge and going clockwise, STC edges **a** and **b** also are part of the same 2-cell. Since they are not directly connected, hexes 96 and 162 contain two distinct faces which share two edges. The chords which must be joined (representing a face seam) are the blind chord and the chord beginning at face 101. However, this cannot be done until hex 157 is removed.

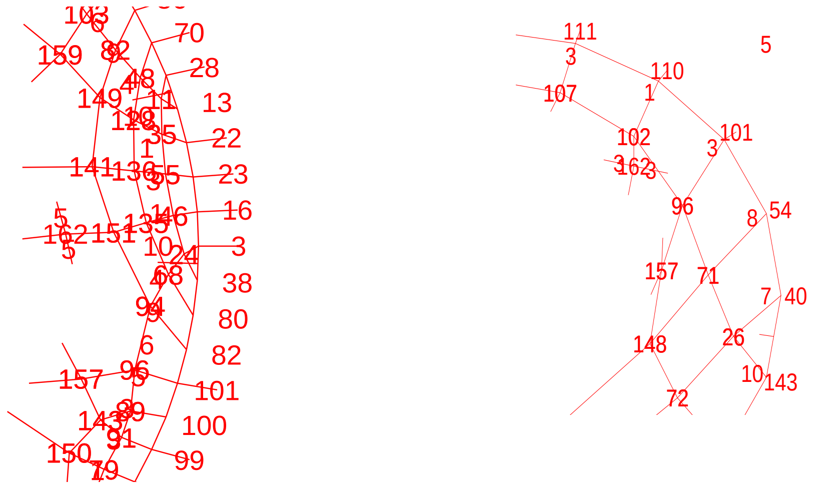

Figure 16. Sheet diagrams showing a remove hex seam case for sheets 3 and 5 of the macaroni problem.

Knife Formation

A knife is a hex-space object which was first introduced in connection with the plastering algorithm [7]*. It has the connectivity of a hexahedron that has had one face collapsed by joining two opposite vertices (see Figure 17). The face opposite the collapsed face is called

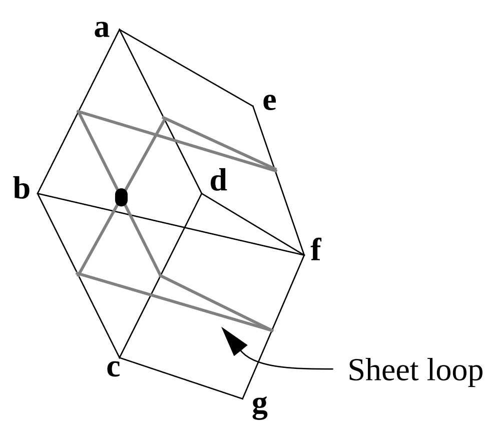

Figure 17. A knife formed from a hexahedron, with nodes labeled **a**-**g**. Knife contains a base face (**abcd**), four side faces (**aedf**, **aebf**, **bcgf**, **cgfd**) and two blade edges (**ef**, **fg**).

the *base face* of the knife (face **abcd** in Figure 17) and the other four faces are called *side faces* (faces **aedf**, **aebf**, **bcgf**, **cgfd** in Figure 17). The collapsed face's remaining edges (edges **ef** and **fg** in Figure 17) are called the blade edges of the knife.

*Knives have previously been referred to as wedges; however, this has lead to confusion in the finite element community, which uses the same term to describe a triangular prism.

In STC space, the chord going through the base face is referred to as the *base chord*. An important property of knives is that the base chord is formed by a self-intersecting sheet; this can be shown by drawing the loops on the surface of a knife as if it were the entire geometry (see Figure 17). This sheet is called the *base sheet*; the other sheet through the side faces is called the *side sheet*.

A knife is indicated in STC space by a self-intersecting chord whose two sheetchords are adjacent and therefore are part of the same STC 2-cell. The implication of this is that the face represented by the end of the base chord shares two edges *with itself*. In other words, two of the edges on that face have been merged into one edge. If these sheetchords are the only two whiskers remaining on a sheet, they have two 2-cells in common, one on either side. In either case, there is a hex element which has had one of its faces collapsed; this element, which is the knife element, is represented by the last STC vertex on the base chord.

The detection of the formation of a knife is based simply on two sheetchords of a self-intersecting chord being adjacent. If this situation occurs, the two sheetchords are joined together in STC space and the knife is left in place. This join operation is not the same as a normal chord join, as the self-intersecting (base) chord propagates into the volume but never comes out; rather, it terminates inside the knife element. After the two sheetchords are joined together, the whisker sheet diagram appears like a normal sheet, and weaving on that sheet can continue. This is true because although a hex element is degenerate, it is composed entirely of quadrilaterals, and meshing can continue as before.

An example of a knife that has been formed during whisker weaving is hex number 123 in the sheet diagram in Figure 18. The base chord for this knife begins at face 25; this chord

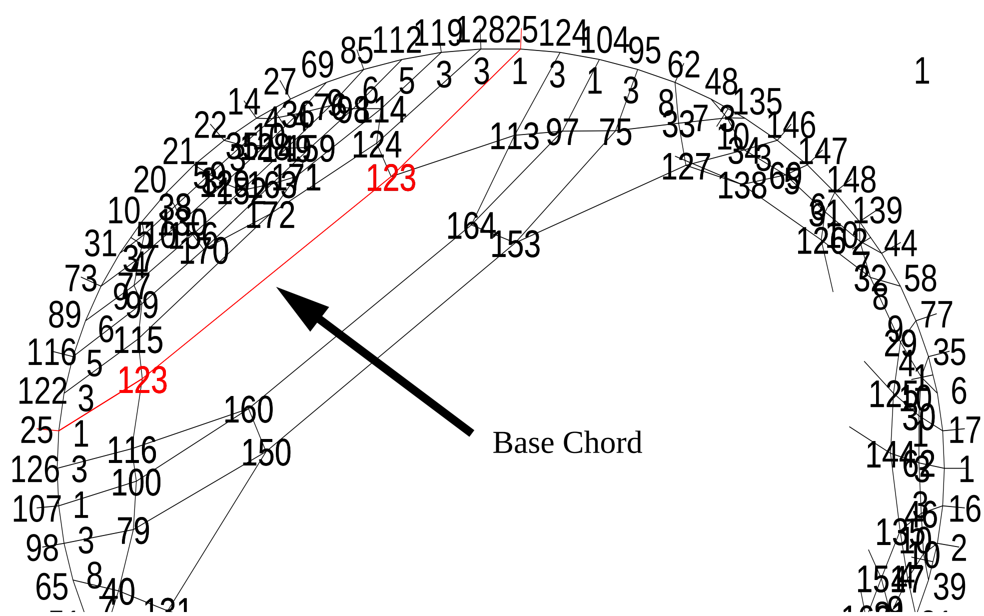

Figure 18. knife formed in sheet 1 of the macaroni problem during whisker weaving, indicated by the joining of two whiskers beginning at face 25.

terminates within the solid, as indicated by the identical beginning face numbers on the base chord at both intersections with the outer loop.

Figure 18 shows that two chord joins have taken place after the formation of the knife, joining the chords that begin at faces 104 and 107 and faces 95 and 98. This demonstrates how whisker weaving is able to proceed, deferring the knife resolution to a separate phase, after all sheets are completely woven. Waiting until the STC is completely woven can also result in better knife resolution, since more information is available [8].

Knife elements are produced by the plastering algorithm as well. These elements allow the generation of hex-dominant meshes for geometries and surface meshes for which an all-hexahedral mesh cannot be generated. For example, a volume bounded by an odd number of quadrilaterals can be closed using mostly hexes and an odd number of knife elements.

The mesh resulting from the whisker weaving algorithm described above can contain invalid elements (knives), but is otherwise valid. Methods for resolving knives in an all-hex mesh have been developed and implemented in the plastering algorithm [11]; these methods consist of either driving knives forward in the mesh or pulling them backward. Analogous methods for driving or pulling knives based only on whisker weaving data have been developed [8]. The important thing to realize is that resolving knives in an otherwise valid all-hex mesh is straightforward from a theoretical point of view. Thus, a meshing problem can be considered solved when whisker weaving has generated an STC, even if the STC contains knives.

6. Examples

The parts of the whisker weaving algorithm described in previous sections have been implemented inside the CUBIT mesh generation environment [13]. This environment is suitable for testing new meshing algorithms because it provides many of the support functions that would otherwise need to be developed; some of these functions are:

- Geometry import from ACIS solid model or FASTQ input [14]
- Limited geometry creation, transformation and boolean operations
- Various surface meshing algorithms including paving and mapping
- Graphics support for mesh and geometry visualization
- Mesh export in EXODUS II format [15]

CUBIT has the capabilities to provide whisker weaving with a volume bounded by an all-quadrilateral mesh.

The small corner hex example was used to demonstrate the basic whisker weaving algorithm in Section 3. In this section, two more examples are given which demonstrate the whisker weaving algorithm, the double-fold problem and the macaroni problem (a 1/4 torus).

Double Fold Problem

The double fold problem has been used to study STC folds and corners, which are STC entities arising from irregular nodes in a surface mesh (e.g. nodes on a surface which are connected to more or less than four mesh edges) [10]. This example also has a simple cube-type geometry, but a more complicated surface mesh; these are shown in Figure 19.

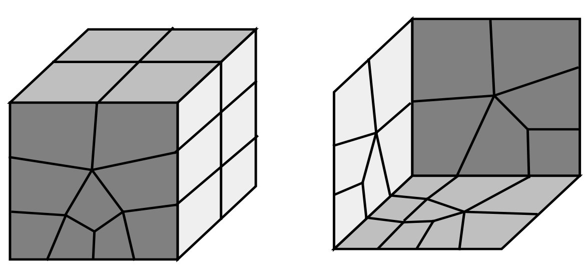

Figure 19. Geometry and surface mesh for the double fold problem.

The mesh resulting from weaving this volume contains 17 hex elements. The sheet diagrams are shown in Figure 20, while the wireframe mesh is shown in Figure 20. Note that there are several 5-sided and 3-sided STC 2-cells, for example on sheets 2 and 3; these higher and lower degree STC 2-cells are also referred to as folds. These cells propagate through the mesh until they either emerge on the surface or meet inside the mesh with other higher or lower degree STC 2-cells; these meeting points are referred to as corners. This

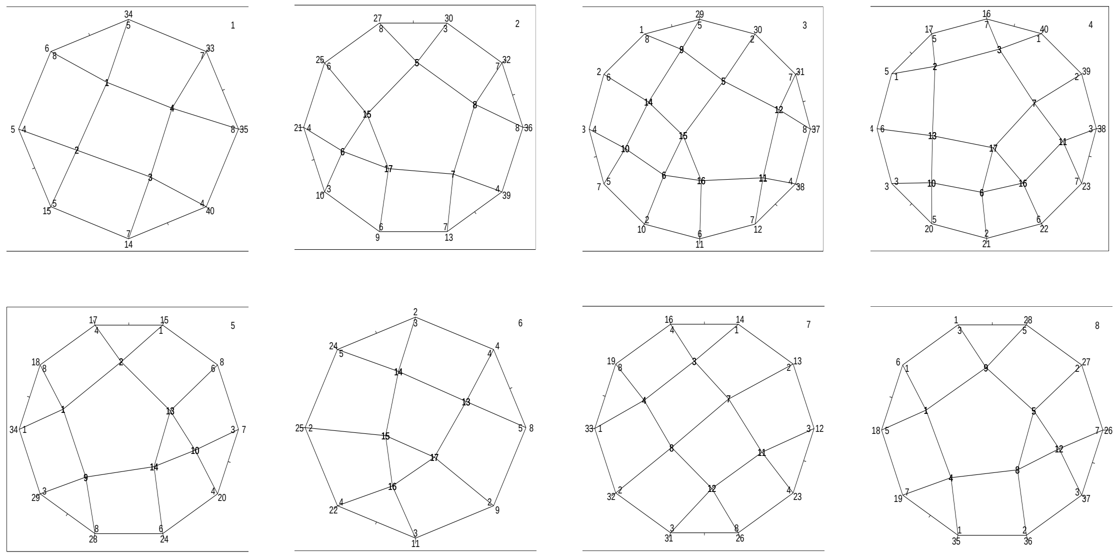

Figure 20. Final sheets for the double fold example.

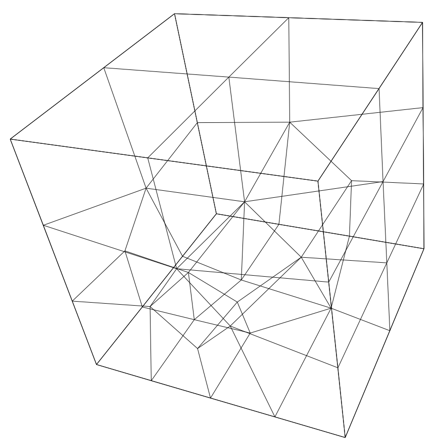

Figure 21. Wire frame view of final mesh for double fold example.

example illustrates the fact that whisker weaving is adept at handling complicated surfaces meshes in a general way.

Macaroni Problem

One of the more difficult geometry and surface meshes that has been completely meshed to date with the whisker weaving algorithm is the macaroni problem. This problem consists of a 1/4 torus geometry, where the inner radius is much smaller than the outer radius. The geometry and surface mesh for the macaroni problem are shown in Figure 22. All three surfaces were meshed using the paving algorithm [9].

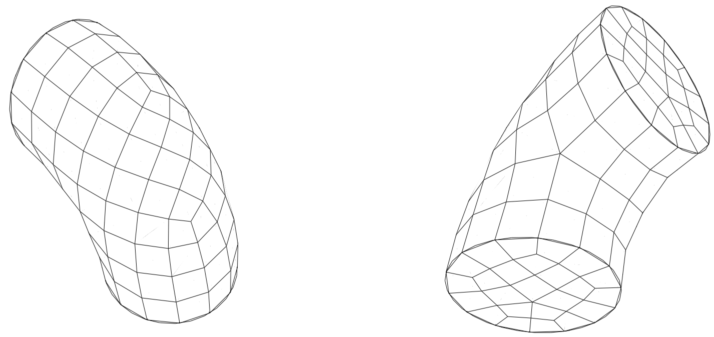

Figure 22. Geometry and surface mesh for the macaroni (1/4 torus) problem.

Meshing the surface with a relatively constant mesh size results in many irregular nodes, which makes this problem more difficult to mesh because of the large difference in inner and outer radii. In terms of STC entities, the irregular nodes and the quad faces surrounding them tend to “redirect” the sheet loops, producing more self intersections. When the surface mesh shown in Figure 22 is looped, 20 self-intersecting faces are produced, four on one sheet and sixteen on another. These are the sheets that prove the most complicated to mesh.

The whisker weaving algorithm closed the macaroni problem with the following metrics:

- 170 hexes
- 6 wedges (left in place)
- 6 removed hex seam operations

Of the nine sheets remaining after the completion of the weave (and after merging two of the initial sheets into one), four of them (sheets 4, 7, 8 and 9) correspond to disks of elements on the two end caps of the geometry. Sheets 5 and 6 are referred to as “stirrup” sheets, because of the topology of their surface loops. Sheets 4-9 are shown in Figure 23.

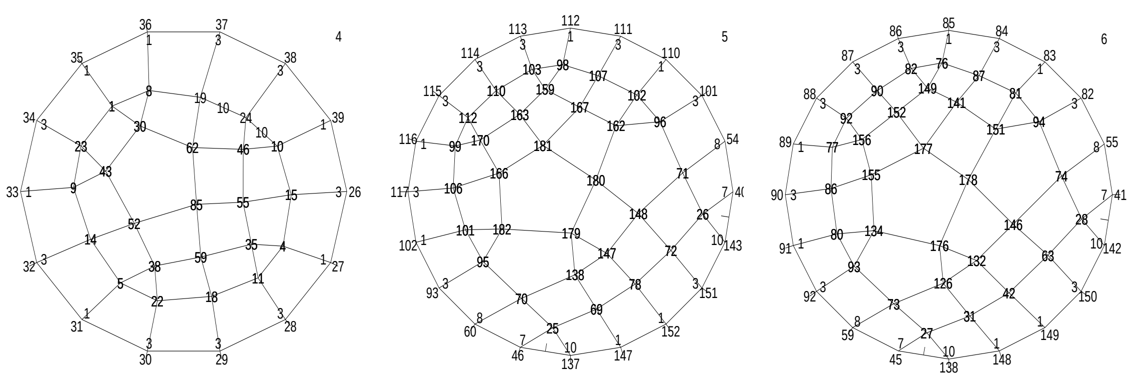

Figure 23. Sheet diagrams for sheets 4-6 (top) and 7-9 (bottom) for the macaroni problem.

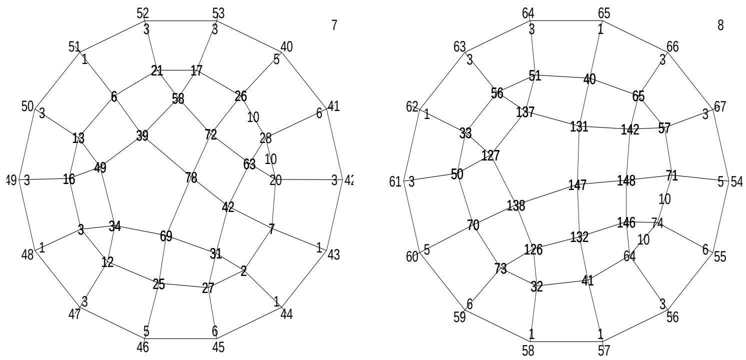

Figure 23. Sheet diagrams for sheets 4-6 (top) and 7-9 (bottom) for the macaroni problem.

Note the similarity between sheets 4, 7, 8 and 9, and between sheets 5 and 6. These similarities can be attributed to similarities in the surface loops and in the crossing sheets.

The final sheet diagram for the merged sheet, shown in Figure 24, is interesting for several

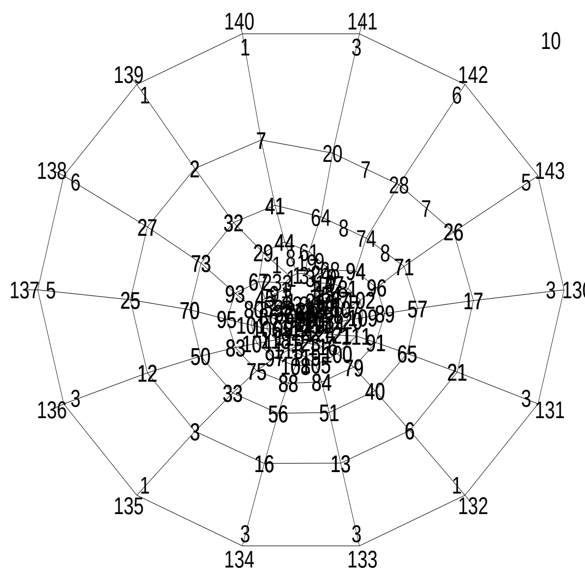

Figure 24. Final sheet diagram for sheet 10 of the macaroni problem, the merged sheet.

reasons. First, the weave is very asymmetric, due to the asymmetry between the inner and outer radii of the macaroni. Second, there are some chords which connect the inner and outer loops, while there are others which begin and end only on the outer loop. The outer loop happens to be on the same end cap as that crossed by the loops of sheets 5 and 6, which do not cross the opposite end cap.

The most interesting sheets for this problem are sheets 1 and 3, which contain all the self intersections (see Figure 25). Sheet 1 contains four self intersections on the face loop. Two of the chords corresponding to these faces join up into a single chord; this chord begins at

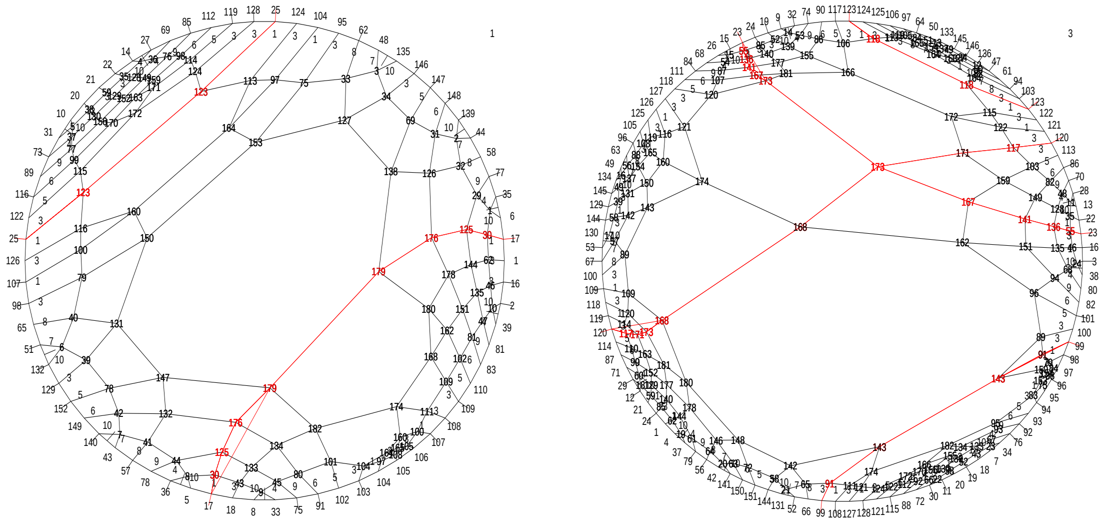

Figure 25. Final sheet diagram for sheets 1 and 3 of the macaroni problem.

face 104 and ends at face 107 in Figure 25. The other two chords beginning at self intersecting faces terminate in wedges inside the mesh. The chord beginning at face 25 ends in a wedge element, which is indicated as vertex 123, while the chord beginning at face 17 ends in a wedge element at vertex 179. These wedges are currently left in the mesh. Recall that a wedge can be easily removed by pulling it out, closing a mesh face on the surface.

Sheet 3 in Figure 25 has 12 self-intersecting chords which pairwise join, forming 6 final self-intersecting chords which begin and end on the surface mesh. Four self-intersecting chords are terminated in the mesh in wedge elements. These chords begin at faces 23, 99, 120 and 123. Pulling all the wedges represented in sheet 3 would result in 4 additional sheets, all much smaller than the original sheet 3, which has 110 faces in its face loop.

7. Conclusions

The mesh dual, contained in the Spatial Twist Continuum (STC), represents the connectivity of an all-hexahedral mesh. The STC consists of an arrangement of surfaces, called whisker sheets. The mesh dual is the facets of intersection of these sheets. Each sheet represents a layer of elements, and each curve of intersection (whisker chord) represents a column of elements. A points of triple sheet intersection (STC vertex) represents a hexahedron, with the sheets bisecting the hexahedron in each of the three principle directions. Sheets are represented using a set of inter-dependent, topologically two-dimensional sheet diagrams. This paper introduces an algorithm, called whisker weaving, that operates in STC space to build an unstructured all-hexahedral mesh given an all-quadrilateral surface mesh.

The basic whisker weaving algorithm consists of four steps; 1) constructing the initial sheets by identifying loops of faces on the surface mesh, 2) choosing three uncompleted chords to make a whisker hex, 3) actually making the hex by crossing the chords, and 4) checking for and resolving mesh invalidities, hopefully resulting in the completion of some chords. Steps 2-4 are repeated until there are no more uncompleted chords, meaning the mesh connectivity is completely specified. Besides the basic types of invalidities, some additional cases arise and are resolved for problems with an irregular surface mesh. Most of these connectivity problems were similar to those encountered in the plastering algorithm[7], and are perhaps germane to advancing front algorithms.

The whisker weaving algorithm is a powerful method for deriving the connectivity of an all-hex mesh given a surface mesh. By considering only connectivity constraints and ignoring most geometric issues, the whisker weaving algorithm reduces the meshing task to a much easier problem. The algorithm is also fast, in that operations are local and expensive geometric embedding and intersection calculations are avoided. The algorithm is boundary sensitive, meaning that it places any poorly shaped elements far from the boundary. This is an advantage in structural analysis, since the geometric boundary is where analytic boundary conditions and loads are usually applied. The algorithm is orientation insensitive as well, so that rotating the input and re-meshing results in the same mesh.

At present, the whisker weaving implementation produces knife elements, which it leaves in place. Although collapsing these knives is implemented and produces a fully valid mesh, this introduces mesh distortion, especially on the surface. An approach based on driving these knives using the STC data has been implemented, and there are a number of other approaches that are being explored as well.[8]

The whisker weaving algorithm outputs the complete connectivity of an all-hexahedral mesh. An algorithm for deriving nodal connectivity and position has been described in another paper [16].

The whisker weaving algorithm has been shown to weave several non-trivial geometries and/or surface meshes. Typical mesh sizes are in the range of 100-200 elements. Further research is needed to enable the algorithm to handle more complicated problems. For topologically ball geometries, there is no fundamental reason why whisker weaving cannot generate an all-hex mesh given any even surface mesh (no algorithm can generate a compatible all-hex mesh if the surface mesh has an odd number of quadrilaterals). Various strategies are being explored for extending the algorithm to topologically non-ball volumes (i.e. volumes with handles). For these volumes, in order for there to exist an all-hex mesh, the surface mesh must satisfy certain additional constraints.[12]

8. References

1. W. A. Cook, W. R. Oakes, “Mapping Methods for Generating Three-Dimensional Meshes”, *Computers in Mechanical Engineering*, Aug. 1982, pp 67-72.

2. T. Tam, C.G. Armstrong, “2D Finite Element Mesh Generation by Medial Axis Subdivision”, *Advances in Eng. Soft.*, Vol 13, 5/6, pp 312-324, Sept./Nov. 1991.

3. T. D. Blacker, M. B. Stephenson, J. L. Mitchiner, L. R. Philips, Y. T. Lin, “Automated Quadrilateral Mesh Generation: A Knowledge-Based System Approach”, ASME paper No. 88-WA/CIE-4 (1988).

4. A. P. Gilkey, G. D. Sjaardema, “GEN3D: A GENESIS Database 2D to 3D Transformation Program”, SAND89-0485, Sandia National Laboratories, Albuquerque, New Mexico, March 1989.

5. PDA Engineering, “PATRAN Plus Users Manual”, Release 2.4, Jan 1990.

6. M. S. Shephard, M. K. Georges, “Automatic Three-Dimensional Mesh Generation by the Finite Octree Technique”, *Int. J. Numer. Methods Eng*., 32:709-749 (1991).

7. Stephenson, M.B., Canann, S.A., Blacker, T.D., Meyers, R.J., “Plastering progress report I”, SAND89-2192, Sandia National Laboratories, Albuquerque, New Mexico, 1992.

8. Ted D. Blacker, S. A. Mitchell, T. J. Tautges, P. Murdoch, S. Benzley, “Forming and Resolving Wedges in the Spatial Twist Continuum”. Engineering with Computers, submitted 1994, to appear.

9. T. D. Blacker, M. B. Stephenson, “Paving: A New Approach to Automated Quadrilateral Mesh Generation”, *Int. J. Numer. Methods Eng*., 32:811-847 (1991).

10. P. Murdoch, S. S. Benzley, T. D. Blacker, S. A. Mitchell, “The Spatial Twist Continuum Captures The Global Connectivity of an All–Hexahedral Finite Element Mesh”, manuscript.

11. T. D. Blacker, R. J. Meyers, “Seams and Wedges in Plastering: A 3-D Hexahedral Mesh Generation Algorithm”. *Engineering with Computers*, 9:83-93 (1993).

12. S. A. Mitchell, “A Characterization of the Quadrilateral Meshes of a Surface Which Admit a Compatible Hexahedral Mesh of the Enclosed Volume”. In proc. 13th Symposium on Theoretical Aspects of Computer Science (STACS `96), Lecture Notes in Computer Science, Springer 1996, pp 465-476.

13. T. D. Blacker et al., “CUBIT Mesh Generation Environment, Volume 1: User’s Manual”, SAND94-1100, Sandia National Laboratories, Albuquerque, New Mexico, May 1994.

14. Ted D. Blacker, “FASTQ Users Manual Version 1.2”, SAND88-1326, Sandia National Laboratories, Albuquerque, New Mexico, August 1993.

15. Larry A. Schoof, Victor R. Yarberry, SAND92-2137, Sandia National Laboratories, Albuquerque, New Mexico, September 1994.

16. Timothy J. Tautges, Scott Mitchell, “Whisker Weaving: Invalid Connectivity Resolution and Primal Construction Algorithm”, in Proc. 4th International Meshing Roundtable, SAND 95-2130, Sandia National Laboratories, Albuquerque, New Mexico, September 1995.